Volume 34 Number 3 October 2010 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Semantic Informational Technologies

Guest Editor:

Vladimir A. Fomichov

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Editorial

Semantic Informational Technologies

The term *semantic informational technologies* (or, shorter, *semantic technologies*) emerged in the 2000s as a generic concept for the qualification of a group of quickly progressing technologies including, in particular, semantics-oriented natural language processing technologies, the use of ontologies in the Semantic Web project and in many other projects of applied intelligent systems, cross-language conceptual information retrieval, ontology-based images recognition and retrieval, the generation of natural language (NL) texts, proceeding from the inner representations of their meanings, the elaboration of content representation languages as a part of agent communication languages in multi-agent systems, and the development of formal means for representing the records of e-negotiations and forming the contracts in the subfield of electronic commerce called e-contracting.

The common features of the technologies from this group is either processing of NL-texts with respect to the fact that lexical items have the meanings (i.e., are associated with one or several semantic items) or/and processing information with respect to an ontology (i.e., with respect to a set of interrelated formal records corresponding to the concepts and the connections of concepts underpinning natural language processing by people).

Overview of the issue

This special issue of *Informatica – an International Journal of Computing and Informatics* contains 7 papers submitted by the researchers from Bulgaria, Czechia, France, Russia, Serbia, Singapore, and Slovenia. The papers were carefully selected on the basis of peer reviews.

Two distinguished features of this issue as a whole are as follows. Firstly, the papers from this special issue describe the studies pertaining to the main branches of semantic informational technologies and, as a consequence, give a rather good initial look at the current state of this field. Secondly, the spectrum of described and discussed subjects is very large: from the industrial applications of the methods and models developed under the framework of the Semantic Web project to the strategy and formal tools of transforming the existing World Wide Web into a Semantic Web of a new generation.

Biology and medicine (biomedicine) are the fields where the methods of semantics-oriented natural language processing (NLP) are being very intensively developed and applied to solving practical tasks. The paper "Obtaining Status Descriptions via Automatic Analysis of Hospital Patient Records" by S. Boytcheva, I. Nikolova, E. Paskaleva, G. Angelova, D. Tcharaktchiev, and N. Dimitrova from Sofia, Bulgaria pertains just to these fields. The paper describes the progress of the study aimed at automatic extraction of patient status data from medical texts in Bulgarian language. It is shown that certain patient-related facts can be relatively easily extracted from the texts.

The paper "Corpus and Web: Two Allies in Building and Automatically Expanding Conceptual Classes" by N. Béchet, J. Chauché, V. Prince, M. Roche (Montpellier, France) describes an original application of the methods of NLP to building and expanding conceptual classes. To find the effective solutions to this problem is important not only for biomedicine but also for many other fields. The main method of the study was to investigate a semantic-syntactic dependency in a sentence between a verb *Vb1* and its object *Ob1*, proceeding from the semantic dependency between a semantically close verb *Vb2* and its object *Ob2*. As a whole, the paper contributes to bridging a gap between Web-based and corpus-based approaches to forming and expanding conceptual classes.

The paper "Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web" by V.A. Fomichov (Moscow, Russia) formulates an original strategy of transforming the existing Web into a Semantic Web of a new generation with the well-developed mechanisms of understanding NL-texts (or a Meanings Understanding Web, or a Multilinguistic Semantic Web). Besides, the paper indicates the basic formal tools being necessary for the realization of this strategy. Firstly, the paper grounds the possibility of using a mathematical model being the kernel of the theory of K-representations and describing a system of 10 partial operations on conceptual structures for building semantic representations (or text meaning representations) of, likely, arbitrary sentences and discourses in English, Russian, French, German, and other languages. The possibilities of using SK-languages (standard knowledge languages), defined by the theory of K-representations, for building semantic annotations of informational sources and for constructing semantic representations of discourses pertaining to biology and medicine are illustrated.
Secondly, the paper describes the correspondence between the inputs and outputs of an original algorithm of semantic-syntactic analysis and indicates its advantages; the semantic representations of the input texts are the expressions of SK-languages. The input texts can be the statements, questions, and commands from the sublanguages of English, Russian, and German.

The next paper "Wikipedia2Onto – Building Concept Ontology Automatically, Experimenting with Web Image Retrieval" by H. Wang, X. Jiang, L.-T. Chia, and A.-H. Tan from Nanyang Technological University, Singapore describes an original approach of the authors to using ontology for better understanding the visual images stored on the Web. This approach includes the construction of a large-scale multi-modality ontology

from Wikipedia for Web images classification. The generated ontology allows for extracting additional information from the Web pages and for increasing the accuracy of concept detection.

The paper "A Service Oriented Framework for Natural Language Text Enrichment" by T. Štajner, D. Rusu, L. Dali, B. Fortuna, D. Mladenić, and M. Grobelnik (Ljubljana, Slovenia) sets forth an original method of complementing the free NL-texts with an enrichment being a set of the triplets of the form *subject, predicate, object*. Due to the use of the triplets, the enrichment can be presented with the help of RDF – one of the basic languages of the Semantic Web project. On the basis of this set of triplets, a semantic graph of a text is constructed. As an example, an enrichment of a short article from Wikipedia is considered. The document's semantic graph is a starting point for automatically generating a document summary. The proposed method is implemented in the applied computer system Enrycher. Several directions of experimenting with this system are outlined.

Two final papers of this special issue will be of particular interest to many readers, because these papers describe the industrial applications of the methods, models, and language means elaborated under the framework of the Semantic Web project. The paper "Applications of Semantics in Agent-Based Manufacturing Systems" by M. Obitko, P. Vrba, V. Mařík, M. Radakovič, and P. Kadera (Prague, Czech Republic) shows the advantages of using semantic models of application domains in the design of distributed intelligent control systems in comparison with traditional centralized manufacturing architectures. One of the precious features of the paper is that it contains a substantial discussion of the role of semantics, RDF-based and OWL-based ontologies, and architectures of Semantic Web Services in the design of distributed intelligent industrial systems. A new ontology for manufacturing domain is described; this ontology provides a semantic model of production planning and scheduling, material handling, and customer order specification. The integration of this model with an agent-based simulation and control system MAST is set forth.

The subject of the paper " The Role of the Semantic Web for Knowledge Management in the Construction Industry" by Igor Svetel and Milica Pejanović (Belgrade, Serbia) is the applications of RDF and OWL-based ontologies in the architecture, engineering, and construction industry (AEC industry). It is shown that the principal advantage of this approach is the contribution to preventing construction time delays, unforeseen work and, as a consequence, the exaggerated cost of buildings. The paper gives an overview of the standards developed for providing interoperability and flexibility in the AEC industry and of the standards elaborated under the framework of the Semantic Web project.

The guest editor would like to thank Professor Matjaz Gams for providing the opportunity to prepare this special issue on Semantic Informational Technologies. Finally, many thanks to the authors of the papers for their contributions and to all of the referees for their precious comments ensuring the high quality of the accepted papers and making the reading as well the editing of this special issue a rewarding activity.

Vladimir A. Fomichov
Professor of Computer Science
Department of Innovations and Business
in the Sphere of Informational Technologies
Faculty of Business Informatics
State University – Higher School of Economics
Kirpichnaya street 33, 105679 Moscow, Russia
Email: vfomichov@hse.ru

Obtaining Status Descriptions via Automatic Analysis of Hospital Patient Records

Svetla Boytcheva
State University of Library Studies and Information Technologies, Sofia, Bulgaria
E-mail: svetla.boytcheva@gmail.com

Ivelina Nikolova, Elena Paskaleva and Galia Angelova
Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria
E-mail: {iva, hellen, galia}@lml.bas.bg

Dimitar Tcharaktchiev
University Specialised Hospital for Active Treatment of Endocrinology "Acad. I. Penchev"
Medical University, Sofia, Bulgaria
E-mail: dimitardt@gmail.com

Nadya Dimitrova
National Oncological Hospital, Bulgarian Cancer Registry, Sofia, Bulgaria
E-mail: dimitrova.nadia@gmail.com

Keywords: automatic natural language processing, information extraction, hospital patient record, patient status, template filling, structured representation

Received: November 16, 2009

Abstract. This article describes the automatic processing of medical texts in order to extract important patient characteristics, thus turning the free text description into a structured internal representation. Shallow text analysis is implemented due to the medical language complexity. The paper sketches the information extraction process and discusses the role of domain knowledge in text analysis. The approach to domain model construction is presented. Evaluation results concerning extraction of patient diagnoses and status are summarised.

Povzetek: Predstavljena je metoda za gradnjo semantičnih podatkov o pacientih iz nestrukturiranega besedila.

1 Introduction

Medical patient records are important documents that were created, processed and stored since the ancient times. They keep the patients' diagnoses, treatments, manipulations etc. Nowadays their role is growing together with the increasing potential for collecting, storing and processing of medical information. Much data values are structured by the Hospital Information Systems – for instance, the numeric values of lab tests are automatically entered in predefined fields, and the drugs prescribed to the patient are maintained via the so-called Computerized Physician Order Entry. However, essential findings are traditionally stored as free text descriptions. In this way the automatic text analysis is viewed as an information technology of vital importance, because it enables automatic generation of databases with structured patient data that can be explored for improving the diagnostics, care decisions, the personalised treatment of diseases, maintenance of adverse drug events, healthcare management and so on. There are major advances in several directions of medical text processing. One important task is to implement tools for automatic extraction and coding of patient-related information with respect to some established classification schemes, such as ICD (the International Classification of Diseases); in this scenario the automatic extraction can provide essential optimisation of health management tasks. Another important objective is to support knowledge discovery in medicine by doing research on disease causes and symptoms, since the automatic text analysis enables searching for effective treatment methods in patient records' texts. In this approach the medical texts are "translated" to internal formalised representations; then inference algorithms can reveal interconnections and regularities between facts and concepts that could remain unnoticed otherwise. Unfortunately most of the medical documents are available as free texts only, which is a major obstacle to the automatic Information Extraction (IE). Despite the difficulties and challenges, however, there is a growing number of industrial systems and research prototypes in

many natural languages, which perform information extraction from patient-related texts. So the application of language technologies to Patient Records (PRs) free text is viewed nowadays as a must in health informatics.

This paper describes an IE prototype which is applied to PR texts in Bulgarian language. The extraction tasks run on anonymised records for hospital treatments of diabetic patients. Section 2 summarises related research dealing with IE from medical texts. Section 3 presents our prototype: the linguistic and conceptual resources and the IE phases for extraction of patient status data. Explicitly-declared domain knowledge enables application of constraining rules and inferences. Section 4 summarises recent evaluation results. The experiments are run within an integrated multifunctional prototype which supports constant collection of new training data. The conclusion and plans for further work are given in Section 5.

2 Related work

Information Extraction is a popular Natural Language Processing (NLP) approach which was proposed in the 1980s as a flexible technology for analysis of domain texts. It extracts only the *relevant* information and ignores the rest, assuming that it is either too difficult to be captured by shallow techniques or consists of irrelevant words (and hence, by default deals with topics which are irrelevant to the problem in question). Relevant information is communicated by relevant words, so there are clear signs where to look and what to analyse in the message. In this way the IE systems are tailored to the extraction of specific facts only, by knowing in advance the words that can signal the entities and relationships of interest. As the overview [1] points out, IE requires "deeper analysis than key word searches but focuses on surface linguistic phenomena that do not require deep inference". In this way IE represents a midpoint between keyword identification and full text understanding. The classical rule-based IE paradigm involves Named Entity Recognition, extraction of entities after morphological analysis, recognition of phrasal expressions and shallow syntactic analysis, recognition of (co-)references, creation of databases, and filling event templates [2].

Recent IE systems typically achieve more than 90% accuracy in Named Entity Recognition, about 80% in template elements construction and about 60% in scenario template production. Most often IE is limited to "the 60% barrier" because of erroneous system choices in the recognition of coreferences between entities and events; another possibility is that this barrier is due to the shallow analysis potential since IE avoids interpretation of implicit relationships and deep inference [1]. In specific domains, however, and with suitably defined IE targets, the automatic extraction features higher precision and recall. Nowadays IE is the common approach to automatic text analysis in biomedicine, but more fundamental research is needed to advance automatic text understanding in principle; there are high expectations that the NLP progress would enable radical improvements in the clinical decision support, biomedical research and the healthcare sphere in general [3].

Current systems for automatic text analysis are usually focused on specific topics only due to domain complexity and the very large number of entities and relationships there. The technology is applied in various prototypes which are constructed to perform different extraction tasks from medical documents, including the following ones:

- ***Processing of patient symptoms and diagnosis treatment data***: the system CLEF (Clinical E-Science Framework) extracts data from clinical records of cancer patients [4]; AMBIT acquires Medical and Biomedical Information from Text [5]; MiTAP (MITRE Text and Audio Processing) monitors infectious disease outbreaks and other global events [6]; the system caTIES (Cancer Text Information Extraction System) processes surgical pathology reports [7]. Other recent systems are HITEx (Health Information Text Extraction), an open-source NLP system [8] and cTAKES (clinical Text Analysis and Knowledge extraction system) [9];
- ***Building of medical ontologies***: IE is applied for construction of ontology in pneumology in the PertoMed project. The approach is based on terminology extraction from texts according to the differential semantics theory - distributional analysis and recognition of semantic relationships by lexico-syntactic patterns [10]. ODIE (Ontology Development and Information Extraction) is a software toolkit which codes document sets with ontologies or enriches existing ontologies with new concepts from the document set. It contains modules for Named Entity Recognition, coreference resolution, concept discovery, discourse reasoning and attribute value extraction [11];
- ***Automatic assignment of ICD codes to diagnoses extracted from patient records***: the article [12] summarises the results of the 2007 Computational Medicine Challenge, a competition which was run on anonymised radiology reports. The top coding systems achieved 89% accuracy and the mean was 76,7%. The three top systems processed the negation, hypernyms and synonyms in some way and exploited the UMLS structure [13]. All three systems performed symbolic computations and two of them had in addition some machine-learning components. The overview [12] notes the importance of rule-based text analysis in the coding-oriented NLP tasks.

Current IE systems are often based on shallow analysis by regular expressions and pattern matching. Some patterns are manually produced and their adaptation to new domain requires much efforts. Other patterns are semi-automatically produced using general meta-rules but they are not too precise [14]. The integration of machine-learning approaches, like e.g. classification of sentences, enables recognition of patient attributes with high precision and recall [15].

In addition we should notice the importance of linguistic and conceptual resources and their integration in the IE tasks. The paper [16] discusses the automatic

entity recognition in biomedical texts using a gold standard corpus of 77 English documents with 2124 entities of five types. The authors consider various methods, ranging from dictionary look-up to machine learning approaches, with maximal success of 83% in entities recognition and conclude that dictionary look-up is a promising basic strategy for terminology recognition (which is the technique chosen in our project too). The system MedScan demonstrates the advantages of ontology-driven approaches to medical IE [17]. MedScan processes sentences from MEDLINE abstracts and produces a set of semantic structures representing the meaning of each sentence. In 2003 it extracted information about pathways and molecular networks, so it was tuned to process sentences containing the relevant words in these areas. After parsing, each sentence is represented as semantic frame; an ontological interpreter evaluates the outputs of the NLP component and converts the valid ones into ontological representation. The following accuracy is reported: processed 4,6 million sentences, with 34% correctly parsed sentences but the analysis of errors shows that with larger lexicon and better grammar the system can extract protein function information with precision above 90% [17]. MedScan applies the ontology as a filter to select correct semantic sentence structures and to skip text units which are irrelevant to the target subject.

Most of the presented IE techniques cannot be directly adapted to our project, because we deal with documents in Bulgarian and many language-processing activities start from scratch. For instance, no Named Entity Recognition module has been implemented for Bulgarian entities in the medical domain; the regular expressions for shallow sentence analysis are constructed for the first time and so on. Therefore we need to select some priorities, i.e. which topics are to be treated first. From medical point of view, a significant task is to analyse the hospitalisation effects: what happens to a patient when he or she enters the hospital in status A and leaves it in status B, i.e. how the hospital treatment affects the patient status. Therefore an important activity is the automatic IE of patient status data, especially the diagnoses and the status extraction for organs which are referred to in the PRs of patients with diabetes.

3 Obtaining patient status data from Bulgarian PR texts

In this section we present our approach to extraction of patient status based on cascades of regular expressions. The PR text is split into relevant fragments using a declarative conceptual model of medical entities and relationships among them. We briefly discuss the raw input texts, the linguistic resources, and the domain model construction.

3.1 Corpus of PRs and system resources

The length of PR texts in Bulgarian hospitals is usually 2-3 pages. The record is organised in the following standard sections: *(i)* personal details; *(ii)* diagnoses of the leading and accompanying diseases; *(iii)* anamnesis (personal medical history), including current complains, past diseases, family medical history, allergies, risk factors; *(iv)* patient status, including results from physical examination; *(v)* laboratory and other tests findings; *(vi)* medical examiners comments; *(vii)* discussion; *(viii)* treatment; *(ix)* recommendations. So the patient status description is clearly seen in the text, it facilitates the application of IE algorithms.

The PR text contains medical terminology in Latin alphabet (about 1% of all term tokens in our present corpus), sometimes with different transcriptions in Cyrillic alphabet. There are specific term abbreviations both in Bulgarian and Latin (about 3% of the tokens), numerical values (16% of the tokens) etc. In the hospital PRs, complete sentences are rare, since the text contains primarily sentence phrases only. Sometimes there is no agreement between the sentence parts, and the punctuation marks are not properly placed. Further specific problems are due to the highly-inflexional Bulgarian morphology; the terms occur in the text with a variety of wordforms. Our present raw text training corpus consists of 197 anonymised PRs of diabetic patients which contain 166336 word occurrences or 146900 tokens after the elimination of enumerations, tables, indices and repeating wordforms. Actually the training corpus contains some 6400 words, with about 2000 of them being medical terms. The test corpus contains 1000 anonymised PRs of diabetic patients.

In order to capture the patient-related information, we use a terminological bank of medical terms derived from ICD-10 in Bulgarian language. The International Classification of Diseases (ICD-10) contains 10970 terms. The Bulgarian version of ICD-10 has no clinical extension, i.e. some medical terms need to be extracted from additional resources like a partial taxonomy of body parts, a list of drugs etc. A lexicon of 30000 Bulgarian lexemes, which is part of a large general-purpose lexical database with 70000 lexemes, completes the necessary dictionary for morphological analysis of Bulgarian medical text. In addition to the lexicons compiled from different sources, the following linguistic and conceptual resources are integrated in the *resource bank* of our system and support the text analysis:

- semi-automatically prepared regular expressions which enable recognition of particular language constructions;
- rules for negation treatment;
- sets of possible and default values for each attribute for each anatomic organ as well as observations about attribute correlations for each anatomic organ. These values – words and phrases - are collected in advance from the representative training corpus of PRs, textbooks, consultations with medical experts etc.;
- templates to be filled in by organ descriptions with associated list of obligatory and optional fields;
- domain model of concepts and relationships relevant to diabetes, including ontology of body parts (see section 3.3) and ontology of diabetes complications which is adopted from the BioPortal resources [18];

- list of drug names and names of relevant medical appliances.

3.2 Shallow analysis of PR texts

The IE procedure for a predefined entity of interest is initiated when a word signalling an entity description occurs in the PR section *(iv)* "patient status". Let us consider an example where the extraction is to be performed for certain anatimic organ (***AO***), where e.g. ***AO***="*крайници*" (limbs) is identified by the morphological analyser. Then the IE system finds in the resource bank the set of ***AO*** characteristics ***Ch***, let in our case ***Ch***={*ankle, leg, peripheral artery, feet, skin, nail*}. Actually the resource bank contains in the domain model the list of all organs related to the chosen ***AO*** which is stored in the domain ontology: especially for lower limbs, the status explanation text can contain details about different limb vein condition, toes, etc. Thus the set ***Ch*** is enlarged to ***Ch'*** for the processed ***AO*** including the other related anatomic organs which potentially can be discussed in the text. Finally the IE system selects from the resource bank the set ***V*** of all relevant characteristics' values. The sets ***V*** and ***Ch'*** contain not only Bulgarian terminology but also Latin terms and their Cyrillic transliteration as well as term abbreviations both in Latin and Bulgarian language. For instance, the reference to the term "*глезен*" (*ankle*) can be found in the PRs also as "*перималеоларни*" (*perimaleolar*) and the term "*периферните артерии*" (*peripheral arteries*) can be represented also as "*a. dorsalis pedis*" and "*аа. дорзалис педис*". Please note that we do not discuss here the possible spelling errors in the text which need to be automatically corrected before the actual IE processing starts; spell-checking should be treated as a technical pre-processing problem in this case.

The further step of the IE algorithm is to determine the scope of the text descriptions where the status of the chosen ***AO*** is presented. Usually this information is given in several consecutive phrasal descriptions or sentences, in a compact manner about one anatomic organ. In our particular example, the system has to decide which adjacent sentences and phrases describe the limb status; the IE analysis for *limbs* will be run only on the selected text fragment. Scoping is made by using the terms and the corresponding concepts in the domain model. There are several rules for scope recognition, let us list two of them:

- ***AO* followed by its characteristics, AO_1 followed by its characteristics, ..., AO_n followed by its characteristics ...**

where AO_n is the first organ in the paragraph not related to the processed ***AO***. For instance:

"*Крайници – отслабени пулсации на a. dorsalis pedis двустранно. Претибиални и перималеоларни отоци. Онихомикоза, tinea pedis. Сукусио реналис – (-) отр. двустранно"* (*Lower limbs – reduced dorsal pedal pulse on both feet. Pretibial and perimaleolar edema. Onychomycosis, tinea pedis. Succusio renalis – (-) bilateral negative*).

The IE system finds in the second sentence a reference to body parts, which are related to limbs - "*претибиални" (leg)* and *"перималеоларни" (ankle).* The third sentence contains the terms *"онихомикоза" (onychomycosis)* - fungal infection of the nails – which is also related to limbs parts and "*tinea pedis",* denoting fungal foot infection. The fourth sentence contains *"succusio renalis"*, which describes a test for pain in the kidney area, and this is a signal that the limbs description is completed.

- ***AO* followed by its characteristics, some characteristics and values not belonging to the set *V* of *AO***. For instance:

"*Крайници - без отоци, варикозни промени, запазени пулсации на периферните артерии, запазени повърхностна, термо и вибрационна чувствителност. Затруднена и болезнена походка, използва помощни средства*" (*Lower limbs – without oedema, varicose changes, palpable peripheral arteries pulse, preserved tactile, thermo and vibratory sensation. Walks with difficulty, algetic gait, uses assistive devices*).

Here the occurrence of the word "*gait*" signals the completion of the limbs description. Further considerations of our heuristic strategy for recognising the irrelevant terms and concepts are given in [19]: if the IE process runs for a term/concept *X*, only concepts linked to *X* by relations *isa*, *part-of*, *has-location* and *associated-with* are considered relevant. In this way the selection of topic-relevant text fragments is done by integral evaluation of linguistic units in the particular input text and corresponding conceptual entities in the domain model.

The shallow syntactic analysis of the selected text fragment is made by application of regular expressions modelling PR phrasal patterns. The IE system finds in the grammatical resources the greediest regular expression that will recognise the maximal part of the sentences selected at the previous step. The system applies the available regular expressions to the text units one by one until a perfect match is found. In case of partial recognition for all of them, the one that fits to the maximal text fragment is selected. We present below two types of regular expressions for *limbs*, out of six types actually used in our IE system. Let us consider the ***AO***s, their characteristics ***Ch*** and their attribute features ***F***. Then the status-related expressions can be grouped into categories, for instance:

- Description of one ***AO***, all its characteristics and their features presented in one sentence:

AO [-] ['with'|'of' F] Ch1, ['with'|'of' F] Ch2, ...

"Крайници без отоци, запазени периферни пулсации, онихомикоза" (*Lower limbs without oedema, preserved peripheral pulse, onychomycosis*).

- Description of one ***AO***, all its characteristics and their features presented in several consecutive sentences:

AO [-] ['with'|'of' F] Ch1. ['with'|'of' F] Ch2. ...

"Крайници – без отоци. Запазени пулсации на периферните артерии" (*Lower limbs without oedema. Palpable peripheral arteries pulse*).

About 96% of all PRs in our training corpus contain limbs descriptions in this format, which excludes the application of deeper syntactic analysis at least to the text paragraphs concerning organ descriptions. The above-listed regular expressions are acquired from the training PR corpus, taking into account some typical prepositions and phrasal constructions.

Unrecognised text fragments which contain relevant words are processed by extra rules in order to capture some negative statements. The IE system considers the negated descriptions as one expression, following a study of negative forms in Bulgarian hospital patient records [20]. For instance:

"Крайници - без отоци или варикозни промени, запазени пулсации на периферните артерии" (*Lower limbs – without oedema or varicose changes, palpable peripheral arteries pulse*),

"*Крайници - без отоци, варикозни промени, запазени пулсации на периферните артерии*" (*Lower limbs – without oedema, varicose changes, palpable peripheral arteries pulse*).

In the first sample the negation "without" refers to "*oedema*" and "*varicose changes*" together, but in the second sample the negated word "*without*" refers to "*oedema*" only and staments about the existence of "*varicose changes*" for this patient is positive.

Some more complicated cases are recognised by the rules for resolving the scope of the characteristics and their values. For instance:

"*Крайници - без отоци, липсващи периферни пулсации на аа.дорзалес педис и тибиалес постериор, суха ливидна, атрофична кожа на стъпалата, ливидни студени пръсти, инфектирани разязвявания на дясно стъпало*"

(*Lower limbs – without oedema, absent dorsal pedal and posterior tibial pulses, dry livid atrophic skin of the feet, livid cold toes, infected ulcers of the right foot*).

In this sample we find six different characteristics: the scope of *"absent peripheral pulses"* concerns the "*dorsal pedal arteries"* and *"posterior tibia's artery*". There is only one characteristic for two anatomic organs. Another case is "*dry livid atrophic skin of the feet",* where we have three characteristics for one anatomic organ within one text phase.

Sometimes status descriptions are mising especially when no pathological changes are observed or the examining medical expert relies on tacit knowledge. In a previous paper we have proposed to collect information concening the attribute correlations by making observarions about attribute interdependencies [21]. In this way we can add most probable values in the template fields which have remained empty, because no explicit statements were found in the PR text. To study the correlation of values for different organ characteristics, the medical experts in the project have developed a scale of *normal*, *bad* and *worst* conditions. Some words from the PR texts are chosen as representative for the corresponding status scale and the other text expressions are automatically classified into these typical status grades. Table 1 illustrates the scales for *limbs* and gives examples for words signalling the respective status. The regular expressions which have been developed for shallow analysis of limbs status map the explicit text descriptions about limbs into the chosen categories. In this way all word expressions are turned into numeric categories, and it becomes possible to study the deviations from the normal condition. The mapping process is not trivial and requires quite precise elaboration and testing of the regular expressions which enable the recognition of the text descriptions. Our approach has similarities to the one presented in [15], where the patient smoking status is classified into 5 categories.

Scale	Ankle	Leg	Peripheral artery pulsation
0	*normal*	*normal*	*normally present*
-1	*(light) swelling*	*oedema*	*reduced*
-2	*solid swelling*	*solid swelling*	*absent*

Table 1: Limbs Characteristics Categorisation

Finally, the IE algorithm has to choose the appropriate template for the captured information, because each template has versions without and with optional fields. Templates are designed after a careful study of the training corpus. For instance, about 99% of the processed PRs discuss explicitly the status of patient *ankle*, *leg* (*ankle* and *leg* status is usually described together) and the *peripheral artery*. Due to the importance of these organs in the status of diabeticians, they are defined as obligatory fields in the limb-status templates. Dynamic generation of template field is possible, to capture the more detailed descriptions of organ status.

Finally, at the last step, the default values are filled in, in case there are obligatory template fields which cannot remain empty. Default values are defined to cope with missing descriptions in the patient's clinical notes. For instance, 77% of the PRs in our training corpus do not discuss explicitly the skin hydration; only 42% discuss the turgor and the elasticity; but 62% discuss the fat tissue and 63% - the skin colour [21]. Therefore, we need to prelist the default status values, to ensure the proper filling of obligatory template fields.

Further details about linguistic particularities of the PR texts, the shallow text analysis and the dynamic template extension are presented in [20], [21] and [22].

3.3 Building domain model to support IE from diabetic patients' PRs

Without making deep inference, the IE applications integrate some kind of ontological resources to consistently interpret the semantic relationships existing among the entities identified in the text. Often these domain models are constructed using standard or widely-used public controlled vocabularies. However, the manual acquisition is time-consuming and non trivial, therefore we need semi-automatic methods for corpus-based term collection and expansion of the controlled vocabulary by conceptual relations. Our domain model has to support the IE tasks as well as further search of

conceptual patterns by providing general and sibling concepts which enable to identify similarities among case histories. In addition we deal with terms in Bulgarian, which are to be mapped to ontological labels in the IE interpretation phase; therefore we need a conceptual resource with labels in Bulgarian. Only the flat nomenclatures ICD-9 and ICD-10 are translated to Bulgarian and can be directly used as a basic terminological lexicon in the IE tasks. Therefore the development of an IE prototype requires conceptual model construction at least for the domain of diabetes.

Starting from the Bulgarian corpus of PR texts and using the Bulgarian terms of ICD-10, we have performed the following automatic steps which facilitated the corpus-based construction of relevant Bulgarian terms:

(i) We have found all corpus wordforms that do not belong to the Bulgarian lexicon of 70000 entries which contains general lexica. Some 75% of them are manually classified as relevant terms (and another 3% are due to spell-errors);

(ii) We have mapped all corpus wordforms to ICD-10 to find domain terms that participate in the nomenclature;

(iii) We have applied a clunker of Bulgarian phrases to the morphologically-analysed PR text which groups single wordforms into phrases. These phrases are mapped to the ICD-10 terms too.

After manual refinement of the joint term collections, we have constructed a list *Diab-Term-Bg* of 1098 terms, which are potential Bulgarian ontological labels in the conceptual model we need to construct. Applying bilingual Bulgarian-English dictionaries and manual correction by medical experts, these terms are translated to English in order to use them as entries for accessing public semantic resources labelled by English vocabulary. Having at hand this list, named *Diab-Term-Eng*, we can search in the UMLS resources, including MeSH, SNOMED, ICD and so on.

The medical nomenclatures, controlled vocabularies and ontologies in UMLS are not readily suited for our purposes. For instance, MeSH (Medical Subject Headings) - the USA National Library of Medicine's controlled vocabulary thesaurus is a polytree, a hierarchical structure containing 22568 descriptors. The top level concepts are labeled by broad categories such as *Anatomy*, *Diseases*, *Organisms*, etc. The MeSH hierarchy is a forest with 16 heads and depth 11. It contains concepts and relations of synonymy, near-synonymy, and closely related concepts. The MeSH thesaurus was initially proposed for indexing, cataloguing, and searching for biomedical documents. Recently MeSH terms are actively used to e.g. improve information retrieval (by query expansion) but it is hard to apply them as NLP ontological backbone, since most concepts have no property-value specifications, and many available properties convey either very general relationships or relationships that are hard to interpret in the NLP context [23]. Therefore we combine automatic extraction of important UMLS fragments and manual reviewing and editing in order to reduce the ambiguity and to assert the conceptual relations needed to support the IE tasks in the diabetes domain.

For mapping English medical terms to UMLS concepts we use the UMLS tool Metamorphosys and the UMLSKS server. In this way we retrieve the term's concepts with their synonyms, definition, semantic types and sub-concepts together with pointers to the different vocabulary sources. There could be several concepts corresponding to a given term, and manual editing is needed to filter the *isa*-hierarchy and tailor it for our application-tailored domain model. We extract and process hierarchies starting from top categories like *Disease or Syndrome* and *Anatomical Structure*. Figure 1 illustrates the adjusted hierarchical structures we obtain after manual editing of UMLS fragments. As background annotation, we store markers pointing to the UMLS resources which are reviewed in the acquisition process.

In addition to the hierarchical refinements, we need to construct the relations among the concepts of interest. UMLS contains two basic relation types: the hierarchical *isa* and the relation *associated_with* with five sub-relations: *physically_related_to*, *spatially_related_to*, *functionally_related_to*, *temporally_related_to* as well as *conceptually_related_to*. The tree of *associated_with* has depth 4 and contains 52 subrelations, some of them shown at Figure 2. For instance, the important *part_of*

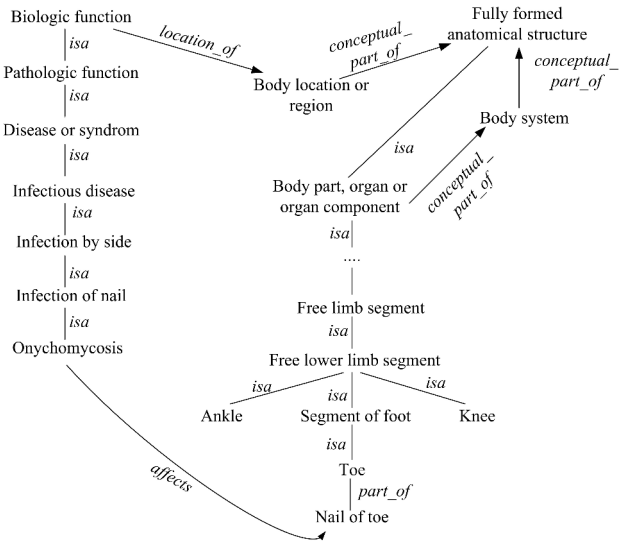


Figure 1. Semantic network for diseases and anatomic organs constructed using automatically extracted UMLS fragments

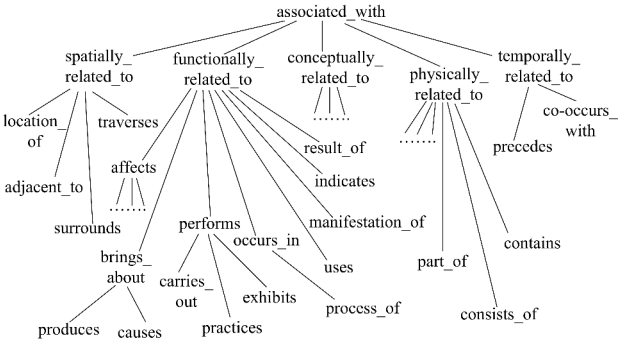


Figure 2. UMLS relations: 28 subrelations of *associated_with*

relation is a subtype of *physically_related_to* and has siblings such as *consists-of*, *contains*, *connected_to* etc. Using the extracted hierarchical structures, we link concepts by UMLS relations. Currently this process is completed for about 300 concepts needed to support the IE of diabetic patient status data. Finally, in our acquisition workbench we provide labelling of domain concepts by the relevant Bulgarian terms. This is necessary, since the IE tasks run on Bulgarian PR texts, and they map input words to domain concept labels during the IE interpretation phase.

Another domain model part concerns the templates where the IE system captures the extracted status data. Usually the IE templates are tables and database entries, but in the medical domain we take into consideration the available archetypes (patterns of standartised structures which normalise the descriptions of various medical artefacts). Archetypes are developed by the openEHR Foundation, an international body which aims at the development of interoperable electronic health records in Europe [24]. They are regarded as an obligatory element of the future EU eHealth framework.

4 Evaluation of the IE prototype

Successes and failures of IE performance are measured by special evaluation exercises which prove the feasibility of the approach to perform partial analysis only, tackling selected entities and relationships. The IE performance is assessed in terms of three classical measures. The *precision* is calculated as the number of correctly extracted entity descriptions, divided by the number of all recognised entity descriptions in the test set. The *recall* is calculated as the number of correctly extracted entity descriptions, divided by the number of all available entity descriptions in the test set (some of them may remain unrecognised by the particular IE module). Thus the precision measures the success and the recall – the recognition ability and "sensitivity" of the algorithms. The F-measure (harmonic mean of precision and recall) is defined as

$F = 2 \times Precision \times Recall / (Precision + Recall)$.

We have developed a prototype which integrates various functions for maintaining the training and test corpus of anonymised PR texts. The IE tasks include browsing and searching functionality, visualisation of the internal templates to the user (see Figure 3) and options for manual editing especially when diagnose codes are assigned [25]. This integrated prototype serves as a convenient unified software environment which is used by project developers and medical experts. We present recent evaluation figures concerning various IE tasks.

At first we summarise the detailed evaluation of patient status IE which is presented in [22]. Table 2 shows the precision, recall and the F-measure of correctly extracted descriptions of the anatomic organs *skin*, *neck*, *thyroid gland* and *limbs*, as well as statements about the patient *age*. We remind that during the analysis and recognition process, the status values are classified as *good*, *fair* and *serious*, which is visually reflected in the interface at Figure 3 by white, yellow and red colours of the respective fields. Green fields contain default values which are automatically filled in for missing obligatory attributes. We see that shallow analysis by regular expressions works relatively well, and the figures shown in Table 2 are comparable to the accuracy of the IE systems presented in section 2. The cases of incorrect analysis are due to more complex syntactic structures in the PR text which need to be analysed by a deeper syntactic parser and semantic processing. Further efforts are also needed to tackle complex language constructions including scope of quantifiers, temporal qualifications etc.

Training set	Skin	Neck	Thyroid gland	Limbs	Age
Precision	95,65	95,65	94,94	93,41	88,89
Recall	73,82	88,00	90,36	85,00	90
F-measure	83,33	91,67	92,59	89,01	89,44

Table 2: Precision, recall and f-measure of extracted patient characteristics

Another important IE task concerns the automatic assignment of disease codes using ICD-10 terms, in order to support the manual coding of patient information and the delivery of health management data. Diagnoses are declared in the PR section *(ii)* "diagnoses of the leading and accompanying diseases". This section contains enumeration of various disease names separated by the punctuation mark full stop. In other words, this section consists of separated, clearly disconnected phrases which are to be mapped to the ICD disease names. In general the diseases in section *(ii)* are not formulated according to the standardised ICD terms, sometimes the disease description might have no common words with the respective ICD term at all. Further mismatches between diseases descriptions in PR texts and the standardised ICD terms are discussed in [26].

The training set for this IE task contains 197 PRs, and the evaluation was performed for a test corpus of 250 unknown PRs. Almost all PRs in the test corpus cite more than one disease per patient, and the number of diseases ranges from 1 to 20. However, when numerous diseases are listed, their phrasal descriptions are often mixed in complex syntax groups; therefore we have performed the evaluation task for 20 test corpus subsets grouping PRs with equal number of diagnoses together. The evaluation results are illustrated by Figure 4 and Figure 5. The *x*-axis of both diagrams represents the twenty PR "test families" consisting of PRs with 1-20 diseases. Figure 4 summarizes the results from the PR perspective. For some PRs, part of the diagnoses are correctly encoded and others are wrong, so Figure 4 shows the ratio of PRs with fully associated diagnoses vs total number of PRs tested. The evaluation can be also made from the perspective of recognised individual diagnoses. Figure 5 presents the ratio of correctly associated codes for diagnoses compared to the total number of diagnoses included in the corresponding PR set.

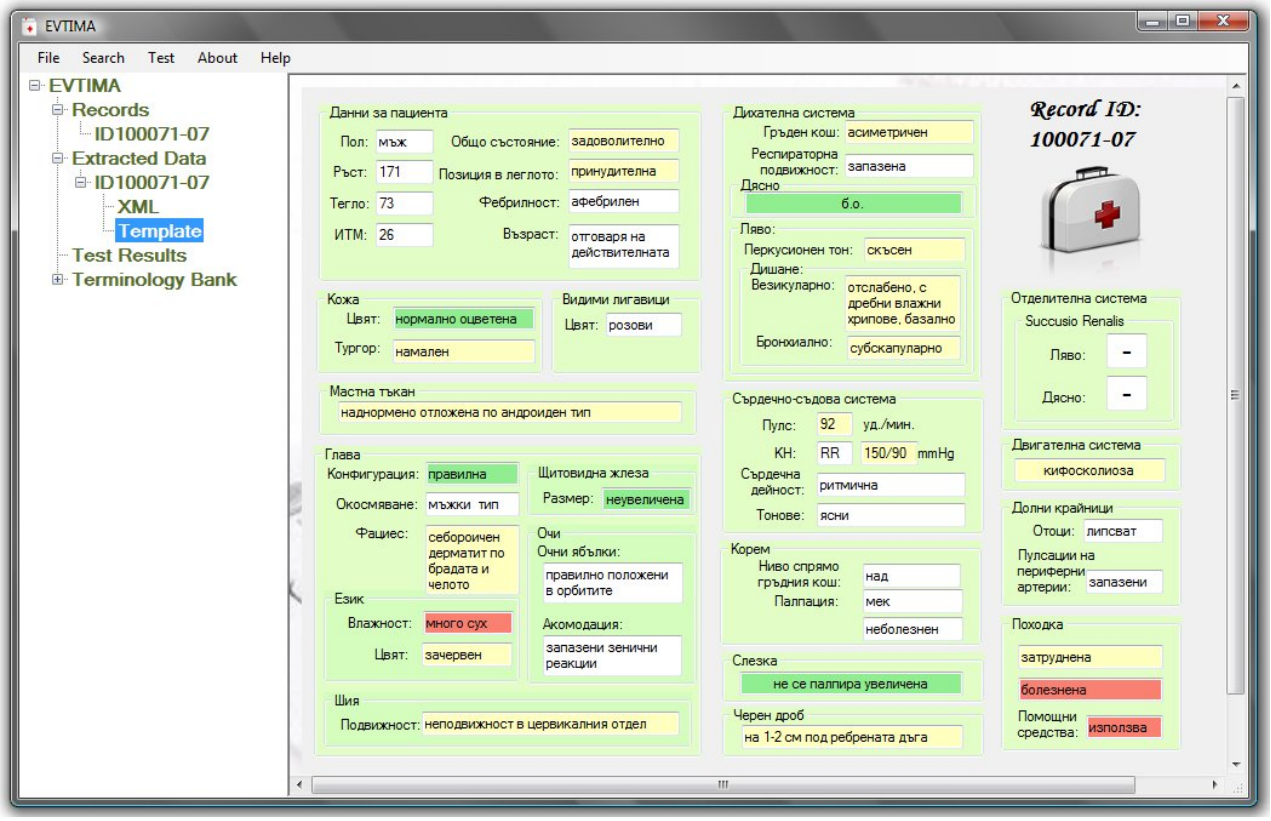


Figure 3: Structured description of patient status data supported by the IE prototype

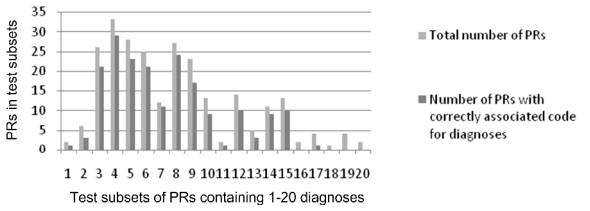


Figure 4. Percentage of PRs with correctly associated ICD-10 codes

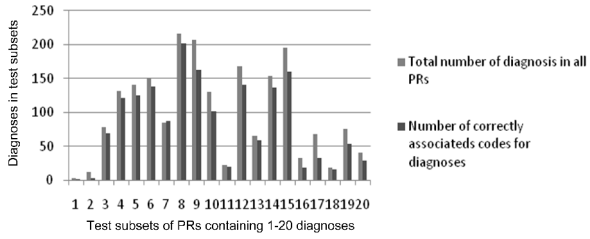


Figure 5. Percentage of diagnoses with correctly associated ICD-10 codes

The evaluation results at Figures 4 and 5 are further explicated at Table 3. Column 2 shows the performance assessment when the whole PR section *(ii)* is submitted to the assigning module as a single text fragment. Column 3 displays the results when the assignment is done phrase by phrase, i.e. every string between two separators in PR section *(ii)* is processed separately. The accuracy is higher when single phrases are considered.

Training set	Sets of diagnoses in PR section *(ii)*	Single diagnose
Precision	81,32	85,73
Recall	76,28	83,96
F-measure	78,72	84,84

Table 3: Precision, recall and f-measure of automatically assigned ICT-10 codes

5 Conclusion

The article describes current results in extraction of patient status data from medical text. It shows the complexity of medical text processing which is due to the complexity of the medical domain and the particularities of the medical texts written in specific, well-established style. The role of explicitly-declared domain knowledge is shown; it supports the information extraction algorithms by providing constraints and inference mechanisms. Construction of domain knowledge resources is a highly expensive, effort-consuming and tedious task, therefore we try to reuse available public resources as much as possible. At the same time the article illustrates the obstacles to build semantic systems in the medical domain: this requires much effort for construction of the conceptual resources as well as the lexicons and grammatical knowledge in case of text processing. Much knowledge in the medical documents is implicit, and its explication in the IE process is a real interpretation challenge.

Despite the difficulties, the paper shows that certain facts can be extracted relatively easily. These promising results support the claim that the Information Extraction approach is helpful for the obtaining of specific medical statements which are described in the PR texts. As future work, we plan to develop algorithms for discovering more complex relations and other dependences among the PR entities.

Acknowledgements

The research work presented in this paper is partly supported by grant DO 02-292/December 2008 "Effective search of conceptual information with applications in medical informatics", funded by the Bulgarian National Science Fund in 2009-2011. The primary PR anonymisation is done by the Hospital Information System of the University Specialised Hospital for Active Treatment of Endocrinology "Acad. I. Penchev", part of Medical University - Sofia.

References

[1] Hobbs, J. and E. Riloff (2010) Information Extraction. In: Indurkhya, N. and F. J. Damerau (Eds.), *Handbook of Natural Language Processing*, 2nd Edition, Chapman & Hall/CRC Press, Taylor & Francis Group.

[2] Cunningham, H. (2005) Information Extraction, Automatic. In: Brown K. (Ed.), *Encyclopedia of Language and Linguistics*, 14-Volume Set, Elsevier, Second edition.

[3] Demner-Fushman, D., W. Chapman and C. McDonald (2009) What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, Elsevier, 42(5), pp. 760-772.

[4] Harkema, H., A. Setzer, R. Gaizauskas, M. Hepple, R. Power, and J. Rogers (2005) Mining and Modelling Temporal Clinical Data. In *Proceedings of the 4th UK e-Science All Hands Meeting*, Nottingham, UK.

[5] Gaizauskas, R., M. Hepple, N. Davis, Y. Guo, H. Harkema, A. Roberts, and I. Roberts (2003) AMBIT: Acquiring Medical and Biological Information from Text. In S.J. Cox (ed.) *Proceedings of the 2nd UK e-Science All Hands Meeting*, Nottingham, UK.

[6] Damianos, L., J. Ponte, S. Wohlever, F. Reeder, D. Day, G. Wilson, and L. Hirschman (2002) MiTAP for Bio-Security: A Case Study. *AI Magazine*, AAAI, 23(4), pp. 13-29.

[7] Cancer Text Information Extraction System (caTIES), see https://cabig.nci.nih.gov/tools/caties, last visited August 2010.

[8] Health Information Text Extraction (HITEx), see https: //www.i2b2.org/software/projects/hitex/hitex_manual.html, last visited August 2010.

[9] Savova, G. K., K. Kipper-Schuler, J. D. Buntrock, and Ch. G. Chute (2008) UIMA-based Clinical Information Extraction System. *Proceedings of LREC-08 Workshop W16: Towards enhanced interoperability for large HLT systems: UIMA for NLP*, ELRA, May 2008.

[10] Baneyx, A., J. Charlet and M.-C. Jaulent (2005) Building Medical Ontologies Based on Terminology Extraction from Texts: Methodological Propositions. In S. Miksch, J. Hunter, E. Keravnou (Eds.) *Proc. of the 10th Conference on Artificial Intelligence in Medicine in Europe (AIME 2005)*, Springer LNAI 3581, pp. 231-235.
Ontology Development and Information Extraction tool, last visited August 2010 at https://cabig-kc.nci.nih.gov/Vocab/KC/index.php/Ontology_Development_and_Information_Extraction_(ODIE)

[11] Pestian J, C. Brew, P. Matykiewicz, DJ Hovermale, N. Johnson, K. B. Cohen, and D. Wlodzislaw (2007) A shared task involving multi-label classification of clinical free text. In: *ACL'07 workshop on biological, translational, and clinical language processing* (BioNLP'07), ACL, pp. 36–40.

[12] *Unified Medical Language System*, US National Library of Medicine, National Institutes of Health, last visited August 2010 at http://www.nlm.nih.gov/research/umls/

[13] Yangarber, R. (2001) Scenario Customization for Information Extraction. PhD thesis, New York Univ., NY.

[14] Savova, G., P. Ogren, P. Duffy, J. Buntrock and C. Chute (2008) Mayo Clinic NLP System for Patient Smoking Status Identification. *Journal of the American Medical Informatics Association*, 15(1), pp. 25-28.

[15] Roberts, A., R. Gaizauskas, M. Hepple and Y. Guo (2008) Combining terminology resources and statistical methods for entity recognition: an evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation* (LREC 2008), ELRA, May 2008.

[16] Novichkova, S., S. Egorov, and N. Daraselia (2003) MedScan, a natural language processing engine for MEDLINE abstracts. *Bioinformatics*, Oxford University Press, 19(13), pp. 1699–1706.

[17] BioPortal, http://bioportal.bioontology.org/visualize/13578/Diabetes_Mellitus, last visited April 2010.

[18] Angelova, G. (2010) Use of Domain Knowledge in the Automatic Extraction of Structured Representations from Patient-Related Texts. In: Croitoru, M., S. Ferre, and D. Lucose (Eds.): *Conceptual Structures: from Information to Intelligence*, Springer, LNAI 6208, pp. 14-27.

[19] Boytcheva, S., A. Strupchanska, E. Paskaleva, and D. Tcharaktchiev (2005) Some Aspects of Negation Processing in El. Health Records. In Paskaleva, E. and S. Piperidis (Eds) Proceedings of the International Workshop *Language and Speech Infrastructure for Information Access in the Balkan Countries* held in conjunction with RANLP-05, INCOMA, pp. 1-8.

[20] Boytcheva, S., I. Nikolova, E. Paskaleva, G. Angelova, D. Tcharaktchiev and N. Dimitrova (2009) Extraction and Exploration of Correlations in Patient Status Data. In: Savova, G., V. Karkaletsis and G. Angelova (Eds). *Biomedical Information Extraction*, Proceedings of the International Workshop held in conjunction with RANLP-09, INCOMA, pp. 1-7.

[21] Boytcheva, S., I. Nikolova, E. Paskaleva, G. Angelova, D. Tcharaktchiev and N. Dimitrova (2010) Structuring of Status Descriptions in Hospital Patient Records. In the *Proceedings 2nd International Workshop on Building and Evaluating Resources for BioMedical Text Mining,* associated to the 7th Int. Conf. on Language Resources and Evaluation (LREC-2010), ELRA, May 2010, pp. 31-36.

[22] Nirenburg, S., M. McShane, M. Zabludowski, S. Beale, C. Pfeifer (2005) Ontological Semantic Text Processing in the Biomedical Domain. *University of Maryland Baltimore County, Institute for Language and Information Technologies, Working Paper 03-05*. Available at http://naboo.ilit.umbc.edu/ILIT_Working_Papers/ILIT_WP_03-05_Biomed_Mesh.pdf, last visited August 2010.

[23] http://www.openehr.org, see *Clinical Models* and *Archetype Authoring,* last visited August 2010.

[24] Boytcheva S., G. Angelova, I. Nikolova, E. Paskaleva, D. Tcharaktchiev and N. Dimitrova (2010) EVTIMA: a System for IE from Hospital Patient Records in Bulgarian. In: Dicheva, D. (Ed.): *AI and Knowledge Societies: Learning, Sharing, Amplifying*, Proceedings of AIMSA-2010, the 14th Int. Conference on Artificial Intelligence – Methodology, Systems, Applications, Springer, LNAI, to appear in September 2010.

[25] Boytcheva S. (2010) Assignment of ICD-10 Codes to Diagnoses in Hospital Patient Records in Bulgarian. In: Alfred, R., G. Angelova and H. Pfeiffer (Eds.). *Proceedings of the International Workshop "Extraction of Structured Information from Texts in the Biomedical Domain" (ESIT-BioMed 2010)*, associated to the 18th Int. Conference on Conceptual Structures (ICCS-2010), Kuching, Sarawak, Malaysia, Published by MIMOS BERHAD, pp. 56-66.

Corpus and Web: Two Allies in Building and Automatically Expanding Conceptual Classes

Nicolas Béchet, Jacques Chauché, Violaine Prince and Mathieu Roche
Équipe TAL, Laboratoire d'Informatique de Robotique et de Micro-électronique de Montpellier
UMR 5506, CNRS, Univ. Montpellier 2
34 392 Montpellier Cedex 5 - France
E-mail : {bechet, chauche, prince, mroche}@lirmm.fr

Keywords: terminology, conceptual classes, expansion, web validation

Received: March 30, 2010

In this paper, the approaches to building and expanding conceptual classes are presented. The classes are built with syntactic and semantic information provided by a corpus. Then, expansion is addressed by using the objects of syntactic relations found in the corpus. Relations between classes are thus designed. They are called induced relations. Then we use objects of induced syntactic relations (called complementary objects) to expand conceptual classes. We propose an automatic experimental protocol to measure the relevance of the provided concepts. The protocol helps alleviating the judgment effort of a human expert. The expansion method is evaluated and mixed in order to provide the most reliable technique in expanding conceptual classes.

Povzetek: V prispevku je opisan postopek izgradnje konceptualnih dreves s pomočjo spleta in korpusov.

1 Introduction

Several NLP (Natural Language Processing) applications use terminology. The latter can be defined as the study of technical terms of a field, as well as their signification. Two kinds of terminology studies can be proposed: one which is called ‘semasiologic’ and the other, ‘onomasiologic’. The first focuses on term signification to study sense. The second proposes to start from the conceptual level, and attaches terms as linguistic instantiations of concepts.

Concepts have born several definitions. One of the most general ones describes a concept 'as the mind representation of a thing or an item' [Desrosiers-Sabbath, 1984]. Within a given domain such as ours, which deals with ontology building, semantic web, and computational linguistics, it seems quite appropriate to stick to the Aristotelian approach of a concept and see it as a set of knowledge gathering of common semantic features. Features choice and gathering design are dependent upon criteria that we will try to explain hereafter.

Starting from concepts needs to have, at start, an extensive representation of the terminology associated with each concept. Thus the onomasiologic approach better deals with restricted thematic fields (e.g. 'meteorology', 'tomato growth in agriculture', etc.). Concepts are first established and agreed upon, and terminology is associated with concepts. Afterwards, all types of processes could be undertaken with such a knowledge base. This approach outcomes are tied with the domain closure.

In an open, or yet incompletely browsed domain (such as Web pages might induce), onomasiology is less capable. Thus such cases are preferably investigated with semasiologic tools. The existing data are analysed and bring forth term which significations are otherwise arranged in order to create gatherings. Both concepts and terminology are incrementally enhanced, and shaping is a loop process with an important feedback. Very obviously, Semantic Web is better approached by the semasiologic method. However, such a method creates new problems as side effects. If onomasiology is better served by restricting the field, semasiology performs better when restricting the task. Tasks involve information retrieval (IR), text indexing, question answering, summarizing, translating, etc...Thus, the terminology built for a given task must not be used in other tasks without some care or partial rebuilding [Roche, 2005].

In this paper, we propose to build conceptual classes, expand them, and directly attach terminology under the framework of a semasiological process. The restrictions of semasiology are however alleviated by the fact that NLP techniques for classes building and term attachments are used on both domain corpora and cross-domains Web pages. Naturally, the most fitting task is IR, but to an extent, other tasks could be addressed by tuning the building and expansion process.

First, we uggest building specific conceptual classes by focusing on knowledge extracted from corpora. Conceptual classes are shaped through the study of syntactic dependencies between corpus terms (as described in section 2). Dependencies tackle relations such as Verb/Subject, Noun/Noun Phrase Complements, Verb/Object, Verb/Complements, and sometimes Sentence Head/Complements. In this paper, we focus on

the Verb/Object dependency, because it is a good representative of a field. For instance, in computer science, the verb *to load* takes as objects the nouns of the conceptual class *software* [L'homme, 1998]. This feature also spreads to '*download*' or '*upload*' which have the same verbal root.

Corpora are rich or in which mining for terminological information is fruitful. A terminology extraction of this kind is similar to a Harris-like distributional analysis [Harris, 1968] and literature displays an abundant set of works undergoing a distributional analysis to acquire terminological or ontological knowledge from textual data (e.g [Bourigault and Lame, 2002] for law, [Nazarenko et al., 2001], [Weeds et al., 2005] for medicine).

After building conceptual classes, we describe an approach to expanding the classes by using the corpus to discover new terms (in section 3). These terms are then ranked and proposed to an expert in a sorted list.

2 Conceptual classes building

2.1 Principle

A class can be defined in our approach as a gathering of terms having a common field. In this paper, we focus on objects of verbs judged to be semantically close regarding a measure. Thus, these objects are considered as instances of conceptual classes.

The first step of building conceptual classes consists in extracting Verb/Object syntactic relations as explained in the following section.

2.2 Mining for verb/object relations

Our corpora are in French since our team is mostly devoted to French-based NLP applications. However, the following method is portable to any other language, provided that a quite reliable dependency parser is available.

In our case, we use the SYGFRAN parser developed by [Chauché 1984]. As an example, in the French sentence "Thierry Dusautoir brandissant le drapeau tricolore sur la pelouse de Cardiff après la victoire." (translation : 'Thierry Dusautoir brandishing the three colored flag on Cardiff lawn after the victory'), there is a syntactic relation verb-object: "verb: brandir (to brandish), object: drapeau (flag)", which is a good candidate for retrieval.

The second step of the building process corresponds to the gathering of common objects related to semantically close verbs.

Semantic Closeness Assumption
The underlying linguistic hypothesis is the following: **Verbs having a significant number of common objects are semantically close**.

To measure closeness, the ASIUM score [Faure and Nedellec 1999], [Faure 2000] is used. This type of work is akin to distributional analysis approaches such as [Bourigault et al. 2002].

Therefore, conceptual classes instances are the common objects of close verbs, according to the ASIUM proximity measure.

3 Expanding conceptual classes

3.1 Principle

In order to expand conceptual classes, the main difficulty is to obtain new terms which can be instances of a conceptual class. The basic idea here is to use the corpus itself to acquire new instances with the same approach as in building classes (see 2.1). As it was said before, the process admits as instances of a class the common objects of close verbs. Thus expanding conceptual classes is a two steps procedure:

1) Retrieving **complementary objects (**to be explained hereafter)

2) Asserting the relevance of complementary object as a possible instance of a concept.

Both steps are introduced in the next sub-section.

3.2 Step 1: Extraction of object features

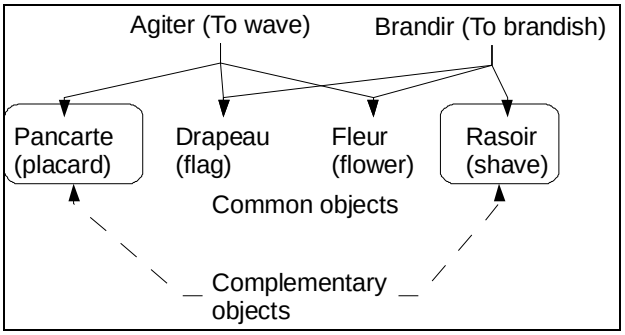


Figure 1: Complementary and Common objects of verbs to *wave* and to *brandish*.

Two types of objects appear as an output of the preceding action: Objects that are **common** to two given verbs, and objects that are called **complementary** since they appear in association with one but not with the other. In Figure 1, the considered pair of verbs is ('to *brandish'*, 'to *wave*'). Their common objects are in the pair ('*flag'*, *'flower'*) (either given from start, or already retrieved from a corpus by a previous step of the process). "*Flag"* and *flower*" are instances of a concept "*symbol*", the gathering class of '*brandish'* and '*wave'* objects. Their complementary objects, i.e., objects that appear either with one or the other, are ('*placard*', '*shave*') where '*placard*' is a retrieved object of '*wave*', and '*shave*' is a retrieved object of ' *brandish*'.

To measure the quality of our expansion approach, we propose to answer the question: Are complementary objects relevant instances of the conceptual class defined by common objects? To answer this, we have several ways to provide an evaluation protocol, and this paper will show different methods. But first, a human evaluation determines what is likely and what is not.

3.3 Step 2: Human evaluation of the quality of complementary objects

The procedure is the following.

A few concepts are selected, since they are addressed by a given corpus. For instance, in Figure 1, concept '*Symbol'* is chosen.

Conceptual classes are built with verb common objects as explained in 3.1. Here, '*Symbol'* is populated with '*flag'* and '*flower'*.

Then complementary objects are considered, and human judges have to evaluate to which extent there terms are relevant instances of the concerned concept. In the example, '*placard'* and '*shave'* are judged as possible instances of '*Symbol'*.

Evaluation consists in selecting a figure associated to one of the following propositions:

- 2: Completely relevant
- 1: Possibly relevant
- 0: Not relevant
- N: No opinion

The principle underlying this method is the following: We assume that complementary objects retrieval is a good way to discover new terms of conceptual classes because some of complementary objects are possible instance of concepts.

Human evaluation was undertaken (see experiment in section 5.1) to assert the likelihood of such an assumption. Complementary objects could very possibly be of no use for conceptual classes expansion.

Once the benefit of such an assumption acknowledged, however accurate, human evaluation might prove to be tedious, time consuming and difficult to undertake (as reported in the experiment). Thus, we have designed a **filtering procedure** that automatically sorts complementary objects by decreasing relevance. This procedure introduces a ranking function, and relies on a pre-formal data structuring called **Induced Syntactic Relation** (ISR), presented in the next section.

4 Induced syntactic relations (ISR): definition, and relevance

4.1 Defining induced relations

According to the *semantic closeness assumption*, 'to *wave'* and 'to *brandish'* (in Figure 1) are supposed to be rather close (and closeness is measured) since they have common objects. An important add-on of our approach is to **assess the status of complementary objects**. More precisely, we call **induced syntactic relation (ISR)** the following relation:

Definition:
Let v_1 and v_2 be two semantically close verbs. Let V/O be a Verb-Object relation.
Let CO_1 be the complementary object of v_1. V/O (v_1,CO_1) is true (there is a Verb/Object relation between them).
Let CO_2 be the complementary object of v_2. V/O (v_2,CO_2) is true.
V/O(v_1, CO_2) and V/O(v_2, CO_1) are the syntactic relations induced by the semantic closeness of v_1 and v_2. They are proposed as new knowledge, and their validity is evaluated.

In Figure 1, '*Placard'* and '*Shave'*, complementary objects need to be validated as possible instances of '*Symbol'*. Presently, object '*Shave'* is not a valid instance of the concept "*Symbol'*. As a filtering procedure, the automatic procedure will examine the two following induced syntactic relations:
To brandish a placard
To wave a shave

If these utterances are to be considered, by a way or another, as likely, then this is a good clue to consider '*placard'* and '*shave'* as possible instances of '*Symbol'*. So, induced syntactic relations (called ISR from now on) relevance needs to be be defined and assessed.

ISR Relevance Assumption
Let v_1 and v_2 be two semantically close verbs.
Let (KO_1, KO_2, …, KO_m) be their common objects. By definition, V/O(v_1, KO_j) is true, and V/O(v_2, KO_j) is true, for j={1,..,m}. Let K_a be their common concept (the KO_j are instances of K_a). K_a is assumed to be the conceptual class of v_1 and v_2.
CO_1 and CO_2 are possible instances of K_a if V/O(v_1, CO_2) and V/O(v_2, CO_1) are relevant.

In other words, we suppose that the complementary object is a *valid instance of the concept* defined by the common objects of the two verbs if an IRS is *relevant.* By the result presented in section 5, we have proved that our hypothesis is relevant.

Relevance is the first step before assessing complete validity. Next section shows how it is dealt with.

4.2 Ranking functions

ISR can be submitted to human approval, as objects could be submitted (see 3.3), but this is not the point: ISR has been introduced in order to pre-filter possible objects, and not to add complexity. So the best method was to examine functions that might rank ISR according to their assumed relevance [Béchet et al., 2009a].
Therefore, we need to describe the three following items:
1-How do we define the semantic relevance of a complementary object to a conceptual class
2-How this semantic relevance is computed: The methods and measures that have been chosen to achieve computation
3-Last, how IRS has been ranked according to each measure.

4.2.1 Semantic relevance definition

Definition
Let v_k be a verb.
An item I_n is assumed to belong to the conceptual class of v_k objects, if:
It has appeared as such in a corpus and has been retrieved, i.e. V/O (v_k, I_n) is satisfied.
I_n has not been retrieved but is a semantically relevant object of v_k.

4.2.2 Semantic relevance measuring process

Semantic relevance is measured as such:
1-A semantic representation of the original Verb/Object relation is computed for complementary objects. This representation is based either on a vector model, or is a digital output representing a statistical information. Both measures are detailed hereafter.
2-The same semantic representation is produced for the IRS.
3-A distance measure (or more precisely a closeness measure) is then used to assess the possible similarity between the IRS and the original relation.
3-The expected result is: **The closest the IRS and the original relation are, the more relevant to the verb, is the CO**.
For instance, in Figure 1, we measure the proximity between both syntactic relations "*to wave a placard*" (original relation) and "*to brandish a placard*" (induced relation).

4.2.3 Semantic measures

Two semantic measures belonging to two semantic modelling paradigms have been determined: *Semantic Vectors*, and Corpus co-occurrence also called *Web Validation*. Both are briefly described hereafter.

Semantic Vectors (SV): Contribution of a Vector Model to the Verb Object Relation Representation

A Semantic vector is built by projecting one or many terms on a close space vector of 873 concepts. Concepts are taken out of an ontology defined in the French Larousse Thesaurus [Larousse, 1992], a Roget-based dictionary indexing all language entries with one or several items taken from the 873 concepts ontology. For instance, the French verb "*brandir*" (to *brandish*) is associated with the concept of "*agitation*" and the noun "*drapeau*" (*flag*) is indexed by the concepts of "*paix (peace), armée (army), funérailles (funerals), signe (sign)",* and *"cirque (circus)*". The ISR vector is the result of a linear combination between verb and object vectors. Coefficients take into account the syntactic structure of the relation [Chauché, 1990]. The vector closeness is finally evaluated by a **cosine** computation between both semantic vectors (vector of the original and the induced relations).

The Web Validation (WV)

The second approach method uses the Web to measure the dependency between a verb and an object of an IRS. It is based on Turney's method [Turney, 2001] summarized as follows: A string is submitted as a query to a search engine. The number of returned results defines the dependency measure. In addition, different statistical measures such as Mutual Information [Church and Hanks, 1990] or Dice's coefficient [Smadja and al., 1996] are employed. With these measures, one can weight the IRS relevance, depending on the verb and the object composing the relation. Here, only Mutual Information is run on experiments, since this measure performed the best in previous works. The Mutual Information measure, adapted for this task, is defined as:

$$MI(v,o) = \frac{nb(v,o)}{nb(v)nb(o)}$$

where nb(v), nb(o), and nb(v,o) are respectively the number of returned results by the search engine with the submission of the verb *v*, the object *o*, and the syntactic relation *vo*. The Web validation process uses external knowledge to measure the relevance of a candidate to a concept. Thus, this validation allows for a more global evaluation of relevance for the final concepts.

Combining Measures

Combining measures has been contemplated in order to improve accuracy. Two different procedures have been defined.
- Combination 1: The first combination introduces a scalar k ∈ [0, 1] to reinforce one approach or the other. The results obtained with SV (Semantic Vectors) and WV (Web Validation) methods are first normalized. Next, both results (the figures are named SV and WV after their methods) are combined with the following formula for a syntactic relation *c*:

$$combined\_score_c = k * SV + (1 - k) * WV$$

- Combination 2: The second combined system between *SV* and *WV* has been computed. First, syntactic relations are ranked with *SV.* Then, the *n* first syntactic relations obtained with *SV* are ranked with *WV*. This second process (*WV* applied on the ranked relations with *SV*) enables to accurately sort these syntactic relations. Thus, with this adaptive combination, *SV* provides a global selection using semantic resources, and *WV* handles this first selection.

The next section presents experiments we made to measure the quality of these validation methods.

5 Experiments

5.1 Experimental setting and goals

We use a French corpus from Yahoo's site (*http://fr.news.yahoo.com/)* composed of 8,948 news

(16.5 MB) from newspapers (called corpus *T)*. Experiments are performed on 60,000 induced syntactic relations [Béchet and al. 2009b]. We have selected *manually* five concepts. Instances of these concepts are the common objects of verbs defining the concept[1]. The French selected concepts are presented in Table 1.

Concepts	Organisme /Administration (Civil Service)	Fonction (work)	Objets symboliques (symbols)	Sentiment (feeling)	Manifestation de protestation (protest)
Instances	parquet (prosecution)	négociateur (negotiator)	drapeau (flag)	mécontentement (discontent)	protestation (remontrance)
	mairie (city hall)	cinéaste (filmmaker)	fleur (flower)	souhait (wish)	grincement (grind)
	gendarme (policeman)	écrivain (writer)	spectre (specter)	déception (disappointment)	indignation (indignation)
	préfecture (prefecture)	orateur (public speaker)		désaccord (disagreement)	émotion (emotion)
	pompier (fireman)			désir (desire)	remous (swirl)
	O.N.U. (U.N.)				tollé (collective protest)
					émoi (commotion)
					panique (panic)

Table 1. The five selected concepts and their instances.

The goal of these experiments are the following:
1- Evaluating the consistency of a procedure manning conceptual classes with complementary objects: A human evaluation of the complementary objects quality has been conducted as a feasibility study
2- Evaluating the quality of the filtering procedure based on semantic measures. The aim is to select the best complementary objects before giving them to a human expert. Thus, CO are ranked according to SV, WV and combined measures, as presented in section 4.2. Then the quality of the resulting lists of ranked objects is measured with experimental protocols presented in following sub-sections.

5.2 Human evaluation of the quality of complementary objects

Eight human judges have undergone the following protocol: An evaluation form was available on a specific Web page. This form allowed them to judge whether given terms can be instances of a given concept as explained in section 3.

Figure 2 gives a screen-copy of the submitted form. Resulting scores can be computed by submitting the different results to a voting system. So a term is positive if a percentage of *p* judges estimate the term to be relevant. A relevant term for a judge gets the value 1 or 2. We fix *p* at 75%.

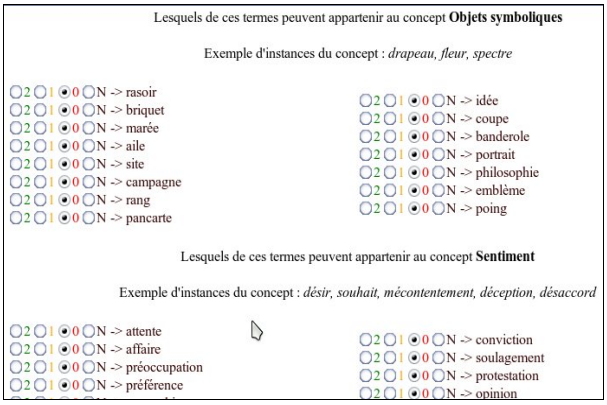


Figure 2. Screen-copy of the French form.

We obtain an accuracy score definition as the number of complementary objects divided by the number of relevant term according by the judges. The obtained score obtained is **0,14** (75 relevant terms divided by 553 complementary objects). This result shows the interest of complementary objects as instances of conceptual classes. It also shows that the number of potential candidates is high, and that an automatic procedure needs to be performed, as an aid to experts.

5.3 Evaluation of induced syntactic relations

We focus in this section on the quality of the ranking function presented in section 4.2. Here, the asset is to assert the reliability of the process, as a 'good' filter for sorting complementary objects.

Thus, we present two different evaluation protocols: A human and an automatic.

5.3.1 Evaluating relevance

Automatic Evaluation (AUTO)

The method we used to automatically measure the quality of IRS focuses on the use of a second French corpus, bigger than the first one, created from the French newspaper Le Monde (called corpus V). It contains more than 60,000 news (123 MB). It helps to determine if those ISR found in corpus T are relevant. Corpora T and V come from the same field. Thus, the first step is to automatically recover the ISR of corpus T existing in corpus V. If an ISR of corpus T appears in corpus V, it is marked down as positive (existing as a real object for the other verb), else it is negative.

Let us note that negative relations can be false negatives. Actually, a syntactic relation not found in the corpus V is not inevitably a negative relation. In addition, a relevant complementary object from an induced syntactic relation can be an irrelevant instance for a concept which has been defined 'on the spot', after the features of existing common objects. Therefore, a manual evaluation protocol, relying on human approval, is needed.

Human Evaluation (VOTING)

The human evaluation is the same as presented in subsection 5.2, except that we measure here the quality

1 From those concepts which have obtained an Asium score higher than 0.7 [Faure, 2000]

of validation approaches and not the quality of complementary objects. Thus, a relevant term for a judge gets the value 1 or 2.

The notion of 'relevant term' being defined for both AUTO and VOTING protocols, the quality of the ranked relations list is evaluated with ROC curves.

5.3.2 Evaluating ranking functions

ROC curves (Receiver Operating Characteristic), detailed in [Ferri02] are often used in medicine to evaluate the validity of diagnosis tests. ROC curves show in X-coordinate the rate of false positives (in our case, the rate of irrelevant IRS) and in Y-coordinate the rate of true positives (rate of relevant IRS). The surface under the ROC curve (AUC - Area Under the Curve), can be seen as the effectiveness of a measurement of interest. The criterion related to the AUC is equivalent to the statistical test of Wilcoxon-Mann-Whitney [yan03].

In the case of the ISR ranking using SV and WV measurements, a perfect ROC curve corresponds to a configuration where all relevant ISR are at the beginning of the list and all irrelevant syntactic relations at the end. This situation corresponds to AUC=1. The diagonal corresponds to the performance of a random system, progress of the rate of true positives being accompanied by an equivalent degradation of the rate of false positives. This situation corresponds to AUC=0.5. Figure 3 is an instance of a ROC Curve with in diagonal a random system distribution.

If the ISR are ranked by decreasing interest (i.e. all relevant ISR are after the irrelevant ones) then AUC=0.

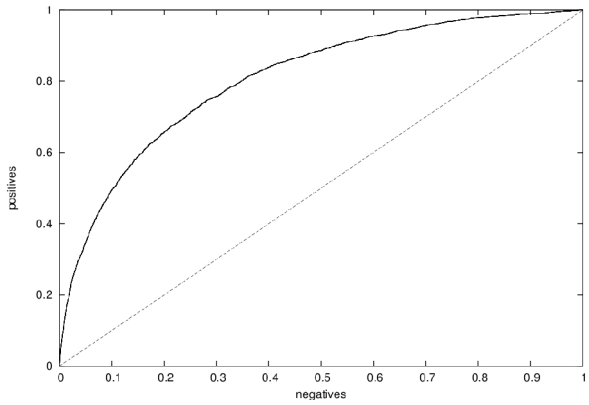


Figure 3: Example of a ROC Curve with a random distribution in diagonal.

An effective measurement of interest to order ISR consists in obtaining an AUC the highest possible value. This is strictly equivalent to minimizing the sum of the rank of the positive examples.

The advantage of the ROC curves comes from its resistance to imbalance (for example, an imbalance in number of positive and negative examples). The interest of this measure is developed in [Roche and Kodratoff, 2006].

Term	***Manual validation***
Conviction (conviction)	+
Opinion (opinion)	+
Préférence (preference)	-
Attente (waiting)	-
Colère (anger)	+

Table 2: Example of evaluation of terms for French concept "sentiment" (*feeling*).

Table 2 presents an example of ranked terms by the second combination approach. Terms are rated by a manual evaluation for the French concept "sentiment" (*feeling*). The resulted ROC Curve is given in figure 4. We finally get an AUC of 2/3 with this example (in blue in figure 4).

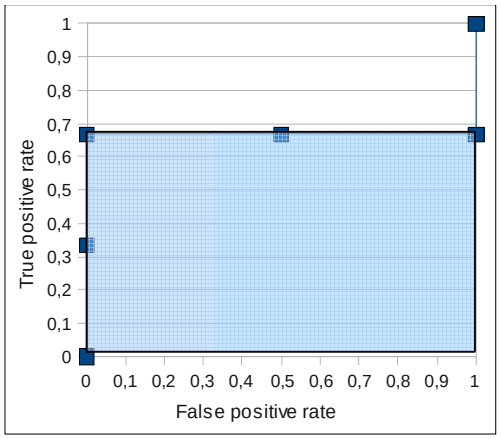


Figure 4: ROC Curve resulting of the example in Table 2

5.3.3 Experimental results

SV, WV and combined approaches propose to validate induced semantic relations by offering sorted lists of relations. The number of induced syntactic relations is taken into account by introducing a threshold of considered relations.

A fixed threshold at 100 means that AUC is computed for the only 100 first ranked (with our validation approaches) induced syntactic relations. Table 3 presents the obtained AUC for both approaches "Web validation" and "combination 2". This table compares manual and automatic evaluations. We present AUC obtained for the automatic validation by using a second validation corpus. We also present results obtained with the manual evaluation by using the voting system. A positive relation is validated if 75% of the judges give the score of 2. Better results in the Table 3 are given by "combination 2".

The manual evaluation gives good results for the second combination (AUC up to 0.83) with the first induced syntactic relations (i.e. small thresholds). Results are fair up to a threshold of 350 to finally decrease with all the induced syntactic relations (AUC close to a random distribution 0.5). Thus, we cannot provide an expert with a complete sorted list of relations but only with a selected part. So we favor the precision and the quality of the sorted list by reducing the number of possible instances to a concept.

	Web Validation		***Combination 2***	
	Vote	**Auto**	**Vote**	**Auto**
Threshold	***AUC***		***AUC***	
50	0,64	0,59	**0,81**	**0,90**
100	0,50	0,60	**0,83**	**0,87**
150	0,62	0,66	**0,80**	**0,84**
200	0,61	0,65	**0,76**	**0,79**
250	0,56	0,66	**0,71**	**0,75**
300	0,51	0,65	**0,70**	**0,74**
350	0,57	0,67	**0,69**	**0,75**
400	0,59	0,67	**0,67**	**0,74**
450	0,61	0,67	**0,65**	**0,71**
500	0,56	0,68	**0,57**	**0,70**
550	**0,52**	**0,69**	**0,52**	**0,69**

Table 3: AUC scores for the Web validation and the combination 2, with the manual (Vote) and the automatic evaluation.

We also compare the manual evaluation and the automatic scores given in Table 3. Results are similar for both evaluations. Actually, results of combination 2 for both evaluation protocols are relevant for small thresholds and decrease with all relations. Web Validation gives regular results close to 0.60 with the manual evaluation and 0.65 with the automatic.

As a conclusion about IRS relevance measure, we can say that:

Combination 2 has the best scores for all threshold values, thus is the best semantic measure among the studied ones

The first 150 ranked IRS have an AUC of and over 0,80, whatever the evaluation method is, so this means that if we retrieve the first 150 IRS with combination 2, these are a valuable material for retrieving complementary objects being possible instances of our conceptual classes, as termed in the IRS relevance assumption.

However the obtained scores are too highly rated with the automatic evaluation. These differences can be explained by the fact that two aspects are addressed by the protocols. The manual protocol addresses the relevance of a given term as an instance of a concept. The automatic protocol tries to measure the relevance of a syntactic relation built with a verb and a complementary object. These close tasks have not the same goals. Actually, automatically measuring the quality of a terms belonging to a concept is a more difficult task than measuring the quality of a syntactic relation.

6 Conclusion

This paper aims at showing and evaluating procedures that help building and expanding conceptual classes. Those tasks are quite common in terminology and ontology design. As several others, this research mines textual knowledge to do so. However, unlike others, NLP knowledge is not restricted to lexical relations but engulfs syntactic knowledge, focusing here on the verb-object dependency as a valuable relation for building and expanding conceptual classes.

One of the original features is to build classes by using common objects of semantically close verbs in a given corpus. Semantic closeness is measured with the ASIUM measure. Then, classes are expanded with complementary objects, being those ‘left over’ data, since they are not common objects.

This information source has proved to be interesting through a feasibility study conducted with a human evaluation protocol (see section 5.2). However, since it is a very abundant set of knowledge, ploughing it manually must not be considered as a possible task, since it is tedious, and time and effort consuming.

This consideration has led us to contemplating an automatic filtering procedure that would rank objects according to their relevance to the conceptual classes. Several methods could have been performed, however, we wanted here to pursue further in the light of the verb object relation, by studying the consistency of what we called the Induced Syntactic Relation, i.e., when the ‘local’ (complementary) object of a verb is exported to close verb. We made the assumption that if that Induced Syntactic Relation was to be relevant then this complementary object should play the same role as a common object, and thus should be a possible instance of the conceptual class (IRS relevance assumption).

So the problem shifted from populating a conceptual class towards measuring and asserting semantic relevance of a verb-object relation.

The second original feature of this paper was to unite both Web-based and Corpus-based techniques in order to fetch as many possible occurrences of an IRS, or to assume its inconsistency when not finding any clue about it. Among the several possible semantic models for corpora data, semantic vectors were chosen since they mix syntactic and semantic representations in a same numeric structure. Also among Web queries measures, it is Turney’s approach that has been chosen. Experiments have shown that a particular combination of measures (combination 2) proved to be the most efficient. Measures and evaluation protocols have shown that the first 150 relations, ranked with combination 2, have the best AUC scores (over 0,80), which means that they are utterly reliable.

Although very concluding, these filtering methods could be improved, at least by introducing contextual information. For Web Validation, context could be introduced in the search engine queries. With the semantic vectors approach, contextual vectors can be used. These vectors take into account the morpho-syntactic structure of the sentence containing the terms to be validated. Thus, combination 2 results might hopefully increase, either by increasing the number of acceptable IRS (AUC over 0,8) or by improving the AUC value for a fixed number of IRS.

Anyway, the IRS relevance assumption not being invalidated by experiments, we think that other dependency relations could also be contemplated: Why not the Verb-Subject relation, or the Verb-other Complements one, depending on the type of terminology or ontology one needs to retrieve. Other works have provided results in ontology populating by using the Subject-Verb-Object relation pattern in a specialized domain (e.g., Makki et al. 2009). Here, we go further by assuming ‘non retrieved’ but likely relations and ranking

them. This tends to show that NLP techniques have still a lot to offer to Web Semantics, and Ontology Design and Population.

7 References

[1] Béchet, N., Roche M., Chauché J. (2009a) A Hybrid Approach to Validate Induced Syntactic Relations. In *AINA Workshops 2009*, pp. 727-732.

[2] Béchet, N., Roche M., Chauché J. (2009b) Towards the Selection of Induced Syntactic Relations. In *ECIR'09* (poster), pp. 786-790.

[3] Bourigault D., Lame G. (2002) Analyse distributionnelle et structuration de terminologie. Application à la construction d'une ontologie documentaire du Droit, in *TAL*, pp. 43-51.

[4] Chauché, J. (1984) Un outil multidimensionnel de l'analyse du discours. In *Proceedings of COLING*, Standford University, California, pp. 11–15.

[5] Chauché, J. (1990) Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance. In *TA Information*, pp. 17–24.

[6] Church, K. W. and Hanks P. (1990) Word association norms, mutual information, and lexicography. In *Computational Linguistics*, Vol. 16, pp. 22–29.

[7] Desrosiers-Sabbath (1984) *Comment enseigner les concepts* - Sillery: Presses de l'Université du Québec.

[8] Faure D., Nedellec C. (1999) Knowledge Acquisition of Predicate Argument Structures from Technical Texts Using Machine Learning: The System ASIUM. In *EKAW 1999*, pp. 329-334.

[9] Ferri, C., Flach P., and Hernandez-Orallo J. (2002) Learning decision trees using the area under the ROC curve. In *Proceedings of ICML'02*, pp. 139–146.

[10] Harris, Z. (1968) *Mathematical Structures of Language*, New-York, John Wiley & Sons.

[11] Larousse, T. (1992) *Thésaurus Larousse - des idées aux mots, des mots aux idées*. Ed.Larousse, Paris.

[12] L'Homme M. -C. (1998) Le statut du verbe en langue de spécialité et sa description lexicographique. *Cahiers de Lexicologie,* 73, pp. 61-84.

[13] Makki, J., Alquier A-M., Prince V. (2009) Ontology Population via NLP Techniques in Risk Management. *International Journal of Humanities and Social Sciences* **3**, 3, pp. 212-217.

[14] Nazarenko A., Zweigenbaum P., Habert B, Bouaud J. (2001) Corpus-based Extension of a Terminological Semantic Lexicon. In *Recent Advances in Computational Terminology*, pp. 327-351.

[15] Smadja, F., McKeown K. R., and V. Hatzivassiloglou (1996) Translating collocations for bilingual lexicons : A statistical approach. *Computational Linguistics*, Vol. 22, No. 1, pp.1–38.

[16] Roche C. (2005) Terminologie et ontologie. *Revue Langages,* No. 157, Éditions Larousse, mars 2005, pp. 48-62.

[17] Roche M., Kodratoff Y. (2006) Pruning Terminology Extracted from a Specialized Corpus for CV Ontology Acquisition. In *Proceedings of onToContent'06 workshop (Ontology content and evaluation in Enterprise) - OTM'06*, Springer-Verlag, LNCS, october 2006, Montpellier, France, pp.1107-1116.

[18] Turney, P. (2001) Mining the Web for synonyms : PMI– R versus LSA on TOEFL. In *Proc. of ECML*, LNCS, 2167, pp. 491–502.

[19] Weeds J, Dowdall J., Schneider G., Keller DaB., and D.J. (2005) Weir Using distributional similarity to organise biomedical terminology. *Terminology*, Vol 11, No. 1, pp. 107-141.

Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web

Vladimir A. Fomichov
Department of Innovations and Business in the Sphere of Informational Technologies
Faculty of Business Informatics State University – Higher School of Economics
Kirpichnaya str. 33, 105679 Moscow, Russia
E-mail: vfomichov@hse.ru, vfomichov@gmail.com

Keywords: semantic web of a new generation, multilingual semantic web, semantics-oriented natural language processing, semantic representation, text meaning representation, theory of K-representations, SK-languages, semantic annotation, algorithm of semantic-syntactic analysis, bioinformatics, bioNLP

Received: March 15, 2010

A comprehensive theoretical framework for the development of a Semantic Web of a new generation, or of a Multilingual Semantic Web, is outlined. Firstly, the paper grounds the possibility of using a mathematical model being the kernel of the theory of K-representations and describing a system of 10 partial operations on conceptual structures for building semantic representations (or text meaning representations) of, likely, arbitrary sentences and discourses in English, Russian, French, German, and other languages. The possibilities of using SK-languages defined by the theory of K-representations for building semantic annotations of informational sources and for constructing semantic representations of discourses pertaining to biology and medicine are illustrated. Secondly, an original strategy of transforming the existing Web into a Semantic Web of a new generation with the well-developed mechanisms of understanding natural language texts is described. The third subject of this paper is a description of the correspondence between the inputs and outputs of the elaborated algorithm of semantic-syntactic analysis and of its advantages; the semantic representations of the input texts are the expressions of SK-languages (standard knowledge languages). The input texts can be the statements, questions, and commands from the sublanguages of English, Russian, and German. The algorithm has been implemented by means of the programming language PYTHON.

Povzetek: Predstavljena je formalizacija multilingualnega semantičnega spleta.

1 Introduction

Due to the stormy growth of the Internet, a huge number of the projects realized in the 2000s in Life Sciences and Health Care, and due to several other factors the users of the Internet and of specialized computer networks have received the access to an immense variety of information sources in many natural languages and to a number of knowledge bases formed with the help of ontology languages, first of all, the language OWL.

With respect to this situation, many specialists in various countries suppose that the only real way of realizing an effective interaction of people throughout the world with these natural language (NL) based information sources and with knowledge bases is the development of appropriate NL-interfaces and semantics-oriented advanced search systems.

In favour of this conclusion says the successful experience of designing in the 2000s several NL-interfaces to databases (see, e.g., [19]) and NL-interfaces to Semantic Web (SW) data repositories (see, e.g., [5, 6, 16]).

Since Web-based informational sources are formed with the help of many natural languages, it is high time to intensively develop the theoretical foundations of multilingual, semantics-oriented information retrieval on the Web and to expand the foundations of designing (for many natural languages) the NL-interfaces to SW data repositories.

On one hand, it is one of the central tasks for Web science, defined in [3] as the science of decentralized information systems. On the other hand, it seems that this task is a part of more general, large-scale problem – the problem of developing a Semantic Web of a new generation.

During several last years, it has been possible to observe that the achieved state of Semantic Web and a state to be relatively soon achieved are considerably different from the state of affairs outlined as the goal in the starting publication on Semantic Web by T. Berners-Lee, J. Hendler, and O. Lassila [2].

The principal reason for this conclusion is the lack of large-scale applications implemented under the framework of Semantic Web project. This situation is implied by the lack of a sufficiently big amount (of "a critical mass") of formally represented content conveyed by numerous informational sources in many fields. This means the lack of a sufficiently big amount of Web-sources and Web-services with semantic annotations, of

the visual images stored in multimedia databases and linked with the high-level conceptual descriptions, rich ontologies, etc.

This situation is often characterized as *the lack of a critical mass of semantic content.* That is why it has been possible to observe the permanent expansion in the scientific literature of the following opinion: a Semantic Web satisfying the initial goal of this project will be created in an evolutionary way as a result of the efforts of many research groups in various fields. In particular, this opinion is expressed in [1].

It is important to underline that this point of view is also expressed in the article "Semantic Web Revisited" written by the pioneers of Web: N. Shadboldt, W. Hall, T. Berners-Lee [22]. In this paper, the e-science international community is indicated as a community playing now one of the most important roles in quick generation of semantic content in a number of fields. The activity of this community seems to give a sign of future success of Semantic Web project.

One of the brightest manifestations of the need of new, strong impulses to developing Semantic Web is the organization of the First International Symposium on Incentives for Semantic Web under the framework of the Semantic Web International Conference – 2008 (Germany, Karlsruhe, October 2008).

The content of this paper is to be considered in the context of the broadly recognized need of the incentives for Semantic Web. Continuing the line of the papers [12 - 15] and the monograph [9], this paper outlines a comprehensive theoretical framework for the development of a Semantic Web of a new generation; it may be also called a Meanings Understanding Web [13] or a Multilingual Semantic Web with respect to [17].

Firstly, the paper grounds the possibility of using a mathematical model introduced in the monograph [9] and describing a system of 10 partial operations on conceptual structures for building semantic representations (in other terms, text meaning representations) of, most likely, arbitrary sentences and discourses in English, Russian, French, German, and other natural languages (texts pertaining to arbitrary spheres of professional activity). This model is the kernel of the theory of K-representations (knowledge representations).

Secondly, the paper sets forth an original strategy of transforming the existing Web into a Semantic Web of a new generation with the well developed mechanisms of understanding natural language texts.

For the realization of this strategy, the theory of K-representations provides a number of broadly applicable formal tools. *The third subject* of this paper are the peculiarities and input-output characteristics of the elaborated algorithm of semantic-syntactic analysis forming one of the principal constituents of the theory of K-representations. The outputs of this algorithm are the semantic representations of the input NL-texts being the expressions of SK-languages (standard knowledge languages). The input texts of this algorithm belong to the sublanguages of English, Russian, and German languages. For the development of a program implementation of this algorithm, the programming language PYTHON has been used.

2 The need of an advanced language platform for semantic Web

In [22], N. Shadbolt, W. Hall, and T. Berners-Lee ground the use of RDF as the basic language of the Semantic Web project with the help of the principle of least power: "the less expressive the language, the more reusable the data". As it is well known, the basic data structure of RDF is the triplets of the form subject – predicate – object. However, it seems that the stormy progress of, first of all, e-science urges us to find a new interpretation of this principle in the context of the challenges faced nowadays by the Semantic Web project. E-science (in particular, bioinformatics) needs to store on the Web the semantic content of the definitions of numerous notions, the content of scientific articles, technical reports, etc. The similar requirements are associated with semantics-oriented computer processing of the documents pertaining to economy, technology, medicine, law, politics, sport. In particular, it is necessary to store the semantic content of the articles from newspapers, of TV-presentations, etc.

The substantial discussions of the role of semantics-oriented natural language processing mechanisms for constructing a Semantic Web satisfying the demands of numerous end users can be found in the papers [12 – 15] and in the monographs [7, 9].

That is why it can be conjectured (see also [14]) that, in the context of the Semantic Web project, the following new interpretation of the principle of least power is reasonable: an advanced language platform for Semantic Web is to allow for modeling a system of operations on conceptual structures enabling us to build semantic representations (SRs) of practically arbitrary texts in Natural Language (NL) pertaining to arbitrary field of professional activity.

3 Shortly about ten conceptual operations considered by the theory of SK-languages

The question immediately emerges what a system of operations on conceptual structures satisfying the mentioned requirement might look like. A possible answer to this question is given by the theory of K-representations (knowledge representations) stated in the monograph [9]. The basic mathematical model of this theory describes a system consisting of 10 partial operations on conceptual structures [7 - 9]. The model determines a new class of formal languages for building semantic representations (SRs) of sentences and complex discourses in NL – the class of SK-languages (standard knowledge languages). An early version of this model set forth in [10, 11] determines the class of RSK-languages (restricted standard knowledge languages).

Let’s consider the central ideas of determining the class of SK-languages At the first step (consisting of a

rather long sequence of auxiliary steps), a class of formal objects called *conceptual bases (c.b)* is defined. Each c.b. *B* is equivalent to a system of the form $(c_1, \ldots, c_{15})$ with the components $c_1,\ldots, c_{15}$ being mainly finite or countable sets of symbols and distinguished elements of such sets. In particular, $c_1 = St$ is a finite set of symbols called sorts and designating the most general considered notions (concepts); $c_5 = X = X(B)$ is a countable set of strings used as elementary blocks for building knowledge modules and semantic representations (SRs) of texts; *X* is called a primary informational universe; $c_6 = V$ is a countable set of variables; $c_8 = F$ is a subset of *X* whose elements are called functional symbols.

Each c.b. *B* determines three classes of formulas, the first class *Ls(B)* being considered as the principal one and being called *the SK-language (standard knowledge language) in the basis B*. Its strings (they are called K-strings) are convenient for building SRs of NL-texts. We'll consider below only the formulas from the first class *Ls(B)*.

For determining for arbitrary c.b. *B* three classes of formulas, a collection of inference rules P[0], P[1], ... , P[10] is defined. The rule P[0] provides an initial stock of formulas from the first class. E.g., there is such c.b. B_1 that, according to P[0], $Ls(B_1)$ includes the elements *car1, green, city1, fin-set, India, 14, 14/cm, all, any, Height, Distance, Staff, Suppliers, Quantity, x1, x5*.

For arbitrary c.b. *B*, let *Degr(B)* be the union of all Cartesian m-degrees of *Ls(B),* where $m \geq 1$. Then the meaning of the rules of constructing well-formed formulas P[1], ..., P[10] can be explained as follows: for each k from 1 to 10, the rule P[k] determines a partial unary operation *Op[k]* on the set *Degr(B)* with the value being an element of *Ls(B).*

Example. There is a conceptual basis B possessing the following properties. The primary informational universe X = X(B) includes the conceptual items *China, India, Sri_Lanka*. Hence the value of the partial operation *Op[7]* (it governs the use of logical connectives $\wedge$ - AND and $\vee$ - OR) on the four-tuple

$< \vee$, *China, India, Sri-Lanka* >

is the K-string *(China* $\vee$ *India* $\vee$ *Sri-Lanka).*

Besides, X(B) includes the items article1 (a paper), article2 (a manufactured article), and h1 = article2, h2 = Kind1(certn article2, ceramics), h3 = (Country1(certn article2) ≡ (China $\vee$ India $\vee$ Sri-Lanka)), h4 = article2 * (Kind1, ceramics) (Country1, (China $\vee$ India $\vee$ Sri_Lanka)) are the elements of Ls(B). Then the K-string h4 is the result of applying the partial operation P[8] to the operands h1, h2, h3.

Ls(B) includes the string *h5* of the form *certn h4*, being the result of applying the operation P[1] to the operands *certn* and *h4.* The item *certn* denotes the meaning of the expression "a certain", and the string *h5* is interpreted as a designation of a manufactured article being a kind of ceramics and produced in China, India, or Sri-Lanka.

Let h6 be the string of the form *(Height(h5)* ≡*14/cm).* Then *h6* belongs to *Ls(B)* and is the result of applying the partial operation P[3] to the operands *Height(h5)* and *14/cm.*Thus, the essence of the basic model of the theory of SK-languages is as follows: this model determines a partial algebra of the form

(*Degr(B), Operations(B)*) ,

where *Degr(B)* is the carrier of the partial algebra, *Operations(B)* is the set consisting of the partial unary operations *Op[1], ..., Op[10]* on *Degr(B).*

The volume of the complete description in [9] of the mathematical model introducing, in essence, the operations *Op[1], ..., Op[10]* on *Degr(B)* and, as a consequence, determining the class of SK-languages considerably exceeds the volume of this paper. That is why, due to objective reasons, this model can't be included in this paper. The short characteristics of these partial operations on conceptual structures can be found, in particular, in [13].

4 The use of SK-languages for building semantic representations of complex biomedical discourses

During several last years, the significance of natural language processing (NLP) technologies for informatics dealing with the problems of biology and medicine has been broadly recognized. As a consequence, the term BioNLP interpreted as the abbreviation for Natural Language Processing in Biology and Medicine was born [20]. The formalization of natural language semantics is a very acute problem of BioNLP. That is why let's illustrate the new expressive possibilities provided by SK-languages on the example of building a semantic representation of a rather complex discourse pertaining to biomedicine.

Let D1 = T1.T2, where T1 = "The scientists know that a sequence of three bases (triplet) contains the message to call for the attachment of a specific amino acid in the protein chain", and T2 = "For example, the mRNA base code sequence GUC (guanine, uracil, cytosine) on mRNA calls for the attachment of the amino acid valine, while the mRNA base code sequence AUG (adenine, uracil, guanine) calls for the attachment of the amino acid methionine".

Let Semrepr1 = Situation(e1, knowing * (Agent1, certn set * (Qual-compos, scientist) : S1)(Content1, Contain2(arbitr sequence * (Numb, 3)(Qual-compos, base1) : x1, certn info-piece * (Determinator1, certn attachment1 * (Dynamic-object, specific amino-acid : x3)(Goal-object, certn chain1 * (Qual-compos, protein1) : x4)) : x2) : P1)).

Let us interpret the formula *Semrepr1* as a possible K-representation of the first sentence T1, that is, as a semantic representation (SR) of T1 being an expression of the SK-language determined by a certain conceptual basis. In the formula *Semrepr1*, the variable *P1* plays the role of a mark of the meaning of the principal part of the first sentence T1.

Let *Semrepr2* be the formula

*Example(P1, 1, Call-for(arbitr sequence * (Numb, 3)(Qual-compos, base1)(Compos-seq, (<1, G>* $\wedge$ *<2, U>* $\wedge$ *<3, C>)) : x5, certn attachment1 * (Dynamic-object,*

*specific amino-acid * (Name1, "valine")(Location, certn mRNA : x6) : x7) (Goal-object, certn chain1 * (Qual-compos, protein1) : x8))).*

Let *Semrepr3* be the formula

*Example(P1, 2, Call-for(arbitr sequence * (Numb, 3)(Qual-compos, base1)(Compos-seq, (<1, A> ∧ <2, U> ∧ <3, G>)) : x9, certn attachment1 * (Dynamic-object, specific amino-acid * (Name1, "methionine") : x10) (Goal-object, x8))),*

and let *Semdisc1 = (Semrepr1 ∧ Semrepr2 ∧ Semrepr3).*

Then the formula *Semdisc1* can be interpreted as a possible K-representation of the discourse D1. This formula provides the possibility to indicate several important advantages of the K-representations theory in comparison with first-order predicate logic and the Theory of Conceptual Graphs.

SK-languages allow for describing semantic structure of the sentences with direct and indirect speech and of the discourses with the references to the meanings of phrases and larger parts of a discourse, for constructing compound designations of the notions, sets, and sequences.

As far as one can judge on the available scientific literature, now only the theory of K-representations explains the regularities of structured meanings of, likely, arbitrary sentences and discourses pertaining to biomedicine and other fields of professional activity of people.

5 A universal tool for constructing semantic annotations

The analysis of a number of publications studying the problem of transforming the existing Web into Semantic Web allows for drawing the following conclusion: an ideal configuration of Semantic Web would be a collection of interrelated resources, where each of them has both an annotation in natural language (NL) and a formal annotation reflecting the meaning or generalized meaning of this resource, i.e. a semantic annotation. NL-annotations would be very convenient for the end users, and semantic annotations would be used by question-answering systems and advanced search engines.

Most likely, the first idea concerning the formation of semantic annotations of Web data would be to use the formal means for building semantic representations (SRs), or text meaning representations, of NL-texts provided by mathematical and computational linguistics.

However, the analysis shows that the expressive power of the main popular approaches to building SRs of NL-texts, in particular, of Discourse Representation Theory, Theory of Conceptual Graphs, and Episodic Logic is insufficient for effective representing contents of arbitrary Web data, in particular, of arbitrary biological, medical, or business documents.

First of all, the restrictions concern describing semantic structure of: (a) infinitives with dependent words (e.g., representing the goals, commitments, and the intended manners of using things and procedures); (b) constructions formed from the infinitives with dependent words by means of the logical connectives "and", "or", "not"; (c) the complex designations of sets; (d) the fragments where the logical connectives "and", "or" join not the designations of assertions but the designations of objects ("the product A is distributed by the firms B1, B2, ..., BN"); (e) the explanations of the terms being unknown to an applied intelligent system; (f) the fragments containing the references to the meanings of phrases or larger fragments of a discourse ("this method", etc.); (g) the designations of the functions whose arguments and/or values may be the sets of objects ("the staff of the firm A", "the number of the suppliers of the firm A", etc.).

Taking into account this situation and the fact that the semantic annotations of Web-sources are to be compatible with the format of representing the pieces of knowledge in ontologies, a number of researchers undertook the efforts of constructing computer intelligent systems, using the languages RDF, RDFS or OWL for building semantic annotations of Web-sources [18, 21].

However, the expressive power of RDF, RDFS or OWL is insufficient for being an adequate formal tool of building semantic annotations of scientific papers, technical reports, etc.

Meanwhile, the formulated idea of where to get the formal means for building semantic annotations from is correct. The main purpose of this section is to illustrate some principal ideas of employing the SK-languages for building semantic annotations of informational sources, in particular, Web-based sources.

Example. Let's consider a possible way of employing SK-languages for building a semantic annotation of the famous paper "The Semantic Web" by T. Berners-Lee, J. Hendler, and O. Lassila published in "Scientific American" in May 2001 [2].

Suppose that there is a Web-source associating the following NL-annotation with this paper: "It is proposed to create such a net of Web-based computer intelligent agents (CIAs) being able to understand the content of almost every Web-page that a part of this net will be composed by CIAs being able to understand natural language".

A semantic annotation corresponding to this NL-annotation can be the K-string of the form

*certn inf.ob * (Kind1, sci_article)(Source1,*
*certn journal1 * (Name1, "Scientific_American") : x1)*
(Year, 2001)(Month, May))(Authors,
*certn group1 * (Numb, 3)(Elements1,*
*(< 1, certn scholar * (First_name, "Tim")(Surname,*
*"Berners-Lee") : x2 > ∧ < 2, certn scholar **
(First_name, "James")(Surname, "Hendler") : x3 >
*∧ < 3, certn scholar * (First_name, "Ora")(Surname,*
"Lassila") : x4 >)) : S1) (Central_ideas,
(< 1, Semrepr1 > ∧ < 2, Semrepr2 >)) : v ,

where the variable *S1* designates the group consisting of all authors of this article, v is a variable being a mark of the constructed semantic annotation as an informational

object, and *Semrepr1, Semrepr2* are the K-strings defined by the following relationships:

*Semrepr1 = Proposed(S1, creation1 * (Product1 , certn family1 * (Qual-compos, intel_comp_agent * (Property, web-based)(Ability, understanding1 * (Inf_object , Content(almost_ every web_page)))) : S2) (Time, certn time_interval * (Part1, Nearest_future(decade1, #now#))) ,*

*Semrepr2 = Proposed(S1, achieving_situation * (Description1, (Exists (S3, set) ∧ Subset(S3, S2) ∧ Qual-compos (S3, intel_comp_agent * (Property , web-based)(Ability, understanding1 * (Inf_object , almost_every text * (Language1, certn language * (Belong, NL_family))))))) .*

To sum up, a comprehensive formal tool for building semantic annotations of Web data is elaborated. This tool is the theory of SK-languages. A very important additional expressive mechanism of SK-languages in comparison with the mechanisms illustrated in the example above is the convenience of building semantic representations of discourses with the references to the meanings of phrases and larger parts of a discourse.

The analysis of expressive power of the class of SK-languages (see the chapters 5 and 6 of [9]) allows for conjecturing that it is both possible and convenient to construct semantic annotations of arbitrary Web data by means of SK-languages. That is why the theory of SK-languages can be interpreted as a powerful and flexible (likely, universal) formal metagrammar of semantic annotations of Web data.

6 The formal tools provided by the theory of K-representations

The monographs [7], [9], stating two versions of the theory of K-representations, propose one universal (most likely) and several broadly applicable formal tools for the realization of this strategy.

The *first basic constituent* of the theory of K-representations is the theory of SK-languages (standard knowledge languages), stated, in particular, in [7 - 9]. The kernel of the theory of SK-languages is a mathematical model describing a system of such 10 partial operations on structured meanings (SMs) of natural language texts (NL-texts) that, using primitive conceptual items as "blocks", we are able to build SMs of arbitrary NL-texts (including articles, textbooks, etc.) and arbitrary pieces of knowledge about the world.

The analysis of the scientific literature on artificial intelligence theory, mathematical and computational linguistics shows that today the class of SK-languages opens the broadest prospects for building semantic representations (SRs) of NL-texts (i.e., for representing meanings of NL-texts in a formal way).

The expressions of SK-languages will be called below the K-strings. If T is an expression in natural language (NL) and a K-string *E* can be interpreted as a SR of T, then *E* is called a K-representation (KR) of the expression T.

The *second basic constituent* of the theory of K-representations is a widely applicable mathematical model of a linguistic database (LDB). The model describes the frames expressing the necessary conditions of the existence of semantic relations, in particular, in the word combinations of the following kinds: “Verbal form (verb, participle, gerund) + Preposition + Noun”, “Verbal form + Noun”, “Noun1 + Preposition + Noun2”, “Noun1 + Noun2”, “Number designation + Noun”, “Attribute + Noun”, “Interrogative word + Verb”. The expressive power of SK-languages enables us to associate the lexical units with the appropriate simple or compound semantic units. The model describes the logical structure of linguistic databases being the components of natural-language interfaces to intelligent databases as well as to other applied computer systems (see Chapter 7 of [9]).

The *third basic constituent* of the theory of K-representations is several complex, strongly structured algorithms carrying out semantic-syntactic analysis of texts from some practically interesting sublanguages of NL. More details about these algorithms can be found below (see also Chapters 8 - 10 of [9]).

7 The principles of designing natural language processing systems

Most often, semantics-oriented natural language processing systems, or linguistic processors (LPs), are complex computer systems, their design requires a considerable time, and its cost is rather high. Usually, it is necessary to construct a series of LPs, step by step expanding the input sublanguage of NL and satisfying the requirements of the end users. On the other hand, the same regularities of NL are manifested in the texts pertaining to various thematic domains.

That is why, in order to diminish the total expenses of designing a family of LPs by one research centre or group during a certain several-year time interval and in order to minimize the duration of designing each particular system from this family of LPs, it seems to be reasonable to pay more attention to: (a) the search for best typical design solutions concerning the key subsystems of LPs with the aim to use these solutions in different domains of employing LPs; (b) the elaboration of formal means for describing the main data structures and principal procedures of algorithms implemented in semantic-syntactic analyzers of NL-texts or in the synthesizers of NL-texts.

That is why it appears that the adherence to the following two principles in the design of semantics-oriented LPs by one research centre or a group will contribute, in the long-term perspective, to reducing the total cost of designing a family of LPs and to minimizing the duration of constructing each particular system from this family:

the ***Principle of Stability*** of the used language of semantic representations (LSR) in the context of various tasks, various domains and various software environments (stability is understood as the employment

of a unified collection of rules for building the semantic structures as well as domain- and task-specific variable set of primitive informational units);

the ***Principle of Succession*** of the algorithms of LP based on using one or more compatible formal models of a linguistic database and unified formal means for representing the intermediate and final results of semantic-syntactic analysis of natural-language texts in the context of various tasks, various domains and various software environments (the succession means that the algorithms implemented in basic subsystems of LP are repeatedly used by different linguistic processors).

The theoretical results stated in chapters 1 - 6 of the monograph [9] provide a basis for following-up the principle of stability of the used language of semantic representations. Chapter 4 defines a class of SK-languages (standard knowledge languages) that enable us to build semantic representations of natural language texts in arbitrary application domains. The broad perspectives for following-up the principle of succession of the algorithms of semantic-syntactic analysis of NL-texts are opened by the content of chapters 7 – 10 of [9].

8 A possible strategy of developing a multilingual semantic web

It seems that the Principle of Stability of the used language of semantic representations has much broader sphere of application than the professional activity of any concrete research group or research centre dealing with NLP. There are reasons to believe that following-up this principle can considerably speed-up the progress of the studies bridging a gap between the Semantic Web and NLP.

The process of endowing the existing Web with the ability of understanding many natural languages is an objective ongoing process [23]. It is a decentralized process, because the research centres in different countries mainly independently develop the translators from particular natural languages to semantic representations (or text meaning representations) and the applied computer systems extracting the meanings from texts in particular natural languages or producing summaries of the collections of texts in particular languages.

The analysis has shown that there is a way to increase the total successfulness, effectiveness of this global decentralized process. In particular, it would be important with respect to the need of cross-language conceptual information retrieval and question - answering. The proposed way is a possible new paradigm for the mainly decentralized process of endowing the existing Web with the ability of processing many natural languages.

The principal idea of a new paradigm is as follows. There is *a common thing* for the various texts in different natural languages. This common thing is the fact that *the NL-texts have the meanings.*

The meanings are associated not only with NL-texts but also with the visual images (stored in multimedia databases) and with the pieces of knowledge from the ontologies.

That is why the great advantages are promised by the realization of the situation when a unified formal environment is being used in different projects throughout the world for reflecting structured meanings of the texts in various natural languages, for representing knowledge about application domains, for constructing semantic annotations of informational sources and for building high-level conceptual descriptions of visual images.

The analysis of the expressive power of SK-languages (see the chapters 3 – 6 of [9]) shows that the SK-languages can be used as a unified formal environment of the kind. It is a direct consequence of the following hypothesis put forward by the author in [7 – 9, 13, 15]: SK-languages are a convenient tool of building semantic representations of arbitrarily complex natural language texts (sentences and discourses) pertaining to arbitrary field of professional activity.

This central idea underlies the strategy (described below) of transforming step by step the existing Web into a Semantic Web of a new generation, where its principal distinguished feature would be the well-developed ability of NL processing; it can be also qualified as a Meanings Understanding Web or as a Multilingual Semantic Web. The previous versions of this strategy are published in [9, 15].

The proposed strategy is based on (a) the mathematical model constructed in [9] and describing a system of 10 partial operations on conceptual structures and (b) the analysis of the expressive mechanisms of SK-languages. The new strategy can be very shortly formulated as follows:

1. An XML-based format for representing the expressions of SK-languages (standard knowledge languages) will be elaborated. Let's agree that the term "a K-representation of a NL-text T" means below a semantic representation of T built in this format and that the term "a semantic K-annotation" will be interpreted below as a K-representation of a NL-annotation of an informational source. The similar interpretations will have the terms "a K-representation of a knowledge piece" and "a high-level conceptual K-description of a visual image".
2. The NL-interfaces for different sublanguages of NL (English, Russian, German, Chinese, Japan, etc.) helping the end users to build semantic K-annotations of Web-sources and Web-services are being designed.
3. The advanced ontologies being compatible with OWL and using K-representations of knowledge pieces are being elaborated.

Example. Let T1 = "A flock is a large number of birds or mammals (e.g. sheep or goats), usually gathered together for a definite purpose, such as feeding, migration, or defence". T1 may have the K-representation *Expr1* of the form

*Definition1 (flock, dynamic-group * (Qualitative-composition, (bird ∨ mammal * (Examples, (sheep ∧ goal)))), S1,*

*(Estimation1(Quantity(S1), high) ∧ Goal-of-forming (S1, certain purpose * (Examples, (feeding ∨ migration ∨ defence)))))*.

The analysis of this formula enables us to conclude that it is convenient to use for constructing semantic representations (SRs) of NL-texts: (1) the designation of a 5-ary relationship *Definition1*, (2) compound designations of concepts (in this example the expressions *mammal * (Examples, (sheep ∧ goal))* and *dynamic-group * (Qualitative-composition, (bird ∨ mammal * (Examples, (sheep ∧ goal))))* were used), (3) the names of functions with the arguments and/or values being sets (in the example, the name of an unary function *Quantity* was used, its value is the quantity of elements in the set being an argument of this function), (4) compound designations of intentions, goals; in this example it is the expression *certain purpose * (Examples, (feeding ∨ migration ∨ defence))* .

The structure of the constructed K-representation *Expr1* to a considerable extent reflects the structure of the definition T1.

4. The new content languages using K-representations of the content of messages sent by computer intelligent agents (CIAs) in multi-agent systems are being worked up. In particular, this class of languages is to include a subclass being convenient for building the contracts concluded by the CIAs as a result of successful commercial negotiations.
5. The visual images of the data stored in multimedia databases are being linked with high-level conceptual K-descriptions of these images (see Section 6.3 of [9]).
6. The NL-interfaces transforming the NL-requests of the end users of Web into the K-representations are being designed.
7. The advanced Web-based search and question-answering systems are being created being able (a) to transform (depending on the input request) the fragments of a discourse into the K-representations, (b) to analyze these K-representations of the discourse fragments, and (c) to analyze semantic K-annotations of Web-sources and Web-services.
8. The NL processing systems being able to automatically extract knowledge from NL-texts, to build the K-representations of knowledge pieces, and to inscribe these K-representations into the existing ontologies are being elaborated.
9. The generators of NL-texts (the recommendations for the users of expert systems or of recommender systems, the summaries of Web-documents, etc.) using the SK-languages for representing the meaning of a NL-text to be synthesized are being constructed. Besides, a reasonable direction of research seems to be the design of applied intelligent systems being able to present the semantic content of a message for the end user as an expression of a non-standard K-language being similar to a NL-expression but containing, may be, a number of brackets, variables, markers.

Fulfilling these steps, the international scientific community will create in a reasonable time a digital conceptual space unified by a general-purpose language platform. The realization of this strategy will depend on the results of its discussion by the international scientific community.

9 A new method of developing multilingual semantic-syntactic analyzers of NL-texts

9.1 The advantages of a new method

It seems that the complete potential of semantics-oriented approach to designing multilingual algorithms of processing NL-texts is far from being exhausted. A new implementation of this approach is described in the monograph [9]. In essence, [9] describes a new method of developing the algorithms of semantic-syntactic analysis of NL-texts. This method can be reconstructed from the study of the algorithm *SemSynt1* completely described in Chapters 8 - 10 of [9].

The input texts of the algorithm *SemSynt1* can be the sentences (statements, commands, and questions) from some practically interesting sublanguages of English, Russian (a Latin transcription is used), and German languages. The output of the algorithm is a semantic representation of the input text being its K-representation.

The principal advantages of the new method are as follows: (1) the algorithm *SemSynt1* uses an original formal model of a linguistic database (see Chapter 7 of [9]), this model is problem-independent; (2) an important feature of the algorithm is that it doesn't construct any syntactic representation of the inputted NL-text but directly finds semantic relations between text units; since numerous lexical units have several meanings, the algorithm uses the information from a linguistic database and linguistic *context* for choosing one meaning of a lexical unit among several possible meanings; (3) the other distinguished feature is that this complex algorithm is completely described with the help of formal tools, that is why its description doesn't uses any expressive mechanisms of any concrete programming system; (4) the main procedures of the algorithm (of the upper and middle levels) are the same for the English, Russian, and German languages; (5) the main procedures of the algorithm *SemSynt1* are described with the help of the terms being well known to the programmers (one- and two-dimensional arrays, a string, a set, a binary conceptual relation between two elements) and don't demand a command of complicated linguistic terminology, often being specific for a concrete natural language.

9.2 The input-output characteristics of the multilingual algorithm *SemSynt1*

Let's consider the examples illustrating the correspondence between the natural language sentences in English, Russian (in Latin transcription), and German and their semantic representations (SR) being the expressions of a certain SK-language, that is, being the K-representations of the input texts. In these examples, the SR of the input text T will be the value of the string variable *Semrepr* (Semantic representation). The considered examples illustrate the correspondence between the inputs and outputs of the developed algorithm *SemSynt1*.

Example 1. Let T1eng = "The international scientific conference "DEXA-2009" took place in Linz, Austria, during August 31 – September 4, 2009", T1rus = "Mezhdunarodnaya nauchnaya konferentsiya "DEXA-2009" prokhodila v gorode Linz, Avstriya s 31 avgusta po 4 sentyabrya 2009 goda", T1germ = "Die internationale wissenschaftliche Konferenz "DEXA-2009" war in Linz, Oesterreich waehrend 31. August – 4. September 2009 stattgefunden". Suppose that the used basic semantic items are constructed with respect to the spelling of English expressions corresponding to these items. For instance, the English words "city" and "town", the Russian word "gorod", and the German word group "die Stadt" will be associated with the semantic item *city1*. From the formal standpoint, it means that the elements of the used conceptual basis are built on the basis of English expressions. If this condition is satisfied, the algorithm builds the K-representation

*Semrepr = Situation(e1, taking-place * (Event1, certn conference1 * (Kind-geogr, international)(Kind-focus, science) : x1)(Place1, certn city1 * (Name1, "Linz")(Belongs-to-Country, certn country1 * (Name1, "Austria") : x3) : x2) (Time-interval, <31.08.2009, 04.09.2009>)) .*

Example 2. Let T2 = "Find a description of the programming language PYTHON on the Web-site http://docs.python.org", T3rus = "Naydite opisanie yazyka programmirovaniya PYTHON na veb-sayte http://docs.python.org", T3germ = "Finden eine Beschreibung der Programmiersprache PYTHON auf dem Site http://docs.python.org". Then *Semrepr = (Command (#Operator#, #Executor#, #now#, e1) ∧ Target (e1, finding1 * (Object-file, certn file1 * (Inf-content, certn description1t * (Focus-object, certn progr-lang * (Name1, "PYTHON") : x3) : x2))(Web-source, http://docs.python.org))).*

Example 3. Let T3eng = "Did the international scientific conference "DEXA" take place in Hungary?", T3rus = "Prokhodila li mezhdunarodnaya nauchnaya konferentsiya "DEXA" v Vengrii?", T3germ = "War die internationale wissenschaftliche Konferenz "DEXA" in Ungarn stattgefunden?". Then

*Semrepr = Question (x1, (x1 ≡ Truth-value (Situation (e1, taking_place * (Time, certn moment * (Earlier, #now#) : t1) (Event1, certn conference * (Type1, international) (Type2, scientific) (Name1, "DEXA") : x2) (Place, certn country1 * (Name1, "Hungary") : x3))))).*

Example 4. Let T4eng = "What English scientist discovered penicillin?", T3rus = "Kakoy angliysky uchony otkryl penicillin?", T3germ = "Welcher English Wissenschaftler hat Penizillin entdeckt?". Then

*Semrepr = Question (x1, Situation (e1, discovering1 * (Time, certn moment * (Earlier, #now#) : t1) (Agent1, certn scientist * (Country1, England) : x1) (New-object, certn medicine1 * (Name1, "penicillin") : x2))).*

Example 5. Let T5eng = "What European companies the firm "Rainbow" is cooperating with?", T5rus = "S kakimi evropeyskimi kompaniyami sotrudnichaet firma "Rainbow", T5germ = "Mit welchen europaeischen Kompanien die Firma "Rainbow" kooperiert?". Then

*Semrepr = Question (S1, (Qualitative-composition (S1, company1 * (Location, Europe)) ∧ Description(arbitrary company1* (Element, S1) : y1, Situation (e1, cooperation * (Time, #now#) (Agent2, certn company1 * (Name1, "Rainbow") : x1) (Cooper-partner, y1))))).*

Example 6. Let T6 = "Who produces the medicine "Zinnat"?". Then

*Semrepr = Question (x1, Situation (e1, production1 * (Time, #now#) (Agent2, x1) (Product2, certn medicine1 * (Name1, "Zinnat") : x2))).*

Example 7. Let T7eng = "When and where did Dr. Erik Stein arrive to Zuerich from?", T7rus = "Kogda i otkuda doktor Erik Stein priekhal v Zurikh?", T7germ = "Wann und woher hat Dr. Erik Stein nach Zuerich gekommen?". Then

*Semrepr = Question ((x4 ∧ x1), (Situation (e1, arrival * (Time, certn moment * (Earlier, #now#) : t1) (Start-location, x1)(Agent1, certn person * (Qualif, Ph.D.)(Name, "Erik")(Surname, "Stein") : x2) (Final-location, certn city1 * (Name1, "Zuerich") : x3)) ∧ (x4 ≡ t1)).*

Example 8. Let T8eng = "How many countries did participate in the Olympic Games - 2008?", T7rus = "Skolko stran uchastvovalo v Olimpiyskikh Egrakh – 2008", T7germ = "Wieviel Laender haben an den Olympischen Spielen – 2008 teilgenommen?". Then

*Semrepr = Question (x1, ((x1 ≡ Numb(S1)) ∧ Qualitative-composition (S1, country1) ∧ Description (certn country1 * (Element, S1) : y1, Situation (e1, participation1 * (Time, certn moment * (Earlier, #now#) : t1) (Agent1, y1) (Time, 2008/year)(Event1, certn olymp-game : x2)))).*

Example 9. Let T9eng = "How many times did Professor Bill Jones visit France?", T7rus = "Skolko raz professor Bill Jones posetil Frantsiu", T7germ = "Wieviel Mal hat Herr Professor Bill Jones Frankreich besucht?". Then

*Semrepr = Question (x1, ((x1 ≡ Numb (S1)) ∧ Qualitative-composition (S1, sit) ∧ Description (arbitrary sit * (Element, S1) : e1, Situation (e1, visiting * (Time, certn moment * (Earlier, #now#) : t1) (Agent1, certn person * (Qualif, professor)(Name, "Bill") (Surname, "Jones") : x2) (Place2, certn country * (Name1, "France") : x3))))).*

9.3 Implementation of the algorithm *SemSynt1*

An expanded and modified version of the algorithm *SemSynt1* has been implemented with the help of the programming language PYTHON; as it is shown in [4], this language proved to be a convenient tool of developing NL processing systems. The input language of the elaborated NL-interface SEMANTIKA (E.K. Orlov, Faculty of Business Informatics, State University – Higher School of Economics, Moscow) is broader than the input language of the algorithm *SemSynt1*: it includes the statements, questions, and commands in Russian that can contain the participle constructions and attributive clauses. For instance, the input language of the program SEMANTIKA includes the question "What medicines offered by the pharmaceutical firm "GlaxoSmithKlein" are produced in Poland?".

The predecessor of the *SemSynt1* – the algorithm *SemSyn* described in [7] – was implemented in the Web programming language PHP. Chapter 11 of the monograph [9] contains the examples illustrating the principles of processing NL-texts by the experimental Russian-language interface NL-OWL1, implemented in the Web programming system PHP and developed on the basis of the algorithm *SemSyn*. An particular, the example associating the definition "Carburettor is a device for preparing a gas mixture of petrol and air" firstly with a K-representation and later with an OWL-expression is considered.

10 Conclusion

The main result of this paper is an original strategy of transforming, step by step, the existing Web into a Semantic Web of new generation (SW-2), where the principal distinguished feature of SW-2 would be the well-developed ability of NL processing. That is why SW-2 can be also qualified as a Meanings Understanding Web or as a Multilingual Semantic Web.

The aim of proposing this strategy is to increase the total successfulness, effectiveness of the mainly decentralized global ongoing process of endowing the existing Web with the ability of understanding texts in many natural languages.

The proposed strategy is based on a broad spectrum of new possibilities provided by the theory of K-representations (knowledge representations) developed by the author of this paper and presented in [9]. In particular, the paper illustrates a number of new precious opportunities of using SK-languages for building semantic annotations of informational sources, constructing complex definitions of the concepts in the advanced ontologies, and building semantic representations (or text meaning representations) of complex discourses pertaining to biology and medicine.

The final part of the paper describes the peculiarities and input-output characteristics of a new multilingual algorithm of semantic-syntactic analysis of NL-texts (from the sublanguages of English, Russian, and German languages). This algorithm, called *SemSynt1,* is a part of the theory of K-representations and is presented in Chapters 9 and 10 of [9]. An expanded and modified version of *SemSynt*1 has been implemented with the help of the programming language PYTHON.

References

[1] Angelova, G. (2005). Language Technology Meets Ontology Acquisition. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) International Conference on Conceptual Structures 2005. LNCS, Vol. 3596, Springer, Heidelberg, pp. 367-380.

[2] Berners-Lee, T., Hendler, J., Lassila, O. (2001). The Semantic Web. Scientific American, pp. 34-43.

[3] Berners-Lee, T., Hall, W., Hendler, J.A., O'Hara, K., Shadboldt, N., and D.J. Weitzner (2006). A Framework for Web Science. Foundations and Trends in Web Science, Vol. 1, No. 1, now Publishers Inc.-134 p.

[4] Bird, S., Klein, E., and E. Loper (2009). Natural Language Processing with Python. O'Reily.

[5] Cimiano, P., Haase, P., Heizmann, J., Mantel, M. (2007). ORAKEL: A Portable Natural Language Interface to Knowledge Bases. *Technical report,* Institute AIFB, University of Karlsruhe, Germany.

[6] Duke, A., Glover, T., Davies, J. (2007). Squirrel: An Advanced Semantic Search and Browse Facility. In: *Proc. of the 4th European Semantic Web Conference. Innsbruck, Austria*.

[7] Fomichov, V.A. (2005). The Formalization of Designing Linguistic Processors. Moscow, MAX Press (in Russian).-368 p.

[8] Fomichov, V.A. (2007). Mathematical Foundations of Representing the Content of Messages Sent by Computer Intelligent Agents. Moscow, State University – Higher School of Economics, Publishing House "TEIS" (in Russian).-176 p.

[9] Fomichov, V.A. (2010). Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms. Springer, New York, Dordrecht, Heidelberg, London.-354 p.

[10] Fomichov, V.A. (1996). A Mathematical Model for Describing Structured Items of Conceptual Level. Informatica. An Intern. J. of Computing and Informatics (Slovenia), 20 (1), pp. 5-32.

[11] Fomichov, V.A. (1998). Theory of Restricted K-calculuses as a Comprehensive Framework for Constructing Agent Communication Languages. In: Fomichov V.A., Zeleznikar A.P. (eds.). Special Issue on NLP and Multi-Agent Systems.

Informatica. An International J. of Computing and Informatics (Slovenia), 22 (4), pp. 451-463.

[12] Fomichov, V.A. (2000). An Ontological Mathematical Framework for Electronic Commerce and Semantically-structured Web. In: Zhang Y., Fomichov V.A., Zeleznikar A.P. (eds.), Special Issue on Database, Web, and Cooperative Systems. Informatica. An International J. of Computing and Informatics (Slovenia), Vol. 24, No. 1, pp. 39-49.

[13] Fomichov, V.A. (2008). A Comprehensive Mathematical Framework for Bridging a Gap Between Two Approaches to Creating a Meaning-Understanding Web. International J. of Intelligent Computing and Cybernetics (Emerald Group Publishing Limited, UK), Vol. 1, No. 1, pp. 143-163.

[14] Fomichov, V.A. (2009a) Theory of K-representations as a Source of an Advanced Language Platform for Semantic Web of a New Generation. Web Science Overlay J. On-line Proceedings of the First International Conference on Web Science, Athens, Greece, March 18-20, 2009; available at http://journal.webscience.org/221/1/websci09-_submission_128.pdf.

[15] Fomichov, V. A. (2009b). A Scheme and Formal Tools for Transforming the Existing Web into Semantic Web of a New Generation. In: Pre-Conference Proceedings of the Focus Symposium on Knowledge Management Systems (August 4, 2009, Focus Symposia Chair: Jens Pohl) in conjunction with InterSymp-2009, 21st International Conference on Systems Research, Informatics and Cybernetics, August 3 – 7, 2009, Baden-Baden, Germany), Collaborative Agent Design Research Center, California Polytechnic State University, San Luis Obispo, CA, USA, pp. 39-50.

[16] Frank, A., Krieger, H.-U., Xu, F., Uszkoreit, H., Crysmann, B., Jrg, B., Schaefer, U. (2007). Question Answering from Structured Knowledge Sources. *Journal of Applied Logic,* Vol. 5, No. 1, pp. 20-48.

[17] Multilingual Semantic Web (2009) CFP: 1st Workshop on the Multilingual Semantic Web (collocated with WWW 2010); received on Monday, 21 December 2009; http://lists.w3.org/Archives/Public/semantic-web/2009Dec/0065.html; retrieved 12.03.2010.

[18] Navigli, R., Velardi, P. (2006). Through automatic semantic annotation of on-line glossaries. In Proc. of European Knowledge Acquisition Workshop (EKAW)-2006, LNAI 4248, pp. 126-140.

[19] Popescu, A.-M., Etzioni, O., Kautz, H. (2003). Towards a Theory of Natural Language Interfaces to Databases. In: *Proc. of the 8th International Conference on Intelligent User Interfaces,* Miami, FL , pp. 149-157.

[20] Prince, V., Roche, M., eds (2009). *Information Retrieval in Biomedicine: Natural Language Processing for Knowledge Integration.* IGI Global. 460 pp.

[21] Reeve, L., Han, H. (2005). Survey of semantic annotation platforms. Proc. of the 20th Annual ACM Symposium on Applied Computing and Web Technologies.

[22] Shadboldt, N., Hall, W., Berners-Lee, T. (2006). Semantic Web Revisited. IEEE Intelligent Systems, Vol. 21, No. 3, pp. 96-101.

[23] Wilks, Y., Brewster, C. (2006). Natural Language Processing as a Foundation of the Semantic Web. Foundations and Trends in Web Science, Vol. 1, No. 3 - 4, now Publishers Inc.-129 p.

Wikipedia2Onto – Building Concept Ontology Automatically, Experimenting with Web Image Retrieval

Huan Wang, Xing Jiang, Liang-Tien Chia and Ah-Hwee Tan
School of Computer Engineering, Nanyang Technological University, Singapore
E-mail: wa0004an, jian0008, asltchia, asahtan@ntu.edu.sg

Keywords: wikipedia, semantic concept, ontology, web image classification

Received: May 15, 2009

Given its effectiveness to better understand data, ontology has been used in various domains including artificial intelligence, biomedical informatics and library science. What we have tried to promote is the use of ontology to better understand media (in particular, images) on the World Wide Web. This paper describes our preliminary attempt to construct a large-scale multi-modality ontology, called AutoMMOnto, for web image classification. Particularly, to enable the automation of text ontology construction, we take advantage of both structural and content features of Wikipedia and formalize real world objects in terms of concepts and relationships. For visual part, we train classifiers according to both global and local features, and generate middle-level concepts from the training images. A variant of the association rule mining algorithm is further developed to refine the built ontology. Our experimental results show that our method allows automatic construction of large-scale multi-modality ontology with high accuracy from challenging web image data set.

Povzetek: Prispevek opisuje izgradnjo velike multimodalne spletne ontologije AutoMMOnto.

1 Introduction

Real-world images always involve pictures with various backgrounds, object aspects, poses and appearances. Taking the animal classes in Figure 1 as an example, human-beings can easily differentiate the four classes. However, computers are not able to identify the difference in the same way. The varied background environment of the same *Arctic Fox* class can introduce great variance in global image features, while the subtle fur color difference between *Arctic Fox* and *Fennec Fox* makes it difficult to classify from local image features. It is also hard to identify the different distribution of colors over *Maned Wolf* or *Dhole* from spatial features. On the other hand, cues from the text on the corresponding web page could make a substantial contribution to the performance of image classification. For example, even a single keyword *Kashmir* could indicate the *Dhole* class, as *Kashmir* is the habitat of *Dhole*. Similar useful relationships which help to narrow down the final concepts include name, diet, and distribution relationships. Therefore, an effective way is to combine the images features with the text information for image retrieval, where ontology is utilized for this purpose.

Ontology, which clearly defines concepts and their relationships in a domain, has been widely used in many information retrieval fields, including document indexing, i.e., extracting semantic contents from a set of text document, image retrieval and classification, i.e., using concepts either from image features or surrounding text for content representation, and video retrieval, i.e., using text in video captions for semantic concept detection. Note that most of the approaches involve external lexical dictionary or online category as ontologies. They certainly improve the performance. However, they also introduce the following major questions:

1. Is ontology just as same as a hierarchical collection of concepts?
2. Ontology has to be manually built, which is extremely time consuming. Can it be done automatically?
3. Can Ontology be scalable when it is extended to large domains?

Through research on the use of ontology to better understand media information, we have provided our answers to the aforementioned questions:

1. Ontology is not just a hierarchical collection of concepts with parent-child relation. Details of the differences will be apparent as you read the details in this paper.
2. We do agree that one main difficulty that hedges against the development of ontology approaches is the extra work required in ontology construction and annotation. But there is a hope, and this paper describes our original attempt to use both structural and content features of Wikipedia to build a proposed hierarchy with not only hyponymy(*is-a*) or meronymy(*part-of*) relationships but also more real-life relationships. Therefore, the resulting semantic concept hierarchy of the built ontology, called AutoOnto, is consistent with

real world knowledge and can be used to map text information on the web page to detect semantic concepts. 3. Scalability is indeed a problem when a single party or a consortium tries to create a whole ontology structure. However, the problem could be solved when we can import existing ontologies or newly created ontologies to merge with other ontologies. The important issue here is to understand and handle the similarities/dissimilarities of concepts existing in the respective ontologies, which is current interest to relevant groups in AI and ontology-related areas.

Figure 1: An example of web image classes in our data set. Even though these images are portraying 4 difficult animal classes, it is easy for human-beings to identify the classes: *Arctic fox* has light-colored fur; *Fennec fox* has a pair of grotesque ears and ET-Style face; *Maned wolf* is featured with its black long legs; and *Dhole* has white fur spreading from its jaw to abdomen. However, it is not easy for image processing approaches to tell the classes apart due to the lack of discriminative low-level features.

Note that we have manually built a text ontology, called ManuOnto, and shown that it can effectively help machine understand multimedia in a better way in our previous work [9]. In this paper, we first show that AutoOnto captures the relationships between concepts as well as, if not better than the manually-built ontology with bigger knowledge coverage and higher efficiency. Then, we train classifiers according to our 164 dimensional features (SIFT with opponent color angle) and generate middle-level concepts from the training result and integrate the AutoOnto to form AutoMMOnto (Auto Multi-Modality Ontology). The MAP results of our experiment on Google (top 200 retrievals) Image search, AutoOnto and AutoMMOnto (AutoOnto+ visual descriptions) are 0.7049, 0.8942 and 0.9125 respectively. We have therefore shown that our method allows automatic construction of large-scale multi-modality ontology with high accuracy from a challenging web image data set.

Our contribution in this paper is concluded as follows: We propose a method to build large scale concept ontology from Wikipedia in a cost effective way. The generated ontology is able to extract additional information from the web pages and increase the concept detection accuracy. We also propose an association rule mining algorithm to refine relationships in the ontology. The resulting relationship set are more concise with higher precision.

The rest of this paper is organized as follows. Section 2 introduces the related works. Section 3 discusses the Wikipedia category and structure information. Section 4 discusses how we use Wikipedia to automatically build the concept ontology. In Section 5 we propose an association rule mining algorithm to discover the key semantic relationships. Experiment results on our collected web image database are given in Section 6. We conclude this paper in Section 7.

2 Related works

Due to the dependency between the external knowledge source and semantic concepts, the chosen knowledge source will affect the derived concepts and relationships for ontology construction, which is used to classify the existing ontology construction approaches.

WordNet [1], developed at Princeton University, has been commonly used as such a lexical dictionary as it is able to group words into sets of synsets. The structured network makes it easy to derive semantic hierarchies for various domains. Some typical examples which practise WordNet directly as an ontology for object recognition and video search include [2, 3, 4]. However, the reason why WordNet works fine in these experiments is that only common concepts (e.g., car, dog, grass, tree) and relationships (hypernymy, meronymy) are employed. If concepts and relationships outside the scope of WordNet, they will not be included and cannot be utilized. For example, WordNet has limited coverage of less popular or more specific concepts like "mountain bike" or "bush dog". This limitation decides that WordNet only works on sparse or general concept domains. Also, WordNet is also disconnected from the update of natural language vocabulary which changes with almost everyday. Therefore, it is not able to work on domains with novel topics and concepts.

Besides the above approaches, WordNet has also been used for assisting ontology building. For example, in [16], WordNet is dynamically included to extend a knowledge acquisition tool CGKAT. Particularly, top-level concept of WordNet ontology is subordinated into Sowa Ontology for finalizing an ontology. Similar approach is also proposed in [17], which incorporates general purpose resources like WordNet and open web directory to build large scale ontology for document indexing and categorization. Particularly, it takes two steps to build an ontology. Firstly, an initial ontology consists of two sub-trees from both the web directory and WordNet synset graph respectively. Then, it iteratively fills the gaps and enriches the existing ontology with new concepts and relationships until the ontology is verified by domain expert to be usable. As a result, the ontology building process is only semi-automatic. Also, the proposed method lacks support from solid experiment results and the performance in real application is yet to be evaluated.

With the rapid development of Internet, online categories seem to be a better choice for ontology construction, especially when some popular online categories also provide easy access [5, 6]. By indexing a

huge number of web pages/topics, online categories cover most real world objects, activities, news and documents in a timely manner. Besides the hierarchical structure offered by these categories, web page submitters and category indexers also provide more related concepts with varied relationships, which further extend the coverage. Some approaches which use online resources to construct knowledge base include [7], wherein specific domain knowledge of animal is extracted from online animal category and image features to construct ontology for web image classification. The application indicates that with the evolution of ontology-based applications, finding a proper knowledge source has become an important issue.

Among the existing online categories, there is an increasing interest in using Wikipedia as the resource for knowledge mining. In [18], a refinement on the Wikipedia category network is implemented step by step to generate taxonomy of *is-a* semantic links. All syntax-based, connectivity-based, lexicon-syntactic based and inference-based methods are used to remove noisy links and set up correct *is-a* links. To compare and analyse the performance of the Wikipedia-based ontology, the manually built ontology ResearchCyc [19] and WordNet are used as the performance baseline. The evaluation shows that the automatically-built taxonomy is comparable with the two existing ontologies. However, the construction is discontinued at the level of domains. While the authors put more emphasis on drawing out a taxonomy of 105,418 *is-a* links, the broad coverage also makes the taxonomy insensitive to specific applications, as different applications need different emphasis on the domain knowledge.

Other than using external resources to construct ontology, existing large-scale ontology constructions usually involve mass manual work. For example, LSCOM [8] aims to design a taxonomy with a coverage of around 1,000 concepts for broadcast news video retrieval. This approach is hampered by the tens of millions of human judgments required, which has been proved to be very ineffective and costly. In a word, most ontology constructions are either constructed on dependent domain or still involve mass manual work. And even those semi-automatic construction processes rely heavily on external knowledge resources, like the aforementioned lexical dictionary and online categories. Another disadvantage is apparent as the important merit of either dictionary or category is a hierarchical graph which connects concepts together. As a result, only shallow relationships like hypernymy/hyponymy(*is-a*) or meronymy(*part-of*) could be mined. These relations are not sufficient enough to support information mining from web images, which are usually attached to web pages with text information. Mining through such kind of text corpus involves more than the aforementioned semantic relationships. An ontology with enriched knowledge provides more discriminative information in web image retrieval, classification and annotation.

Referring to the existing work, we can see that an advanced ontology for multimedia research and applications should meet the following requirements: 1) The ontology should be constructed automatically, so that when it is applied to extended domains, the scalability will not become the bottleneck. 2) The ontology should involve more than domain-specific concepts. Also, besides *is-a* or *part-of* relationships, deeper semantic relationships should also be included so that the ontology is a better imitation of human general knowledge.

3 Wikipedia concepts and structure

Wikipedia is by far the biggest online free encyclopedia. It provides definitions for more than 2 million words and phrase concepts. This number is still growing as Wikipedia is based on online collaborative work and anyone can freely access, create and edit the page content of each concept. This open feature makes Wikipedia an up-to-date knowledge source, where even the latest concepts can be found. It also covers many concepts which are not commonly used and included in other electronic lexical dictionaries. In the following subsections, we will introduce some of Wikipedia's features which make it suitable for ontology construction.

3.1 Wikipedia category

The underlying structure of Wikipedia can be described in two network graphs: category graph and article graph. In both graphs, nodes represent articles and edges represent links between articles. Basically, all the Wikipedia web pages are put into a subject category according to general knowledge. This structure is depicted as the category graph which has been proved to be a scale-free, small world graph by graph-theoretic analysis[20]. The category graph is formed following the taxonomy of concepts. Therefore, the links in category graph indicate either *is-a* or *part-of* relationships between the two connected concepts (a sample of the category graph is given in Figure 2). In this sense, the semantic relationships provided by the category graph is quite similar to the relationships provided by WordNet. When referring to specific article, the Wikipedia classification is listed in a separate Categories section. Besides the category graph, there is also an article graph which indicates the cross-references between Wikipedia web pages. In particular, the articles are nodes of the graph, which are hyperlinked to corresponding Wikipedia articles. These links indicate a direct semantic relationship between the two connected concepts. Compared with WordNet which mainly organizes word concepts according to synset, Wikipedia category provides a more formal classification of concepts. As a result, the extracted concepts and relationships are closer to a formal ontology with various semantic relationships.

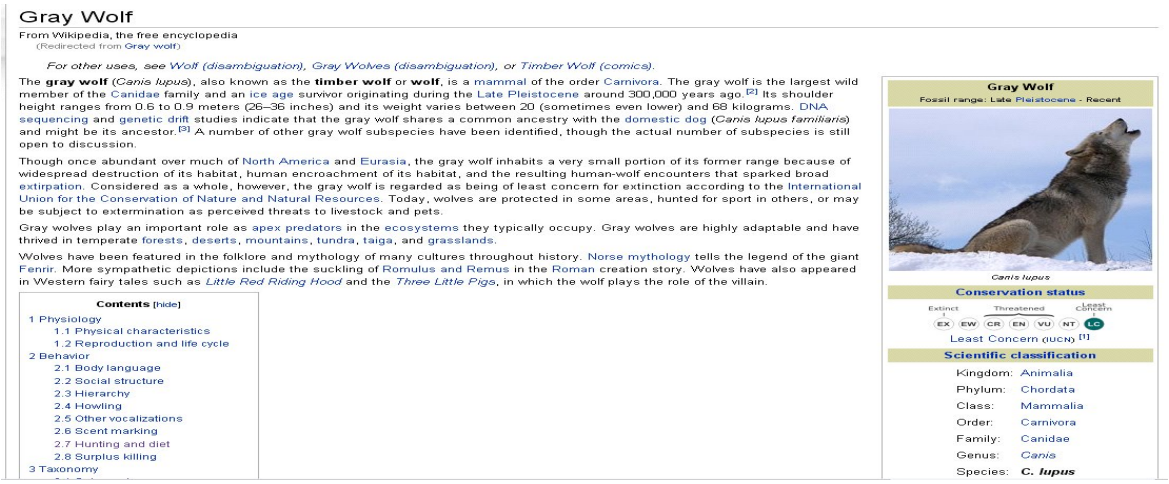


Figure 3: An example of Wikipedia web page with corresponding extracted concept. The extracted concept definition is: (define-concept concept_gray_wolf(or Some animal(all hasName(or gray_wolf timber_wolf wolf))(all has Distribution(or Canada Ireland Kazakhstan the_Middle_East North_America Russia Europe the_United_States India Asia Finland))(all hasDiet (or Herbivore Coyote American_Bison Deer Caribou Moose Yak Ungulate Rodent)))).

3.2 Wikipedia web page

In Wikipedia, each web page defines one concept according to general knowledge. Disambiguation is removed by separating different senses in different web pages. The searching in Wikipedia is straightforward as each web page has already been associated with the keywords. In most cases, the page title is the indexed keywords. The text information on the web page is divided into sections. Each section describes one aspect of the concept in details. Taking the concept Aardwolf as an example (see Figure 3), the main web page content includes physical characteristics, distribution and habitat, behaviour, and interaction with humans. From the viewpoint of concepts, each section is connected to the main concept with semantic relationships depicted as section titles. A concept graph is easily drawn from this web page content structure. On the right, the web page also provides a section of Scientific classification, which lists the zoology taxonomy of the animal. By integrating different concepts under the same domain *Animalia*, a big hierarchy picture can be easily constructed with the concepts positioned under corresponding branches. Compared to our manually built Animal Domain Ontology [7], the hierarchy generated from Wikipedia Scientific classification is more formally defined, and is considered to contain rigid domain information.

3.3 Concept coverage

In comparison to WordNet, whose total number of words is limited to around 147,278, Wikipedia certainly contains more information. For our case, only 12 out of the 20 class names are covered by WordNet. Class names such as *African wild dog, bat-eared fox, black jackal, bush dog, cape fox, Ethiopian wolf, fennec fox, golden jackal* are all missing from WordNet. Such limitations make WordNet an incomplete appropriate resource for ontology learning. On the contrary, Wikipedia is more suitable for this task. The total number of words has

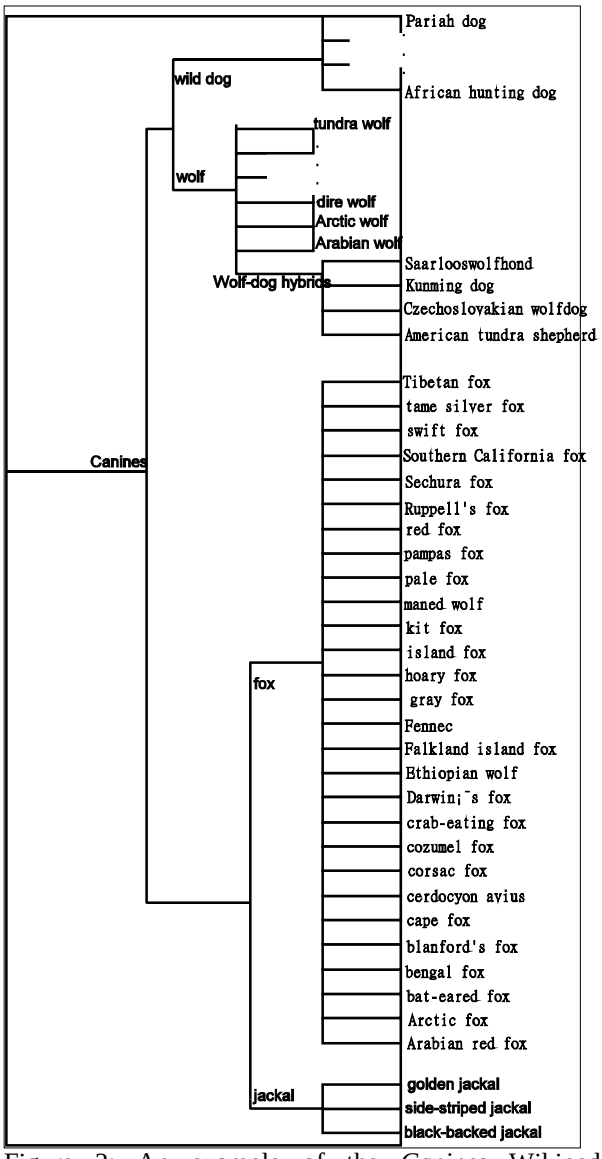


Figure 2: An example of the *Canines* Wikipedia category.

reached 2 million and it keeps increasing significantly daily. It can cover almost all the relevant concepts in our experiment.

4 Automatic ontology construction – Wikipedia2Onto

In this section we discuss the construction of our multi-modality ontology. Similarly to our previous manually construction process, the automatic process includes 3 steps. Particularly, the key concepts in the animal domain and the taxonomic relations are firstly extracted from Wikipedia. Then, the narrative descriptions of particular animals, including relevant concepts and non-taxonomic relations, are extracted. Finally, the *visual description*s of each concept are added. Note that we do not use the XML corpus provided by Wikipedia directly for construction. Instead, we use a web page crawler to download relevant concept web pages before ontology building in advance. Such an approach makes it more flexible to build ontology for specific domain. Meanwhile, a dynamic connection to Wikipedia can ensure "freshness" of our concepts as Wikipedia web pages are edited from time to time.

4.1 Key concepts and taxonomic relations extraction

Wikipedia has provided an entire category of many meaningful concepts, which is formed according to hypernymy relationships between concepts. In other words, Wikipedia category provides taxonomy of general concepts in natural language, which is much more precise than our in-door manually built one. Therefore, our *Animal Domain Ontology*, which is used to describe the taxonomy information of animal concepts, can be directly obtained from Wikipedia category. However, as the Wikipedia concepts under *animal* domain have some special content features, we use the *Scientific Classification* entry on each concept page as a shortcut.

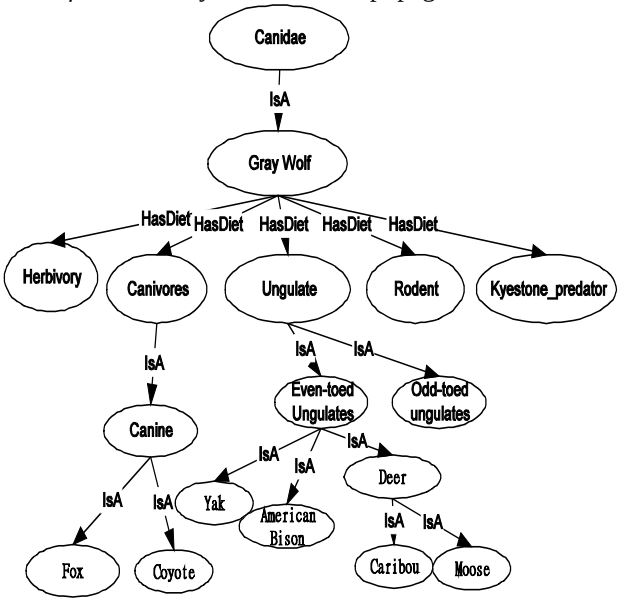


Figure 4: Knowledge resource structure in our system.

This entry provides animal taxonomy in a top-down manner, from *Kingdom*, *Phylum*, *Class*, *Order*, *Family*, *Subfamily*, *Genus* to Species. We then extract the hierarchy structure from this entry and form our *Animal Domain Ontology*. For example, *Phylum* is defined as a sub–class of *Kingdom*, while *Class* is defined as a subclass of *Phylum*. Since our ontology is only defined for general web image classification, we stop at *Family* level and do not go beyond *Subfamily*. Taking *Aardwolf* as an example, this concept belongs to the family of *Hyaenidae*. So when an input query suggests a concept of *Hyaenidae*, Aardwolf will also be considered as a matched concept.

4.2 Narrative descriptions extraction

In the definition of *ontology*, what is alluded to but not formally stated is the modelling of concept relationships. In order to show that ontology is more than a set of related keywords, we have to prove that every concept in the ontology is different from a plain word. It should be understood as concepts supported by structures. When building the *Textual Description Ontology*, our main concerns are twofold: an ontology, which depicts the real world, should contain more descriptive concepts and relationships. These relationships convey general knowledge according to domain knowledge. On the other hand, the related concepts should contain a hierarchical structure, so that when we do concept inference, additional facts could be generated. Here is an example to illustrate the above concerns. *South Africa* is where the species *cape fox* lives. Therefore, *South Africa* is linked to *cape fox* with a named relationship *hasDistribution*. Given two other relations *Zimbabwe* is a part of *South Africa* and *South Africa* is part of *Africa*, one could reasonably infer that *cape fox* can also be found in *Zimbabwe*. And this possibility increases when additional information matches. Therefore, the first step is to find all the important terms. Some pre-process includes crawling Wikipedia web page of relevant concept and using HTML parser to filter irrelevant HTML codes. After that, we analyse the web page content to extract useful concepts and relationships. It is worth noticing that at the beginning of each web page, where a short paragraph is given as a brief introduction of the particular concept, some words are emboldened as alternative name or synonymy to the main concept. By extracting these words, a synonymous set is first constructed for the original concept. We use a *hasName* relationship to link it to the original concept. This relationship extends the naming information. In the next step, by analysing HTML tags of document title, section title and links to other pages, we locate the title of each section. Before we look into the details of the section content, we exam the section title to see if it contains relevant semantic relationships, like information about *Distribution*, *Habitat*, *Diet*, etc. Once the relevant keywords are discovered in the section title, we look into the details of the section and find candidate concepts for that particular relationship. Candidate concepts are defined as those that have their own Wikipedia web pages. For the normal

plain text on the Wikipedia web page, we believe it is of trivial importance, thus has less contribution to the concept detection. Based on this assumption, we extract a set of concepts from the section for each relationship. While not all the candidate concepts are correct, an association rule mining is discussed later to improve the accuracy of the generated ontology.

After the relationships and related concepts are collected, we do further hierarchical construction among all the concepts. This step is done based on the Wikipedia category structure, which offers a systematic categorization of all the concepts. The category information is listed as a separate section at the bottom of each Wikipedia web page. In most cases one Wikipedia concept belongs to several categories, some of which serve for Wikipedia administration purposes, such as *Wikipedia administration*. We remove these categories and keep the rest, which follow different categorical classification. And for each related category, we move one step further to find its parent category. In our current implementation we do five iterations, and construct a hierarchical structure of five levels for each concept. This step helps to formulate the information and introduce more structured concepts on top of the current ontology. To evaluate the performance of the proposed ontology system with other textual aware methods, we also follow the text processing part of [10] and use Latent Dirichlet Allocation(LDA) to find 10 latent topics from the web page text. And we take the top 20 words from each latent topic as the topic representation. However, the resulting clusters of words do not show explicit semantic meanings. We presume that it is due to the relative smaller size of text corpus. Therefore, the ontology approach is more appropriate on our median-sized data set.

4.3 Visual descriptions extractions

In this section we discuss the visual description features for our concept ontology. We collect a median size collection of 4,000 animal web images together with the corresponding web pages as our experiment data set. More specifically, the data set contains 20 animal categories under the domain of canine. For our experiment, we use recognition techniques to build a visual vocabulary and train classifiers using support vector machine (SVM). We do not generate our own object detection techniques as these techniques have been extensively discussed in computer vision researches. Our aim is also to show that instead we follow the object detection techniques whose superiority has been proved in the latest researches[11]. We first use Harris-Laplace detector[12] which is scale invariant and detects corner-like regions in the images as interest point and then use SIFT[13] descriptor to represent the shape information around the interest point. Color descriptor is also combined with SIFT descriptor. A 20 by 20 image patch around the centre of the interest point is generated to extract *opponent angle* features. In addition, a shift along the horizontal or vertical axis is made when boundary is within the patch range. The final descriptor is a vector of dimension 164, where 128 dimensions are from SIFT descriptor and 36 dimensions are from *opponent angle* descriptor. We build a vocabulary of 1,000 visual words based on k-means clustering result of feature vectors from all images. For each image in the data set, a histogram of visual words is calculated and then each image is represented by a vector whose dimension is 1,000. After feature space construction, half of the data set is used as training sample, which is of size 2,000. The training set is further divided into 5 parts for cross validation. After training, the relations between image feature concepts and the *animal* concepts are obtained.

After construction, we use association rule mining to refine the initial ontology.

5 Association rule mining for ontology

Wikipedia is an online collaborative work and the content is maintained by users, therefore a certain level of inherent noise must be expected. When we extract real-world relations besides hypernymy and meronymy relations into the ontology, we are extracting those relations from the Wikipedia web pages with text analysis techniques. A small set of wrong relations could be extracted either due to the complexity or correctness of the texts and the strategy we used for relation extraction. For association rule mining[14], the research has evolved from a flat structure with a fixed support value to variances that consider complex tree or graph structure with different support values. In order to enhance the correctness of semantic relations extracted, we develop a variant of association rule mining method which considers the hierarchical structure of the ontology and propose a new quality measure called Q measure for relation pruning.

Here, we use Figure 5 to illustrate the idea of the *Q* measure. We can see concept *Even-toed Ungulates* has three children in the ontology, namely *Deer*, *Yak*, and *American Bison*. If the relation *Gray_Wolf hasDiet Even-toed Ungulates* is correct, the three relations *Gray_Wolf*

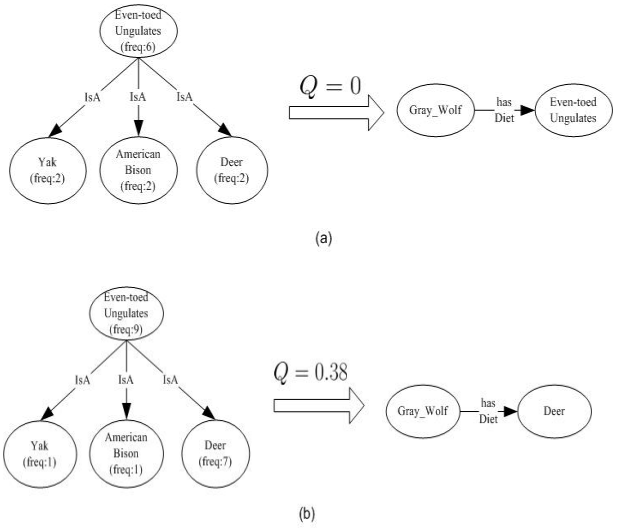


Figure 5: An example for association rule mining.

hasDiet Deer, *Gray_Wolf hasDiet Yak*, and *Gray_Wolf hasDiet American Bison* should also be correct if a minimum support level is present. Given a sufficient large number of documents collected, the three relations should have the same frequencies (i.e., the expected value of 1/3). But in realty and with a smaller number of documents, the three relations have different frequencies. We therefore could compute a variance-like value *Q* by

$$Q = \sum_{i=1}^{N} [\frac{freq(C_i)}{freq(R)} - \frac{1}{N}]^2 \qquad (1)$$

where C_i represents a child rule of a generalized rule R, and N is the number of children rules of R.

For those relations with parent concepts, they would have a lower Q value although they have high frequencies. We therefore can efficiently remove those relations by looking at the Q value and the predefined support threshold.

6 Experiment result

In this section, we evaluate the performance of our approach by using the built AutoOnto and AutoMMOnto for image retrieval.

The matchmaking of concept ontology is defined as a process that requires the user specified domain concept repository to take an image's detected concept as the input, and return all the matched domain concepts which are compatible with the concept generated from the input concept. From the matchmaking result we can conclude which predefined concept the generated image concept corresponds to and what relationship can be find between two given concepts. In this step, reasoners(semantic matchmakers) are used to derive additional facts which are entailed in any optional ontologies and predefined rules, through process and reason over the knowledge encoded in the ontology language. We use both the description logic reasoner RACER[15] and an enhanced ranking algorithm [9] in the experiment. The matched concepts are attached with the web images as semantic labels.

There are 20 classes of web images in our database and each class has 200 web images downloaded from Google Image Retrieval. The performance is computed using Average Precision (AP), which is defined as the average (interpolated) precisions at certain recalls

$$AP = \frac{1}{\min(R,k)} \sum_{j=1}^{k} P(r_i) I_j ,$$

where R is the total number of correct images in the ground truth, k is the is number of current retrievals, $I_j = 1$ if image ranked at jth position is correct and $I_j = 0$ otherwise, $P(r_i) = \frac{R_j}{j}$ is the interpolated precision, and {r,P(r)} are the available recall-precision pairs from the retrieval results. By using AP, the PR curve can be characterized by a scalar. A better retrieval performance, with a PR curve staying at the upper-right corner of the PR plane, will have a higher AP, and vice versa. In the current experiment, we set j = 200. As MAP is sensitive to the entire ranking with both recall and precision reflected in this measurement, we will also give Mean Average Precision (MAP)

We compare our result to both the Google Image Retrieval results and the manually built ontology results(namely ManuOnto and ManuMMOnto). The corresponding comparisons are shown in Table 1 and Table 2 respectively, where the Average Precision (AP) values for each class using different approaches are presented. From Table 1, we can conclude that text ontology improves the retrieval performance by formulating the text information into structured concepts. From Table 2, we can observe that the AutoMMOnto approach gives comparable performance to the ManuMMOnto approach. And in most classes, AutoMMOnto generates even better results by extracting more concepts from the web page text. And the MAP of the Google, ManuMMOnto and AutoMMOnto are 0.7049, 0.8942 and 0.9125, respectively. The result of MAP also shows an overall improvement. It is worth adding, AutoMMOnto requires minimal level of human involvement: Only the main domain concepts, which are the image classes in our case, are given by users according to experimental domain to build up the whole concept hierarchy in the domain. The result is encouraging, as it proves that it is viable to build large-scale concept ontology from Wikipedia automatically for effective web image retrieval. Ranking results from several sample classes are also shown in Figure 6.

7 Conclusion and future works

In this paper we have proposed Wikipedia2Onto, an approach that uses the content and structure features of the online encyclopaedia Wikipedia to build large-scale concept ontology automatically. The constructed ontology has automatically extracted more descriptive semantic relationships than most existing ontologies. More importantly, this ontology is a ready structure that can be used in semantic inference. Through association rule mining, our approach has detected 743 concepts with high accurate corresponding relations.

Finally, it is shown that our approach will help to improve precise retrieval for images (with free text information) for various domains. The proposed approach largely dispenses with the conflict between cost and precision in ontology-based applications. We would also like to conclude by drawing the attention of the readers to Figure 7. The results from our AutoMMOnto search for “wild dog in Kashmir region” further show the potential of ontology in the better understanding of multimedia.

References

[1] Y. A. Aslandogan, C. Their, C. T. Yu, and N. Rishe (1997) Using semantic contents and wordnet in image retrieval. In SIGIR’97: Proceedings of the 20th annual international ACM SIGIR conference

on research and development in information retrieval, pp. 286 - 295, New York, USA.

[2] M. Marszalek and C. Schmid (2007) Semantic hierarchies for visual object recognition.) In *CVPR'07: Proceedings of IEEE conference on computer vision and pattern recognition*, pp. 1-7, Minnesota, USA,.

[3] X.-Y. Wei and C.-W. Ngo (2007) Ontology-enriched semantic space for video search. In MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia, pp. 981-990, New York, USA.

[4] A. Popescu, P.-A. MoÄellic, and C. Millet (2007) Semretriev: an ontology driven image retrieval system. In CIVR'07: Proceedings of the 6th ACM international conference on image and video retrieval, pp. 113 -116, New York, USA.

[5] L. Khan and F. Luo (2002) Ontology construction for information selection. In ICTAI'02: Proceedings of 14th IEEE International Conference on Tools with Artificial Intelligence, pp. 122-127, Washington, USA.

[6] Y. Labrou and T. Finin. Yahoo! as an ontology: using Yahoo! categories to describe documents. In CIKM'99: Proceedings of the 8th international conference on Information and knowledge management, pp. 180-187, New York, USA, 1999.

[7] H. Wang, S. Liu, and L.-T. Chia (2006) Does ontology help in image retrieval?: a comparison between keyword, text ontology and multi-modality ontology approaches. In MULTIMEDIA'06: Proceedings of the 14th annual ACM international conference on Multimedia, pp. 109-112, New York, USA.

[8] M. Naphade, J. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis (2006) Large-scale concept ontology for multimedia. IEEE MultiMedia Magazine, 13(3), pp. 86-91.

[9] H. Wang, L.-T Chia and S. Liu (2007) Semantic Retrieval with Enhanced Matchmaking and Multi-Modality Ontology. In ICME'07: Proceedings of IEEE International Conference on Multimedia and Expo, pp. 516-518, Beijing, China.

[10] T. L. Berg and D. A. Forsyth (2006) . Animals on the web In CVPR'06: Proceedings of IEEE conference on computer vision and pattern recognition, pages 1463-1470, New York, USA, 2006.

[11] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid (2007) Local features and kernels for classification of texture and object categories: A comprehensive study. In International Journal of Computer Vision, 73(2), pp. 213-238.

[12] K. Mikolajczyk and C. Schmid (2004) Scale & invariant interest point detectors. In International Journal of Comput Vision, 60(1), pp. 63-86.

[13] D. Lowe. (2003) Distinctive image features from scale-invariant keypoints. In International Journal of Computer Vision, volume 20, pp. 91-110.

[14] R. Srikant and R. Agrawal (1995) Mining Generalized Association Rules. In VLDB'95: Proceedings of the International conference on very large data bases , pp. 407-718, Zurich, Switzerland.

[15] V. Haarslev and R. Moller (2001) Racer system description. In IJCAR'01: Proceedings of International Joint Conference on Automated Reasoning, pp. 701-705, Siena, Italy.

[16] P. Martin (1995) Using the WordNet Concept Catalog and a Relation Hierarchy for Knowledge Acquisition. In Proceedings of the 4th Peirce Workshop, Santa Cruz, USA.

[17] V. Varma. (2002) Building large scale ontology networks. In Proceedings of Language Engineering Conference, pages 121-127, Hyderabad, India.

[18] S. Ponzetto and M. Strube (2007) Deriving a Large Scale Taxonomy from Wikipedia. In AAAI'07: Proceedings of the 22nd National Conference on Artificial Intelligence, pp. 22-26, Vancouver, CA, 2007.

[19] D. Lenat and R. Guha (1989) Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

[20] T. Zesch and I. Gurevych (2007) Analysis of the Wikipedia Category Graph for NLP Applications. Proceedings of the TextGraphs-2 Workshop, NAACL-HLT, pp. 1-8, New York, USA.

Table 1: Performance of image classification on single-modality text ontology

Class	Aardwolf	CapeFox	BushDog	ArcticFox	Ethiopian Wolf	Coyote	GrayWolf	GrayFox	FennecFox	SpottedHyena
Google	0.5801	0.4958	0.4695	0.715	0.7516	0.5042	0.7513	0.7183	0.8181	0.8365
ManuOnto	0.6209	0.5446	0.7881	0.7905	0.0.844	0.5421	0.7196	0.6336	0.8145	0.8683
AutoOnto	0.6472	0.5231	0.8422	0.7983	0.7973	0.496	0.7316	0.6465	0.8214	0.9024
Class	Dhole	RedFox	ManedWolf	BlackJackal	Bat-Eared Fox	Dingo	KitFox	RedWolf	GoldenJackal	AfricanWildDog
Google	0.6342	0.744	0.7949	0.8872	0.7967	0.67	0.6698	0.7669	0.7092	0.7844
ManuOnto	0.6598	0.781	0.8565	0.8805	0.7914	0.6799	0.6844	0.715	0.7252	0.7723
AutoOnto	0.6835	0.7522	0.8193	0.8959	0.8396	0.7196	0.6791	0.8175	0.7528	0.7869

Table 2: Performance of image classification on multi-modality ontology

Class	Aardwolf	CapeFox	BushDog	ArcticFox	EthiopianWolf	Coyote	GrayWolf	GrayFox	FennecFox	SpottedHyena
Google	0.5801	0.4958	0.4695	0.715	0.7516	0.5042	0.7513	0.7183	0.8181	0.8365
ManuMMOnto	0.8332	0.8911	0.8087	0.9955	0.9218	0.9058	0.8267	0.935	0.857	0.9391
AutoMMOnto	0.8552	0.8835	0.9302	0.9938	0.9447	0.884	0.8561	0.9766	0.8981	0.942
Class	Dhole	RedFox	ManedWolf	BlackJackal	Bat-EaredFox	Dingo	KitFox	RedWolf	GoldenJackal	AfricanWildDog
Google	0.6342	0.744	0.7949	0.8872	0.7967	0.67	0.6698	0.7669	0.7092	0.7844
ManuMMOnto	0.8184	0.966	0.9508	0.9498	0.9134	0.8108	0.9483	0.8537	0.8962	0.8627
AutoMMOnto	0.8535	0.9526	0.938	0.9555	0.9333	0.8334	0.8821	0.9034	0.9166	0.9185

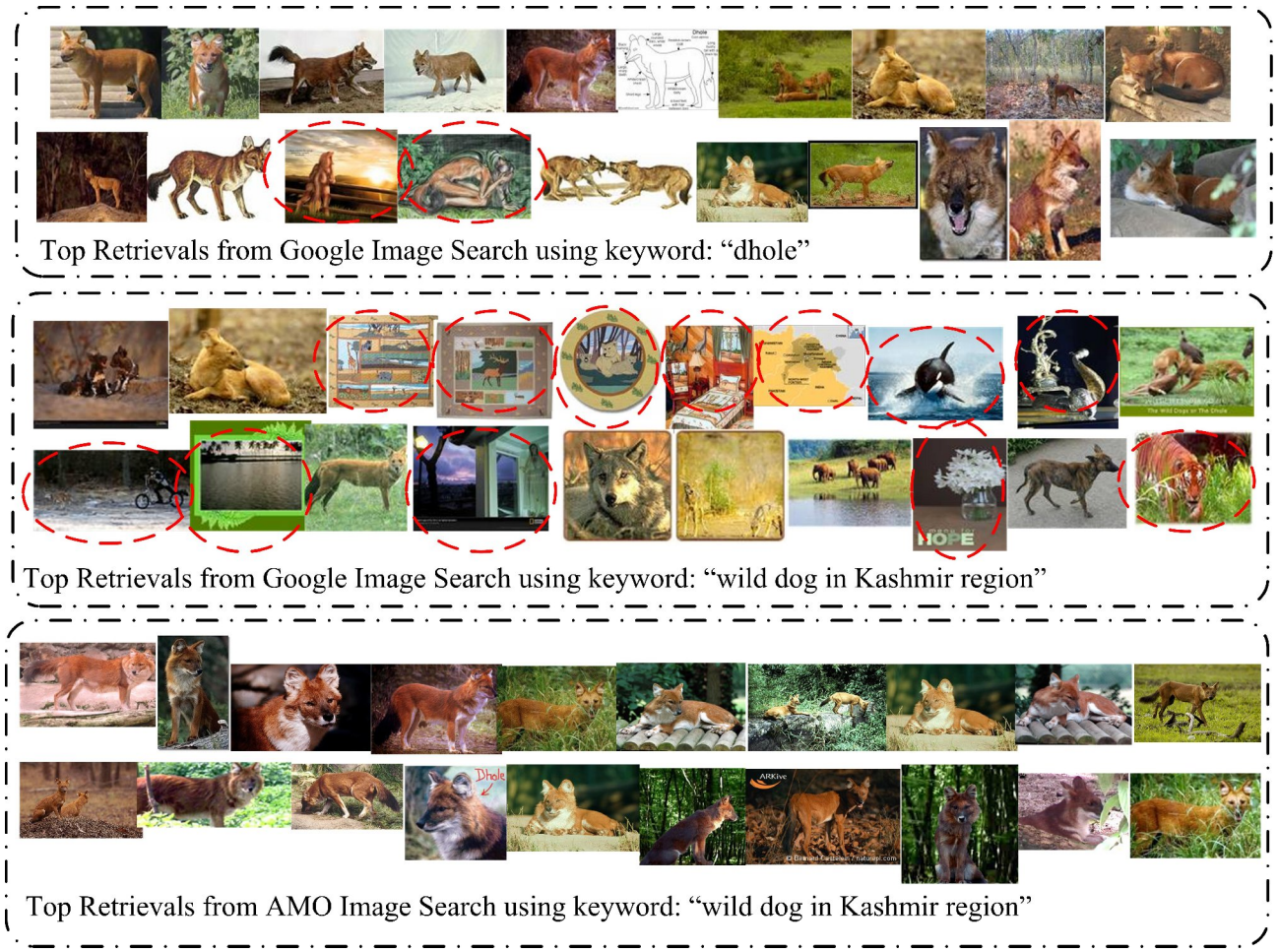
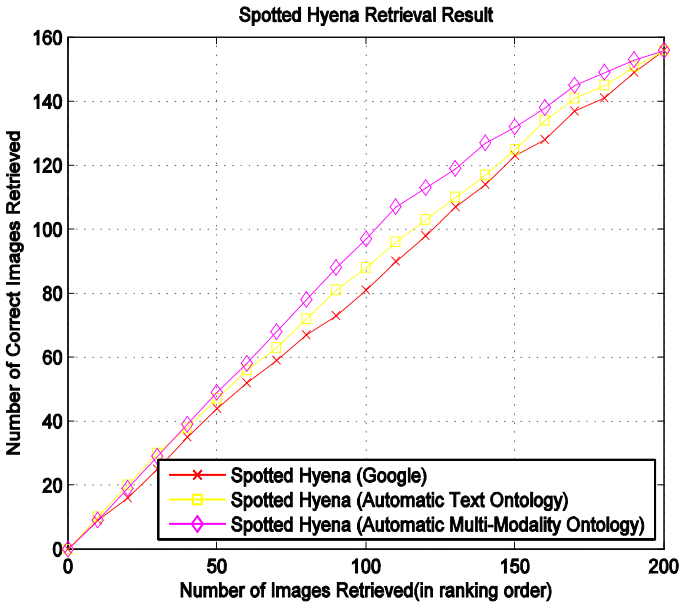
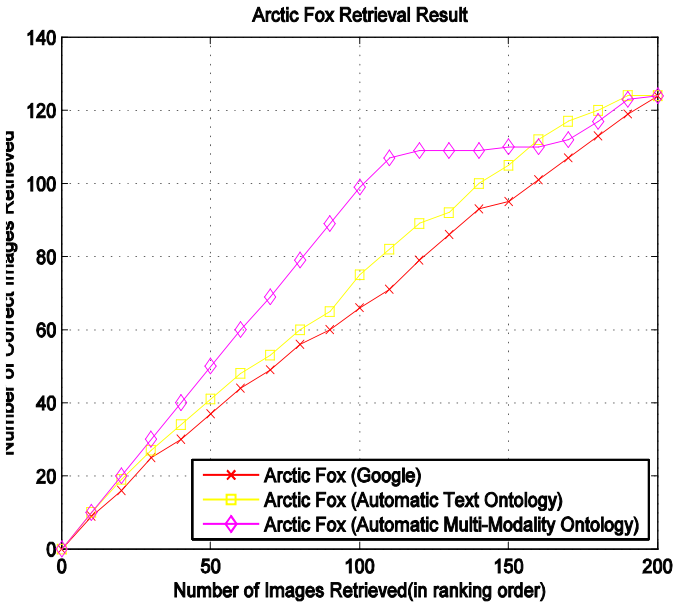
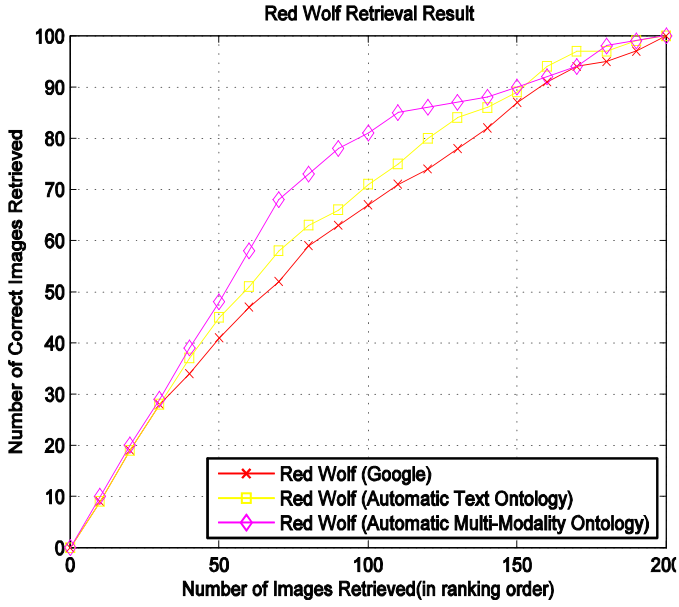
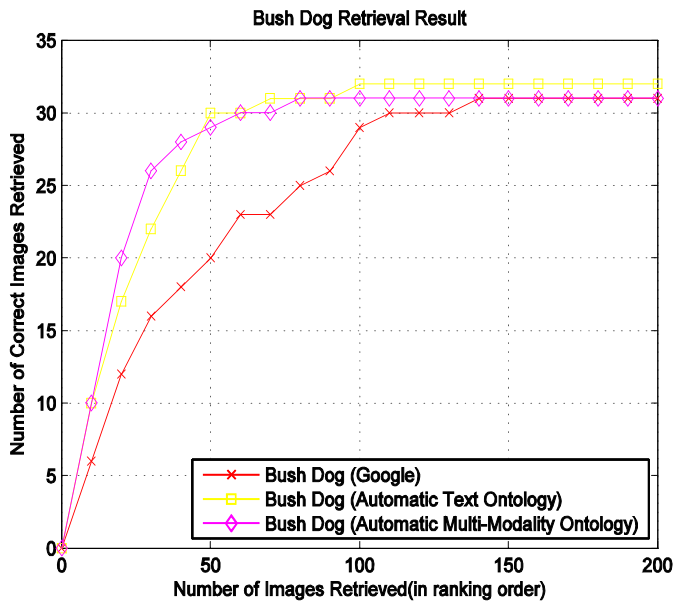


Figure 6: An example of web image classes in our data set. Different results returned for different keywords but it is the same animal from the canine family: Dhole - a wild dog in the Kashimar region.


Spotted Hyena Retrieval Result
Number of Correct Images Retrieved
Number of Images Retrieved(in ranking order)
Spotted Hyena (Google)
Spotted Hyena (Automatic Text Ontology)
Spotted Hyena (Automatic Multi-Modality Ontology)


Arctic Fox Retrieval Result
Number of Images Retrieved(in ranking order)
Arctic Fox (Google)
Arctic Fox (Automatic Text Ontology)
Arctic Fox (Automatic Multi-Modality Ontology)


Red Wolf Retrieval Result
Number of Correct Images Retrieved
Number of Images Retrieved(in ranking order)
Red Wolf (Google)
Red Wolf (Automatic Text Ontology)
Red Wolf (Automatic Multi-Modality Ontology)


Bush Dog Retrieval Result
Number of Correct Images Retrieved
Number of Images Retrieved(in ranking order)
Bush Dog (Google)
Bush Dog (Automatic Text Ontology)
Bush Dog (Automatic Multi-Modality Ontology)

A Service Oriented Framework for Natural Language Text Enrichment

Tadej Štajner, Delia Rusu, Lorand Dali, Blaž Fortuna, Dunja Mladenić and Marko Grobelnik
Department of Knowledge Technologies
Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773419; fax: +386 1 4251038
E-mail: tadej.stajner@ijs.si

Keywords: language processing, information extraction, entity resolution, automatic document categorization

Received: November 18, 2009

This paper describes a text enrichment framework and the corresponding document representation model that integrates natural language processing, information extraction, entity resolution, automatic document categorization and summarization. We also describe the implementation of the framework and give several illustrative use cases where the service-oriented approach has proven to be useful.

Povzetek: Opisan je okvir za obogatitev naravnega besedila.

1 Introduction

Integration and sharing of data across different data sources is the basis for an intelligent and efficient access to multiple heterogeneous resources. Since a lot of knowledge is available in plain text rather than a more explicit format, an interesting subset of this challenge is integrating texts with structured and semi-structured resources, such as ontologies. However, this also involves dealing with natural language, which is an error-prone process.

In our experience, many knowledge extraction scenarios generally consist of multiple steps, starting with natural language processing, which are in turn used in higher level annotations, either as entities or document-level annotations. This in turn yields a rather complex dependency scheme between separate components. Such complexity growth is a common scenario in general information systems development. Therefore, we decided to mitigate this by applying a service-oriented approach to integration of a knowledge extraction component stack. The motivation behind Enrycher[17] is to have a single web service endpoint that could perform several of these steps, which we refer to as 'enrichments', without requiring the users to bother with setting up pre-processing infrastructure themselves.

The next sections will describe the specific components, integration details and some of the use cases that motivated this experiment of integration. Section 2 describes related ideas and their implementations and positions our model within their framework. Section 3 describes the overall architecture, the model and individual components. Section 4 focuses on the applications and use cases of the framework, continuing to the conclusion in section 5 and continueing to the outline of the current and future work that is going on related to the Enrycher framework and system.

2 Related work

There are various existing systems and tools that tackle either named entity extraction and resolution, identification of facts, document summarization. The OpenCalais system [15], for example, creates semantic metadata for user submitted documents. This metadata is in the form of named entities, facts and events. In the case of our system, named entities and facts represent the starting point; we identify named entities within the document, determine the subject - verb - object triplets, and refine them by applying co-reference resolution, anaphora resolution and semantic entity resolution. As opposed to OpenCalais, we continue the pipeline to extract assertions from text which represent newly identified relationships expressed in text. This process enables the construction of a semantic description of the document in the form of a semantic directed graph where the nodes are the subject and object triplet elements, and the link between a pair of entities is determined by the verb. The initial document, its associated triplets and semantic graph are then employed to automatically generate a document summary. The resulting triplets are then in turn used to construct a semantic graph, an effective and concise representation of document content [12].

The enrichment where we provide a set of triplets in the form subject, predicate, object can be also described as a form of information extraction, since we are extracting structure from inherently unstructured data. Information extraction approaches can be distinguished

in several ways. The first distinction comes from the way the background knowledge is being used.

Open information extraction approaches do not assume anything about the relations they might encounter and do not require training data. Their main characteristic is that they work accross a wide array of domains, it is a trait which we wanted to keep in Enrycher. Another prominent example in this area is TextRunner [18],

Supervised information extraction approaches, such as WebKB [19], are suitable for domain-specific text, since they require training data for each relationship that we wish to extract, although more recent approaches tend to focus on approches which require less human supervision.

Semi-supervised information extraction approaches are most often implemented via starting with a bootstrap training set and learning new patterns along the way, for example, the recent ReadTheWeb approach [20]. While it still requires the concrete relations to be specified via the training set, it requires much less human supervision to improve the extraction by learning. Recently, hybrid approaches combining open and bootstrapped information extraction have started to appear [21]. Since the overall tendency in the area of information extraction is reducing the human involvement within the extraction process, we positioned Enrycher's information extraction capabilities to be closer to an open information extraction system which would use not only literal pattern matching for information extraction but also extracting patterns with the help of part-of-speech information. This enables us to extract statements that would be otherwise missed by literal patterns and to provide us with a starting point for language independence.

Another difference within information extraction approaches is the treatment of extracted terms, such as named entities:

Well-defined entities and relationships are a property of the knowledge model which asserts that a single term has only a single meaning. In that case, we refer to terms as entities. We achieve this property by performing entity resolution. However, this is not always necessary and depends on the desired use of the knowledge base. For instance, if a knowledge base is exposed as an information retrieval engine, such as TextRunner [18], StatSnowball [21] and Read The Web [20], ambiguity is not a significant issue. However, if we wish to use the knowledge base to perform logical inference or any other task with similar demands, entity and relationship resolution is necessary, as demonstrated in SOFIE [22]. Enrycher uses a combination of both - named entity extraction may detect more terms that the entity resolution can then resolve into concrete entities, allowing for both to co-exist.

3 Architecture

The process underlying the proposed framework consists of several phases, each depending on the output of the previous one.

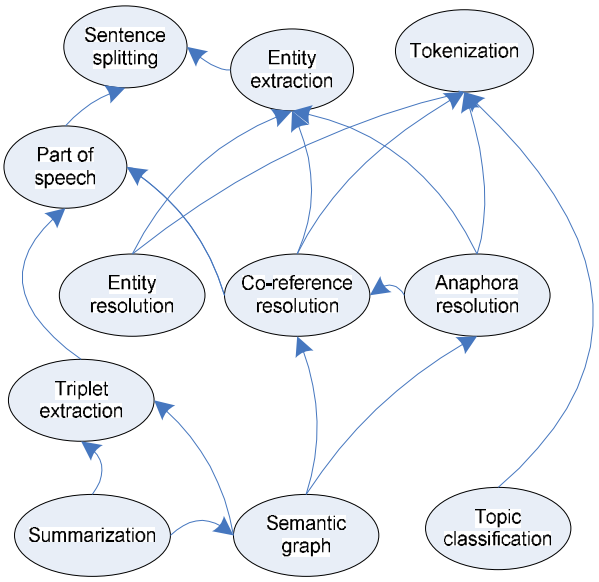


Figure 1: Component dependency graph.

The dependencies between components can be illustrated by the chart in Fig. 1. The architecture can be roughly split into four phases which correspond to four different parts of the proposed document enrichment model.

3.1 Model schema

To summarize, the schema that is used in the inter-service communication is abstracted to the point that it is able to represent:

1. *Text*: sentences, tokens, part of speech tags.
2. *Annotations*: entities and assertion nodes, identified in the article with all identified instances, possibly also with semantic attributes (e.g. named entities, semantic entities).
3. *Assertions*: identified <*subject, predicate, object*> triplets, where subjects, predicates and object themselves are annotations).
4. *Document-wide metadata*: identifier, document-wide semantic attributes (e.g. categories, summary).

The first part of the enrichment model is the linguistic description of the text containing the output of initial natural language. The data structures at this level are blocks (i.e. paragraphs, titles), which are then split into sentences and later into individual words.The second part of the enrichment model is the annotation section, containing information on individual objects that have been detected in the document one way or another, for instance, named entities and other semantic graph nodes. The annotations themselves contain a list of their instantiations within the document and a set of associated semantic attributes, meant for describing them and linking them to an existing ontology

The third part of the enrichment model is the assertions section, containing the triplets that construct the semantic graph. This represents the individual

information fragments that were extracted from the plain text and form the basis for new knowledge.

Furthermore, each document contains a document metadata section storing attributes that apply to the document as a whole, such as categories, descriptive keywords and summaries. The next sections will individually describe each of the parts and outline the approaches that have been employed to implement the actual software that provides data for the enrichment model.

3.2 Language-level processing

While language-level features usually aren't explicitly stated as a requirement in most use cases, they are instrumental to most of the further enrichments that are required in those use cases:

Sentence splitting
Tokenization
Part of speech tagging
Entity extraction

This is also the layer that is the most language-dependent, providing an abstraction to the other layers. Ideally, the tools above this layer should be reasonably lanugage-agnostic, although some concrete implementation approaches might not make this entirely possible. For instance, semantic entity resolution requires a background knowledge base to map from concrete phrases to concepts, requiring the phrases to be in the language that is used. On the other hand, if the triplet extraction layer works only on top of part-of-speech, such as the one implemented within Enrycher, it is reasonably language-agnostic, since it operates at a higher level than matching literal strings.

3.3 Entity-level processing

Whereas the language-level processing step identified possible entities, the purpose of this phase is to consolidate the identified entities. This is done with anaphora resolution, where pronoun mentions are merged with literal mentions, co-reference resolution that merges similar literal mentions and entity resolution, which links the in-text entities to ontology concepts. Since entity extraction is often handled with several domain-specific extractors, the purpose of this layer is to allow multiple extraction mechanisms and consolidate their output into a coherent set of entities and if possible, linking them to ontology concepts.

3.3.1 Named entity extraction

We gather named entities in text using two distinct approaches to named entity extraction, a pattern-based one [9] and a supervised one [10]. This step is done to gather as much annotations as possible to have a rich node set for the semantic graph.

3.3.2 Anaphora resolution

Anaphora resolution is performed for a subset of pronouns *{I, he, she, it, they}* and their objective, reflexive and possessive forms, as well as the relative pronoun *who*. A search is done throughout the document for possible candidates (named entities) to replace these pronouns. The candidates receive scores based on a series of antecedent indicators (or preferences): givenness, lexical reiteration, referential distance, indicating verbs and collocation pattern preference [1].

3.3.3 Co-reference resolution

Co-reference resolution is achieved through heuristics that consolidate named entities, using text analysis and matching methods. We match entities where one surface form is completely included in the other, one sufrace form is the abbreviation of the other, or there is a combination of the two situations described in [1]. This is useful for reducing the size of the graph, as it enables us to merge many nodes into one, such as merging mentions, such as *"Mr. Norris"* and *"Chuck Norris"* into a single entity.

3.3.4 Semantic entity resolution

Rather than just extracting information from text itself, the motivation behind entity resolution is to integrate text with an ontology. This consists of matching previously extracted named entities to ontology concepts. Since named entities are often ambiguous, especially in multi-domain ontologies, such as DBpedia [13], we have to employ sophisticated methods to determine the correct corresponding semantic concept of a named entity. The underlying algorithm uses ontology entity descriptions as well as the ontology relationship structure to determine which are the most likely meanings of the named entities, appearing in the input text. Because the approach is collective, it does not treat distinct entity resolution decisions as independent. This means that it can successfully exploit relational similarity between ontology entities, it means that entities which are more related to each other, tend to appear more ofter together.This is explored in further detail in [11], with concrete implementation details in [23] and [6], where the improvement in entity resolution quality with using heterogeneous graph structure is demonstrated.

3.4 Entity graph processing

3.4.1 Triplet extraction

The triplet is a semantic structure composed of a subject, a verb and an object. This structure is meant to capture the meaning of a sentence. We try to extract one or more triplets from each sentence independently. Two approaches to triplet extraction have been tried, both of which take as input a sentence with tokens tagged with their part of speech.

Triplet extraction by deep parsing was the first approach that was tried. In this approach the sentence is

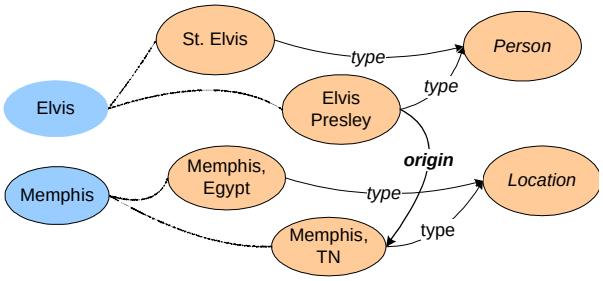


Figure 2: An example of an entity resolution scenario, where the blue nodes represent in-document extracted entities and the pink nodes represent the ontology concepts to which we align our document.

parsed, and then the triplets are extracted based on the shape of the parse tree obtained. The rules of triplet extraction from a parse tree are explained in detail in [5]. *Triplet extraction by noun phrase chunking and pattern matching* was employed in order to avoid the performance bottleneck introduced by deep parsing, we tried another approach where instead of parsing, we only do noun phrase chunking on the input sentence. The result of chunking is a sequence of tags on which pattern matching rules are applied in order to find the triplets which must be extracted. This pattern matching rules are similar to regular expressions applied on text. The difference is that as opposed to regular expressions which have as the processing unit a character, the triplet extraction rules recognise the tags as the smallest units which can be matched. An example:

```
  Triplet <-
{:subject:<NP>}(<PP><NP>)*
{:verb:<VP>(|<JJ.*>)*}
{:object:<NP>+}
```

Figure 2: Triplet extraction pattern

The second approach brings an important speedup to the triplet extraction process. However, due to the sequential structure of the chunked sentence, it loses some of the representational power when compared to the richer structure of a parse tree. This is why it is more difficult, if not impossible, to find some of the triplets in a chunked sentence than finding them in a parsed sentence. Another advantage of the chunked approach is that the pattern matching rules, such as in Fig. 2, are easier to understand and extend.

3.5 Document-level processing

While the language-level processing operates on the token and phrase domain and the entity-level processing operates on the in-text entities and concepts, the document-level processing uses the preceding enrichments to annotate the document as a whole.

3.5.1 Semantic graph visualization

The semantic representation of text is achieved through linking triplet elements together, where the nodes are represented by the subject and object elements, and the relationship between them is linked with the corresponding verb. The yielded graph is a directed one, from the subject element to the object one.

Thus we can represent plain-text in a more compact manner that enables visual analysis, highlighting the most important concepts and the relations among them. An example is illustrated in Fig. 3.

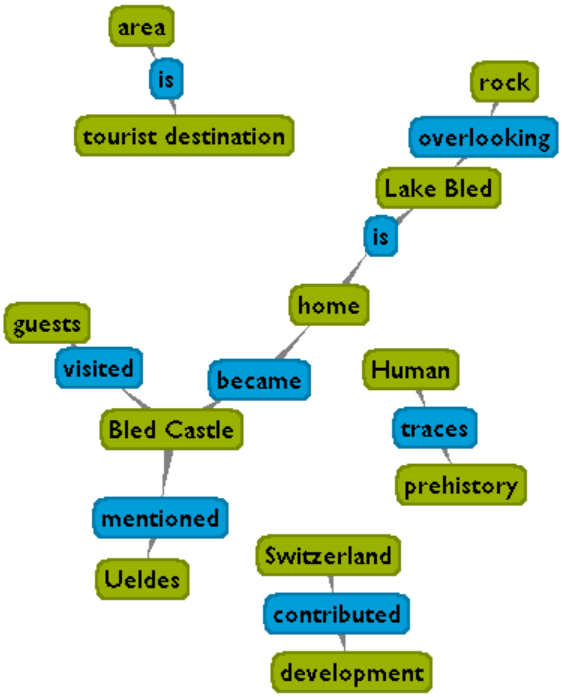


Figure 3: Example of a semantic graph visualization: Wikipedia article on Bled

3.5.2 Taxonomy categorization

A common use case in working with documents is classifying them in categories. This component annotates the input text with a hierarchical centroid classifier which chooses relevant categories based on word and phrase similarity [14]. The current on-line implementation uses the Open Directory as an example of a taxonomy. The same component also provides descriptive keywords, generated from segments of the category names. For instance:

Categories:

- Top/Reference/Knowledge_Management/Knowledge_Representation
- Top/Reference/Knowledge_Management
- Top/Reference/Knowledge_Management/Knowledge_Representation/Ontologies
- Top/Reference/Knowledge_Management/Knowledge_Discovery

Keywords:

- Knowledge Management,
- Knowledge Representation,
- Ontologies

In this way it is possible to provide enrichments that do not necessarily correspond to an explicit mention of an entity but rather to the document as a whole, such as document classification labels.

3.5.3 Content summarization

The document's semantic graph is a starting point for automatically generating the document summary. This summarization approach is based on the idea that more imporant entities are positioned more centrally in the semantic graph which represent the central topic of the document. Therefore, if we reduce the graph in less dense areas, we are removing less important sentences.The model for summary generation is obtained by machine learning, where the features are extracted from the semantic graph structure and content [1].

3.5.4 Other options

Although not directly implemented in the Enrycher system, the document model is sufficient to represent also other enrichments. For instance, other document-wide enrichments can be the automatically detected language of the document, the detected sentiment of the document, references to possibly related documents and similar. The assertions are also able to represent RDF assertions, making it amenable to Semantic Web-oriented applications and knowledge acquisition.

4 Use cases

The system abstracts the setup and workflow from the user by exposing only a single web service endpoint, which in turn pipelines the request thorough other web services. All communication is done with REST-like XML-over-HTTP requests.

4.1 Visual analytics

Visual analysis of documents based on the semantic representation of text in the form of a semantic graph can aid data mining tasks, such as exploratory data analysis, data description and summarization. Users can thus get an overview of the data, without the need to entirely read it. This kind of concept overview offers straightforward data visualization by listing the main facts (the triplets), linking them in a way that is meaningful for the user (the semantic graph), as well as providing a document summary [4].

4.2 Semantic integration of text and ontologies

An important part of information systems integration is providing interoperability of data. This is a major issue when dealing with plain text, because it is inherently unstructured. On the other hand, one of the most pragmatic approaches is representing knowledge in a common ontology. Therefore, we designed our system to not only identify and consolidate named entities in text but use the semantic entity resolution component to match it with ontology concepts, which enables us to represent nodes in the graph as semantic concepts. This can be an important aid in constructing ontologies from textual data.

4.3 Question answering

Document enrichment techniques such as triplet extraction and semantic graphs have been applied to build a question answering system [3]. The use case is that the answer to a natural language question is searched in a collection of documents from which triplets have been previously extracted. Triplets, possibly incomplete, are also extracted from the question, and they are matched against the triplets extracted from the documents to find the answers.

4.4 Story link detection

A task related to news mining and analysis is story link detection [7], where the objective is to identify links between distinct articles that form a coherent story. [2] shows that enriching the text with entity extraction and resolution improves story link detection performance. This indicates that such enrichment on documents may also be beneficial for other topic detection and tracking or semantic search tasks.

5 Conclusion

The main contribution of this paper is an integrrated framework for enriching textual data based on natural language information extraction to include more structure and semantics. We implemented the proposed framework in the system, named Enrycher, which offers a user-friendly way to qualitatively enhance text from unstructured documents to semi-structured graphs with additional annotations. Since the system offers a full text enrichment stack, it makes the system simpler to use than having the user to implement and configure several processing steps that are usually required in knowledge extraction tasks. We described various use cases in both research and applied tasks which we were able to solve with the use of Enrycher as infrastructure.

Furthermore, such systems can be used as infrastructure for knowledge acquisition. Recently, much emphasis in both the academic and industrial communities has been given to the Linked Open Data [24], proposing common RDF-based vocabularies for databases to allow easy integration. The consequence of this for knowledge extraction engines is that it is desired to have a knowledge representation rich enough to be expressed in RDF, that inevitably means resolving phrases to entities and verbs to well-defined relations, identifiable by Unique Resource Identifiers. This push on stricter data representation also brings along more difficult knowledge acquisition. Weaker representations can tolerate wrong assertions more easily, since they make less assumptions about their truth value. On the other hand, ontologies assume that the statements are true in their model, this can lead to erratic behaviour of applications, depending on those assumptions. We therefore expect to see further work on constrained

ontology population from extracted information. On the other hand, our paper barely touches the possibilities that could be employed by using globally identified data approaches, opening way for better data integration, visualization and using annotated documents to enable semantic search. We expect that the proposed semantic article enrichment method will yield even more improvement on tasks that depend on the added semantic information, such as document summarization, triple extraction and recommendation systems.

6 Current and future work

A use case for Enrycher in a related domain of computational linguistics is evaluating local discourse coherence of text. This is an intrinsic measure that indicates readability of text. Since it is automatic, it is also convenient for large-scale evaluation of automatically generated text. The concrete method is based on detecting rough shifts in entity mentions and short entity topics as indicators of poor coherence. As Enrycher supplies grammar roles and entities in triplets, we can match them to the sentences they've been extracted from and evaluate discourse coherence.

Another interesting research area that we are currently tackling is extracting knowledge from large-scale document collections, such as news corpora, where we are exploring possible usability and visualization improvements. Since we extract triplets and possibly resolve their nodes to semantic concepts, we can create new ontologies from corpora of text automatically. Since we are able to do semantic entity resolution, we can also perform alignment of newly extracted ontologies with other ontologies.

Our ongoing work is on developing additional applications that use Enrycher at their cores. One such example is a mobile RSS news reader, which leverages Enrycher to perform text summarization on news items to make them more suitable to consume on a screen space constrained mobile device.

Other future work will focus on using Enrycher as an automated approach to knowledge acquisition which will be able to use the obtained knowledge to improve its output quality. Another path of possible improvement is to test the language-independence aspect of higher-level processing stages, such as anaphora and coreference resolution and also semantic entity resolution and demonstrate whether this sort of framework is able to support multiple different languages withing its processing pipeline, an important requirement for using data sourced from the Web.

Acknowledgement

This work was supported by the Slovenian Research Agency and the IST Programme of the EC under ACTIVE (IST-2008-215040) and PASCAL2 (IST-NoE-216886).

References

[1] D. Rusu, B. Fortuna, M. Grobelnik and D. Mladenić: Semantic Graphs Derived From Triplets With Application. *In Document Summarization. Informatica Journal,* 2009

[2] T. Štajner, M. Grobelnik: Story Link Detection with Entity Resolution. *Semantic Search at WWW2009, Madrid, Spain*, 2009

[3] L. Dali, D. Rusu, B. Fortuna, D. Mladenić, M. Grobelnik: Question Answering Based on Semantic Graphs. *Workshop on Semantic Search at WWW2009, Madrid, Spain*, 2009

[4] D. Rusu, B. Fortuna, D. Mladenić, M. Grobelnik and R. Sipoš, Visual Analysis of Documents with Semantic Graphs. *VAKD '09 at KDD-09*

[5] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, D. Mladenić: Triplet extraction from sentences, Proceedings of the 10th International Multiconference on Information Society, *SiKDD 2007subconference.*

[6] T. Štajner: From unstructured to linked data: entity extraction and disambiguation by collective similarity maximization. In *Identity and reference in web-based knowledge representation at IJCAI 2009*

[7] J. Allan. Introduction to Topic Detection and Tracking. *Kluwer Academic Publishers, Massachusetts, 2002, pp. 1–16.*

[8] J.J. Thomas and K.A. Cook. A Visual Analytics Agenda. *IEEE Comput. Graph. Appl.* 26, 1 (Jan. 2006), 10-13.

[9] H. Cunningham, GATE, a general architecture for text engineering, *Computers and the Humanities, 2002*

[10] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.*

[11] X. Li, P. Morie, and D. Roth, Semantic integration in text: From ambiguous names to identifiable entities,*" AI Magazine. Special Issue on Semantic Integration, vol. 26, no. 1, pp. 45{58, 2005.*

[12] I. Herman, G Melançon, M.S. Marshal: Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics, 2000.*

[13] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, Dbpedia: A nucleus for a web of open data, *Lecture Notes in Computer Science, vol. 4825, p. 722, 2007.*

[14] M. Grobelnik, D. Mladenić. Simple classification into large topic ontology of Web documents. In *Proceedings: 27th International Conference on Information Technology Interfaces, 20-24 June, Cavtat, Croatia, 2005*.

[15] OpenCalais, *http://www.opencalais.com/*

[16] R. Barzilay, M. Lapata. Modeling Local Coherence: An Entity-Based Approach. In *Computational Linguistics*, Vol. 34, No. 1, pp. 1-34, *2008*

[17] Enrycher, *http://enrycher.ijs.si*

[18] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. *Proc. of IJCAI, 2007.*

[19] R. Ghani, R. Jones, D. Mladenic, K. Nigam, and S. Slattery. Data mining on symbolic knowledge extracted from the web. *In Workshop on Text Mining at the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. 56.* Citeseer, 2000.

[20] Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information extraction. *In Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*, 2010.

[21] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.R. Wen. StatSnowball: a statistical approach to extracting entity relationships. *In Proceedings of the 18th International Conference on World Wide Web*, pp. 101-110. ACM New York, NY, USA, 2009.

[22] F.M. Suchanek, M. Sozio, and G. Weikum. SOFIE: a self-organizing framework for information extraction. *In Proceedings of the 18th International Conference on World Wide Web*, pp. 631- 640. ACM, 2009.

[23] Tadej Štajner, Dunja Mladenić: Entity Resolution in Texts Using Statistical Learning and Ontologies, *In Proceedings of the 4th Asian Semantic Web Conference*, pp. 91-104, 2009

[24] C. Bizer, T. Heath, and T. Berners-Lee. Linked data- the story so far. *International Journal On Semantic Web and Information Systems*, 2009.

Applications of Semantics in Agent-Based Manufacturing Systems

Marek Obitko, Pavel Vrba and Vladimír Mařík
Rockwell Automation Research Center
Pekařská 695/10a, Prague, Czech Republic
E-mail: {mobitko, pvrba, vmarik}@ra.rockwell.com

Miloslav Radakovič and Petr Kadera
Department of Cybernetics, Czech Technical University in Prague
Technická 2, 166 27 Praha 6, Czech Republic
E-mail: {radakm1@, kaderp1}@fel.cvut.cz

Keywords: semantics, ontologies, industrial systems, distributed systems, multi-agent systems

Received: October 30, 2009

Distributed intelligent control systems compared to traditional centralized manufacturing architectures provide much more powerful instruments for developing robust, flexible and reconfigurable factory automation systems. The basic characteristic of any distributed system is a communication between the system's components needed for information exchange and coordination of activities for accomplishing collective goals. To achieve effective knowledge exchange and integration in open, reconfigurable environments, an explicit definition of semantics is needed to capture the data and information being processed and communicated. The paper shows how semantics and ontologies can be employed in industrial systems, considering particularly distributed, agent-based solutions. A new manufacturing ontology providing semantic model of production planning and scheduling, material handling and customer order specification is presented. Its integration with an agent-based simulation and control system MAST is demonstrated.

Povzetek: S pomočjo ontologij in semantike je izdelan vmesnik za agentni sistem.

1 Introduction

It is not easy to avoid starting the paper about holonic and multi-agent control systems with the usual words about the growing requirements on flexibility, reconfigurability and robustness, which can be hardly met with traditional centralized control systems [26] . Although the classical architectures based on PLCs and IEC 61131-3 programming languages are still predominantly used in industry and there is a very little number of real deployments of these new approaches [39] , this paper tries to document another significant step that the research in the field of intelligent manufacturing systems is going to take.

This area becomes strongly influenced by the recent advances in semantic technologies, like semantic web and semantic web services. In the past a strong emphasis was put on creation of standards for interoperability in heterogeneous interacting systems. A significant standardization endeavor aimed at physical applications of multi-agent systems was undertaken within the FIPA consortium [http://www.fipa.org/]. The adoption of FIPA standards and usage of compliant agent platforms like JADE or FIPA-OS became very popular for the implementation of intelligent control systems. In particular, the acceptance of Agent Communication Language [13] was seen as a way of ensuring interoperability in heterogeneous agent systems developed by various bodies. However, the interoperability within the FIPA context is guaranteed only at the syntactical level. Well defined syntax of a message offering elements like sender, receiver, communication protocol and content provides a simple way for composing a message structure by the sender and its explicit comprehension by the receiver.

Nevertheless, a full interoperability can be attained only if the receiver understands also the content of the message. This can be achieved by introducing a common semantics providing explicit and machine processable description of the objects and their relationships appearing in the selected domain. Shifting the attention to semantic techniques has been a natural step in the evolution of multi-agent systems. Unfortunately, FIPA did not catch up to this trend what we assume was one of the factors of gradual declination of activities around this organization. It is not only messaging but also gathering, internal representation and processing of knowledge about own states and goals as well as about conditions of the surrounding environment where semantic technologies can be advantageously applied. As it is shown in this paper, there has been a remarkable increase in pilot applications of ontologies in agent-based industrial control systems over the past few years.

In addition, the obsolete FIPA standards for messaging and agent services registration and lookup are also to be replaced. Increasing number of researchers has become to realize many similarities between mutli-agent systems and service-oriented architectures (SoA) [32] and forecast a high potential of the synergy of these two approaches. Basically, agents provide capabilities to other agents in the same way as services are provided in service-oriented systems and also use messages to exchange data. Moreover, when considering increasing requirements for seamless integration of control system with enterprise business systems the SoA represent a suitable technological framework for implementing the agent-based rapidly reconfigurable factory automation systems [24] .

The intention of this paper is to provide an overview of current state-of-the-art in application of semantic technologies in industrial systems with the particular aim at the distributed agent-based solutions. Section 2 provides a general introduction into the world of semantics, ontologies and semantic web. Section 3 gives an overview of recently emerged applications of semantics and ontologies in the industrial domain and gives a summary of general aspects of such applications. Section 4 describes a new modular ontology developed for discrete manufacturing and provides guidelines how to utilize this ontology in multi-agent control systems. Section 5 presents a case study of deployment of the developed ontology for assembly line production control. Section 6 concludes the paper with the presentation of major advantages of using ontologies in multi-agent industrial systems.

2 Semantics and ontologies

The most common type of interaction in any distributed heterogeneous system is the exchange of messages. Obviously, there must be an agreement on both the syntax and the semantics of them. The syntax defines the structure of a language, i.e., a grammar typically in a form of rules that govern the structure of sentences. Semantics is dealing with the aspects of meaning as expressed in the language, i.e., the sense of language elements and their combination, including the relation of these elements to the real world.

The semantics is often captured by an ontology. The term ontology as clarified in [16] comes from philosophy, where it refers to the study of being or existence. It attempts to describe categories and relationships of all existing things. In engineering applications, this is reduced to a model of a part of a selected domain – a model that is processable by a machine and is appropriate for a specific application. The ontology in this context is often defined as a formal explicit specification of a shared conceptualization, where conceptualization is a shared view of how to represent the world. Ontology is then formal description of this view and generally consists of concepts representing classes of objects in the real world, their attributes, relations and constraints. In fact, the ontology provides a static vocabulary describing the general patterns occurring in given reality. Such a vocabulary is then used to describe the actual state of the observed real system and its dynamism – the model is composed of instances of classes defined in ontology and is usually called a knowledge base.

The utilization of ontologies in software engineering is mainly linked with the Semantic Web (SW) and Semantic Web Services architectures. SW aims to provide a common framework that would allow data to be effectively shared and reused. It is an extension of the World Wide Web that brings the semantic description of a content so that it can be found, processed and integrated by software agents more effectively [21] . The core semantic web technologies are Resource Description Framework (RDF) and Web Ontology Language (OWL). Although these technologies were primarily designed for the Web, they have been found suitable for other applications as well [38] .

The RDF is a standard for expressing structured data in a form of simple statements [30] . Each statement is expressed as a triple *subject-predicate-object* and each participant of the triple is a web resource identified by Uniform Resource Identifier (URI). In the place of object, a literal such as string or number can be used. The triples, which can be gathered from distributed sources on the web, are linked together to a searchable graph. For searching within an RDF graph there is the Simple Protocol and RDF Query Language (SPARQL) enabling specification of a required graph pattern for building a query [41] . The Web Ontology Language (OWL) is a widely recognized language for describing ontologies [12] . OWL is based on the description logic that allows sound and complete reasoning in a practically usable time. It defines a semantic description of concepts and roles in a particular domain in form of classes also called T-boxes (terminological boxes), their relations and constraints on the use. The ontology forms in fact a language that is applied for giving the meaning and semantic context to the information observed in a real world. The information is stored in a knowledge base in form of instances (called A-boxes or assertions) of the classes and relations defined in the ontology, which represent particular objects and structure of the real system.

The aim of the classical Web Services is to provide a coherent framework for publishing, discovery and remote execution of services over the Internet. To ensure syntactic interoperability of Web Services, there are core specifications defined by W3C, including message protocol, language for description of the service interface and protocol for registration and searching for services [45] . Semantic Web Services are semantic extension of the Web Services, like Semantic Web is a semantic enrichment of the Web. The main goal of the Semantic Web Services is to provide a semantic interoperability, which is needed for automatic, machine-orchestrated discovery, execution and composition of services. The typical scenario is a decomposition of a complex task into a serial/parallel call of number of various services, where resulting data from one service are used as an input to another service or where data from more service

have to be assembled to create a sophisticated reply to the original request.

Special ontologies were developed for these purposes, such as Web Service Modeling Ontology WSMO [25] or OWL-S [45] . The OWL-S ontology, which is built on top of the Web Ontology Language contains three main parts: the service profile used to describe what the service does, the service model describes how to interact with the service and the service grounding specifies the details of interaction with the service. Such a description is necessary for influencing the broader service-oriented architecture vision to allow truly open architectures that would enable integration of various heterogeneous services. In the internet environment such a vision is often referred to as Internet of Services [43] .

3 Semantics in distributed intelligent industrial systems

Holonic and multi-agent systems have been widely recognized as enabling technologies for designing and implementing next-generation of distributed and intelligent industrial automation systems [8] . These systems are characterized by high complexity and requirements for dynamic reconfiguration capabilities to fulfill demands for mass customization, yet low-volume orders with reduced time-to-market. Self-diagnostics and robustness that allow efficient continuing in operation even if the part of the system is down are other important properties.

The trend of applications of multi-agent systems is apparent at all levels of the manufacturing business. At the lowest, real-time control level, so called holons or holonic agents are usually tightly linked with the real time control programs (implemented in IEC 61131-3 or IEC 61499 standards) through which they can directly observe and actuate the physical manufacturing equipment [7] . Intelligent agents are also used for production planning and scheduling tasks both on the workshop and factory levels [40] . More generic visions of intensive cooperation among enterprises connected via communication networks have led to the ideas of virtual enterprises [9] .

The common principles in industrial deployment of the agent technology is the distribution of decision-making and control processes among a community of autonomously acting and mutually cooperating units – agents. At the shop floor level, for instance, an agent represents and independently controls particular physical equipment, like a CNC machine, conveyor belt or docking station. The substantial characteristic is the cooperation among the agents as they pursue either their individual goals or the common goals of the overall control system. The inter-agent interactions vary from simple information exchanges, for example about the state of processing as the product moves from one machine to another one, through requests to perform a particular operation, for example requesting an automated guided vehicle to transport a product to a particular work station, to complex negotiations based on contract-net protocol or different auction mechanisms.

As the information representation and exchange is the essence of such systems, the need of explicitly defined and shared ontologies becomes apparent. From our experience, the exploitation of semantics and ontologies in the area of agent-based industrial systems seems to be very intensive these days, which was not the case even a few years ago. The researches apparently realized that the syntactical interoperability, predominantly ensured by the adoption of FIPA standards and XML-based messaging, will not be sufficient to keep the pace with the trend towards semantically interoperable knowledge based systems. Thus, the use of semantic web technologies has accelerated significantly in the agent research community over the past few years.

Generally speaking, the formal ontology brings unambiguousness in the sense of explicitly defined vocabulary for manufacturing domain that facilitates communication and cooperation in a distributed industrial system. Also, the formal ontology allows reasoning over the shared knowledge. The intelligent autonomous controllers can process, exchange, search and reason about the knowledge related to the manufacturing plant much more efficiently if the information and data is given a clear semantic context.

3.1 Domain-specific ontologies for agent-based manufacturing control systems

The number of reports about the deployments of ontologies in agent-based manufacturing systems increases. Usually, a domain-specific ontology covering a subset of the manufacturing area, for instance assembly, is developed and utilized only for the purposes of the particular agent-based control application.

In [10] the ontology for shop floor assembly is described. Two basic categories of concepts are proposed: *modules* and *skills*. Modules represent physical processing units or their aggregation. One of the modules is for instance a *workcell*, which is defined as a composition of *workstations*, where a workstation is a composition of *units*. The examples of units are *transforming unit*, *flow unit* and a *verification unit* where transforming unit can be further specialized as pick&place unit or milling machine. The two typical ontological constructs, `composedOf` and `isA`, are used to describe the composition and specialization (inheritance) relations between concepts in ontology. Skills represent abilities to perform manufacturing actions, as for instance *MoveLinear*, where complex skills are represented as a composition of basic skills. The basic element in the multi-agent system which uses ontology as a data model for reasoning about objects and their relations is the Manufacturing Resource Agent (MRA). This agent, representing for instance a robot, searches the ontology after its instantiation for skills it supports using its serial number and type of equipment. Then it registers its capabilities in the yellow pages provided in the multi-agent system by a special Directory

Facilitator (DF) agent, which manages and provides information about services provided by the agents. The particular MRA agents can form coalitions in order to provide combined skills. In such a case there is a Coalition Leader Agent, which registers in the DF all complex skills provided by the coalition and subsequently coordinates the execution of elementary actions by particular coalition members. The proposed solution has been deployed in the NovaFlex laboratory environment installed at the Intelligent Robotic Center at UNINOVA in Portugal. A simple assembly line is composed of two robots, each with four different types of grippers and tools, and an automated transportation system that connects the robots and a storage unit. According to the authors, the proposed solution proved enhanced reconfiguration capabilities. The components can be added to the system at runtime, and thus the line can be easily adapted to new types of products.

An OWL-based ontology developed for agent-based reconfiguration purposes is reported in [1] . The application of ontology is illustrated on a small laboratory manufacturing environment consisting of two machines equipped with different mechatronic devices such as a rotating indexing table, plunger, drill, picker, etc. The basic ontology concepts are *material resource* and *operation*. In fact, this abstraction is identical with the previous example, only with the difference in the used terminology (module vs. material resource and skill vs. operation, respectively). The resources are *machine* and *tool* with corresponding subclasses like *handling machine* and *processing machine* as well as *rotary indexing table*, *drill*, *kicker*, etc. The operations are subdivided into *manufacturing operation* and *logistic operation* with further classification on *sorting*, *hole testing*, *drilling* and *picking*, *kicking* and *rotating*, respectively. References between machine and operation concepts express the facts that machine *enables realization of* an operation. These general concepts from the ontology are then instantiated to capture the real environment, such as the particular machines and their relations. Such a dynamic part of the ontology is also expressed in OWL, thus allowing the agents to reason about the available machines and operations in the semantic context.

Magenta Technology company provides another example of exploration of ontologies in agent-based applications. The details of an Ontology Management Toolset are given in [42] . This set of multi-agent tools enables developers to create and edit the static aspects of ontology as well as the dynamic aspects, here called *scenes*. The ontology developed by this toolset for supply chain and logistic planning is then presented in [2] . The examples of concepts are for instance *factory*, *cross-dock*, *truck*, etc., and relations like *is booked for a demand*. Although it is not explicitly mentioned in the last two cited papers, the Magenta's multi-agent engine provides a mechanism of updating the agent's behavior (i.e., the program code) dynamically as the ontology is being extended. The corresponding piece of code providing an agent with an algorithmical description of its behavior associated with a particular new ontology concept is sent to the agent so that it can subsequently execute the code to react appropriately.

Merdan et al. report on the application of ontologies in a transportation domain [33] . The OWL ontology has been developed for supporting the interactions of agents controlling the palette transfer system, which is a part of the Vienna University of Technology's testbed for Distributed Holonic Control [18] . The agents represent basic components of the transport system such as conveyors, diverters, junctions, index stations and palettes. The agents use the ontology for representation of the real state of the environment, like mutual connections of components, actual position of palettes, failure states, etc. Such data are stored in agent's knowledge base, which is continually updated as the agent perceives the dynamic changes in the environment. Such changes in the knowledge base trigger the rule-based behavior of the agent to properly react to the new facts. More details on the proposed ontology in the broader context of production scheduling in the assembly domain are given in [34] . The *product* is described as a composition of *subassemblies* that further contain *parts* representing raw materials. Each subassembly is produced as a result of step, in which a particular *operation* has to be performed by a manufacturing *resource*. The relations *needsPredecessor* and *isFollowedBy* between steps is used by the Supply Agent that schedules the production in cooperation with resource agents. A related case study presented in [23] examines the usability of the proposed ontology-based agent architecture in the resource allocation tasks.

Hellingrath et al. present the FRISCO ontology designed to support organization of knowledge in automotive supply chains [19] . Five different models have been designed: the Sourcing Model gives information about the products sold to customers and the parts procured from suppliers; the Resource Model contains all manufacturing resources that are relevant for planning like machines, workers, etc., including their capacities; the Adjustment Measure Model provides structures to represent network adaptivity; the Demand Constraint Model describes relations between demand and capacity to allow real-time capable-to-promise processes; and the Time Model provides structures for different calendars in order to create common understanding of dates between customers and companies. A proof-of-concept has been designed to verify the applicability of the proposed knowledge models. The scenario encompasses an OEM producing cars and two suppliers providing parts. The partners in supply chain are modeled as agents in a multi-agent system. The ontology is used for negotiations between car producer agent and part supplier agents about planning of delivery of required parts.

The proposal of an ontology for organizational model of general holonic systems deployment is presented in [11] . The context of an organization is described in terms of project *management*, *manager*, *employee* and the roles such as *supervise* and *assigns*. Other general concepts for the agency and holonic

domain are defined like *agent*, *agent role*, *holon*, *holon role*, etc.

3.2 General-purpose ontologies for manufacturing domain

The previous section documented that even though there are many efforts towards designing ontologies for manufacturing domain, different developers use slightly different vocabularies for describing similar concepts like for example *module* versus *resource* or *skill* versus *operation*. Moreover, developed ontologies cover usually very narrow areas. More complex and consistent general ontologies for manufacturing domain together with a series of complementary, coherent, domain specific ontologies would be helpful for achieving better interoperability and reusability. The existing norms and standards like ANSI/ISA-95 a ANSI/ISA-88 [3] could serve as a good basis in this effort. The ISA-95 „Enterprise-Control System Integration“ standard describes hierarchical model of production organization and the event flow and provides basic concepts for the integration of control system with business systems of the enterprise. The ISA-88 “Batch Control” describes in more detail the batch process production environment. It defines hierarchical model of production system from the enterprise, through areas and units down to control modules. It also defines process model for description of sequenced production phases and actions.

Very promising standardization effort seems to be concentrated around the O^{3}NEIDA (Open Object-Oriented kNowledge Economy for Intelligent inDustrial Automation) consortium that aims at creating the open technological infrastructure for automation components [50] . The goal is to create the architecture for hardware and software compatibility at all levels of automation components market, from device and machine vendors, through system integrators up to industrial enterprises. The basic element in this architecture is the a*utomation object* that is an abstraction of mechanical device with encapsulated intelligence, i.e., software components providing different functionality like control, visualization, simulation, diagnostics, etc. with well defined interfaces. Simple automation object such as sensors, drives and microprocessors can be then used as reusable modules for creating more complex objects (such as machines) that can be further used in the same modular way for building the whole industrial enterprises. The use of ontologies, mainly OWL, is promoted for the description of automation objects. This would allow automatic machine processing and reasoning as well as simplifies search of automation objects in repositories.

A complementary work to OOONEIDA initiative presented in [28] aims at semantic extension of automation objects by applying the semantic web technologies. Two separate ontologies for mechatronic devices reference model (covering both the hardware and the software features) and the IEC 61499 reference model respectively are proposed and merged into an ontology for Automation Objects reference model (proposed by IEC-TC65 group). The basic concepts designed for the lowest level include function blocks, events, I/Os, etc. The device/machine level part of ontology provides concepts like function block application, resource, etc. Two examples of semantic description of automation objects – Conveyor and Lifter – are sketched.

National Institute of Standards and Technology (NIST) devotes considerable standardization effort to manufacturing domain. For instance, shop data model is described using UML diagrams and XML serialization examples [31] . The model includes description of organization, bill of materials, process plans, resources, schedules, etc. Although it is not a formal ontology in the sense described earlier in this chapter, such standards are important as a base for ontologies that would be widely accepted. Another example of NIST activities is the Process Specification Language PSL [15] that is a logical theory that covers generic process representation, which is common to all manufacturing applications. The PSL ontology contains axioms grouped to theories describing aspects such as complex activities and can serve as a solid base or upper ontology for representing processes. The PSL ontology contains primitive concepts like *activity*, *object* or *timepoint*, functions such as *beginningOf* and *endOf* and relations like *between*, *is_occuring_at*, etc.

MASON (MAnufacturing’s Semantic ONtology) presented by Lemaignan et al. [27] represents another contribution in this area. The goal is to develop an upper ontology that would allow seamless integration of more specific ontologies using the common cognitive architecture. The ontology is based on OWL and describes the taxonomy of concepts such as entities, operations and resources and their relations like associating a tool with an operation (property *requiresTool* with the domain *ManufacturingOperation* and a range *Tool*). It is reported that currently the ontology, which is available on-line at http://sourceforge.net/projects/mason-onto, constitutes of more than 220 base concepts and 40 properties. Moreover, a mapper has been developed between OWL ontologies and the internal ontology model used by the popular Java Agent DEvelopment Framework (JADE) agent platform (http://jade.tilab.com/). Although some of the constructs in the ontology seem to be application specific, for example, restricting previous operation in the definition of operation concepts, this work can be seen as an important step towards formalizing the vocabularies used to describe manufacturing domain.

When building the general-purpose manufacturing ontologies, it is obviously necessary to have solid basis in form of well developed foundational (upper) ontologies incorporating for example spatial or time theories. Unfortunately, the direct utilization for manufacturing purposes is limited because these foundational ontologies are often created in very expressive languages without taking care of computability. The formalization of ADACOR ontology (ADAptive holonic COntrol aRchitecture for distributed manufacturing systems) using the DOLCE methodology

(Descriptive Ontology for Linguistic and Cognitive Engineering) is outlined by Borgo and Leitão [6] . ADACOR is originally described using Unified Modeling Language (UML) diagrams and natural language descriptions, while DOLCE uses first order modal logic and aims at capturing the ontological categories underlying natural language and human commonsense, such as physical or abstract objects, events and qualities. The alignment of ADACOR to DOLCE yields well formalized and well founded ontology. The ontology described in ISO 15926 "Industrial automation systems and integration" also uses well founded principles of temporal and spatial representation of objects in a form of four dimensional approach to simplify reasoning in the process engineering domain [4] .

3.3 Common properties of ontologies deployment in agent-based manufacturing systems

Within the frame of semantic extension of multi-agent industrial control applications certain characteristics could be identified that differ from the common usage of ontologies in pure software systems like web applications.

Static and dynamic aspects of ontology – the OWL language allows one to express two kinds of terms. First are static, unchanging concepts, so called T-boxes (terminological boxes) that represent vocabulary for description of selected domain in form of classes and their relations. The latter group are so called A-boxes (assertion boxes) that have the meanings of particular assertions about the described part of the real world and that are formulated using the vocabulary of T-boxes. Each A-box is an instance of corresponding T-box. A set of A-boxes form together a *knowledge base*. The first part that could be called an *ontology* is invariant or is changed rarely. In case of a knowledge base, which is sometimes called also a scene [2] , often changes are supposed as there are dynamic changes in the observed part of real world. An example from the automation area is the set of classes (T-boxes) *machine* and *operation* with relation *providesOperation*. A knowledge base that describes current state of the factory shop floor contains for example instances *machine_M25* and *drilling_D14* as instances of these classes connected together with the mentioned *providesOperation* relation. In case of deployment of agents the ontology or its part is shared by all agents while the knowledge base is created and maintained by each agent individually as the agent perceives the changes in its environment by means of sensors, communication with other agents (the agents share their knowledge bases) or by communication with a human.

Interaction with a real world – an important part of agent's knowledge base, which represents and controls physical manufacturing components, is the information related to actual state of the controlled equipment (readings from sensors) and the status of controlled production process (e.g., actual location of the product). Another important factor to be considered is also a possible physical interaction or collision avoidance with other equipment. The agent has to be aware of the physical effects of its decision making. Its actuation could in negative case lead to a failure or damage of the equipment or to increased number of defected products. The link between an agent and the real world in case of manufacturing control deployment is usually ensured through the interaction with the low-level control (LLC) layer, which might be implemented according to IEC 61131-3 or IEC 61499 standards. The ontology could be advantageously used for designing the semantic model of the low-level layer and corresponding interface. This ensures keeping the agent away from the details of software and hardware implementation of the LLC and thus makes the integration of multi-agent control system with current PLC-based architectures much easier. The first attempts to design the ontology-based interface between agents and LLC is presented in [18] . It is argued that the use of semantic model for these kind of interactions keeps the agent and LLC layers more loosely coupled, rather than tightly coupled as seen for instance in real-time interface described in [29] . Loose coupling is a desired property, because it enables that the LLC layer can be still operational even if the agent layer becomes unavailable or faulty. Also the technology of radio frequency identification (RFID) becomes to play increasingly important role in tracking and localization of parts, semi-products and products in manufacturing environment. The architecture integrating RFID with agents is described in [47] . The use of ontologies in this field is also expected to provide semantic interoperability between applications processing RFID data and events and the RFID infrastructure involving tags, readers, middleware, etc. [22] .

Reactivity – imagine a situation when an agent notes a particular event in the real world, for instance, detects a failure of the controlled machine. It creates a corresponding fact consistent with the ontology (describing relation between a *machine* and *failure* concepts) and stores this information into its knowledge base. The agent's inference engine can then possibly deduce other new facts, but this still does not directly lead to reaction. But often, particularly in agents acting in real world, some action or reaction needs to be taken by the agent – for instance, actuating (stop the drive) or informing other related agents. So the meaning of the particular concept from the ontology is sometimes not only knowledge-based but also "algorithm-based". The ontology should provide the agent also with the explicitly defined rules in a form of algorithms (or directly a program code) to be executed by the agent to react appropriately. This issue is not sufficiently discussed in literature. As mentioned earlier, Magenta agent runtime environment [42] provides such features – program code is sent to the agent at runtime to modify or extend its behavior.

Ontology-based service matchmaking – one of the basic concepts of multi-agent systems is the advertisement of agents' skills and services in the Directory Facilitator (DF), known also as yellow pages

services. Other agent can then search the DF to find out particular service providers. However, the information held in the DF in majority of agent platforms available today, such as JADE [5] , Cougaar [20] or A-globe [44] , can be registered only in a very simple form. It usually contains just type of the service (for instance Drilling) but it cannot be further parameterized (diameter: 10-100 mm, hole depth: 5-20 mm). Obviously, to fully explore the potential of semantics in agent-based systems, ontologies must be deployed for service registration and lookup through DF as well. Within the registration the agent sends the corresponding part of its ontology (services it offers) to the DF. DF can be then queried for finding particular service providers using more complex queries, like "find all machines that can drill a hole of 50 mm diameter and 15 mm depth". The result sent back by DF to the requester (also in form of ontology) might be with convenience supplemented by the message template and protocol to be used in the corresponding inter-agent negotiations, as discussed in [10] . Services provided by agents can be described using OWL-S in a similar way as semantic web services. Matchmaking of services can be then made using OWL reasoning.

Orchestration of manufacturing processes – service integration and composition becomes very attractive topic in the Service-Oriented Architecture (SOA) domain. Authors in [17] describe a solution based on multi-agent and holonic techniques. Community of interacting holons, representing service providers and requestors, can be nested so that requested complex service is automatically orchestrated as a composition of basic services. An important function of a reconfigurable distributed manufacturing system is the distribution of tasks over multiple agents or holons. This goes beyond the simple service matchmaking – the whole process must be decomposed, executed and problems occurring during the runtime must be resolved. For that, manufacturing processes should be also specified in ontologies [24] . We envision the ontology-based recipes compiled as a sequence of elementary operations described in a suitable ontology to allow automatic discovery of equipment that can perform requested operations (see next Section for more details).

Interoperability – represents property required within a manufacturing system as well as between other systems on MES (Manufacturing Execution System) or ERP (Enterprise Resource Planning) levels. Translation between ontologies is a way of integrating systems that use different ontologies [36] . One agent prepares a message in its ontology, and the message is then translated into the ontology used by the receiving agent while preserving the meaning of the message. The details on the translation using semantic web technologies and OWL reasoning are described using transportation domain examples in [38] . The architecture of integrating systems has to be considered as well – the low level control devices would be hardly able to do such translation themselves, and so they need to ask a special service to provide translation for them or the translation has to be made automatically in the message transportation layer [37] .

3.4 Semantic search

One of the core applications of the semantic web is a semantic search, i.e., search within semantically enriched data. The design, operation and maintenance of a manufacturing system is very knowledge intensive task and involves handling of information stored in different forms – for example, function blocks or ladder diagrams describing the real-time control system, SCADA/HMI (Supervisory Control And Data Acquisition/ Human Machine Interface) views, collected historical data, etc.

It is often not easy to search within such information space even using plain text search. However, when the information is accessible in the semantic web form, it is possible to make queries beyond the classical keyword search. We have investigated the use of semantic web technologies for the semantic search within various information sources of an assembly line. Our conclusion is that the RDF/OWL form of data is appropriate for storing the information and for querying [38] . The SPARQL language is capable to express structural queries that are of practical interest for the control system designers as well as maintenance personnel. Example of such a query is to find all projects where specific ladder code instructions (e.g., XIO – eXamine If Open) with specific variable (e.g., Valve13) occur in a rung. In the prototype prepared by Rockwell Automation the data extracted from the control system, such as ladder logic programs and HMI views, are annotated automatically depending on their context, so the process of indexing, which is necessary for structural queries, is fully automatic.

In addition, the implicit information (such as the `part-of` relation) can be made explicit in the ontology describing the manufacturing system. The query engine can employ an OWL reasoner to include this information into query results, so that for example query results containing the part-of relation contain transitive closure of this relation. An important advantage of using RDF is that all the distributed information can be merged into a single RDF graph. This allows asking for connections of information from different sources, such as in the query to find all HMI views that have push buttons connected to a ladder code project that is used in a specified area.

4 Generic manufacturing ontology and related multi-agent architecture proposal

In this section we describe the utilization of ontologies for a multi-agent industrial production system. The system consists of agents that handle various aspects of a typical flexible discrete production system. The goal is to isolate knowledge and semantics so that typical system deployment would mean only extending ontologies without having to reprogram the multi-agent system.

4.1 Usage of ontologies in multi-agent system

The reason for integrating ontologies into multi-agent system is that we would like to express semantics explicitly for agents so that they are able to operate differently when ontology or knowledge base is changed and also so that integration of new agents can really proceed without reprogramming existing agents. Our general testing scenario is flexible and reconfigurable distributed production system that consists of workstations able to provide different manufacturing operations and of transportation between these workstations. The system accepts different highly customized orders, is able to create customized production plans and is able to execute these plans. The plan is executed by selecting workstations that provide required manufacturing operations needed in steps of the plan. The manufactured product is transported between workstation together with material until the plan is finished, i.e., until the final product is produced.

The orders, production plans, capabilities of workstations, transportation system and other components are described in ontologies and knowledge bases and not in agents' code. This is very important factor that makes future extension of the system very easy. It is possible to introduce new product type by adding its production plan or it is possible to introduce a new operation by extending the ontology. Only the update of the ontologies and/or knowledge bases of appropriate agents is required for introducing completely new functionality. For example, when a new product type is added, it is enough to extend the order ontology to cover new product type and its parameters, to add general production plan for that product type, and to add rules for conversion from product order to specific production plan. When a machine with new kind of operation is added to multi-agent system, it is again enough to extend ontologies to cover this new operation.

4.2 Description of ontologies

From the state of the art description we could see that there are some interesting features in existing ontologies. We were inspired by ontologies described earlier and decided to design a new ontology based on existing standards and ontologies such as ANSI/ISA 88, OOONEIDA or MASON. It has been found that none of these ontologies provide a suitable semantic model that would fulfill our requirements for generality, extensibility and deploy-ability in distributed control applications devoted for discrete manufacturing – i.e., handling customized orders, handling customized production plans, handling material and semi-product transportation between production machines and describing these machines. ANSI/ISA 88 is exclusively designed for batch processing industry, OOONEIDA does not provide what we needed for this purpose and MASON seemed to be too limiting for us.

When designing a new ontology, the attention has been paid to two complementary aspects that influence feasibility of a real deployment at a wide spectrum of tasks. First, the ontology has to be as much general as possible in order to provide a common, consistent model of base concepts, on which application specific extension ontologies could be designed. Second, the ontology has to be relatively simple in order to be applicable in real control systems. The common issue of many foundational ontologies that are defined in very expressive languages is that the processing of these ontologies is very computationally demanding. It does not meet constraints imposed by PLC-based architectures and real-time or near-to-real-time fashion of control programs that should work with the ontology.

Our new manufacturing ontology includes three different aspects of automation systems, as illustrated in the upper part of the Fig. 1: (i) specification of customer order, (ii) definition of production plan and (iii) transportation and material handling. The ontology is implemented in OWL as three separate ontology modules that describe these aspects in general. All of them reuse classes and properties from the Common Ontology that for example separates physical and information resources. There are also other ontologies, such as ontology for the configuration of the system. Reusing particular classes or properties from one module in another one is implemented using OWL imports.

These ontologies are intended as a base for the multi-agent system operations. Agents should understand these ontologies and should be able to handle their extensions. As we can see in the Fig. 1, it is possible to extend the ontologies with application specific description of product order and product plan. The advantage of this approach is that for particular product and product plan we only need to extend existing ontologies (i.e., subclass existing classes) and the system is able to handle new product without any other changes.

The lower part of the Fig. 1 shows operation of the multi-agent system together with illustrating which parts of the whole ontology are used. The workflow of the system is as follows. First, the Order Agent receives a customer order. Based on that order, corresponding Product Agent is created that receives production plan created by Production Plan Agent individually for the customer order. The Product Agent then executes the production plan by contacting Workstation and Transportation Agents. Note that multiple agents can process multiple customer orders at the same time.

As we can see, different agents use different parts (modules) of the whole ontology. We present more detailed overview of the ontology modules together with their intended usage in the next sections.

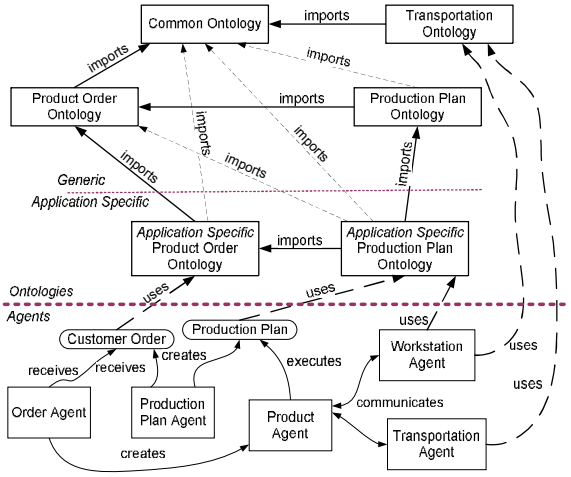


Figure 1: The upper part shows modularized manufacturing ontology with generic and application specific parts including *import* relation between individual ontology modules; the dashed *import* lines show transitive closure of the *import* relation. The lower part shows agents that use different modules of the ontology.

4.3 The order ontology

The ontology module for customer order specification is schematically shown in Figure 2. The dashed line separates the general order ontology (on the left), domain specific extension of the ontology for hypothetical production of a filled bottle (on the bottom) and a knowledge base corresponding to a sample order of three filled bottles (on the right). The ontology as well as knowledge base is expressed in OWL. The ontology is shown schematically without all OWL relations, but in general blue rectangles in ontology correspond to classes and green rounded rectangles correspond to properties. Object properties use normal font, data type properties use italics. In the instance part, black rectangles correspond to instances and black ellipses to RDF literals.

The general concepts of the order ontology are as follows. Order is the top most representation of the customer order; it contains one or more `ProductOrders` to enable ordering of more products of different types in a single order (for example three bottles with water and five cans of coca-cola). `ProductOrder` represents part of the order corresponding to products of the same type (e.g., three bottles of water). The `hasProductOrder` property is used to express the fact that an `Order` is composed of `ProductOrders`. The quantity property represents number of ordered products of the same type; the domain of this property is the `ProductOrder` class and its range is an integer literal. `ProductSpecification` specifies for a particular `ProductOrder` the type of the product (e.g., filled bottle) and parameters or attributes that all the products of the same type should have (e.g., water as a liquid in the bottle, 0.5 volume in liters, etc.). The `hasProductSpecification` relation represents that a `ProductOrder` has assigned `ProductSpecification`. The Product class represents the type of the product being ordered in particular `ProductOrder` (e.g., filled bottle). The control system is supposed to find appropriate production plan (a recipe how to make a product) based on this parameter. Thus this class provides a link between this ontology module and production plan ontology.

To represent different parameters, a class Parameter was introduced to represent a general parameter that can be further specialized. It is supposed that subclasses of this class are defined to represent different, domain specific parameters. The parameters are connected using the `hasParameter` property. An example of parameter is `ParameterBoolean`, which represents a general boolean parameter, and uses booleanValue relation to hold a boolean literal (true or false).

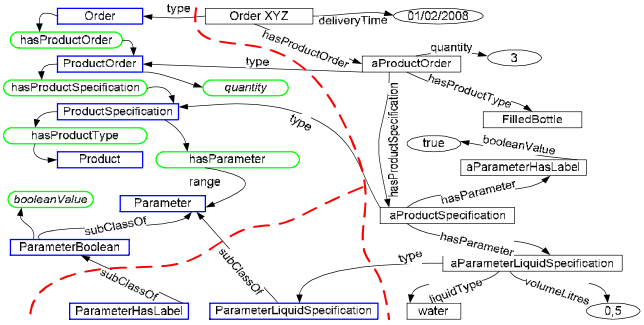


Figure 2: Schema of the ontology for customer order specification (on the left); domain specific extension of the ontology for hypothetical production of a filled bottle (on the bottom); knowledge base corresponding to sample order of three filled bottles (on the right).

The domain specific extension of the general order ontology regarding a hypothetical production of filled bottles includes the following subclasses of the `Parameter` class. `ParameterHasLabel` is a subclass of a `BooleanParameter` for specifying if a label is required on the bottle. `ParameterLiquidSpecification` is a parameter specifying type and amount of a liquid to be filled in the bottle (see the instance part for the intended usage).

A sample instance of order (knowledge base) depicted on right part of Fig. 2 represents a customer order for three bottles filled with 0.5 liters of water and attached label. All the rectangles are instances of classes from the ontology. In this case the order instance (Order XYZ) contains only one `ProductOrder` class instance (`aProductOrder`). It says that the type of the ordered product is filled bottle (`FilledBottle` as instance of Bottle class that is subclass of `Product` class). The specification of the product (`aProductSpecification`) includes a parameter specifying type and amount of liquid (`aParameterLiquidSpecification` instance with relations to water and 0.5 literals). The other parameter is

aParameterHasLabel with relation to true literal specifying that a label should be attached on the bottle.

To summarize, the Product Order ontology serves primarily for description of product parameters in customer orders. These parameters are then used to create production plan, as described in the next section.

4.4 The production plan ontology

The second ontology module shown in Figure 3 provides concepts for description of a discrete production process. It is supposed that a production process involves consecutive execution of steps with defined order, where parallel branches are also supported. In each step there is a particular manufacturing operation associated (like filling) that is performed upon the semi-product, which passes through the process. Performing the operation could require an additional material, like for instance a liquid to be filled to a bottle within the filling process. The ontology includes following concepts.

Product is a class representing a general product of particular type which the production system is able to make. Subclasses of this class are supposed to be introduced (like Bottle) in domain specific extensions of the ontology. The same Product class appears in the previously described order ontology as a type of product that can be ordered. ProductionPlan represents general production plan that describes a consecutive execution of steps that transform raw materials and semi-products into the final product. ProductionStep represents a single production step in a discrete manufacturing process; basically, in each step a particular operation upon the semi-product is executed; a raw material(s) could be required by the operation. The relation hasProductionStep represents that a production plan is composed of several production steps. The precedes relation between two steps A and B is intended for representing the fact that the step A has to be executed prior to execution of the step B (but not necessarily immediately before). It is a transitive relation what means that if step A precedes a step B and step B precedes step C than also step A precedes step C. The inverse relation to precedes is follows. For expressing the fact that no other step can be executed between steps B and A there is relation mustImmediatelyFollow. An example is shown on right part between CapStep and FillStep. The requires relation between two steps A and B expresses the fact that that step A requires a presence of step B in the plan. It is expected that there is a special component in the control system that modifies templates of plans according to user orders (see further). Usually if some feature of product is not required, some steps are deleted from the plan. The requires relation means that if step A is to be preserved in the plan then also step B has to be preserved and cannot be deleted (it does not say anything about order of executing the steps).

Operation represents a manufacturing operation to be executed within a step. It is supposed that an operation is provided in a manufacturing system by a workstation. The operation is usually a complex task that involves execution of a series of sub-operations executed by different machines and tools that are part of the workstation. The relation hasOperation serves to represent that a production step has a single operation associated with it. The hasParameter relation is again intended for expressing various parameters of the operation. The parameters are usually copied or transformed from the order at the time of creating the plan for a particular order from the general product plan.

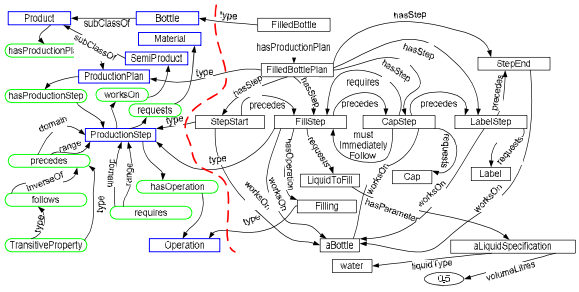


Figure 3: Ontology for description of a discrete production process as a series of consecutive/parallel production steps in which particular manufacturing operations are executed upon the semi-product; right hand side shows sample plan describing production of a filled bottle.

SemiProduct is a class representing a semi-product – an object that is being transformed by operations associated with production steps into a final product. A particular piece of raw material(s) as an input to production step is transformed by associated operation into semi-product(s). For example, the material input into assembly step can be pieces X and Y and the output is a new semi-product XY. Semi-product can then enter other steps as an input, and other steps can further transform it using other operations and additional material. The worksOn relation represents those semi-products that are both input and output ones for a particular production step. Material class represents of raw material that is required as an input for a particular operation in production step or is generated as an output of a production step (for example, in a disassembly process).

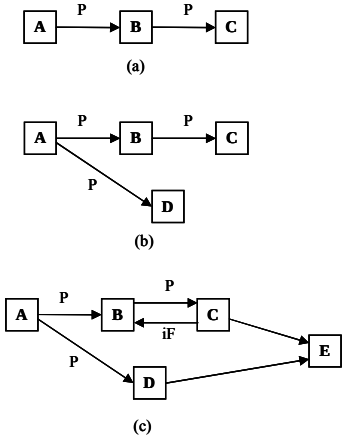


Figure 4: Illustration of various alternatives of precedes (P) and immediatelyFollows (iF) relations between production steps.

The relations between production steps (`precedes`, `follows` and `immediatelyFollows`) allow for expressing either serial or parallel branches or various combinations of those in a production plan. Some of the alternatives are illustrated in the Figure 4. In the case (a) there is a serial plan where the order of executing the steps is A-B-C (step B requires step A to be done before as well as step C requires step B to be done before). In the case (b), A is the first executed, and then B and D can be executed in any order or in parallel. Step C can be executed after step B is finished regardless of the status of D. In other words, the order of execution without considering the possibility of parallel execution of steps B and D can be: A-B-D-C, A-D-B-C or A-B-C-D. In the case (c) there is the `immediatelyFollows` relation between steps C and B, so the step C has to be executed immediately after B. Step D can be executed in parallel with steps B or C or before them or after them. In other words, the order of execution can be thus following: A-B-C-D-E or A-D-B-C-E.

Back to Fig. 3, on its right part there is a sample instance showing a production plan for making a filled bottle. It can be seen that such a plan is developed as a knowledge base composed of instances of classes from the ontology. The plan starts by a step `StepStart` in which a suitable empty bottle is retrieved – it is represented by an instance `aBottle` linked by `worksOn` relation. This bottle instance then enters all other steps. The `FillStep` represents filling of the bottle. It requires a liquid as an input material represented by `LiquidToFill` instance. Specification of the required material is attached as parameter `aLiquidSpecification` instance with subsequent relations `liquidType` to value water and `volumeLiters` to value 0.5. These values are copied from the customer order (see Fig. 2 where the same `aLiquidSpecification` instance is attached). Immediately after filling operation the capping operation (`CapStep`) is required to be executed. The last step is labeling (`LabelStep`) in which a label is attached in the bottle. A `true` value of the `aParameterHasLabel` parameter is again copied from the order. In the case when a label is not required, the label step is removed from the plan.

The plan instance is created by a component of the control system (i.e., agent) by modifying general production plan for a given product type. This agent has access to knowledge base containing general production plans for various types of products. On the basis of an order for particular product piece this component is able to modify the general production plan by rules or other kind of procedural knowledge. The result of the modification is a specific production plan instance tailored to the particular ordered product. This modification is done by copying the parameters from order to corresponding parameters of operations and material in the production plan. Possibly, some steps that are not required because of the specific product feature is not listed in the order are removed from the plan.

4.5 Workstation concept and material handling ontology

Material and operations associated with production steps are provided in the distributed control system by workstations. In other ontologies this concept is also known as work cells, work places, manufacturing cell or just cells. Each workstation is a logical composition of physical manufacturing equipment or devices. It represents an individual, stand-alone manufacturing entity providing different processing capabilities and/or material resources. The workstation is represented by an autonomous control component (workstation agent) that can negotiate about the allocation of its advertised resources with the agents that control the execution of production steps. The single operation provided by a workstation is usually internally decomposed into the execution of several sub-operations carried out by the particular equipment. Such an execution is supervised and controlled by the workstation agent by negotiations with the subordinate equipment agents.

Figure 5 shows a part of the Cambridge's DIAL manufacturing testbed [47] with two workstations – VS1 and VS2 (see also Sect. 5). Each of it contains the following equipment: raw material storage, manipulation robot and docking station(s) connected to a conveyor-based transportation system. One of the operations provided by this workstation is box packing – the robot picks the raw item from the storage and inserts it into a slot in the box. Boxes are carried out on top of palettes moving in the transportation system.

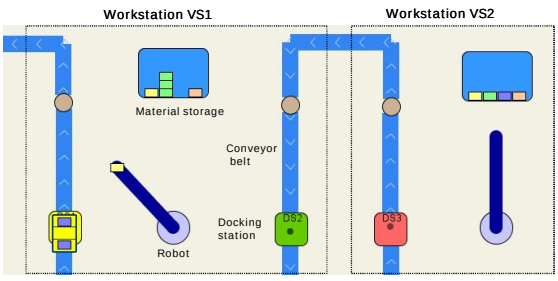


Figure 5: Part of the manufacturing system (Cambridge's DIAL testbed) with two workstations, each composed of material storage, robot and docking station(s).

As can be seen in Fig. 6 showing the third ontology module, the `contains` relation is used to express the fact that an equipment (`Equipment` class) is part of the workstation (`Workstation` class). The `providesOperation` relation is then used to associate all operations (`Operation` class) provided by the workstation (not shown in Fig. 6).

The internal functionality of a workstation can be advantageously described using the same concepts of this third ontology module. In the knowledge base of the workstation there is a production plan (`ProductionPlan`) associated with each operation (`Operation`) externally advertised by the workstation. This plan is again composed of a sequence of steps (`ProductionStep`), while the operations associated

with those steps are performed by the equipment in the workstation. Similar concept of hierarchical classification can be applied also to the production steps themselves. The proposed ontology provides an effective tool that allows one to replace a single operation associated with production step (`ProductionStep`) with the whole production plan (`ProductionPlan`). This plan is again described with the same ontology concepts as a sequence of steps, where each step can again have another production plan associated (see Fig. 7). Using such a simple mechanism enables to handle production plans with different level of granularity at different levels of enterprise. At the roughest resolution it is for example possible to decompose the car production into basic steps like production of engine, gear box, chassis, body, and so on. For each of these steps there is a more detailed plan; engine production is composed for instance from steps for making cylinder head, cylinder body, valves, camshaft, etc. These steps can be further refined using the same ontology concepts down to most fine details at the level of basic operations performed by equipment

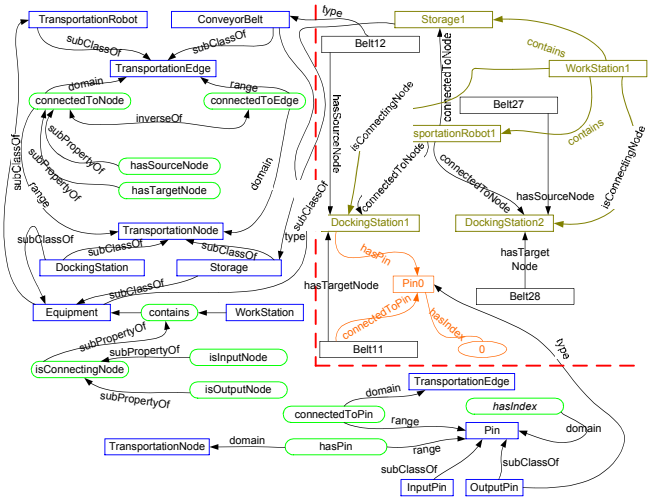


Figure 6: Ontology for description of material handling and transportation aspects including definition of WorkStation concept is shown in the left part of the figure. Sample knowledge-base related to the workstation VS1 of Fig. 5 is shown in the right part of the figure.

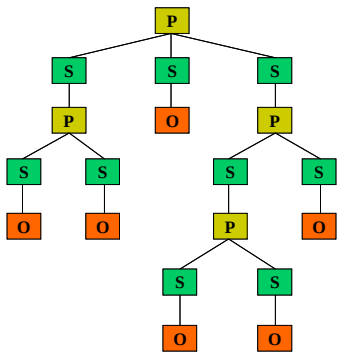


Figure 7: Example of a production plan (P) defined by ontology, where some production steps (S) have an operation (O) associated and some have another production plan associated.

inside workstations.

The third ontology module (Fig. 6) also includes aspects of material handling and transportation. Using the graph theory, the ontology defines a transportation node (`TransportationNode` class), a transportation edge (`TransportationEdge`) and their connection using two different relations, `connectedToNode` and `connectedToEdge`. From the viewpoint of product transportation between workstations, the ontology is used to define which equipment that is part of the workstation is also a transportation node, i.e. an input (`isInputNode`) or output (`isOutputNode`) point of the workstation through which a product or material can be delivered to or out of the workstation respectively. In Fig. 5, docking stations DS1 and DS2 are connecting nodes of the workstation VS1 (connecting node is such a transportation node that is both an input and output node), while docking station DS3 is a connecting node of the workstation VS3. Such an information is used by the components that ensure transportation of material (e.g., transportation agents controlling Automated Guided Vehicles) for planning of optimal transportation paths between the workstations.

5 Case study: MAST system and DIAL scenario

This section demonstrates the integration of developed ontologies in the agent-based manufacturing control system MAST. The functionality is illustrated on a real-world manufacturing scenario – the packing and assembly environment of the DIAL laboratory.

5.1 Ontological extension of MAST system

The Manufacturing Agent Simulation Tool (MAST), developed by Rockwell Automation, was designed with the intention to provide a simulation tool transparently demonstrating the advantages of application of multi-agent system technologies in the industrial control domain [46] . The MAST system consists of agent classes that represent various manufacturing components such as a conveyor belt, diverter, storage, docking station, sensor, etc. A typical task for agents is the transportation of products between work cells through a complex and redundant network of conveyor lanes. The agents use messages to exchange information about optimal routes through the system, about failures or transportation jams and about currently transported products. From the first simulation prototype MAST system matured into a comprehensive simulation and control tool which can interact with the real Programmable Logic Controllers to actually control the real physical equipment [48] .

From the viewpoint of knowledge handling and exchange, the original agent architecture could be characterized as implicit and rigid without the notion of semantics being applied. The agent's representation of the surrounding world was held in local variables, and the content of exchanged messages was encoded in XML. The major deficiency was that the interpretation of

meaning of received messages is not described explicitly but is tightly embedded in the agent program code. This complicates further extension of the system as well as integration of new agents, because it requires reprogramming the existing agents to embrace the new conditions.

To overcome these problems and to allow automated processing of knowledge including reasoning, we have started to extend the MAST system to use explicit semantics. The agents use RDF and OWL ontologies described in Sect. 4 to handle, store and exchange the semantic information about customer orders, production plans and actual schedules [49] .

5.2 Extending ontologies for the DIAL scenario

The holonic packing cell installed in DIAL (Distributed Information and Automation Laboratory), formerly known as CDAC, at Cambridge University's Institute for Manufacturing [14] was used in past as a real-world platform for verification of the MAST agent-based control principles. The system receives customer orders in a form of selection from two box types and from selection of different cosmetic items (gel, razor, deodorant and foam) that should be placed to specific slots in the selected box. The system layout is illustrated in the Fig. 8. There are two independent production stations (marked as Workstation1 and Workstation2) containing a vertical, four-slot storage unit for holding the raw items and a Fanuc M6i robot that picks the items and places them into box. Empty as well as finished boxes are stored in an Automated Storage and Retrieval System (Workstation 3).

To be able to execute this scenario in MAST system, the DIAL-specific extensions has been created on top of the general ontology concepts described earlier. The order ontology has been extended with the description of types of boxes, with the description of different cosmetic items. The production plan ontology has extended by the pack operation that simply puts cosmetic items to a specified slot in a box. The orders are placed through the OrderManager component in the graphical user interface. Its task is to compose a semantic description of the order based on parameters specified by the user in the form of an RDF description compatible with the order ontology. The order agent created for the order then instantiates a new product agent giving it the specification of the order. The product agent then asks a special Production Plan Agent for an instance of production plan for box packing. This agent modifies the general production plan according to the user order. This modification includes association of parameters from the order to particular production steps (type of box, types of inserted items) and optionally deletion of some of the three steps for item inserting in case particular slot is requested to be empty. According to this production description, the product agent plans execution of these steps, i.e. searches the Directory Facilitator for agents providing operations associated with step. The cost-based model, extended contract-net protocol and RDF content language are used for negotiation about resource allocation between the product agent and the workstation agents. In bidding on cost of providing operation the workstation agent considers the availability of raw material (boxes and cosmetic items), capacity and load of workstation and product priority. The product agent selects the lowest cost and subsequently negotiates with the transporter agents representing shuttles about the delivery of product to the selected destination. Once the operation is finished, the product agent starts the planning of the next step defined in the plan ontology.

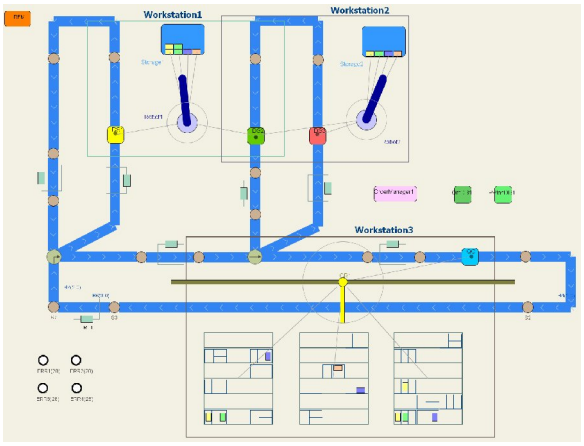


Figure 8: MAST System – DIAL scenario layout.

6 Conclusion

The main added features of the multi-agent system that uses ontologies as a base for information processing are extensibility, modularity and flexibility.

We have shown how the multi-agent production system can be modified for new types of products only by extending ontologies, without any need to reprogram or even shutdown and restart agents. The *extensibility* means that new kinds of physical machines with previously unknown manufacturing operations or new kinds of transportation means can be added to the system, and the system is able to cope with these changes without having to interrupt the system run. Moreover, new products with specific production plans, which are unknown in the design phase of the control system, can be added online without any further modifications needed. Thus it can be stated that *extending the system is only a matter of extending the ontology*.

The *modularity* of ontologies is inspired by software modularity. There are modules that serve to describe classes and properties common to the whole base system; there are modules specific for operation of part of the system; and there are modules that are application specific. Each of the modules can be extended separately – for example, we can extend parameters for customer orders without affecting the rest of the ontology modules. The customer order description extensions can be directly reused for production plan extensions thanks to the modularity and relations between ontology modules. In other words, the extensions are made only where necessary and do not affect the rest of the whole ontology. This is also possible, because an agent that is

designed for a particular task does not need to be equipped with all the ontology modules. The agent needs only selected module or modules.

Even when we do not consider updating agents during runtime, the use of ontologies increases the *flexibility* of the agent-based control system. In the DIAL scenario described in the previous section, the original version of the product agent was hard coded for a particular product type. Although the agent was able to discover alternative routing in the case of conveyor failures, the agent could not be used to control production of other product type without reprogramming. In the described semantically enriched solution, the production process is not described in the program code of the agent, but it is specified explicitly based on the shared ontology. The agent is then able to process such a recipe automatically, in such a way it schedules the execution of steps by negotiating with workstation agents about providing the operations associated with the steps. This means that the agent is much more flexible. This applies also to other agents when comparing the original implementation of MAST system with the semantically enriched version of the system.

The paper shows that the industrial automation domain, and especially intelligent control system, are being more and more influenced by semantic web technologies. The survey of existing ontologies applied in various applications like assembly or supply chain declares a need of standardized upper ontologies that would provide seamless information integration throughout the different levels of manufacturing enterprises. We have presented a compact, yet general ontology for description of discrete manufacturing processes and outlined guidelines for integration of this ontology in a multi-agent control system.

We have described a real-world scenario application of semantic technologies in the MAST system and have shown how semantic technologies can be employed to describe the operation and capabilities of the system not in the code of agents but rather in ontologies and knowledge bases. The advantage is that introducing new product type, introducing new or updated product plan, introducing new kind of machine or operation means only extension of one of the ontologies or updating knowledge base, and the agents in the MAST system do not have to be changed to be able to use such update.

7 Acknowledgements

This paper is extended version of our paper “Semantics in Industrial Distributed Systems” presented at the 17th IFAC World Congress.

This work has been supported by Rockwell Automation, by the Ministry of Education of the Czech Republic within the Research Program No.MSM6840770038: Decision Making and Control for Manufacturing III and by the FIT-IT: Semantic Systems Program, an initiative of the Austrian Federal Ministry of Transport, Innovation, and Technology (bm:vit) under the contract FFG 815132.

8 References

[1] Y. Al-Safi, and V. Vyatkin (2007). An Ontology-Based Reconfiguration Agent for Intelligent Mechatronic Systems. In: *HoloMAS 2007*, LNAI 4659, Springer, Berlin - Heidelberg, pp. 114-126.

[2] V. Andreev, G. Rzevski, P. Skobelev, and P. Shveykin (2007). Adaptive Planning for Supply Chain Networks. In: *HoloMAS 2007*, LNAI 4659, Springer, Berlin - Heidelberg, pp. 215-224.

[3] ANSI/ISA-88.01.1995 (1995). *Batch Control Part 1: Models and Terminology*. The Instrumentation, Systems and Automation Society.

[4] R. Batres, M. West, D. Leal, D. Priced, and Y. Nakaa (2007). An upper ontology based on ISO 15926. *Computers & Chemical Engineering*, Vol. 31, No. 5-6, pp. 519-534.

[5] F. Bellifemine, G. Caire, and D. Greenwood (2007). *Developing multi-agent systems with JADE*. Wiley, Chichester.

[6] S. Borgo, and P. Leitão (2004). The role of foundational ontologies in manufacturing domain applications. In: *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, LNCS 3290, Springer, Berlin - Heidelberg, pp. 670–688.

[7] R. Brennan, P. Vrba, P. Tichy, A. Zoitl, C. Sünder, T. Strasser, and V. Mařík (2008). Developments in Dynamic and Intelligent Reconfiguration of Industrial Automation. *Computers in Industry*, Vol. 59/6, Elsevier B.V, pp. 533-547.

[8] S. Bussmann, N.R. Jennings, and M. Wooldridge (2004). *Multiagent Systems for Manufacturing Control: A Design Methodology*. Springer, Berlin - Heidelberg.

[9] L. M. Camarinha-Matos (2002). Multi-Agent Systems In Virtual Enterprises. In: *Proceedings of International Conference on AI, Simulation and Planning in High Autonomy Systems,* SCS, Lisbon, Portugal, pp. 27-36.

[10] G. Cândido, and J. Barata (2007). A Multiagent Control System for Shop Floor Assembly. In: *HoloMAS 2007*, LNAI 4659, Springer, Berlin - Heidelberg, pp. 293-302.

[11] M. Cossentino, N. Gaud, S. Galland, V. Hilaire, and A. Koukam (2007). A Holonic Metamodel for Agent-Oriented Analysis and Design. In: *HoloMAS 2007*, LNAI 4659, Springer, Berlin - Heidelberg, pp. 237-246.

[12] M. Dean, and G. Schreiber (2004). *OWL Web Ontology Language reference*. http://www.w3.org/TR/owl-ref/ (Accessed 25 September 2007)

[13] FIPA. *ACL Message Structure Specification*. Available at: http://www.fipa.org/specs/fipa00061/index.html.

[14] M. Fletcher, and J. Brusey (2003). The Story of the Holonic Packing Cell. In: *Proceedings of 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, ACM Press, Melbourne, Australia.

[15] M. Grüninger, and J. B. Kopena (2005). Planning and the Process Specification Language, In: *Proceedings of WS2 ICAPS 2005*, pp. 22-29.

[16] N. Guarino, and P. Giaretta (1995). Ontologies and Knowledge Bases – Towards a Terminological Clarification. In: *Towards Very Large Knowledge Bases*, IOS Press, Amsterdam.

[17] Ch. Hahn, and K. Fischer (2007). Service Composition in Holonic Multiagent Systems: Model-Driven Choreography and Orchestration. In: *HoloMAS 2007*, LNAI 4659, Springer, Berlin - Heidelberg, pp. 47-58.

[18] I. Hegny, O. Hummer, A. Zoitl, G. Koppensteiner, and M. Merdan (2008). Integrating Software Agents and IEC 61499 Realtime Control for Reconfigurable Distributed Manufacturing Systems. *International Symposium on Industrial Embedded Systems*, June 2008, pp. 249-252.

[19] B. Hellingrath, M. Witthaut, C. Böhle, and S. Brügger (2009). An Organizational Knowledge for Automotive Supply Chains. In: *HoloMAS 2009*, LNAI 5696, Springer-Verlag Berlin Heidelberg, pp. 37-46.

[20] A. Helsinger, M. Thome, T. Wright, and T. Cougaar (2004). A Scalable, Distributed Multi-agent Architecture. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, pp. 1910-1917.

[21] I. Herman, (2009). W3C Semantic Web Homepage. http://www.w3.org/2001/sw/ (accessed 11 April 2009)

[22] K. Jang-won, J. Keunhwan, J. Dongwon, K. Jinhyung, and B. Doo-Kwon (2006). An Ontology-based RFID System Model for Supporting Semantic Consistency in Ubiquitous Environment. In: *Proceedings of International Conference on Computational Intelligence and Security*, pp. 357-366.

[23] G. Koppensteiner, M. Merdan, A. Zoitl, and B. Favre-Bulle (2008). Ontology-Based Resource Allocation in Distributed Systems Using Director Facilitator Agents. *IEEE International Symposium on Industrial Electronics*, pp. 1721-1726.

[24] J. L. M. Lastra, and I. M. Delamer (2006). Semantic Web Services in Factory Automation: Fundamental Insights and Research Roadmap. *IEEE Transactions on Industrial Informatics*, Vol. 2, No. 1, pp. 1–11.

[25] H. Lausen, A. Polleres, and D. Roman (2005). Web Service Modeling Ontology WSMO. http://www.w3.org/Submission/WSMO/ (Accessed 26 July 2010).

[26] P. Leitao (2009). Holonic Rationale and Self-organization on Design of Complex Evolvable Systems. In: *HoloMAS 2009*, LNAI 5696, Springer-Verlag, Berlin - Heidelberg, pp. 13-24.

[27] S. Lemaignan, A. Siadat, J.-Y. Dantan, and A. Semenenko (2006). MASON: A Proposal for an Ontology of Manufacturing Domain. In: *IEEE Workshop on Distributed Intelligent Systems*, IEEE Computer Society Press, pp. 195–200.

[28] O. Lopez, and J. L. M. Lastra (2006). Using Semantic Web Technologies to Describe Automation Objects. *Int. J. Manufacturing Research*, Vo. 1, No. 4, pp. 482-503.

[29] O. Lopez, and J. L. M. Lastra (2007). A Real-Time Interface for Agent-Based Control. In *Proceeding of International Symposium on Industrial Embedded Systems SIES'07*, pp. 49-54.

[30] F. Manola, and E. Miller (2004). RDF primer. http://www.w3.org/TR/rdf-primer/ (Accessed 25 September 2007).

[31] C. McLean, Y. T. Lee, G. Shao, and F. Riddick (2005). Shop Data Model and Interface Specification, NISTIR 7198.

[32] M. Mendes, P. Leitao, F. Restivo, A. Colombo (2009). Service-Oriented Agents for Collaborative Industrial Automation and Production Systems. In: *HoloMAS 2009*, LNAI 5696, Springer-Verlag, Berlin, Heidelberg 2009, pp. 13-24.

[33] M. Merdan, G. Koppensteiner, I. Hegny, B. Favre-Bulle (2008). Application of an Ontology in a Transport Domain. *IEEE International Conference on Industrial Technology*, Sichuan University, Chengdu, China, pp. 1-6.

[34] M. Merdan (2009). *Knowledge-based Multi-Agent Architecture Applied in the Assembly Domain*. Dissertation thesis, Vienna University of Technology.

[35] M. Obitko, and V. Mařík (2003). Adding OWL Semantics to Ontologies Used in Multi-agent Systems for Manufacturing. In: *HoloMAS 2003*, LNAI 2744, Springer, Berlin - Heidelberg, pp. 189–200.

[36] M. Obitko, and V. Mařík (2005). Integrating Transportation Ontologies Using Semantic Web Languages. In: *HoloMAS 2005*, LNAI 3593, Springer, Berlin -Heidelberg, pp. 189–200.

[37] M. Obitko, and V. Mařík (2006). Transparent Ontological Integration of Multi-Agent Systems. In: *IEEE International Conference on Systems, Man, and Cybernetics*, IEEE SMC, pp. 2488-2492.

[38] Obitko, M. (2007). *Translations between Ontologies in Multi-Agent Systems*. PhD thesis, Czech Technical University, Prague.

[39] M. Pěchouček, and V. Mařík (2008). Industrial Deployment of Multi-Agent Technologies: Review and Selected Case Studies. *Autonomous Agents and Multi-Agent Systems*, Vol. 17, Issue 3, p. 397-431.

[40] M. Pěchouček, M. Rehák, P. Charvát and T. Vlček (2007). Agent-based Approach to Mass-Oriented Production Planning: Case Study. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, Vol. 37, No. 3, pp. 386-395.

[41] E. Prud'hommeaux, and A. Searborne (2008). SPARQL Query Language for RDF. W3C Recommendation. http://www.w3.org/TR/rdf-sparql-query/ (accessed 11 April 2009).

[42] G. Rzevski, P. Skobelev, and V. Andreev (2007). MagentaToolkit: A Set of Multi-agent Tools for Developing Adaptive Real-Time Applications. In:

HoloMAS 2007, LNAI 4659, Springer, Berlin - Heidelberg, pp. 303-313.

[43] C. Schroth, and T. Janner (2007). Web 2.0 and SOA: Converging concepts enabling the internet of services. *IEEE IT Professional*, Vol. 9, No. 3, pp. 36–41.

[44] D. Šišlák, M. Rehák, M. Pěchouček, and D. Pavlíček (2006). Deployment of A-globe Multi-Agent Platform. In: *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1447-1448.

[45] The OWL Services Coalition (2004). *OWL-S: Semantic markup for web services*. http://www.daml.org/services/ owl-s/1.0/owl-s.html (Accessed 25 September 2007).

[46] P. Vrba (2006). Simulation in agent-based control systems: MAST case study. *Int. J. Manufacturing Technology and Management*, Vol. 8, No. 1/2/3, pp. 175–187.

[47] P. Vrba, F. Macůrek, and V. Mařík (2008). Using Radio Frequency Identification In Agent-Based Control Systems For Industrial Applications. *Engineering Applications of Artificial Intelligence*, Vol. 21/3, pp. 331-342.

[48] P. Vrba, V. Mařík, and M. Merdan (2008). Physical Deployment of Agent-based Industrial Control Solutions: MAST Story. In: *Proceedings of IEEE International Conference on Distributed Human-Machine Systems*, Athens, pp. 133-139.

[49] P. Vrba, M. Radakovič, M. Obitko, and V. Mařík (2009). Semantic Extension of Agent-Based Control: The Packing Cell Case Study. In: *HoloMAS 2009*, LNAI 5696, Springer-Verlag, Berlin - Heidelberg, pp. 47-60.

[50] V. Vyatkin, J. Christensen, J. L. M. Lastra, and F. Auinger (2005). OOONEIDA: An Open, Object-Oriented Knowledge Economy for Intelligent Industrial Automation. *IEEE Transactions on Industrial Informatics*, Vol. 1, No. 1, pp. 4-17.

The Role of the Semantic Web for Knowledge Management in the Construction Industry

Igor Svetel
Innovation Center, Faculty of Mechanical Engineering
Belgrade, Serbia
E-mail: isvetel@mas.bg.ac.rs

Milica Pejanović
Faculty of Architecture, University of Belgrade
Belgrade, Serbia
E-mail: pmilica@arh.bg.ac.rs

Keywords: semantic web, building information modelling, BIM, industry foundation classes, IFC, knowledge management

Received: October 31, 2009

The architecture, engineering and construction (AEC) industry is knowledge intensive field. Significant heterogeneity of the forms of knowledge mobilized in the construction industry prevented adoption of IT based knowledge management in the field. Recently, a large international initiative is launched to provide extensive IT support that will enable model-based interoperability among all professions in the AEC industry. Resulting standards coupled with Semantic Web technologies have potential to serve as the foundation for the knowledge management in the AEC field. The paper gives an overview of the both technologies and depicts ways in which they can provide knowledge management support for the AEC industry.

Povzetek: Predstavljena je vloga semantičnega spleta pri upravljanju znanja v industriji.

1 Introduction

The architecture, engineering and construction (AEC) industry faces with continued social pressure to improve quality, responsiveness, reliability and efficiency in its business. Time delays, unforeseen work and costs, and resulting lawsuits among participants become routine events and result in exaggerated prices of buildings. Worldwide reports give an account that approximately 5% of total annual turnover in the AEC industry is lost because of the inadequate interoperability among participants in the industry and a lack of standardization in technology adoption among stakeholders [1] [2] .

Recently, a large international initiative is launched to provide extensive Information Technology (IT) support that will enable model-based interoperability among all professions in the AEC industry. Open standards developed as the part of the initiative have an extended potential to serve as the foundation for the knowledge management in the AEC field, especially in connection with Semantic Web technologies.

The paper gives an overview of the standards developed to provide interoperability in the AEC industry and standards developed under Semantic Web initiative. It also depicts ways in which both can be combined to provide knowledge management support for the AEC industry.

2 Knowledge in AEC

The AEC industry is knowledge intensive field. It encompasses heterogeneous expertise from multiple fields and diverse occupational groups. The knowledge ranges from tacit knowledge of architects on the ways of combining spaces to accommodate various social needs to the practical knowledge of builders on the use of tools and materials to construct physical spaces.

Research on the knowledge management in the AEC industry identified many specific features of the process [3] . The process of aligning the symbolic and the material is essential feature of the knowledge in the AEC industry. Up to the phase of the physical assembly of the building, the whole process is based on the symbolic representations in the form of plans, calculations and descriptive texts. Still, in minds of the participants those representations are constantly linked to the real physical entities. This feature was main reason for the rejection of IT systems in AEC community because those systems supported only symbolic processing.

Another significant feature is the importance of talk and communication in the process of knowledge sharing. This aspect is often ignored and consequently IT solutions support only formal representations. However, observations and field research demonstrate that formal representations serve as the stage around which series of

informal verbal discussions occur that are essential for the enlargement of the knowledge about design [4] .

The importance of external influences is also a feature that lacks support in traditional IT systems for the AEC. The large amount of the knowledge that constitutes one AEC project is produced elsewhere and is continuously attended and integrated into the existing procedures and activities. Many new products that constitute finished building are permanently produced, as well as procedures and regulations that account for their proper use. AEC project is an ongoing effort at aligning a variety of heterogeneous resources and practices.

3 Open standards for AEC

Based on years of research on a general data model for AEC [5] [6] the term Building Information Modelling (BIM) denotes a process of using IT to model and manage data encompassing the whole facility lifecycle [7] . The BIM concept means to build a facility virtually, prior to building it physically, in order to work out problems and simulate and analyse potential impacts. It is easier to fix a problem by moving element with a mouse than to demolish and rebuild elements on a construction site. The commercially developed BIM applications support creation of the computer-based facility model using parametric three-dimensional (3D) components with attached descriptive parameters that are necessary to identify particular elements. Still, those applications typically use proprietary data formats to represent facility models thus keeping all information locked in distinct software.

The need to establish interoperability among applications dealing with different phases of the facility lifecycle, such as architectural design, civil engineering, HVAC design, building construction, and facility management (FM), was met with the development of the Industry Foundation Classes (IFC) standard [8] . The currently available model is Version 2, Revision 3 and is registered as the ISO/PAS 16739:2005 standard. IFC is a neutral and open model whose development is conducted by the International Alliance for Interoperability (AIA), which has 550 member organizations in 24 countries. The standard provides the following basic functionality:

- Data interchange without information loss among all AEC and facility management (FM) applications.
- Unified model-based description of all building components.
- Information on the graphical representation of components.
- Description of relationships with other components and their location in the whole structure.
- Link to property and classification data and access to external libraries.

The open specification of the IFC data model allows commercial software developers to write interfaces for their software that enable exchange and sharing of the same data in the same format with other software applications, regardless of the internal data structure of any individual software application. All leading software companies like Autodesk, Bentley System, Graphisoft, Nemetschek, Data Design System, Solibri, Tekla, Archimen Group, Vector–Works, etc. support IFC in their applications.

Being an object-oriented data model, the IFC standard is comprised of class definitions representing all things and events occurring in the facility lifecycle. At the top of the hierarchy is a domain layer that describes classes related to basic functional units: building controls, plumbing and fire protection, structural elements, structural analysis, heating, ventilation and air conditioning, electrical circuits, architecture, construction and facilities management. Below that layer rests the interoperability layer that defines all classes essential for connection and cooperation among disciplines. Next is the core layer, containing basic model classes depicting controls, products, and processes. The resource layer is at the bottom, embodying classes that represent all building elements. Elements encompass not only physical components, as traditional models, but also actors and their roles, time, price, approval, etc.

The IFC standard does not produce one monolithic data model encompassing the whole lifecycle. Instead, many separate models are generated. In the context of IFC, a View is a defined as a subset of the IFC Object Model that a number of implementers have agreed to support in their implementations. The software certification process is conducted according to IFC Views. Depending on agreement, many IFC Views can exist with partially overlapping content or with entirely different contents. The data exchange between applications should occur within the scope of a specific View. The entire facility lifecycle is represented across multiple Views.

The IFC standard relates to the representation of a particular instance of the facility, its components, properties, and relationships. Using the vocabulary of object-oriented modelling, it can be said that it deals with object instances. It does not allow representation of the object classes and their relationships (i.e., that part is covered by the International Framework for Dictionaries (IFD), registered as the ISO 12006-3:2007 standard [9]).

IFD is the classification system for all information in the AEC/FM field. It is an object-oriented framework that defines objects, collections and their relationships. It is intended to work as the overarching structure that will provide support for the development of the unified AEC/FM vocabularies at the national, regional or domain levels. Since all share the same structure it will be possible to translate terms between languages and domains, preferably using automated software agents. IFD identifies each object in the model and this provides the capability to define context within which a concept is going to be used. Each object can have multiple names providing for the definition of synonyms or usage in different languages. An object is related to a formal classification system using references. The standard supports the following types of objects: Subjects, Activities, Actors, Units, Measures with Units, and Properties. Relationships are divided into Association, Collection, Specialization, Composition, Involvement, Property Assignment, Sequence and Measure

Assignment. Employing these mechanisms, the user can create a model-based definition of all concepts in AEC/FM including facts about classification systems, information models, object models, and process models. In other words, IFD functions as the IFC metamodel. In addition, IFD provides a unique global reference for any AEC/FM concept. The mechanism that relates IFC and IFD standards is scheduled to be published in the IFC 2x4 standard revision. The actual realization of IFD is the IFD Library, an international initiative currently run by four nations: Canada, Netherlands, Norway and USA. The purpose of the library is to provide semantic knowledge to the construction industry in a global and uniform way.

In addition to the above-described standards, a second type of interoperability formats has been developed based on another open standard - eXtensible Markup Language (XML). XML is a general-purpose specification capable of describing published data. The data description mechanism is based on the insertion of tags in the traditional text and the user can choose any term to define a particular tag. The language permits representation of arbitrary data arranged as an hierarchical tree with one element serving as the tree root [10] XML enables the structured representation of any kind of information but does not provide any mechanism to infer the meaning of the terms used in tags. One approach to the definition of a tag's meaning is the XML schema. It is a language that provides a description of a type of XML document, usually articulated in terms of constraints on the structure and content of related XML documents. Many schemas have been developed for the AEC/FM field. The gbXML (Green Building XML) schema is used for describing data relating to the building energy efficiency of the facility and its impact on the environment. The aecXML schema is used for depicting all building data in design, engineering and construction disciplines, and the CityGML schema is used for geo-spatial data representation. In addition, IFC data can be represented with the ifcXML schema. Since the IFD is an EXPRESS model, the EXPRESS to XML Schema Converter [11] can be used to obtain the XML schema for IFD.

Open standards developed for the AEC/FM industry relate to the problem of interoperability, since this is the most obvious obstacle in the industry. These standards enable the highly structured representation of information about buildings but do not consider the problem of information reuse outside of the context of a particular facility lifecycle or the automatic creation of new information for later reuse. Since all described standards have suitable XML schema representations, extension of the knowledge management capabilities can be achieved with technologies developed under the Semantic Web initiative.

4 Semantic Web, ontology and knowledge management

The recent advent of the Semantic Web has given a new impulse to the old knowledge management research field. The goal of the endeavour is to build a unified information medium that is both understandable for people and computers and that can be used for the automatic deduction of meaningful inferences [12] . To function effectively, the Semantic Web should be built on structured collections of information and sets of inference rules. Research on knowledge representation conducted as the part of the long time effort to build artificial intelligence systems has already developed many useful technologies. Unfortunately, those systems are centralized, relying on limited sets of rules to describe narrowly defined domains making the reuse of rules in new domains impractical. Similarly to the hypertext concept, when existing knowledge representation concepts were coupled with the global information system, the full potential of the technology was realized and this spurred a new wave of interest in the knowledge management field.

The new attempt to create universal knowledge representations is based on the layered structure of representation standards. The upper layer exploits functionality of lower layers and provides greater semantic expressiveness. At the bottom level resides XML. Meaning is expressed in the next layer containing the Resource Description Framework (RDF), a data model for representing information about entities in the Web [13] . In the Semantic Web standards, an entity or thing is referred as the resource. RDF achieves its functionality by using triples, a structure consisting of subject, predicate and object. This formation states that a particular thing (subject) has a property (predicate) with a particular value (object). The Universal Resource Identifier (URI) identifies subject and predicate and their value is either URI or literal. URIs ensure that concepts are not just bare terms devised by someone, but are connected to unique definitions on the Web. When multiple triples point to the same resource, they start to form a network of information about related things. That way information that defines a single entity is not held in one place but is spread over the Web forming a distributed web of data. However, RDF has no mechanism for determining that two or more dissimilar terms point to the same concept.

The next level of the semantic expressiveness is achieved with ontology. In the Semantic Web domain, ontology is identified as the formal representation that defines relationships among terms. The first level of ontological functionality is achieved with the RDF Schema (RDFS) [14] . Like other schema languages, RDFS provides information about basic RDF structures. It accomplishes this task by supplying constructs that allow the declaration of classes, subclasses, property, and subproperty relationships among resources. Constructs domain and range describe the relationship between properties and classes. These definitions are expressed using RDF triples. RDFS provides a limited set of inference rules that are restricted to the transitive closure of the hierarchies.

The Web Ontology Language (OWL) currently provides the highest level of ontological functionality among Semantic Web technologies. It is a family of

languages based on two semantics. OWL Lite and OWL DL are based on Description Logic semantics that guarantee completeness of reasoning. OWL Lite is a restricted version of OWL DL and is intended as a quick migration path for thesauri and other taxonomies. OWL Full provides maximum expressiveness and the syntactic freedom of RDF, but does not support complete or efficient reasoning. The language provides constructs like class, property, property restrictions, Boolean combinations, enumerations and instances. A wide range of services like reasoners and editing tools enable users to express and test knowledge using this formalism leading to the widespread acceptance of this technology [15] .

The level of expressiveness and functionality realized in the Semantic Web development surpasses previous attempts to model computer-based knowledge management systems. It is the idea of an open community, essential to the notion of the Web, which attracts so many people to the field and generates so many results. Anyone can use open standards to develop personal systems, use open source software to express his/her knowledge, or can engage in the development of standards. This openness of the development process has resulted in the remarkable range and richness of topics covered by the Semantic Web. Moreover, the organic development that grows from the interests and energy of the supporting community persuades an increasing number of researchers to adopt both Web standards and the open development principle as the foundations for development in their professional domains.

5 Semantic Web based knowledge management in AEC

Even if the IFC standard was not developed with the knowledge management in mind, it contains many supporting features. First, it represents a standardised object-oriented model of the building around which all professions in the AEC industry can focus their collaboration. In addition, since the globally unique identifier (GUID) identifies every object in the model, the IFC model provides unique definition of the IFC objects on the Web. If the GUID is connected with the physical building element by attaching an RFID tag to the element, it provides unique identification of the elements in the real world. That way a "symbol grounding problem" [16] that arises when a meaning is assigned to a symbol system can be solved to the level of real physical entities in the AEC field.

Multi-disciplinary professionals are involved in AEC projects, with various viewpoints, goals, priorities, and backgrounds. Diverse terms are employed to represent similar concepts or a single term for different concepts. By providing a unique global reference for any concept in the AEC field, the IFD supports communication among participants. If any dispute arises around denotation of some term, the IFD serves as the central authority. In addition, it provides mapping mechanism between usage of terms in different AEC professions and occupations enabling participants to understand model elements from their point of view.

Still, full knowledge management support can be obtained only by extending these standards with Semantic Web technologies. The IFC can describe only particular object instances, and IFD provides domain ontology, the structure of concepts and the relationships between them. They lack power to express classes, aggregations and rules.

The Semantic Web technologies provide all functionality necessary to support importance of external influences in AEC knowledge management.

The search for information on required materials and building elements represents around 18% of time spent on the building's design [17] . The Semantic Web can reduce this time by automatically acquiring links to relevant resources on the Web providing information about needed products that matches projects requirements. This process will use both Semantic Web technologies to acquire meaning of the information published on the web, and IFD to connect that information to the terms used in the IFC model. The automatic discovery and invocation of building product information additionally improves design process by enabling retrieval of cost and performance criterions, regulatory standards or component availability or delivery schedule that enables designer to give more precise predictions about building performances and construction schedules to his clients.

The scale of the waste from reinvention in design firms is also around 18% of time [17] . The Semantic Web technologies coupled with IFC and IFD standards provide a mechanism that will keep all information on previous projects in the form that will enable automatic retrieval of required information in future projects. The information on the project will serve not only particular firm, but also if published on the Web it will become a global reference available to all interested parties.

So far, few prototype systems that link Semantic Web and IFC standard are developed. The easier technique to extend existing AEC standard formats and enable knowledge management functionality is to add semantic annotations using RDF. The method is demonstrated in the system for conformity checking in construction [18] . The norms are extracted from the electronic regulations and formalized as SPARQL queries in terms of the IFC model. The conformity checking process is based on matching an RDF representation of a project to a SPARQL conformity query. The project's RDF representation is extracted from the ifcXML schema and later manually enriched with domain knowledge.

More projects are using OWL to add knowledge management functionality. One notable example is the Sydney Opera House facility management model [19] . The basic IFC model is transformed using an IFC-OWL converter [20] Existing tools are used to manually enrich this OWL representation with ontology and rules. The OWL representation is associated with the IFC model enabling the publication of performance by selecting spaces in the 3D model.

The InteliGrid project [21] also uses a custom developed IFC-OWL transformer [22] to enable the representation of expertise in addition to the basic representation of building elements and services.

The SWOP project [23] uses a custom developed Product Modelling Ontology (PMO) that is sufficiently rich to represent product ontology for any parametric product type. PMO is a layer on top of the RDF, RDFS and OWL hierarchy and models the product from the ontology point of view, meaning that an IFC model or any other product model can be obtained automatically as the result of the modelling process.

Today many open source converters for Semantic Web formats can be found. XSD2OWL provides transformation of an XML Schema into an OWL ontology and XML2RDF enables transformation of XML into RDF [24] . These tools together with the availability of many open source editors and development environments that support Semantic Web standards provide an opportunity to add meaning and enrich open AEC standards.

6 Conclusion

Integration of knowledge management capabilities with BIM methodology has great potential, especially since the current open standards approach shares the same technological background as the recent Semantic Web initiative. The openness of both activities motivates many researchers and software developers to join this initiative and make their contributions. This process guaranties the best adaptation of the technologies to the users' needs and their widest support and endorsement. However, the approach also has its drawbacks. The number of technologies and their variants is massive. Moreover, the pace of change is so rapid that systems based on today's most recent technology become obsolete over a two to three-year period, requiring developers to update constantly their products to take advantage of the latest version of the applied technology.

Acknowledgement

The Ministry of Science and Technological Development - Republic of Serbia supported this work under grant TP-16025. It is a part of the project 'Pilot Project: Application of the IFC (ISO/PAS 16739:2005) Standard in Drywall Systems.' The project director is Dr Igor Svetel.

References

[1] Ercoskun, K., and Kanoglu A. (2003). Bridging The Gap Between Design and Use Processes: Sector-Based Problems of a CRM Oriented Approach. *9th EuropIA International Conference: EIA9: E-Activities and Intelligent Support in Design and the Built Environment*, Istanbul, Turkey, pp. 25-30.

[2] Gallaher, M.P., et al. (2004). *Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry*. NIST GCR 04-867 Report.

[3] Styhre A. (2009). *Managing Knowledge in the Construction Industry*. Taylor & Francis, Oxon.

[4] Lawson B. (2004). *What Designers Know*. Architectural Press, Oxford.

[5] Eastman, C.M., Bond, A. and S. Chase (1991). A Data Model for Engineering Design Databases. *Artificial Intelligence in Design '91*. (ed. J.S. Gero), Butterworth-Heinemann, pp. 339-365.

[6] Björk, B.-C. (1989). Basic structure of a proposed building product model. *Computer-Aided Design.* vol. 21, no. 2, pp. 71-78.

[7] Lee, G., Sacks, R., and C.M., Eastman (2006); Specifying parametric building object behavior (BOB) for a building information modeling system; *Automation in Construction*, vol. 15, no. 6, pp. 758-776.

[8] Liebich, T. et al. (eds.) (2007). *IFC2x Edition 3 TC1*. International Alliance for Interoperability. Available from http://www.iai-tech.org/ifc/IFC2x3/TC1/html/index.htm

[9] IFD (2008). *IFD Library White Paper.* Available from http://www.ifd-library.org/images/IFD_Library_White_Paper_2008-04-10_I_.pdf

[10] Harold, E.R., Means, W.S. (2004). *XML in a Nutshell.* Third Edition, O'Reilly Media, Inc., Sebastopol, CA.

[11] EXC (2002). *EXPRESS to XML Schema Converter.* Available from http://cic.vtt.fi/projects/ifcsvr/index_exc.html (May 2009)

[12] Berners-Lee, T., Hendler, J., and Lassila O. (2001). The Semantic Web. *Scientific American,* Vol. 284, No. 5, pp. 34-43.

[13] Klyne, G. and J.J., Carroll (eds.) (2004). *Resource Description Framework (RDF): Concepts and Abstract Syntax.* W3C Recommendation 10 February 2004, Available from http://www.w3.org/TR/rdf-concepts/

[14] Allemang D. and Hendler J.A. (2008). *Semantic web for the working ontologist modeling in RDF, RDFS and OWL.* Morgan Kaufmann, Burlington.

[15] McGuinness D.L., van Harmelen F., (eds.) (2004). *OWL Web Ontology Language Overview.* W3C Recommendation 10 February 2004, Available from http://www.w3.org/TR/owl-features/.

[16] Harnad, S. (1990). The Symbol Grounding Problem. *Physica D 42.* pp. 335-346.

[17] Bartholomew D. (2008). *Building on Knowledge*. John Wiley & Sons Ltd, Chichester.

[18] Yurchyshyna, A., et al. (2008). Towards the Knowledge Capitalisation and Organisation in the Model of Conformity-Checking Process in Construction. *Knowledge-Based Intelligent Information and Engineering Systems. KES 2008*. Part I, vol. 5177 (eds. I. Lovrek, R.J. Howlett, and L.C. Jain), Springer, pp. 341–348.

[19] Schevers, H., et al. (2007). Towards Digital Facility Modelling for Sydney Opera House Using IFC and Semantic Web Technology. *Itcon*, Vol. 12, pp. 347-362.

[20] Schevers, H. and R. Drogemuller (2005). Converting IFC Data to the Web Ontology

Language. *Proc. International Conference on Semantics, Knowledge and Grid.* Beijing, China pp. 73-73.

[21] Dolenc, M., et al. (2007). The InteliGrid Platform for Virtual Organisations Interoperability. *Itcon*, Vol. 12, .pp. 459- 477.

[22] Beetz, J., van Leeuwen, J. P., and B. de Vries (2005). An Ontology Web Language Notation of the Industry Foundation Classes. *Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction.* (eds. R.J., Scherer, P., Katranuschkov and S.-E. Schapke) CIB publication no. 304, Dresden, pp. 193-198.

[23] Böhms, M. et al. (2008). *The SWOP Semantic Product Modelling Approach.* STRP NMP2-CT-2005-016972 report. Available from http://www.swop-project.eu/swop-solutions/semantic-product-modelling-approach/SWOP_D23_WP2_T2300_TNO_2008-04-15_v12.pdf (May 2009)

[24] ReDeFer (2009). *ReDeFer - compendium of RDF-aware utilities.* Available from http://rhizomik.net/redefer (May 2009)

Cryptanalysis of a Simple Three-party Key Exchange Protocol

He Debiao, Chen Jianhua and Hu Jin
School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei 430072, China
E-mail: {hedebiao, chenjh_ecc, hujin_ecc}@163.com

Keywords: key exchange protocol, secure communication, password, dictionary attack, Diffie-Hellman assumption

Received: November 5, 2009

Key exchange protocols allow two or more parties communicating over a public network to establish a common secret key called a session key. Due to their significance in building a secure communication channel, a number of key exchange protocols have been suggested over the years for a variety of settings. Among these is the so-called S-3PAKE protocol proposed by Lu and Cao for password-authenticated key exchange in the three-party setting. In the current work, we are concerned with the password security of the S-3PAKE protocol. We first show that S-3PAKE is vulnerable to an off-line dictionary attack in which an attacker exhaustively enumerates all possible passwords in an off-line manner to determine the correct one. We then figure out how to eliminate the security vulnerability of S-3PAKE.

Povzetek: Prispevek se ukvarja z varnostjo v protokolu S-3PAKE.

1 Introduction

In 1992, Bellovin and Merritt [1] proposed a two-party encrypted key exchange protocol based on user passwords. Since then, many two-party password-based authenticated key exchange (2PAKE) protocols have been proposed. Because 2PAKE protocols are only suitable for the client − server architecture, some researchers extended 2PAKE protocols into 3PAKE protocols for three-party environments. Most existing 3PAKE protocols are designed for the client− client− server architecture, in which each client (user) shares his password with a trusted server and resorts to the server to authenticate the peer for establishing a session key. In 2004, Lee et al. [2] presented two enhanced three-party encrypted key exchange (3PEKE) protocols without using public key techniques, and showed that their protocols can resist several attacks, achieve mutual authentica-tion, and provide perfect forward secrecy. In 2005, Wen et al. [3] proposed a 3PAKE protocol using Weil pairing and claimed that their protocol is provably secure against active adversaries in the random oracle model. However, Nam et al. [4] showed that Wen et al. ' s protocol cannot resist a man-in-the-middle attack, and then interpreted their attack in the context of the formal proof model. Recently, Lu and Cao [5] proposed a new simple 3PAKE (S-3PAKE) protocol based on the chosen-basis computational Diffie-Hellman (CCDH) assumption. They claimed that their protocol is superior to similar protocols with respect to security and efficiency.

According to recent works [6-9], S-3PAKE is vulnerable to various attacks like man-in-the-middle attacks [6, 7], an unknown key-share attack [8], undetectable on-line dictionary attacks [7-8], and off-line dictionary attacks[9].

The above-mentioned attacks are not the only ones that can compromise the security of the S-3PAKE protocol. We found that S-3PAKE is not secure against an off-line dictionary attack. The present work reports this new (and more serious) security problem with S-3PAKE and, in addition, shows how to fix it.

In this paper, we will first review the S-3PAKE protocol, and then demonstrate the attacks on the S-3PAKE protocol. Furthermore, we will suggest a countermeasure to enhance the security of the S-3PAKE protocol against the attacks.

2 The S-3PAKE protocol

The notations used in the S-3PAKE protocol are described as in the following:

- (G , g , p) represents a finite cyclic group G generated by an element g with prime order p .
- M and N denote two elements in G .
- S represents a trusted server.
- A and B represent the initiator and the responder, respectively, of a protocol run.
- pw_A denotes the password shared between A and S .
- pw_B denotes the password shared between B and S .
- $H()$ and $H'()$ denote two distinct secure one-way hash functions.

The security of the S-3PAKE protocol mainly relies on the chosen-basis computational Diffie-Hellman CCDH) assumption [5], which is a variation of the computational Diffie-Hellman (CDH) assumption. In he CDH assumption, given g^u and g^v, where u and v are drawn randomly from Z_p, it is computationally infeasible to compute g^{uv}, which is denoted by CDH(g^u, g^v). And, the CCDH assumption is defined as in the following: The adversary is given three random elements, M, N, and X in G, with the goal of finding a triple of values (Y, u, v) such that u = CDH(X, Y) and v = CDH($X / M, Y / N$). The idea behind the CCDH assumption is that the adversary may be able to successfully compute either u (by choosing $Y = g$ to obtain $u = X$) or v (by choosing $Y = g \cdot N$ to obtain $v = X / M$), but not both. Note that all modular operations in this paper are performed under modulo p, and we drop the operator $\bmod p$ for clearness. Suppose that A and B request to authenticate each other and then resort to S for a session key agreement. The steps of the protocol, can be briefly described as in the following:

Step 1. A chooses a random number $x \in Z_p$, computes $X = g^x$ and $X^* = X \cdot M^{pw_A}$, and sends $A \| X^*$ to B.

Step 2. B selects a random number $y \in Z_p$, computes $Y = g^y$ and $Y^* = Y \cdot N^{pw_B}$, and sends $A \| X^* \| B \| Y^*$ to S.

Step 3. Upon receiving $A \| X^* \| B \| Y^*$, S first recovers X and Y by computing $X = X^* / M^{pw_A}$ and $Y = Y^* / N^{pw_A}$. Next, S selects a random number $z \in Z_p$ and computes $\overline{X} = X^z$ and $\overline{Y} = Y^z$. S then computes $pw_A^* = H(A \| S \| X)^{pw_A}$, $pw_B^* = H(B \| S \| Y)^{pw_B}$, $\overline{X}^* = \overline{X} \cdot pw_B^*$, $\overline{Y}^* = \overline{Y} \cdot pw_A^*$ and sends $\overline{X}^* \| \overline{Y}^*$ to B.

Step 4. After having received $\overline{X}^* \| \overline{Y}^*$, B computes $pw_B^* = H(B \| S \| Y)^{pw_B}$, $K = (\overline{X}^* / pw_B^*)^y$, $\alpha = H(A \| B \| K)$, and sends $\overline{Y}^* \| \alpha$ to A.

Step 5. With $\overline{Y}^* \| \alpha$ from B, A computes $pw_A^* = H(A \| S \| Y)^{pw_A}$, $K = (\overline{Y}^* / pw_A^*)^x$, and verifies that α is equal to $\alpha = H(A \| B \| K)$. If the verification fails, then A aborts the protocol. Otherwise, A computes the session key $SK_A = H'(A \| B \| K)$ and sends $\beta = H(B \| A \| K)$ to B.

Step 6. B verifies the correctness of β by checking that the equation $\beta = H(B \| A \| K)$ holds or not. If it holds, then B computes the session key $SK_B = H'(A \| B \| K)$. Otherwise, B aborts the protocol.

The correctness of S-3PAKE can be easily verified [5].

3 Off-line dictionary attack on S-3PAKE

Lu and Cao [5] claim that their S-3PAKE protocol is secure against off-line dictionary attacks. This argument may hold if there only exist honest clients who stick to the protocol specification. But, there could be malicious clients who deviate from the protocol.

Indeed, we found that S-3PAKE is not secure against an offline dictionary attack in the presence of a malicious client. Assume that B is a malicious client, and wants to find out the password of client A. Then the following description represents our off-line dictionary attack mounted by B against A's password.

In the S-3PAKE protocol, the user B can easily obtain valid information $A \| X^*$, since all transcripts are transmitted over an open network. Then he/she does the following steps.

Phase 1. The attacker B runs the protocol with the server S while playing dual roles of B itself and the victim A.

1) B computes $Y^* = 1 \cdot N^{pw_B}$. Then B sends $A \| X^* \| B \| Y^*$ to S.
2) Upon receiving $A \| X^* \| B \| Y^*$, S first recovers X and Y by computing $X = X^* / M^{pw_A}$ and $Y = Y^* / N^{pw_A} = 1$. Next, S selects a random number $z \in Z_p$ and computes $\overline{X} = X^z$ and $\overline{Y} = Y^z = 1$. S then computes $pw_A^* = H(A \| S \| X)^{pw_A}$, $pw_B^* = H(B \| S \| Y)^{pw_B}$, $\overline{X}^* = \overline{X} \cdot pw_B^*$, $\overline{Y}^* = \overline{Y} \cdot pw_A^* = pw_A^* = H(A \| S \| X)^{pw_A}$ and sends $\overline{X}^* \| \overline{Y}^*$ to B.

Phase 2. When B receives ($\overline{X}^* \| \overline{Y}^*$), he/she can carry out the off-line dictionary attack using $X^* = X \cdot M^{pw_A}$ and $\overline{Y}^* = H(A \| S \| X)^{pw_A}$. The process of the off-line dictionary attack is as follows.

Step1. B selects a password s from a uniformly distributed dictionary D.

Step2. B computes $X' = X^* / M^s$ and $\overline{Y}^{*\prime} = H(A \| S \| X')^s$.

Step3. B then verifies the correctness of s by checking that $\overline{Y}^{*\prime}$ is equal to $\overline{Y}^*$.

Step3. B repeats steps 1, 2, and 3 of this phase until a correct password if found.

This off-line dictionary attack may lead to devastating losses of passwords, because it can be mounted against any registered client and does not even require the participation of the victim, and the steps for verifying password guesses can be performed in an off-line manner by an automated program.

It's easy to say that if B let Y^* be $(p-1)\cdot N^{pw_B}$, he/she can get the correct password pw_A by the same way. In a similar way, A can guess the password of the client B if A is a malicious client.

4 Countermeasure

The random number z is used in the S-3PAKE protocol to randomizing X and Y, but if Y^* is set $1\cdot N^{pw_B}$ or $(p-1)\cdot N^{pw_B}$, the function of the random number z is destroyed, then a malicious client can get the password pw_A through the off-line dictionary attack. Fortunately, in order to make the S-3PAKE protocol against the attack, we just need let the S check

$$X = 1, -1 \text{ and } Y = 1, -1$$

hold or not. If one of above equations holds, then S stops the protocol.

Theorem 1. The modified S-3PAKE scheme is secure against the off-line dictionary attack described in section 3.

Proof: When the server S checks $X = 1, -1$ and $Y = 1, -1$ hold or not. He will find the off-line dictionary attack by the malicious client. He stops the session, and records the attack in his database. After finding the attack for several times (three time for example), S stop the service supplied to the malicious client.

So, our countermeasure can withstand the off-line dictionary attack.

5 Conclusion

Herein, we have demonstrated that Lu-Cao's S-3PAKE protocol is potentially vulnerable to an off-line dictionary attack. The reason for the attack is due to the fact that the function of randomization is destroy when X or Y is set some special value. To enhance the security of the S-3PAKE protocol, we have suggested a countermeasure to resist our described attacks while the merits of the original protocol are left unchanged.

Reference

[1] S.M. Bellovin, M. Merritt (1992) "Encrypted key exchange: password-based protocols secure against dictionary attacks," *Proc. 1992 IEEE Symposium on Research in Security and Privacy*, pp. 72– 84.

[2] T.F. Lee, T. Hwang, C.L. Lin (2004) "Enhanced three-party encrypted key exchange without server public keys," *Computers Security* 23 (7), pp. 571–577.

[3] H.A. Wen, T.F. Lee, T. Hwang (2005) "Provably secure three-party password-based authenticated key exchange protocol using Weil pairing," *IEE Proceedings Communications* 152 (2) , pp. 138–143.

[4] J. Nam, Y. Lee, S. Kim, D. Won (2007) "Security weakness in a three-party pairing-based protocol for password authenticated key exchange," *Information Sciences* 177 (6), pp. 1364– 1375.

[5] R. Lu, Z. Cao (2007) "Simple three-party key exchange protocol," Computers Security 26 (1), pp. 94– 97.

[6] H.-R. Chung and W.-C. Ku (2008) "Three weaknesses in a simple three-party key exchange protocol," *Inform. Sciences* 178(1), pp. 220–229.

[7] H. Guo, Z. Li, Y. Mu, and X. Zhang (2008) "Cryptanalysis of simple threeparty key exchange protocol," Computers & Security, 27(1), pp. 16–21.

[8] R. C.-W. Phan, W.-C. Yau, and B.-M. Goi (2008) "Cryptanalysis of simple three-party key exchange protocol (S-3PAKE)," Inform. Sciences,178, (13), pp. 2849–2856.

[9] J. Nam, J. Paik, H. Kang, et al. (2009) An Off-Line Dictionary Attack on a Simple Three-Party Key Exchange Protocol, *IEEE COMMUNICATIONS LETTERS*, 13(3), pp. 205-207.

KP-Lab System for the Support of Collaborative Learning and Working Practices, Based on Trialogical Learning

Ján Paralič
Centre for Information Technologies, Technical University of Košice, 042 01 Košice, Slovakia
E-mail: jan.paralic@tuke.sk, http://people.tuke.sk/jan.paralic/paralic-a.html

František Babič and Jozef Wagner
Deparment of Cybernetics and Artificial Intelligence, Technical University of Košice, 042 01 Košice, Slovakia
E-mail: {frantisek.babic, jozef.wagner}@tuke.sk, http://web.tuke.sk/kkui/kkui-a.html

Peter Bednár and Marek Paralič
Centre for Information Technologies, Technical University of Košice, 042 01 Košice, Slovakia
E-mail: {peter.bednar, marek.paralic}@tuke.sk, http://web.tuke.sk/fei-cit/index-a.html

Keywords: trialogical learning, collaborative system, shared objects, semantics, knowledge practices

Received: November 12, 2009

This paper presents an approach in the domain of collaborative systems for working and learning practices called KP-Lab System. This system provides integrated multifunctional application with interesting end-user functionalities as manifold semantic based manipulation possibilities with shared objects of activities in real time, a support for management and analysis of knowledge practices, tools for synchronous and asynchronous communication, tools for personal organization and customization of working spaces, etc. Theoretical background for presented system is provided by trialogical learning, an approach in the domain of collaborative learning or working, with several similar aspects to existing constructivist approaches to learning. These approaches and some other theories, e.g. activity theory and knowledge building had a strong influence on specification of trialogical learning characteristics and their analysis in real settings. Presented research and development results have been achieved in FP6 IST project called KP-Lab (Knowledge Practices Laboratory). KP-Lab is an ambitious project that focuses on developing a theory, methods and tools aimed at facilitating innovative practices of sharing, creating and working with knowledge in education and workplaces. Research and development are integrated into co-evolutionary process that consists of collaboration between various types of project partners and other stakeholders This paper focuses mainly on the technological results of this project, the KP-Lab System, presenting its architecture, main tools and interesting features provided by this system, e.g. its strong semantics-based character.

Povzetek: Predstavljen je sistem KP-Lab za sodelovanje pri učenju.

1 Introduction

The process of research and development in the domain of technology supported learning brings continuously new methodologies, methods or approaches how to adapt on the new appearing situations posing changing requirements on information and communication technologies. This adaptation process is tightly connected with utilization of existing or newly emerging information technologies, e.g. Semantic web, Web 2.0, as well as the ever changing practices of learning and work. We can identify many existing theories with relevant supporting solutions as Virtual Learning Environments (VLE), Learning Management Systems (LMS) or Computer Supported Collaborative Learning (CSCL) and Computer Supported Collaborative Work (CSCW).

In this paper we describe a new collaborative application called KP-Lab System that can be compared with existing similar systems to identify advantages of the proposed solution. KP-Lab System provides a complex user virtual environment with integrated end-user tools based on the theoretical approach called Trialogical Learning (TL). This approach was derived from several existing theories in education areas to provide theoretical framework for realization of collaborative processes, focusing on the knowledge creation aspect of learning processes.

This system is one of the main KP-Lab[1] project outcomes. KP-Lab (Knowledge Practices Laboratory) is an IST project examining theoretical background of TL, identifying user requirements for its exploitation based on new technological solutions and accompanying

[1] http://www.kp-lab.org/

knowledge practices, implemented and tested on various pilot cases in different learning and working environments across the Europe and in Israel. The described system lays stress on support of collaborative and cooperative activities, work around shared objects of these activities, broader scale of communication possibilities (synchronous or asynchronous), manifold use of semantic information, service-oriented architecture and others.

The rest of the paper is organized as follows. Second section provides a short state of the art about existing approaches in education domain, which mainly influenced the concept of trialogical learning. This section provides also a short comparison between proposed collaborative system and some of the selected main applications in relevant domain. The third section is devoted to description of trialogical learning and its main aspects and principles as they have been identified and formulated in EU project KP-Lab, which is briefly described in the section four. The main technological outcome of this project, the KP-Lab System is described in section 5. KP-Lab System consists of two main parts: platform and virtual user environment with integrated end-user tools called KP-Environment (KPE). Section six provides detail comparison of the proposed KP-Lab System with several existing comparable solutions representing various types of VLE, LMS or collaborative working spaces such as Moodle, BSCW, SAKAI and Google Apps. Google Apps was selected as representative of new Web 2.0 technologies. Results of this comparison were used for declaration of innovative elements and advantages of the KP-Lab System. The main contributions of this paper are summed up in last, conclusions section.

2 State of the art

Trialogical learning (TL) can be defined as a theoretical framework in the domain of collaborative approaches to learning or working practices. TL is not supposed to be a „super-theory" on the basis of different background theories or replace any of these widely adopted approaches but it pinpoints certain kinds of phenomena which are prevalent and central nowadays: how people organize their work for developing some shared, concrete objects. This means situation that individuals (or groups of people) are developing some shared objects of activity within some social or cultural settings. As a representational example of trialogical activities can be mentioned the way how the wiki pages are collaboratively developed. It is a long-term effort of developing something for communal use on the basis of individual initiative. The interaction happens through shared objects (wiki pages) on the basis of other peoples' efforts. These main characteristics of trialogical activities represent possible similarity with other existing approaches in examined domain, e.g.:

- Carl Bereiter's knowledge building approach which emerged from cognitive studies in the educational context [16] ;
- Yrjö Engeström's theory of expansive learning based on Activity Theory [6] ;
- Nonaka and Takeuchi's model of organizational knowledge creation [12] ;
- Constructivist approaches representing paradigm that people create new knowledge based on their past experiences and interactions with other people or surrounding environment [3] , [4] .

Constructivism understands learning process not as passive receiving information and storing into the brain, but an active construction of knowledge and skills. Learning is a self-directed process, and the teacher can only help and support the learner to acquire new knowledge. This paradigm and relevant approaches are described within many works, e.g. [10] , [18] , [14] .

The trialogical learning takes some of the ideas from these approaches, adds some new ones and makes different focus of learning activities towards knowledge advancement (for details see next section). In order to validate this learning approach, the KP-Lab System has been designed and implemented.

KP-Lab System provides interesting features that make it possible to compare it with systems belonging to representatives of e.g. VLE, LMS CSCL or CSCW. We carried out some internal tests to identify advantages and potential disadvantages of our system with existing applications in mentioned domains. Some interesting findings are presented here, but detailed comparison with selected systems is described further in chapter six.

- Awareness of performed activities in user environment is implemented in different formats within most of the collaborative systems, but only BSCW and Moodle provide persistent storage of all performed events and some basic statistics of users' behavior too. KP-Lab System provides separate database for event logs and this historical data are used for several types of analyses, e.g. time-line based analyses or visual analysis of statistical summaries based on user preferences [15] .
- Basic free term search is implemented in any collaborative system, but BSCW provides in addition also semantic search, supporting different metadata standards and offering faceted browsing and filtering. KP-Lab System extends these functionalities with possibility to save search queries and their results.
- KP-Lab System provides possibility to create dynamic process models, not only predefined and rigid structure of created courses.
- KP-Lab System is strongly oriented on semantic information, the main source of data is represented by knowledge repository that stores ontologies as main semantic framework for the whole system; created and modified shared objects with relevant properties represented by metadata; and relations between relevant concepts of the whole system.

Actually it would be possible to compare KP-Lab System with new initiative in this field called Google Wave[2], but actual version is still available only for invited Google users.

3 Trialogical learning

The theoretical foundation of TL is based on knowledge-creation metaphor of learning, contrasting two most typical previous ones, i.e. knowledge acquisition metaphor and participation metaphor of learning (see Figure 1). These two core approaches (referenced also as monological and dialogical) describe two basic ways of understanding the area of learning by Anna Sfard [17] :

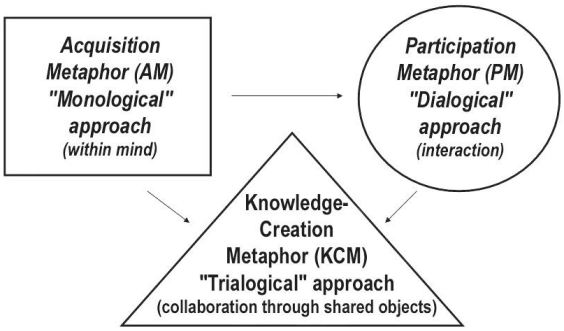


Figure 1: Three metaphors of learning, taken from [7] .

The acquisition (*monological*) metaphor of learning relies on the idea that knowledge is a characteristic of the individual mind, and the individual is the basic unit of knowing and learning.

The participation (*dialogical*) metaphor of learning understands learning as an interactive process of participating in cultural practices and shared learning activities that structure and shape cognitive activity in many ways.

From these two previous approaches third metaphor of learning has emerged that is aimed at overcoming the dichotomy between the acquisition and participation metaphors. It is knowledge-creation metaphor of learning [8] . While the acquisition metaphor represents a monological view on human learning as a mental within-mind process and the participation one represents a dialogical view based on interaction between humans and the cultural environment, the knowledge-creation perspective may be tagged as trialogical.

Definition describes TL as process where learners are collaboratively developing shared objects of activity (such as conceptual artefacts, practices, products) in a systematic fashion [7] . It concentrates on the interaction through these common objects (or artefacts) of activity (see Figure 2), not just among people or within one's mind.

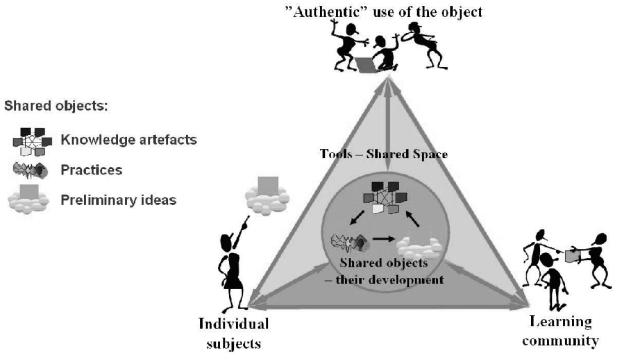


Figure 2: Structure of trialogical process, taken from [13] .

The main goal of trialogical approach is to facilitate innovative knowledge practices within educational or professional communities. A central characteristic of these practices is a collaborative pursuit of complex problems, sustained knowledge advancement, and transformation of practices within heterogeneous networks of actors representing both application areas. The following six, interrelated basic features characterize TL [7] :

1. Focus on shared objects of activity whether those are knowledge artefacts, concrete products or practices to be reflected on;
2. Sustained and longstanding pursuit of knowledge advancement;
3. Interaction between personal and collective knowledge advancement efforts;
4. Cross-fertilization of knowledge practices between educational and professional communities;
5. Development through transformation and reflection;
6. Flexible technology mediation designed to scaffold collective creation, building, and sharing of knowledge.

4 Knowledge practices laboratory

Theoretical as well as practical aspects of TL have been created and are continuously researched in Finland, at the University in Helsinki (Centre for Research on Networked Learning and Knowledge Building[3]) mainly. From this institution the idea has been distributed in the (not only) European educational community, e.g. within the Knowledge Practices Laboratory (KP-Lab) project, UH (University in Helsinki) being the project coordinator. KP-Lab is an ambitious project that focuses on developing a theory, methods and tools aimed at facilitating innovative practices of sharing, creating and working with knowledge in education and workplaces.

This five years long IST project is sponsored by European Commission within the "Sixth Framework Programme" and has been launched 1st of February 2006. The first three years the main focus was on the research and development of methods, practices and supporting

[2] http://wave.google.com/help/wave/about.html

[3] http://www.helsinki.fi/science/networkedlearning/eng/

tools. New tools were designed, developed and tested against user requirements in real situations within various pilot cases (in e.g. Finland, Holland, Switzerland, Hungary or Israel). The last two years have been devoted mainly to finish longitudinal experiments, improvements of existing tools based on experimental outcomes, dissemination activities and exploitation planning.

KP-Lab technological background consists of emerging technologies, such as Semantic web, Web 2.0, web services and service–oriented architecture in general, real-time multimedia communication, and ubiquitous access using wireless or mobile devices.

The multinational consortium integrates expertise from various domains, including pedagogy, psychology and engineering as well as end-users and key representatives from the corporate/business sector to provide authentic environments for research and piloting. The project involves 22 partners from 14 countries providing a suitable variety of universities, companies, work places and other prospective end-users.

The trialogical approach must be researched, developed and evaluated differently in different contexts. TL gives direction and ideas for developing existing practices and models to have more elements of collaborative knowledge creation, and both needs and problems of existing practices as well as the theoretical ideas have given requirements and guidance for the technology developed. This means that the basis for KP-Lab project is a challenging co-design model which must combine theoretical development, pedagogical research and models, and technology research & development. The main aim of its strategy is to provide a platform for researchers, developers and end users to create a shared understanding of current knowledge practices and to envision, design and evaluate novel applications and methods and thereby contribute to the facilitation of innovative knowledge practices.

4.1 Project objectives

The KP-Lab project aims at developing theories, tools, and practical models that elicit deliberate advancement and creation of knowledge (the trialogical knowledge-creation approach) as well as corresponding transformation of knowledge practices in education and workplaces.

In parallel with changes in society, conceptual frameworks, practices in school and at work, and social organization of learning also have to be transformed to facilitate development of corresponding individual and cultural competencies. The KP-Lab project examines these knowledge practices, i.e., innovative processes, routines, and procedures of working with knowledge. Knowledge practices represent socially constituted, rather than merely individual activities.

The following three objectives are central in framing the co-evolutionary nature of the KP-Lab project.

4.1.1 Theoretical development and modeling

Theoretical development aims at bridging a gap between individualistic and social approaches on learning and cognition by building on approaches focusing on knowledge creation processes, that is, how people collaboratively develop new artefacts, products, and ways of working in long-term processes. Based on close collaboration with empirical research and technology development the aim is to understand and conceptualize transformation of knowledge practices with the use of novel technology in education and workplaces. The emerging trialogical framework is disseminated at academic and professional arenas and publications.

4.1.2 Educational and professional knowledge practices

The general objective of the pedagogical research is to develop a set of pedagogical methods to foster knowledge creation in educational and workplace settings and to specify possibilities of their implementation. Within the field of higher education, the focus is on the development of methods for symmetric knowledge advancement. Symmetric knowledge advancement is realized when communities of learners cross the boundaries of a classroom or an organization and promote one another's advancements rather than emphasize a one-directional flow of knowledge and competence from old-timers to newcomers.

4.1.3 Technological development and research

The general objective of the technological research and development is to design and implement a modular, flexible, and extensible ICT system that supports the trialogical pedagogical methods to foster knowledge creation in educational and workplace settings. The system provides tools for collaborative work around shared objects of activity, and for knowledge practices in the various settings addressed by the project. The technological framework provides an operational technical architecture for KP-Lab tools and services, software modules allowing for interfacing KP-Lab tools with third-party software. A set of guidelines and reference documents to support the implementation will also be provided in the last part of the project.

5 KP-Lab system

KP-Lab project provides a modular, flexible, and extensible system with a set of integrated and cooperative applications to support TL in educational and workplace settings. The project has developed and maintained a framework of shared technological solutions enabling the development and integration of inter-operable and extendable set of tools and services. KP-Lab System (see Figure 3) consists of two parts, namely platform and user environment (KPE) with integrated end-user tools.

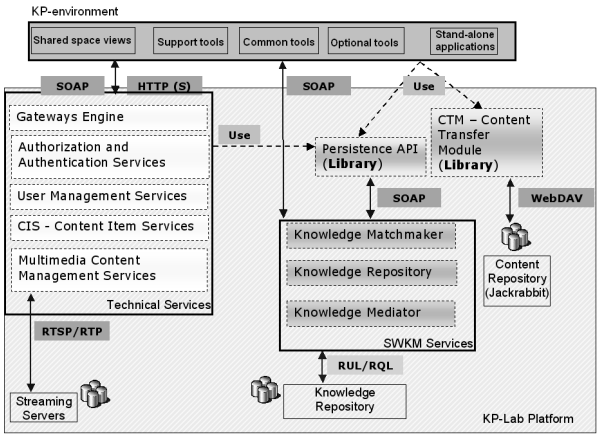


Figure 3: Architecture of the KP-Lab System [9] .

5.1 KP-Lab platform

The KP-Lab platform (see Figure 4) is based on a flexible service-oriented architecture that aims at facilitating the integration and interoperability of different end users tools as well as interactions with middleware functionalities.

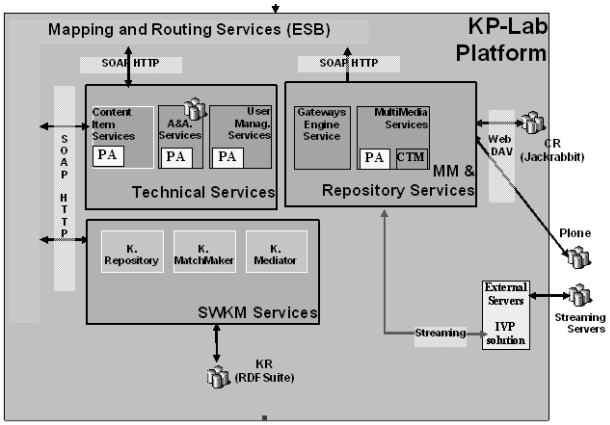


Figure 4: Architecture of the KP-Lab platform [9] .

KP-Lab platform architecture is composed from several groups of services or libraries:

- Semantic Knowledge Middleware Services (*SWKM Services* in Figure 4) provide storage and management of the metadata created by the KP-Lab tools. This metadata are stored into RDFSuite that is used as the knowledge repository [1] . RDFSuite is being developed at FORTH -ICS in Greece and comprises the Validating RDF Parser (VRP), the Schema-Specific Data Base (RSSDB) and interpreters for the RDF Query Language (RQL) and RDF Update Language (RUL).
- Content Management Services (*Repository Services* in Figure 4) are dedicated to creation and management of regular content (documents in various formats) used in shared objects (content described by metadata), either towards KP-Lab's own content repositories or external content repositories. KP-Lab content repositories are implemented through Jackrabbit for the compatibility with the JSR-170 standard.
- The Multimedia Services (*MM Services* in Figure 4) provide functionalities for manipulation and management of dynamic content such as streamed material for audio and video function to be supplied to the KP-Lab tools (e.g. Semantic Multimedia Annotation Tool).
- *Technical services* cover those middleware support services, dedicated to the authorization and identity management, the user management, routing etc.

The services and applications interfaces build on common semantic data models that describe the semantics of shared objects, with the Trialogical Learning Ontology (TLO, see next chapter) as the core ontology.

5.2 KP-Lab ontology architecture

The common knowledge model for the integration and semantic enabled manipulation with various shared objects of activity in the KP-Lab System is provided by ontologies that are implemented in the following three layers (see Figure 5)

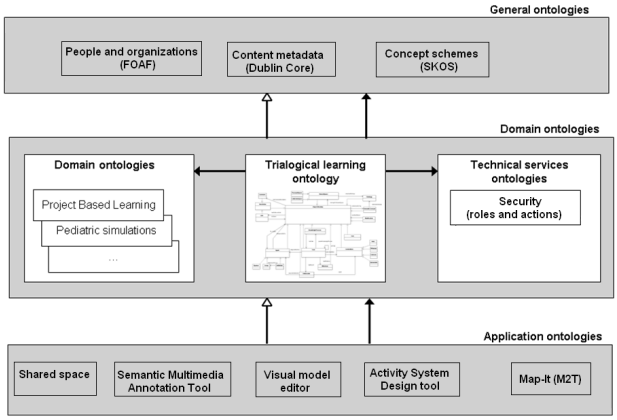


Figure 5: Architecture of KP-Lab ontologies [11] .

General ontologies describe in common sense (i.e. reusable across application domains) objects, actors and activities, e.g.: persons and organizations for general description of people and their affiliations, etc.

Domain ontologies provide semantic framework for relevant courses or interesting application domains of KP-Lab System and trialogical learning approach, e.g. Project Based Learning ontology (PBLO). This ontology was created based on evaluation of performed courses by project pedagogical partners in Netherlands and Finland. It defines the basic concepts and relations needed for construction of whole courses and their implementation and further evaluation.

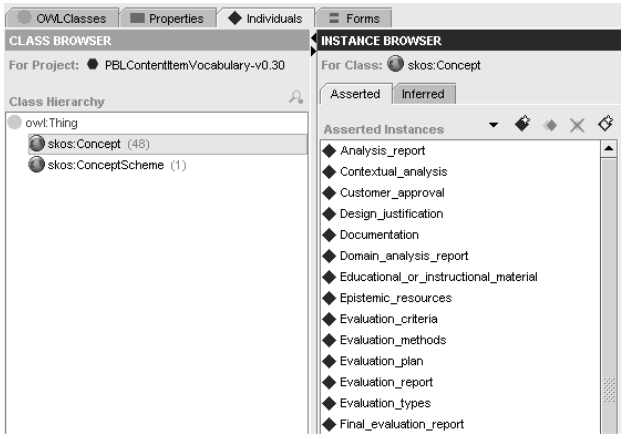


Figure 6: Example of PBLO in Protégé.

Technical ontologies that model technical concepts related to the services provided by the KP-Lab platform, integration of tools, e.g. security ontology to define different user roles in the whole system with personal information as login, password, etc.

Application ontologies represent set of ontologies relevant to end-user applications in KP-Environment as Shared space, Semantic Multimedia Annotation Tool, Visual Model Editor, etc., in order to model relevant self concepts of these applications.

Trialogical learning ontology (TLO) defines core concepts and principles of the TL and is shared by all applications and tools in KP-Lab System in order to provide the common semantics needed for the data interoperability. It defines basic classes for construction of user virtual environment with relevant parts as shared objects, users and integrated tools. Properties defined in this ontology provide common semantic framework for description of all types of objects in the system, as title, description, date, status, etc. The whole TLO (see Figure 7) can be divided into several subparts based on relevancies to particular end-user tools or functionalities that are represented with.

Class	Relations
Agent	subclass of http://www.kp-lab.org/ontologies/foaf#Agent ObjectOfActivity
Annotation	subclass of ObjectOfActivity
CollectiveSpace	subclass of SharedSpace
Comment	subclass of Annotation
ContentItem	subclass of ObjectOfActivity
ContentItemOrganizer	subclass of ObjectOfActivity
ExternalContentItem	subclass of ContentItem
GoogleDocsContentItem	subclass of RTCollaborativeContentItem
Group	subclass of http://www.kp-lab.org/ontologies/foaf#Group Agent
Individual	subclass of http://www.kp-lab.org/ontologies/foaf#Person Agent
Milestone	subclass of ObjectOfActivity
Modification	subclass of ObjectOfActivity
ObjectOfActivity	
Ontology	subclass of ObjectOfActivity
Organization	subclass of http://www.kp-lab.org/ontologies/foaf#Organization Agent
PersonalSpace	subclass of SharedSpace
RTCollaborativeContentItem	subclass of ContentItem
Relationship	subclass of Annotation
ScormImsContentItem	subclass of ContentItem
SemanticConcept	subclass of ObjectOfActivity
SharedSpace	subclass of ObjectOfActivity
System	subclass of Agent
Task	subclass of ObjectOfActivity
Template	subclass of ObjectOfActivity
TodoItem	subclass of ObjectOfActivity
Tool	subclass of ObjectOfActivity
UploadableContentItem	subclass of ContentItem
VML Concept	subclass of ContentItem
VML Relation	subclass of Relationship
VisualModel	subclass of ContentItem
VisualModellingLanguage	subclass of ContentItem

Figure 7: Trialogical learning ontology, extract through SWKM browser[4]

5.3 KP-Environment

The Knowledge Practices Environment (KP-Environment - KPE) represents virtual user environment to support creation, management and evaluation of different user activities as whole courses, simple collaborative processes or individual actions with selected goals. Actual implemented version (see Figure 8) is a result of co-design process based on initial analyses, case studies, generic scenarios and requirements identification, to enable simple and intuitive environment with relevant features for experienced users with collaborative systems and novices too. KPE is built on conceptual ideas underlying the proposed learning approach (TL), such as collaboration, shared objects, boundary crossing, etc.

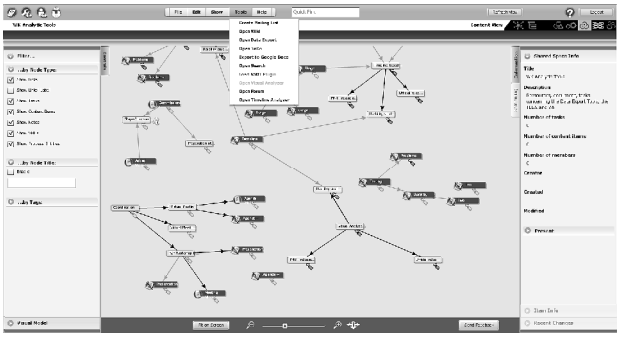


Figure 8: KP-Environment

The user is provided with four basic perspectives (views) when working with KPE: Network view (visualization of shared spaces network), Content view (organisation and management of different shared objects), Process view (management of shared objects relevant to the processes, as task, subtask, milestones, etc.), and Community view (management and

[4] http://kplab.fei.tuke.sk:8080/swkm-explorer/

visualization of relations between relevant users). Moreover, Tailored view is a customized individual virtual space which can be created by any user. Various tools and functionalities are highly integrated in these basic views to enable versatile and flexible creation, connection, organization and reflection of the material in shared spaces (personal or individual):

- working with the shared objects, e.g. creating, editing, storing, sharing, commenting, discussing, semantically annotating using existing vocabularies and/or using users' own tags;
- managing different types of processes; e.g. creating, changing and executing process models supporting dynamic structure and adaptation;
- displaying a status information about each logged user, status information about actual working environment, present and performed users' actions, etc;
- context-based chat as one type of synchronous communication;
- Note editor for creation of small notes by users;
- proactive to-do list in combination with Google Calendar to support personal management of time and work;
- possibility to collaborative creation of shared documents through tight integration with Google Docs;
- export and import features for SCORM and IML packages;
- Semantic search [2] with possibility to save search queries and their results, advanced faceted window based on the metadata and content of shared objects and support user-defined, flexible visualization as well as semantics based classification and clustering of search results.

KPE has strong support for semantic features by providing integrated use of semantic functionalities, like tagging, filtering, grouping and searching. Users have the opportunity to flexibly create their own conceptual models and conceptualizations.

5.3.1 Examples of other KP-Lab tools

Semantic Multimedia Annotation Tool (SMAT) is a rich internet application that offers the possibility of annotating any multimedia content item. It is adaptive to the user's domain based on the use of ontologies to define the type of annotated item (i.e. video, audio), the granularity of the item fragments (i.e. page, image, timestamp) and the domain ontology (i.e. actors' activity in a scene, actors' decisions, actors' behaviour, objects in a scene).

Activity System Design Tool (ASDT) was designed and implemented in order to help the participants in making their practices and activity dynamics visible. The idea of ASDT is to enable users to analyse the history, present and future of their work activity in a way that helps address issues critical for deliberate transformation of prevailing practices based on an intervention method called DWR (Developmental Work Research [5]).

Map-It is a tool that provides features for preparation, execution and evaluation of meetings (face-to-face and remote). Synchronous and asynchronous interactions are possible through the collaborative elaboration of "discussion maps" that capture user interactions through performed meetings. Map-It allows the use of meeting templates, in-advance individual preparations, sharing of materials, planning and follow-up of actions, automatic generation of meeting minutes in various formats.

The *Visual Model Editor (VME)* allows users to create, share, use, and update visual models as well as the underlying visual modelling languages. It provides easy and customizable environment for collaborative semantic modelling in diverse domains of interest.

The *CASS-Query tool* is a Java-application for collecting process and context-sensitive data that supports mobile users for the purposes of further analyses in KPE.

Wiki based on MediaWiki[5] engine with semantic extension that is integrated in KP-Environment.

6 Comparison

KP-Lab System represents modular and extensible collaborative system based on trialogical learning principles with many integrated end user features so it is possible to compare it with other existing frequently used systems such as Sakai[6], BSCW[7] and Moodle[8]. We extend proposed list for comparison with Google Apps[9] as representative of innovative information and communication technologies based on Web 2.0 paradigm. Results of our comparison (May 2008) are presented in Table 1.

Organization of user virtual space (feature)	
1 (KP-Lab)	Network view, basic list, different filters, favorites lists
2 (SAKAI)	individual lists of user's pages based on their roles
3 (BSCW)	spaces are organized as folders
4 (Moodle)	courses divided into categories
5 (Google Apps)	the sites are presented as list formats
Relations between spaces	
1	links between different spaces with descriptions of these relations

[5] http://www.mediawiki.org/wiki/MediaWiki
[6] http://sakaiproject.org/
[7] http://www.bscw.de/copyright.html
[8] http://moodle.org/
[9] http://www.google.com/a/help/intl/en/var_1c.html

2	not supported
3	not supported
4	courses organized only hierarchically
5	not supported
Awareness	
1	online status, history of performed actions or changes, notifications
2	supported
3	supported
4	supported
5	supported
Search	
1	free term and full text search, faceted browsing and filtering, saving of queries or search results
2	basic free term search
3	basic and semantic search
4	basic free term search
5	basic free term search, possibility so save performed queries or their results
Process modeling	
1	Gantt chart, dynamic process structure, process templates
2	supported
3	supported
4	supported
5	unverified
Personal management	
1	To-do list and Google Calendar integrated directly into whole system
2	simple to-do list
3	supported
4	supported
5	supported
Analyses	
1	statistical summaries, social network analysis, time-line based analyses (evolution of process creation)
2	not supported
3	not supported
4	basic user statistics
5	basic statistics
Monitoring and logging	
1	persistent storage of performed events in virtual environment – historical data for analyses
2	supported
3	supported
4	Only log files in insufficient format
5	not supported
Commenting	
1	visualization of comments in threaded manner, awareness of ongoing discussion
2	not supported
3	supported
4	supported
5	not supported
Tagging	
1	free tags or tags from vocabularies, consistency check, recommendation service
2	not supported
3	supported
4	not supported
5	simple tags in Google Docs
Semantic wiki	
1	extension of wiki engine with semantic features (tags, search, ontologies)
2	not supported
3	not supported
4	supported
5	not supported
Real-time Collaborative Document Editing	
1	Integration of Google Docs
2	not supported
3	not supported
4	not supported
5	supported
User Community view	
1	Grouping, social relations with visualization within networks, multiple roles
2	not supported
3	not supported
4	not supported
5	simple group formation
Data export	
1	Export information from knowledge and awareness repository for further analyses as Excel sheets
2	not supported
3	unverified
4	supported
5	not supported
Import/Export of IMS packages	
1	Reuse of courses made in some other compatible systems
2	not supported
3	unverified
4	supported
5	not supported
Meeting support	
1	Creation of discussion map, preparation of meeting materials, generation of meeting outputs
2	not supported
3	export relevant data for conferencing application
4	not supported
5	not supported
Visual models and languages	
1	Creation and operations with semantics by means of self-defined visual modeling languages and models
2	not supported
3	not supported
4	not supported

5	not supported
Semantic multimedia annotation	
1	Possibility to annotate uploaded video clips based on created ontologies
2	Project Pad
3	not supported
4	not supported
5	not supported
Specialized tools supporting reflection activities	
1	Analysis of performed activities represented by records in text, sound, video or other formats based on DWR theory.
2	not supported
3	not supported
4	not supported
5	not supported

Table 1: The comparison table.

Based on the comparison summarized in Table 1 we can observe several main and important advantages of the KP-Lab System, mainly:

- Multifunctionality
- Extensive use of semantic features
- Extendable and interoperable solution with existing approaches

6.1 Multifunctionality

No one open-source system provides so many features as KP-Lab System. To sum up the functionality comparison (also with some other systems):

- Dokeos[10] supports videoconferencing, RSS, calendar with agenda, session information, and integration with Google search but lacks much other functionality provided by KP-Lab System, e.g. the semantic aspects and visual organization possibilities.
- FLE3[11] consists of three main parts that provide features for knowledge building, sharing objects and collaborative construction of digital objects but does not provide possibility to create and manage processes in an easy way (e.g. Gantt chart), it does not provide possibilities for virtual meetings in context of the selected objects, does not provide tagging possibilities, etc.
- Moodle is strongly oriented on the area of integrated modules while the semantic aspects and possibilities to analyze user's practices /activities are weakly developed.
- Claroline[12] is a simple tool and has only basic functionalities offering complex and complicated information about courses (users list, accesses to course, utilization of the tools, documents usage, forum contributions), but hardly addresses the management and flexible modification of processes.
- SAKAI places emphasis on the development phase and like Moodle can be extended by relevant modules with new features. SAKAI provides several similar functionalities as KP-Environment, such as: shared workspace, job scheduler with calendar, portfolio, but lack focus on semantics of the used objects, e.g. semantic search, commenting or tagging, semantic wiki and different visualization possibilities.
- BSCW seems to be a strong commercial system that provides advanced functionalities as tagging, communities, templates, search on different indexing services, while editing tags or the indexes is poorly developed or does not exist, also collaborative idea generation tools are not available.

6.2 Usage of semantic features

Another strong point of KP-Lab System is management and utilization of metadata and semantic features. For example, comparable systems provide features to share and save objects (documents in different formats, multimedia, etc.) in different types of repositories as transaction databases or content management systems. KP-Lab System provides possibility to work with different objects that are stored within content repository (content of relevant object: text file, video, audio, package, etc.) and semantically described by self defined or predefined properties stored in knowledge repository. It means that objects' description consists of two parts:

1. Metadata (semantic information) that is saved into knowledge repository, implemented within RDFSuite based on predefined ontologies. Basic object properties are defined within Dublin Core standard, e.g. Title, Description, etc.
2. The content is saved in content repository based on Java technologies and access services are provided by designed and implemented specialized gateways (G2CR) with supporting of versioning and possibility to select relevant repository based on content type or actual availability.

This is an important point since the metadata is a crucial mean to achieve real semantic integration, i.e. the different tools are not only „co-located" in a single environment but also share the same semantics, e.g. all tools know what an object is, hence this information can be reused across tools. It enables to work with one shared object in different contexts and different processes. Moreover, one shared object can have different semantic interpretations in different processes.

Semantic features are implemented within end user tools as the tag-vocabulary editor (user has possibility to use existing tags from different vocabularies (e.g. PBLO) or create the new ones based on the actual

[10] http://www.dokeos.com/
[11] http://fle3.uiah.fi/
[12] http://www.claroline.net/

needs), semantic search (integrated part of faceted GUI is filtering the results through semantic properties to divide search space based on user requirements and improve performance measures of search algorithms), visual model editor (possibility to create own visual model – ontology, e.g. conceptualization of examined domain or used approach based on identification of main concepts of this domain with relevant relations – the goal is not to replicate existing application as Protégé[13] or OntoEdit[14]), but provide a simple mean for definition of any visual concept model. Semantic wiki (tagging the whole wiki pages or selected internal sections of the pages) and analytical tools as data export (export required data from knowledge repository, e.g. overview summary about created and uploaded shared objects during examined activity with their description) or time-line based analyses (visualization of the whole evolution process of selected share objects based on relevant user actions stored in awareness repository).

6.3 Extendable and interoperable with existing solutions

KP-Lab System is easily extendable and highly interoperable with existing solutions by the following means.

- SWKM and G2CR provide not only simple access to the saved data, e.g. G2CR provides access to different types of content repositories, thus providing potential integration with existing databases from e.g. Moodle.
- The end-users that currently use some different (IMS compatible) systems for their work or learning, will not have problems in migrating to KP-Lab System, and can execute the migration without necessity to redesign their shared objects, because of the possibility to import IMS packages.
- Tight integration with Google Docs and Calendar to provide features for real-time collaborative creation and modification of shared documents and for personal time and work management.
- Proposed logging mechanism can be used for other collaborative system to store performed events in virtual user environment and collect historical data for analyses in KP-Lab analytic tools.
- Data export provides also the possibility to extract data from knowledge and awareness repository for further analyses in third party analytic tools like e.g. SPSS.

7 Conclusion

KP-Lab System represents an interesting collaborative system which is based on the trialogical learning principles. The architecture and interesting aspects of this system (mainly its semantic character) have been presented in this paper. Our comparison with other collaborative tools available nowadays showed that collaborative systems usually provide several common functionalities for this type of tools and a few specific ones dependent on application domain or theoretical background behind. KP-Lab System provides all of these common functionalities and many specific ones in one environment as multifunctional application. Furthermore, KP-environment outperforms other collaborative systems with respect to their semantics nature, interoperability with other tools and analytical possibilities provided for knowledge creation processes. Actual version is available on http://2d.mobile.evtek.fi/shared-space/.

[13] http://protege.stanford.edu/
[14] http://www.ontoknowledge.org/tools/ontoedit.shtml

Acknowledgments

The work presented in this paper was supported by: European Commission DG INFSO under the IST program, contract No. 27490; the Slovak Research and Development Agency under the contract No. VMSP-P-0048-09; the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under grants No. 1/0042/10 and 1/0131/09; project implementation Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120020) and project implementation Development of Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120030) supported by the Research & Development Operational Programme funded by the ERDF.

The KP-Lab Integrated Project is sponsored under the 6th EU Framework Programme for Research and Development. The authors are solely responsible for the content of this article. It does not represent the opinion of the KP-Lab consortium or the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

References

[1] Alexaki, S., et al.: The ICS-FORTH RDFSuite: managing voluminous RDF description bases. In: 2nd International Workshop on the Semantic Web, pp. 1-13, Hong Kong, 2001.

[2] Babič, F., Paralič, J.,Furdík, K., Bednár, P., Wagner J. (2009). Use of semantic principles in a collaborative system in order to support effective information retrieval. Computational Collective Intelligence, Semantic Web, Social Networks and Multiagent Systems (ICCCI 2009). LNAI 5796/2009, Springer Berlin / Heidelberg, pp. 365-376, ISBN 978-3-642-04440-3.

[3] Bednar, A.K., et al.: Theory into practice: How do we link? In T.M. Duffy and D.H. Jonassen (Eds.), Constructivism and the technology of instruction: A conversation. Hillsdale, NJ: Lawrence Erlbaum Associates, 1995.

[4] DeVries et al.: Developing constructivist early childhood curriculum: practical principles and activities. Teachers College Press: New York, 2002.
[5] Engeström, Y., et al.: The Change laboratory as a tool for transforming work. Lifelong Learning in Europe, 1(2), 10-17, 1996.
[6] Engeström, Y.: Innovative Learning in Work Teams: Analyzing Cycles of Knowledge Creation in Practice. Cambridge, MA: Cambridge University Press, 1999.
[7] Hakkarainen, K., Paavola, S.: From monological and dialogical to trialogical approaches to learning. A paper at an international workshop "Guided Construction of Knowledge in Classrooms", February 5-8, 2007, Hebrew University, Jerusalem, 2007.
[8] Hakkarainen, K., et al.: Communities of networked expertise: professional and educational perspectives. Amsterdam: Elsevier, 2007.
[9] Ionescu, M., et. al.: KP-Lab Platform Architecture Dossier - Release 4. KP-Lab public deliverable D4.2.4. June, 2008.
[10] Jonassen, D.H.: Constructivist Learning Environment on the Web: engaging students in meaningful learning. Paper presented at the Educational Technology Conference and Exhibition, SUNTEC City, Singapore, 1999.
[11] Markkanen, H., Holi, M.: Ontology-driven knowledge management and interoperability in trialogical learning applications. Article in the KP-Lab book, 2008.
[12] Nonaka, I., Takeuchi, H.: The Knowledge Creating Company. Oxford University Press, New York, 1995.
[13] Paavola, S., Lipponen, L., Hakkarainen, K.: Models of Innovative Knowledge Communities and Three Metaphors of Learning. Review of Educational Research 74(4), 557-576, 2004.
[14] Papert, S., Harel, I. (eds).: Constructionism: research reports and essays 1985 - 1990 by the Epistemology and Learning Research Group, the Media Lab, Massachusetts Institute of Technology, Ablex Pub. Corp, Norwood, NJ, 2001.
[15] Paralič, J., Babič, F., Wagner, J., Simonenko, E., Spyratos, N., Sukibuchi, T. (2009). Analyses of knowledge creation processes based on different types of monitored data, In: Proc. of the ISMIS 2009. Foundations of Intelligent Systems. LNCS 5722/2009, Springer Berlin / Heidelberg, pp. 321-330, ISBN 978-3-642-04124-2.
[16] Scardamalia, M., Bereiter, C.: Knowledge building. In Encyclopedia of Education, 2nd ed., pp. 1370-1373. New York: Macmillan Reference, USA, 2003.
[17] Sfard, A.: On two metaphors for learning and the dangers of choosing just one. Educational Researcher, 27(2), pp. 4–13, 1998.
[18] Vygotsky, L. S.: Thought and language Cambridge [Mass] : M.I.T. Press, 1962.

An LPGM Method: Platform Independent Modeling and Development of Graphical User Interface

Jan Kryštof
Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic
E-mail: jan.krystof@mendelu.cz

Keywords: GUI, HCI modeling, MB-UID, UML, adaptive modeling tool

Received: September 12, 2009

This paper introduces a new method in the area of platform independent modeling and the development of graphical user interfaces. The method bridges the gap between traditional MB-UIDEs and the modern web methodologies by enabling the modeling and development of both traditional and web user interfaces. The method is based on a proposed Presentation model and a Task Action Model which drive the development process. The modeling notation in both models is done with use of UML, and the development process is supported by a UML-compliant adaptive modeling tool. Descriptions of both the model and the method of application are included. An evaluation done using a JavaEE and a Swing widget toolkit is also mentioned.

Povzetek: Predstavljena je nova metoda za izdelavo platform za razvoj grafičnih vmesnikov.

1 Introduction

In the course of developing software for a user interface (UI), a developer frequently recognises that a similar UI has been created previously, perhaps in a different context and with different visual aspects, but nearly identical in concept. This research investigates the possibilities of reusing UIs.

UIs can be made more readily reusable by elaborating the specifications for them in a form that is independent of platform. Such platform-independent tools and methodologies have been developed, but, unfortunately, the results have never achieved widespread adoption and successful application in industry [12, 38, 17].

UI development is a difficult and time-consuming procedure [37] that involves a collection of different activities. UI development deals with the interaction between humans and computer and specifies how software will function across this i.e. the tasks of the user and the system. The physical user interface is subsequently assembled with respect to the tasks identified for the user and the system. The UI should have appropriate ergonomics and appearance and it must communicate with the underlying application layer. The process of UI development is not properly described in traditional software development methodologies (Waterfall, Spiral). For example, the Unified Process provides advice for UI development only by recommending the build-up of a prototype [19]. The UI prototype in the Unified Process serves only as a tool for better understanding the particular use-case and its functionality. The methodological framework Rational Unified Process [28] goes a step further, extends the number of artifacts used for UI modeling, and introduces the UI storyboards [43]. However, neither the UP nor the RUP adds methodological guidelines for UI development.

The lack of development guidelines for the UI development was partially covered in traditional methodologies by the concept of Model-Based User Interface Development (MB-UID), which unified development of applications using a traditional UI. The development is based on intensive modeling of the different aspects of all part of the application, including aspects of UIs. With the emergence of web-based development, a number of new web methodologies have been proposed [23, 7, 55, 50]. These define the entire development process for web applications, including issues of UI development. However, these modern web methodologies and MB-UID represent disparate branches, which can be used for either traditional or web user-interface development.

In this paper a new approach for UI modelling will be presented along with the architecture of development environment for this approach. The approach aims to provide a method for producing platform-independent modelling and development of graphical user interfaces using pure UML. The following section will deal with the current state of MB-UID and some of the drawbacks which led to this new approach. The section three describes the approach and later in section four, steps for applying the approach will be presented. Some examples will be included in the section five.

2 Current state

2.1 Model based user interface development

Systematizing of UI development is a challenging and important prerequisite for the quality of development, and the concept of MB-UID supports it. MB-UID is characterized by a set of declarative models and a way of interpretating them [48]. The MB-UID approach is supported by software environments which are called MB-UIDEs (Model-Based User Interface Development Environments).

The UI development process is focused purely on constructing models which describe different areas of application. Models are built incrementally, describing "what" without explaining "how", thus hiding the method of implementation. However, approaches within MB-UIDs are not yet mature, and proposals for the range and nature of the models supported differs significantly [17]. The development process varies with environment, since each particular MB-UIDE defines its own set of models. Thanks to this diversity, the models mostly commonly encountered are [53, 48]: the domain model, the application model, the task model, the user model and the presentation model.

Many different notations are used for MB-UID, because no uniform standard for all MB-UIDEs exists. In general, notation has been developed specially for each the MB-UIDE [53] which makes it difficult for developers to get oriented in other forms of notations and causes compatibility problems: a model created using a particular tool can not be processed using a different tool. Silva divided MB-UIDEs into two generations [53]. Second generation environments are oriented more towards industrial standards and are more receptive to new user-interface features. Despite enhancements, interoperability remains rather low, and the MB-UIDEs are not in widespread use among developers. There are also addressed two more drawbacks of the MB-UID approach [38]. Firstly, the generated UIs are often not as good as those that could be created using conventional programming techniques. Secondly, heuristics are often involved and the connection between the specification and the final result can be quite difficult to understand and control. This makes the results unpredictable. We assume that efforts to generate "final" and "ready to run" products cannot succeed and make extension of any particular MB-UIDE to support a new platform very hard. The reusability of models is associated with the whole application of MB-UIDEs, so it is not possible to make use of a single model. With regard to the specification of a presentation model in MB-UIDEs, we can address one significant drawback which is connected with the separation of concerns [39, 15, 26]. Concerns are often merged together with visual appearance, layout or content specified within a single presentation model, which makes such a form useful only for the original requirements. Furthermore, the layout of UIOs is sometimes specified in terms of the absolute positioning [34]; this is the possibility, that constraints of screen and resolution will prevent the proper display of the user interface.

2.2 Modern web methodologies

A similar approach to the generation of applications in development is driven by modern web methodologies such as OOHDM [50], WebML [7], UWE [23, 24] or OOWS [14]. These also provide methodological guidelines for specifying sets of declarative models which drive subsequent development. Therefore they fit the concept of MB-UID. In order to make a clear distinction between web and traditional development, we will use the terms "web MB-UID", and "traditional MB-UID" respectively. Some of these methodologies (UWE, WebML, OOHDM) also provide software environments (ArgoUWE [22], WebRatio [58] and OOHDM-Web [49]) in order to support the modeling approach by means of a set of frequently used functions in the context of model construction or code generation. Thus we can classify them as web MB-UIDEs.

Like traditional MB-UIDEs, web MB-UIDEs suffer from low interoperability since they also use their own modeling notation, which makes the interchange of model data between different environments impossible. On the other hand, some web methodologies have already employed UML for modeling notation. UML is the de-facto industrial standard object-oriented modeling language [13]. The notation is familiar to many developers, and there are a lot of resources such as documentation and software support in the form of modeling and CASE tools. The UML profiles mechanism [40] is also used sometimes to provide new modeling facilities. Since UML profiles are based on UML, it is not difficult for any software designer with a background in UML to understand a model based on a UML profile [24]. Regarding the summary of the modeling notations employed in web methodologies published in [11], the UML notation is fully employed in OOWS while some other methodologies (e.g. OOHDM, UWE, WebMl) combine UML with other forms of notation (e.g. OO, OMT, ERDs, DFD), and the rest do not use UML notation at all. The set of declarative models is nearly the same in web methodologies compared to the model sets in MB-UIDs, except for the navigation model which is tightly connected with the hypertext paradigm. The MDA (Model Driven Architecture) [16] concept is used in some environments (ArgoUWE, WebRatio) in order to interpret models and support code generation.

2.3 Characteristics of traditional and MB-UID in summary

From the overview that has been carried out the preceding sections, we want to point out several positive and negative characteristics of current MB-UID.

Model-based development. The UI development in MB-UIDEs and web methodologies is based on the construction of different models. Employing modeling approaches in UI development has proved that modeling represents a good way to support user-interface development.

Development environment. MB-UID is supported by graphical software environments which enable rapid model construction and utilization of models. Using graphical environments also reduces the cognitive burden on human programmers [47].

Diversity of modeling notations. Many varied notations employed in different approaches do not support interoperability, since data obtained from the model can be read and modified only in the original environment.

Heavy-weight solutions. Solutions produced in MB-UID tend to provide "ready to run" software. Attempts to cover the presentation, application and data layer which result in the great complexity of traditional MB-UIDEs and make any extension of such environments difficult.

Low separation of concerns. The separation of concerns in presentation models is very low both at the model and source code levels, which impacts the reusability. Thus models can hardly be used on different platforms.

Disjunction of development of web and traditional user interfaces. Modern web methodologies and traditional MB-UID have built two disjoint branches that focus on either web or traditional UI. There is no middle ground where both scopes can be developed.

3 Modeling approach

The summary mentioned in the previous section has contributed to forming our approach to the scope of MB-UID: our approach is built with the respect to the pros and cons of the current state of MB-UID. In this section, our modeling approach is introduced and described along with an argument for chosen methods which are involved.

The goal of our approach is to provide facilities enabling the modeling and development of platform-independent user interfaces. In order to provide appropriate facilities for modeling user interfaces and user-computer interaction, we need to choose a modeling notation which enables us to model these domains. Basically, we can design our own notation and create a domain-specific language (DSL) [1, 54]. This approach is recommended for cases where the modeling domain is large, the modeling area is well charted, and there is only a small probability of further evolution. The main disadvantage of the DSL approach is that in setting up a new notation we might be considered to adding another stone to the tower of Babel of modeling languages. Furthermore, we want to have a language that is easily extensible, since the field of user interfaces is constantly evolving. Therefore we chose an approach employing UML profiles – a light-weight extension of UML [47] which preserves compatibility with UML. The UML profiles mechanism is currently the most utilized approach thanks to the large number of CASE tools [1] and other support (UML-compliant tools, extensive documentation, and a wide base of users) available for it. Many projects [20, 23, 9] have employed UML profiles to model diverse domains and benefited from the high level of interoperability, thanks to the XMI (XML Metadata Interchange) data format [41]. For these reasons, we chose the UML and its profiles to define and provide modeling facilities, and we created a UML profile called "LPGM" (Lightened Profile for GUI Modeling), which gave its name to our approach. This profile is presented in fig. 3 and fig. 4.

Our profile provides facilities for building two kinds of models: the task-action model focused on aspects of the interaction and the presentation model focused on the structure of the user interface.

3.1 Task-action model

Since all user interfaces are associated with an underlying application layer that performs a particular business [36], we wanted to link the user-interface to it and express the business in terms of the user-interface interaction.

The functionality conveying associated business is usually specified in the form of a use-case model and its documents. The user-interface interaction is also recorded here in the text form of scenarios. Scenarios often include references to particular interaction objects of the relevant user interface [43], as is common in use-case in the Rational Unified Process [28].

Many styles of writing scenarios (common narrative style, partitioned narratives, pseudo-code, interaction diagrams, etc.) are summarized in [10]. The most comprehensive style was formed by Wirfs [59]; in it the scenario is captured in a two-column dialog between the system and the user. This style of interaction capture is very natural, since the user-interface interaction is a kind of dialog consisting of the user's action and the system's reaction.

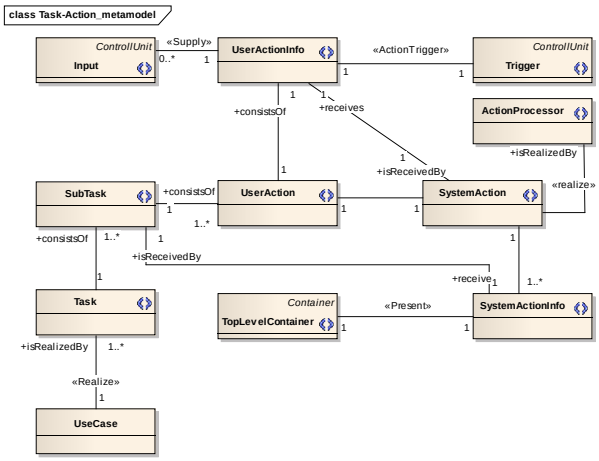


Figure 1: Task-Action Meta-model.

In order to specify the user-interface interaction, we synthesized a two-column dialog scenario capture with a

behavioral diagram and formed the Task-Action Model (TAM) [31]. The goal of TAM is to convey the user-interface interaction by capturing 1) the goal of the user, 2) the user's responsibility to the interface and 3) the system's responsibility to the interface.

We chose the UML Activity diagram as modeling facility because of its simple notation compared with common interaction diagrams. A UML activity diagram is normally used to represent the dynamic view of a system as control and data flow from activity to activity [6]. In our case we have used it to depict the flow of actions performed by the user and the system. The Activity diagram has also been successfully employed in user-interface storyboarding in RUP [43] and it has been proposed as a suitable diagram for CTT (Concur Task Trees) [3, 42], a widely used notation for task modeling. However, we want to model tasks in the context of the user and the system to show how these tasks should be performed in terms of elementary actions as well as to show which data are transferred during the steps of interaction. The TAM, specified in the meta-model shown in fig. 1, is based on our proposed meta-model for the presentation layer [32]. All of the meta-model elements are described in the table 1. The TAM is commonly constructed after analysis of a particular use-case where at least one task having a goal has been identified. We consider the terms “task” and “goal” as they are defined in Hierarchical Task Analysis (HTA): a task is an activity that a user does to reach a goal, while the goal is a desired state of the system [21]. Each task can be broken down into several subtasks. Each subtask has associated with it a particular container which represents a set of user interface objects (UIO). The subtask is a composition of one or more atomic actions which are associated with particular interaction objects (IO). An action associated with a subtask is called a User action and denotes a user responsibility with the respect to one or more IOs. We model two kinds of interaction: 1) Supply interaction, which represents providing input data for a current task and 2) Trigger interaction, which causes termination of a current subtask and transition to a connected System action. The System action is

Meta-model object	Description	UML	Location
UseCase	Use-case associated with one or more task.	Original	Use-case model
Task	The task is bound to a particular use-case through a dependency «Realize». The task has one or more SubTasks.	Activity, stereotype «Task».	Task Action Model
SubTask	The Subtask represents one or more steps which belongs together within a task. It has an input (SystemActionInfo) which holds a reference of UI displayed within this subtask. It has one or more UserActions.	Activity, stereotype «SubTask».	Task Action Model
UserAction	The UserAction represents a user-interface interaction which has one UserActionInfo.	Action, stereotype «UserAction».	Task Action Model
UserActionInfo	The UserActionInfo is the specification of a particular UserAction and conveys more information about the interaction. The UserActionInfo can have one or more UIOs of the ControlUnit (from LPGM structural model) type associated through «ActionTrigger» or «Supply» dependency. The UserActionInfo. This object is received by a SystemAction which processes the UserAction.	ActionPin, stereotype «UserActionInfo».	Task Action Model
Input	An input object (TextField, CheckBox, etc. from LPGM structural profile) used during a user-interface interaction for obtaining data from a user. It is connected with the UserActionInfo through the «Supply» dependency.	Class, stereotype «Input» and its descendants.	Presentation model
Trigger	The object (from the LPGM structural model) used during a user-interface interaction for triggering a SystemAction. It is connected with a UserActionInfo through the «ActionTrigger» dependency.	Class, stereotype «Trigger» and its descendants.	Task Action Model
Supply	The dependency between a user and UserActionInfo and a particular Input object (e.g. TextField, CheckBox). It denotes the user's responsibility for providing data to the current SubTask.	Dependency, stereotype «Supply».	Task Action Model
ActionTrigger	The Dependency between a UserActionInfo and a particular Trigger object (e.g., Button, MenuItem). It denotes a user operation which terminates the current SubTask.	Dependency, stereotype «ActionTrigger».	Task Action Model
SystemAction	The SystemAction represents an abstraction of the system action responsible for processing the previous SubTasks. It is responsible for processing the previous SubTask and providing a UI as a response.	Action, stereotype «SystemAction».	Task Action Model
SystemActionInfo	The SystemActionInfo is a specification of a particular SystemAction. It holds a reference to a method ActionProcessor and TopLevelContainer that is generated and displayed in the subsequent task.	ActionPin, stereotype «SystemActionInfo».	Task Action Model
ActionProcessor	The ActionProcessor is a method which represents a physical implementation of the SystemAction. It is responsible for processing the data provided by the previous SubTask.	Operation, stereotype «ActionProcessor».	Application model
TopLevelContainer	TopLevelContainer is a UIO which is generated as a response and passed to the subsequent SubTask.	Class, stereotype «TopLevelContainer».	Presentation model
Presents	The dependency between SystemActionInfo and generated UIO. The dependency between the user and UserActionInfo and a particular Input object (e.g., TextField, CheckBox). It denotes the user's responsibility for providing data to the current SubTask.	Dependency, stereotype «Presents».	Task Action Model

Table 1: Description of the Task-Action Meta-model.

responsible for processing the finished subtask through a delegated method denoted as the Action processor. This method generates a user interface which is represented by a container. One interaction step is finished at this point and a new one begins by sending the generated container to the following subtask. The whole process is repeated until the last subtask is finished and the goal associated with the current task has been achieved. Example of the TAM is presented in the fig. 2.

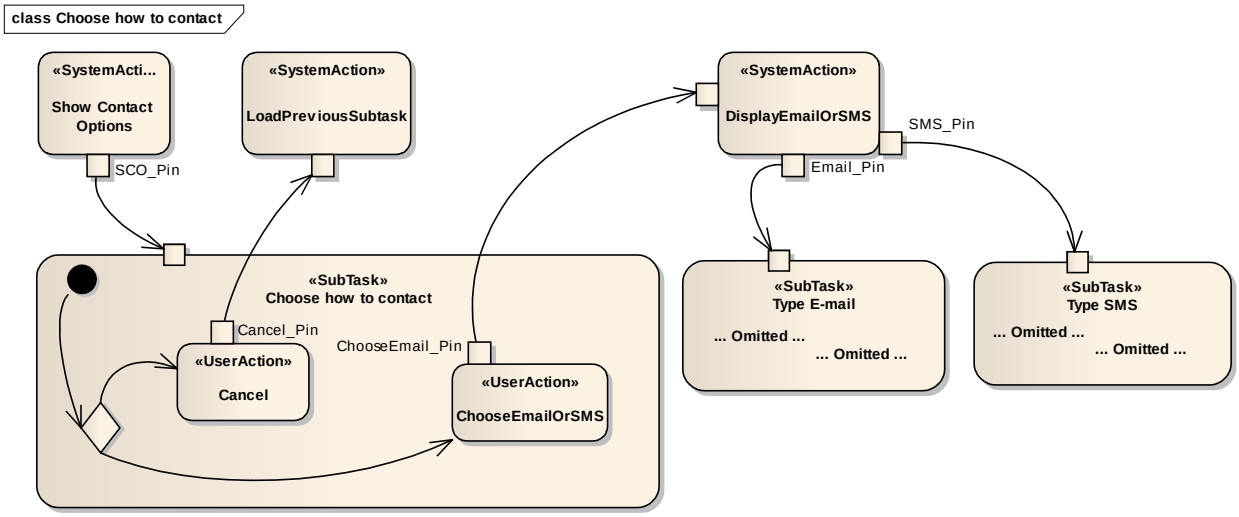


Figure 2: Example of Task-Action Model, showing the subtask "Choose how to contact".

3.2 Presentation model

The goal of the Presentation model (PM) in our approach is to describe the **structure** of the user interface. By structure we mean a set of widget types (i.e., buttons, icons, forms, etc.) and the specification of the containment hierarchy [2].

Rules for the construction of the PM are based on the meta-model for the presentation layer presented in [32] and have been included in the LPGM profile. The profile for the PM contains a hierarchy of stereotypes representing concrete and abstract interface objects. The hierarchy has a root in GElement which is a stereotype extending the UML meta-class and serves as a common parent for all inherited stereotypes that form a hierarchical tree. Tree leafs represent concrete interface objects (CIO) while tree nodes may represent abstract interface objects (AIO).

In the tree, three basic classes of UIOs are defined: containers, presentation units, and control units. These are the AIO and the parents of, e.g., Form, Label and ComboBox, respectively. Both AIOs and CIOs can be found in other approaches, such as [8, 17, 51] but we offer a richer set of UIOs: The UMLi approach (focused on both web and traditional UI) [51] provides three UIOs; the TEALLACH approach (focused on traditional UI) provides five UIOs [52], and the UWE approach (focused on web UI) provides ten UIOs [25]. If we need to eliminate a particular CIO, we can replace it by using the most appropriate AIO, which can be the nearest parent of the node in the hierarchical tree. Since all UIOs are defined in a UML profile, extension of to the set of UIOs is possible and easy.

We consider that the PM is a platform independent and reusable component that cannot include any information other than a structure. Specifying any of the geometrical aspects of the UIOs (location, width, height) or their appearance (color, font, alignment) a premature commitment to a specific look and feel. Therefore we decided to consider our PM as an artifact capturing the structure of the UI and nothing more. For us a structure means a set of UIOs and the logical relations among them. We have proposed in [29] a hypothesis, with which we can model the structure of a UI using hierarchical and neighborhood relations. In order to formalize a the description of the UI structure, we have formulated definitions that contribute to the definition of the UI structure.

Definition 1.
Let g denote a sorted couple (id, t) where the id is an indentifier and the t is a data type.

Definition 2.
Let G be a set containing all g elements.

Definition 3.
Let C be a set of containers: $C = \{(id,t) | (id,t) \in G \wedge t = container\}$

Definition 4.
Let VN denote a Vertical Neighborhood relation $VN \subseteq G^2$. This VN relation must satisfy Constraint 1.

Constraint 1.

$$\forall g_i, g_j : VN(g_i, g_j) \Rightarrow ((\neg\exists g_k : k \neq j \wedge VN(g_i, g_k)) \wedge (\neg\exists g_l : l \neq i \wedge VN(g_l, g_j))$$

If a g_i is in VN relation with a g_j then g_i cannot be in a VN relation with any other element.

Definition 5.

Let VN denote a Horizontal Neighborhood relation $HN \subseteq G^2$. This VN relation must satisfy Constraint 2.

Constraint 2.

$$\forall g_i, g_j : HN(g_i, g_j) \Rightarrow ((\neg\exists g_k : k \neq j \wedge HN(g_i, g_k)) \wedge (\neg\exists g_l : l \neq i \wedge HN(g_l, g_j))$$

If a g_i is in an HN relation with a g_j then this g_i cannot be in HN relation with any other element.

Relations of Horizontal Neighborhood and Vertical Neighborhood have additional constraint 3.

Constraint 3.

$$VN \cap HN = \varnothing$$

Definition 6.

Let H denote a relation $H = VN \cup HN$.

Definition 7.

Let ParentOf denote a relation $ParentOf \subseteq C \times G$. The ParentOf relation must satisfy the Constraints 4 and 5.

Constraint 4.

$$(c, g) \in ParentOf \Rightarrow \neg\exists d, d \neq c \wedge ParentOf(d, g)$$

No element g can have more then one parent c.

Constraint 5.

$$ParentOf(c, g) \Rightarrow \neg VN(c, g) \wedge \neg HN(c, g)$$

Neither c nor g can take part in any VH or HN relation.

We have expressed all defined relations in terms of UML and created stereotypes «ParentOf», «Neighbour», «H_Neighbour» and «V_Neighbour» as extensions of the UML Associaton meta-class. In our PM, we use the «ParentOf» stereotype to denote the first owned element of a container. The «Neighbour» stereotype denotes an ordered pair of elements. «H_Neighbour» and «V_Neighbour» are specializations of the «Neighbour» and correspond to the Horizontal and Vertical Neighborhood relations. We bind two UIOs by «H_Neighbour» or «V_Neighbour» when we want our model to represent these elements laid out horizontally or vertically, respectively, within a common container. With the use of these relations, we can model the containment hierarchy and the relations contributing to the UI layout.

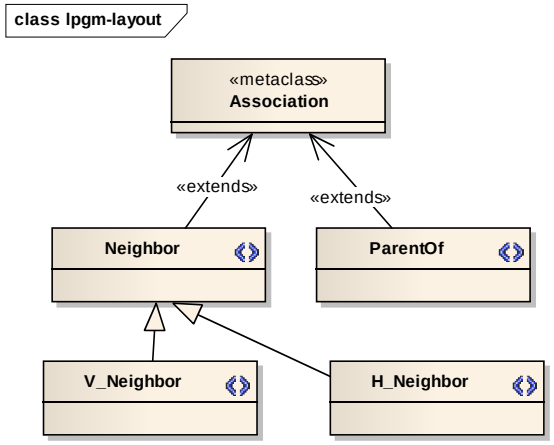


Figure 3: Stereotyped associations of the UML profile for the Presentation model.

3.3 Development environment

As we have mentioned, MB-UID is usually supported by a software tool that provides a graphical environment. Since we are focused on UML, we have explored several UML compliant modeling tools [30], [33] focusing on the level of their extension in order to support our modeling approach. These tools, which we have called **adaptive**, can be extended and adapted to specific purposes different from the original. In [33] we set forth the following requirements which must be satisfied in adaptive UML compliant modeling tools.

- The tool must support UML profiles and stereotypes as specified in [46]. We want to emphasize support for features enabling the application of alternative icons to stereotypes because appropriate icons can better convey the modeling domain thus and make the modeling more intuitive and clear.
- The tool must provide an API (Application Programming Interface) which enables access to the UML repository and manipulation of the data of the model.
- The API must provide a mechanism for establishing a channel of communication to show which action is being performed on the model data, e.g., element creation or model deletion. Such interactive observations enable keeping track of a development process and better controlling it.
- If there exists an adaptive modeling tool, we recommend implementing an environment that provides such functions as model transformations, generation of unique internal identifiers for new model elements and checking the names of elements according to naming conventions. The environment should also behave as a container for storing approach-compliant resources (transformation rules, type mapping, UML profiles, etc.).

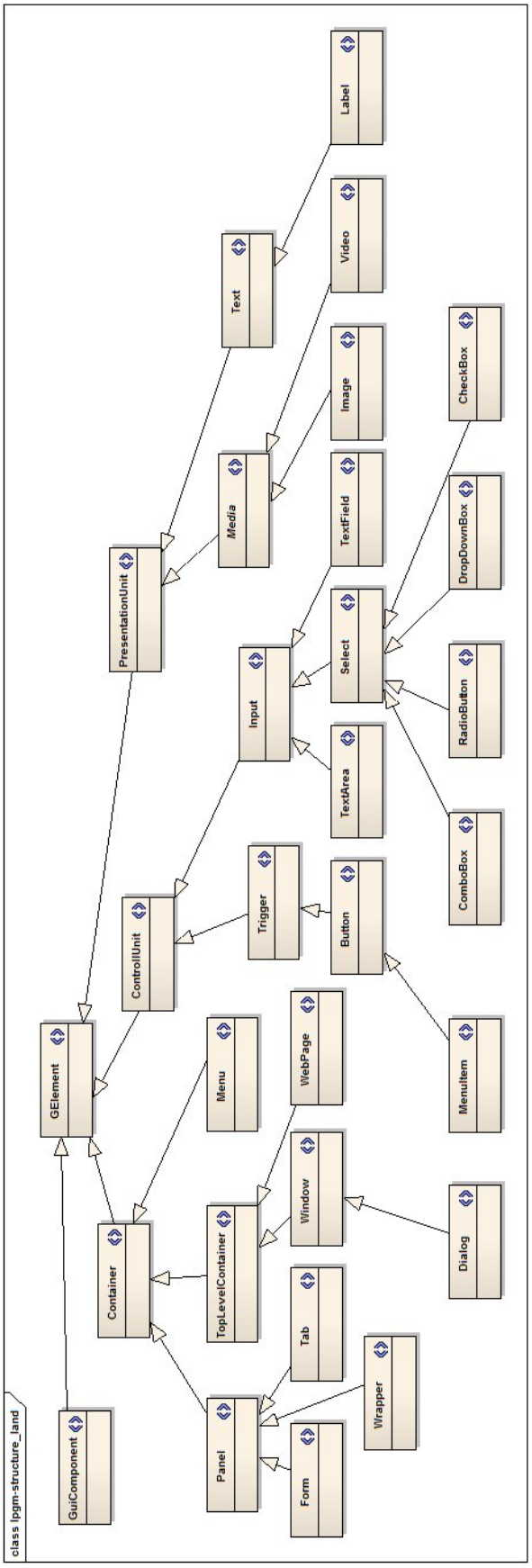


Figure 4: Stereotyped classes of the UML profile for the Presentation model.

4 Method application

The application of our approach consists of constructing models and generating source code for a particular platform. Models are built using the LPGM profile and are created without including any platform-specific information. The models created are considered as PIM models (Platform Independent Model) in the MDA. We will illustrate in this section how these models are constructed and utilized.

4.1 Model construction

Our approach to the process of modeling a user interface is based on the existing use-case model. After the functionality of the entire application has been specified, we identify all the goals of the given use-case for which the TAM is being constructed. Each goal is associated with a particular task, which is modeled as a «Task» activity. This activity is connected to the original use-case through «Realize» dependency. The task is broken down into «SubTasks» activities which contain «UserAction» actions. The first IOs are identified at this point: the PM is constructed in parallel, to enable it to immediately bind action pins «UserActionInfo» to UIs through «Supply» or «ActionTrigger» dependencies. A «SystemAction» action is constructed for each «SubTask», and a «SystemActionInfo» action pin is added to the action through «Present» dependency, the «SystemActionInfo» is associated with a container which is generated and displayed for the user in the next subtask. The container is produced by the «ActionProcessor» method, which represents physical implementation of the method used to generate the UI for the following task. The method is connected to the «SystemAction» through «Realize» dependency.

When the TAM is finished, the PM is completed by adding additional UIOs (images, icons, labels) to the IOs which were created during the construction of the TAM. All UIOs must be connected to their neighbors and parents through «ParentOf», «H_Neighbour», and «V_Neighbour» dependencies until the specification of the structure is finished. The PM is then done.

The system action "Show Contact Options" provides the “Controlls_Cont” container of the «Form» stereotype which is bound to the output action pin “SCO_Pin” through the «Present» association. This container is subsequently displayed in the “Choose how to contact” subtask, wherein the user can use two radio buttons “Email_RBtn” and “SMS_RBtn” in order to choose the way of communication or cancel the subtask. These two options are performed within “ChooseEmailOrSMS” and “Cancel” user actions, respectively. Thus the “ChooseEmailOrSMS” user action has the “CES_Pin” action pin, which is associated by means of two «Supply» associations with the “Email_RBtn” and “SMS_RBtn” radio buttons. Furthermore, the

"CES_Pin" has an «ActionTrigger» association connected with the "Next_Btn" button. The "Cancel" user actions its "Cancel_pin" action pin connected with the "Dismiss_Btn" button through «ActionTrigger» association. After using the "Next_Btn", the user action "DisplayEmailOrSMS" is executed and one the "Type E-mail" or "Type SMS" subtasks is displayed according to the user's choice.

4.2 Model transformations

The TAM and PM do not contain any information related to implementation since the target platform is not yet known. After the platform is specified we should transform current models in order to get new and richer models having a form optimal for the straightforward and effective generation of code. We have proposed several transformations which will be depicted in the next sections. The transformations are model-to-model and model-to-text and are applicable for both web and traditional UI.

4.2.1 Layout normalization of presentation model

After finishing the PM, we have the ideal model from the point of view of a developer. The developer need not focus on any implementation issues and the PM is thus created with respect its function and not technical issues. However, such a form of model is hard to interpret in the context of code generation. The problem is represented by so-called "**corner elements**". Corner elements are UIOs which take part in both V_Neighborhood and H_Neighborhood relations. Corner elements "Controlls_Cont", "Next_Btn", "Email_RBtn" and "SMS_RBtn" are seen in the fig. 5. Common containers of user interfaces can hold and arrange objects in only one direction, i.e., either horizontally or vertically: QT/C++ (HorizontalLayout and VerticalLayout), Swing/Java (BoxLayout.X_AXIS and BoxLayout.Y_AXIS), HTML/Web (div and span). Therefore we need to eliminate all corner elements in order to shift the model a bit towards an implementation form. We have proposed and implemented an algorithm [29] which breaks every corner element into one element and one new wrapping container. The element is later removed from the H_Neighborhood or V_Neighborhood relation and the relation is inherited by the new container (see fig. 6). This process is called "**layout normalization**" and after it is done all H_Neighborhood and V_Neighborhood relations must satisfy the constraints 6 and 7.

Constraint 6.

$$\forall g_i \in G : \exists g_j \in G : VN(g_i, g_j) \Rightarrow$$
$$\neg \exists g_k \in G : HN(g_i, g_k)$$

No element g_i can be in both VN and HN relations.

Constraint 7.

$$\forall g_i \in G : \exists g_j \in G : HN(g_i, g_j) \Rightarrow$$
$$\neg \exists g_k \in G : VN(g_i, g_k)$$

No element g_i can be in both VN and HN relations.

4.2.2 Model enrichment

The PM contains no additional information beside that information regarding the structure of the UI, so we need to add information through a transformation step which we call "model enrichment". Model enrichment is performed partly on the PIM level and causes the transition of the PIM to a PSM (Platform Specific Model), when the process of model enrichment begins to add platform-specific information. This enrichment is based on mapping "key - new information", where the key is a unique identifier of the model element being processed. New information can be added to the model in the form of tagged values, as has been demonstrated in [27].

Transformation at PIM level. Since no information related to appearance or content has been specified, we propose to add this through use of the tagged values `appearance`, `text` and `resource`. The tagged value `appearance` contains a link to the definition of appearance. It is not necessary to generate the `appearance` tagged value for all UIOs. It is enough to generate this for the top-level containers and distribute appearance information to their descendants at the run time (as we show in the next section). The tagged valued `text` contains either a text which will be displayed at the run time or a key referring to a resource that has a corresponding text value. The later approach enables flexible management of the content (e.g. localization) in future. This tagged value can be presented only by a UIO with the stereotypes Text and Label. The tagged value `resource` is generated for all types of Presentation units (i.e., Media, Image, etc.) and defines the location of an associated resource (e.g., multimedia file, image file).

Transformation at PSM level. Once the target platform has been chosen, we recommend enriching it immediately with additional, implementation-related information. This can typically be the data type for each UIO. For this purpose, we propose to set a `tptype` (target platform type) tagged value that refers to the fully qualified name of a data type for a UIO of a particular stereotype. Other tagged values can specify a namespace (C#) or package (Java) for top-level containers which are considered to be transformed into a class. We also propose to perform another enrichment which adds a new tagged value containing a text value that corresponds to an identifier suited to the target platform to prevent problems during source code compilation. The alternative name may be derived from the original one and can conform to a particular naming convention.

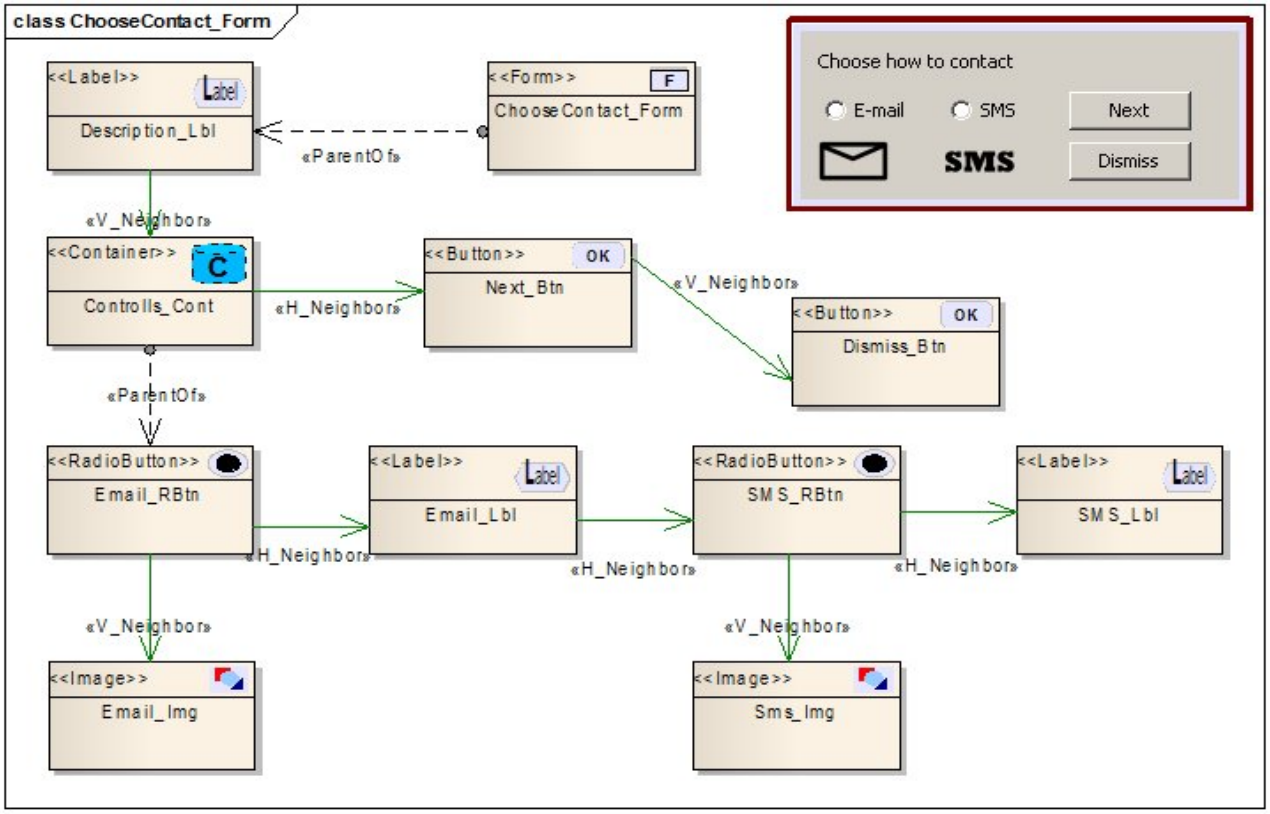


Figure 5: Depiction of the Container "ChooseContact_Form", an example of a PM. (A mock-up of it is shown in the upper right-hand corner.)

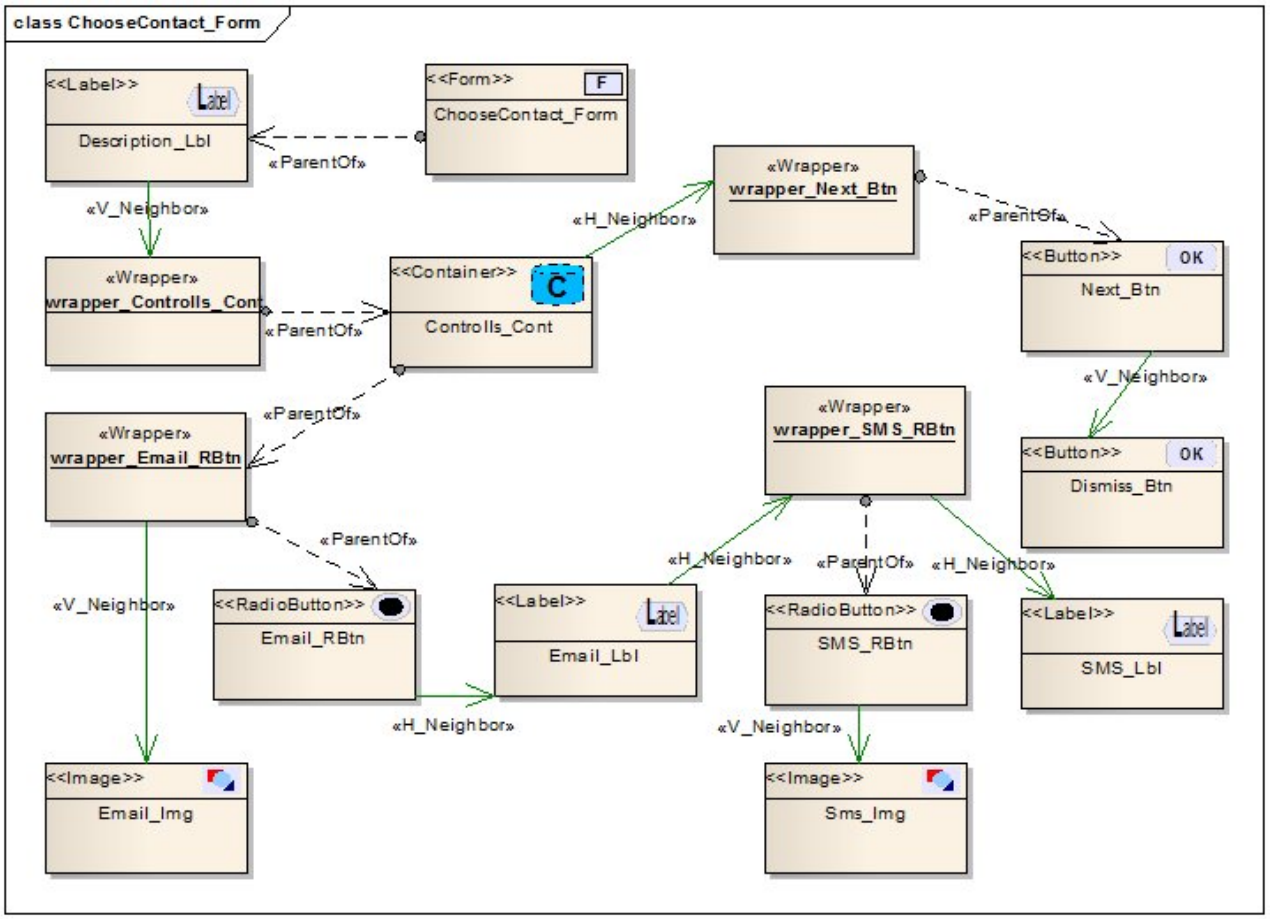


Figure 6: Normalized layout of the "ChooseContact_Form" container, which contains no corner-elements.

4.2.3 Source code generation

When the last model-model transformation has been completed, we can proceed to generate source code, remembering that one of our goals is to separate concerns as much as possible.

Based on the TAM, we propose to generate an XML file named Task-Action Descriptor, see fig. 10. This file provides information which can drive the application flow without the need to hard-code such information into the application logic or the presentation logic. By keeping track of both the last action triggered by the user (e.g., button press) and the TAD we can determine the corresponding action of the system and launch it dynamically.

We use the PM to generate source code for the user interface. Since we have not included any spatial information regarding the layout within the PM, we need to reconstruct this from the definition of the UI structure and place UIOs at the right positions within the top-level container. In traditional MB-UID, the UI layout is sometimes generated from a task model [35, 52, 5] or based on some grouping relations [24, 57, 52]. Some of these approaches use particular strategies that give a solution for automated placement. Techniques like the double-column strategy or right-bottom strategy [5], [56] provide good results under certain circumstances and only partially, so they cannot be employed widely without corrections [44]. The problem with these strategies comes from the endeavor to solve this issue complex and in their own hook. Therefore we decided to avoid generating source-code, including the command for the automated placement of UIOs. On the other hand, we propose to generate a UI layout with the use of containers which control the placement of UIOs on their own. This strategy can be applied in a variety of widget toolkits which support the concept of Layout Managers [18]. The great advantage of using layout managers is that they can adjust layout dynamically, e.g., during changes in the size of the screen.

5 Evaluation

We have already done some experimental evaluation of our method in the areas of traditional and web development. The first tests focused on generating a traditional UI for the Swing platform, where we employed our PM. The second test focused on web applications, particularly on the J2EE platform, where we employed both models with emphasis on the TAM. We used the reflection mechanism [45] intensively during this evaluation because our method is heavily dependent on it.

5.1 Development environment

In order to provide software support for our approach, we have implemented the software environment LPGM4EA, see fig. 7. We focused on contemporary UML-compliant modeling tools used in the commercial sphere because we wanted to explore the possibility that our approach could be adopted without forcing anybody to abandon a tool currently in use. After comparing the modeling tools Visual Paradigm, Enterprise Architect and Rational Rose, we have implemented our environment in the Enterprise Architect modeling tool. The EA is widely used in the community of software developers and provides some important features which put it into the class of adaptive modeling tools.

The LPGM4EA environment is written in the .NET, has its own presentation layer, and runs in its own window outside the EA graphical environment. The LPGM4EA is connected to the running instance of the EA through a bidirectional communication channel which is based on listener which propagates user actions performed in the LPGM4EA to and from the EA. The application layer of the LPGM4EA is also able to access the UML repository of the EA without the running instance of the EA. This offline access mode is also supported in Rational Rose, but it is not supported in Visual Paradigm. This lack of offline UML data processing precludes processing data non-interactively which can cause the development process to break down: there can be a lot of models in the UML repository, and thus it should be possible to process data automatically without user intervention, as a batch.

The LPGM4EA provides functionality which enables the running model-model and model-text transformations. These transformations are template-based [4] for both model-model and model-text transformations. It also watches the UML repository and manages newly added or deleted model elements. Thus we are able to decorate new elements with an `lpgmid` tagged value which holds our internal identifier, generated uniquely for PM and TAM elements and to provide some assistance in correctly naming model elements.

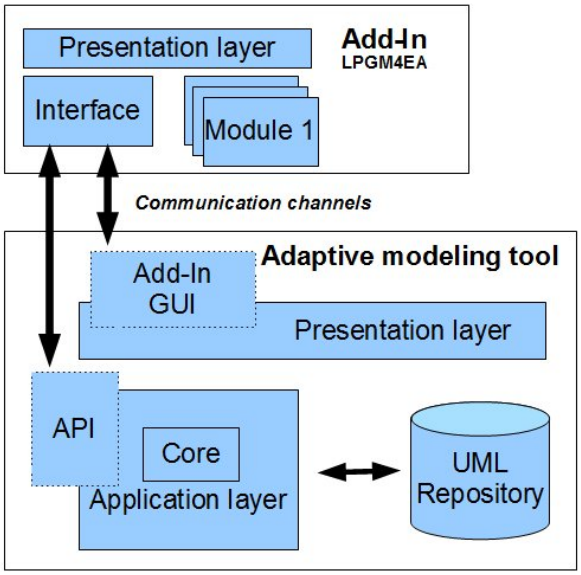


Figure 7: Scheme of the LPGM4EA modeling and development environment.

5.2 UI generation of Swing components and HTML tags

The application logic responsible for UI code generation is realized through the `IWireframe2CodeTransformation` interface, see fig. 8. A note on naming of our classes and interfaces: we think of the PM as a sort of (UML) wireframe and therefore we use the term "wireframe" within our implementation. The implementation of the interface is quite simple, and it is necessary to implement it on every platform we want to support. This interface is used by the interface `ITransformation2CodeController` which is able to read the normalized model. It drives the generation and calls methods placed in implementations of the `IWireframe2CodeTransformation`. At this moment, we have two implementations: `Wireframe2HtmlTransformationImpl` and `Wireframe2SwingTransformationImpl`. Each implementation has a template for the StringTemplate[1] library. Templates are designed with respect to the target platforms and provide parameterized generation of code.

5.2.1 Swing

User Interfaces for the Swing library were generated against top-level containers. A new java class was generated for each top-level container in the form of a file. This file (class) contains common sections such as a package name, an imports section, a class skeleton, attribute declarations and a constructor.

The import section is generated from the tagged value `tptype` of all the nodes and leafs in containment hierarchy of the top-level container. The class skeleton is generated for the top-level container and extends the platform type of the container (e.g. `class Foo extends JFrame`).

The section attribute declarations contains declaration expressions for all nodes and leafs (UIOs) in the containment hierarchy of the top-level container.

The constructor contains three blocks of commands. The first block contains commands for the initialization of all UIOs. The second block contains the commands responsible for building the containment hierarchy and setting the proper layout. We use BoxLayout with constants `BoxLayout.X_AXIS` and `BoxLayout.Y_AXIS` to lay out UIOs horizontally and vertically, respectively. The third block contains commands responsible for setting texts and resources for textual and multimedia UIOs. Figure 9 shows the part of the generated code that is relevant to figure 6.

The appearance of the components generated is set separately. It is achieved by creating a simple text file for each top-level container where in the appearance properties are specified for the particular UIO or group of objects. This definition is parsed and processed at run time. Then we proceed to set the appearance values of objects of the hierarchy of top-level containers. We get a reference to the top-level container object and traverse

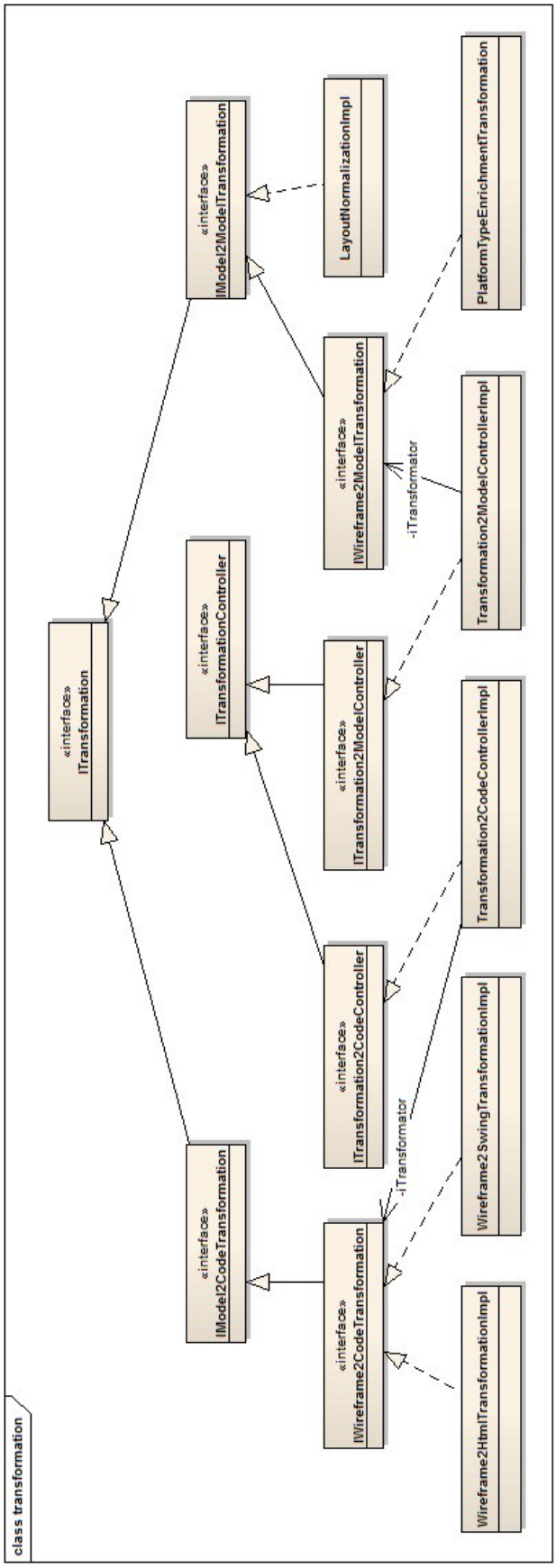


Figure 8: Depiction of the infrastructure of class and interface in the transformation process

[1] http://www.stringtemplate.org/

```
class ChooseContact_Form extends JPanel {
  ...
  private JPanel iControlls_Cont;
  private Container iWrapper_Email_RBtn;
  ..
  public ChooseContact_Form() {
    ...
    iControlls_Cont.setLayout(new BoxLayout(
    iControlls_Cont, BoxLayout.Y_AXIS ));
    iControlls_Cont.add(iWrapper_Email_RBtn);
    iControlls_Cont.add(iEmail_Img);
    iWrapper_Email_RBtn.setLayout(new BoxLayout(
      iWrapper_Email_RBtn, BoxLayout.X_AXIS ));
    iWrapper_Email_RBtn.add(iEmail_RBtn);
    iWrapper_Email_RBtn.add(iEmail_Lbl);
    ...
  }
}
```

Figure 9: Example of the code generated to set up the containment hierarchy for the diagram shown in figure 6.

the entire tree from the root to the bottom at the leafs level. Along the way we set all appearance properties to objects which are selected by a selector in the appearance-definition file. Each processed object is inspected via the reflection mechanism by checking for the existence of a method conforming to the name of the property and having an appropriate set of formal parameters. For instance, if there is a property "`font-style: courier, italic, 12`", we seek a method void `setFontStyle(String, String, int)`. If such a method exists, we perform its execution and supply the specified values.

In the generated file, the appearance is separated from the structure definition which makes the source code more modular and readable and easier to maintain. Furthermore, we claim that this strategy keeps up a unified appearance, thanks to the selector mechanism: `JLabel font: Font.BOLD` will apply the bold font style to all labels in the user interface. This prevents us from forgetting to set it, as we might if we were using the common manual approach.

5.2.2 HTML

During the PM evaluation, we also tested the generation of tags for HTML. This generation is easier thanks to the fact that HTML is a declarative language. In order to generate the structure of a UI in HTML, we used the `DIV` and `SPAN` tags to lay out UIOs vertically and horizontally, respectively. We were able to generate common HTML forms or menus for a content management system, where in the TAM was also utilized. The appearance was set in common way by using CSS (Cascade Style Sheet documents).

5.3 Dynamic flow control in web JavaEE applications

The TAM was tested during the development of a content management system on a JavaEE platform by using servlet and JSP technologies. We have designed a format for the XML file to hold information from the TAM. This document is called the Task-action descriptor.

The format of the document is self explanatory and corresponds to the TAM. Information in the document helps us to control the flow of applications. The descriptor contains records corresponding to the actions of a user and system which are bound via the `lpgmid` identifier. The utilization of the descriptor is performed according to the following scenario.

The user performs an action using a particular IO with associated lpgmid. The web browser generates an HTTP request and the `lpgmid` value is sent to the server as a parameter. The HTTP request is processed by a servlet, which extracts the lpgmid value and seeks the corresponding record in the TAD using `userAction`. The `userAction` found contains an attribute actionTrigger referencing a `systemAction`. The `systemAction` has a method name and the fully qualified name of the parent class. The method (`ActionProcessor`) is executed by the servlet via a reflection mechanism, and the `HttpServletRequest` is passed on as an argument. The method performs common steps such as extracting parameters, and calling application logic, and it generates an HTTP response (`HttpServletResponse`). The response contains a UI within the JSP specified as a view attribute. The UI is generated from the PM.

This way of processing an HTTP request replaces common techniques, where in long blocks of "`if-elseIf-...`" are used within the servlet code. Furthermore, if we need to change a flow order or UI generated for a particular subtask, we can do it manually by editing the TAD, without needing to compile compilation the servlet source code.

```
<systemActions>
  ...
  <systemAction="DisplayEmailOrSMS"
      userAction="375048">
    <nextSubtask="214781" parameters="Email_RBtn"
        view="/jsp/EmailForm.jsp"/>
    <nextSubtask="811626" parameters="SMS_RBtn"
        view="/jsp/SMSForm.jsp"/>
    </systemAction>
    ...
</systemActions>

<tasks>
  ...
  <task name="Notify co-workers" lpgmid="277193">
    ...
    <subtask name="Choose how to contact"
        lpgmid="724793" >
        <userAction name="ChooseEmailOrSMS"
            actionTrigger="375048"/>
        <userAction name="Cancel"
            actionTrigger="800064"/>
    </subtask>
    <subtask name="Type E-mail" lpgmid="214781"
        ...
  </task>
    ...
</tasks>
```

Figure 10: Depiction of the Task-action descriptor for the "Choose how to contact" subtask.

6 Future work

Our research and development within the LPGM approach is not finished. Our future activities will focus on more extensive utilization of the TAM with emphasis on generating the source code of event handlers in the scope of traditional UI or automating the extraction of parameters from an HTTP request and validating them. The models will also be used to generate technical and user documentation for the interface.

We also want to use the TAM to generate tasks for collaborative user interface agents [12]. We believe this is a good way of providing assistance to help users and support the user experience.

7 Conclusion

In this paper, we have introduced our approach for modeling and development of user interfaces. The approach can be classified as MB-UID since it is based on a set of models which are interpreted and used for transformations. The approach is suited to the field of traditional and web user interfaces. The constructed models can be used with a particular platform. The modeling approach focuses on task modeling in the context of a user and a system. Furthermore, it provides facilities to model the UI structure with the use of a PM. The TAM and PM have been formalized with the use of meta-model and algebraic formulas. These models can be processed automatically in order to perform a series of transformations resulting in the source code of the user interface for a particular platform. Processing of the models is supported by a software environment which provides assistance during the construction of the model and the generation of source code. Therefore the environment can be classified as MB-UIDE.

Our approach differs from other MB-UID approaches in several ways. Firstly, we use UML modeling notation in both our models, so they can be read and processed in other environments. This is a step towards interoperability and compatibility with industry standards. Secondly, our models can be considered as reusable components and can be used for the development of both web and traditional interfaces. It is not our goal to generate “ready to run applications” but just reasonable and useful fragments for the development of user interfaces.

We have demonstrated the utilization of our models with the support of our developing environment, which we have integrated into an adaptive modeling tool EA. The way we generate source code and integrate it into other source codes supports the separation of concerns. Such code is modular and easily maintained.

Acknowledgment

The paper is written as a part of solution of a research plan PEF MZLU MSM 6215648904/03/03/02.

References

[1] Abouzahra, A.; Bézivin, J.; Fabro, M. D. D. & Jouault, F. (2005), A Practical Approach to Bridging Domain Specific Languages with UML profiles, in 'In Proceedings of the Best Practices for Model Driven Software Development at OOPSLA’05'.

[2] Batory, D.; Sarvela, J. N. & Rauschmayer, A. (2003), Scaling step-wise refinement, in 'ICSE '03: Proceedings of the 25th International Conference on Software Engineering', IEEE Computer Society, Washington, DC, USA, pp. 187--197.

[3] den Bergh, J. V. & Coninx, K. (2007), From Task to Dialog Model in the UML, in 'TAMODIA', pp. 98-111.

[4] Boas, G. E. (2004), 'Template Programming for Model-Driven Code Generation', http://www.softmetaware.com/oopsla2004/emdeboas.pdf.

[5] Bodart, F.; Hennebert, A.-M.; Leheureux, J.-M. & Vanderdonckt, J. (1994), Towards a dynamic strategy for computer-aided visual placement, in 'AVI '94: Proceedings of the workshop on Advanced visual interfaces', ACM, New York, NY, USA, pp. 78--87.

[6] Booch, G.; Rumbaugh, J. & Jacobson, I. (2005), Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series), Addison-Wesley Professional.

[7] Ceri, S.; Fraternali, P. & Bongio, A. (2000), 'Web Modeling Language (WebML): a modeling language for designing Web sites', Comput. Netw. 33(1-6), 137--157.

[8] Chesta, C.; Patern?, F. & Santoro, C. (2004), 'Methods and Tools for Designing and Developing Usable Multi-Platform Interactive Applications', PsychNology Journal 2(1), 123-139.

[9] Conallen, J. (2000), Building Web applications with UML, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[10] Constantine, L. L. & Lockwood, L. A. D. (2001), 'Structure and style in use cases for user interface design', 245--279.

[11] Domingues, A. L.; Bianchini, S. L.; Costa, M. L.; Ferrari, F. C. & Maldonado, J. C. (2007), eb application development methods: a comparison, in 'Workshop on Business Process Management'.

[12] Eisenstein, J. & Rich, C. (2002), Agents and GUIs from task models, in 'IUI '02: Proceedings of the 7th international conference on Intelligent user interfaces', ACM, New York, NY, USA, pp. 47--54.

[13] Engels, G.; Heckel, R. & Sauer, S. (2000), UML -- A Universal Modeling Language?, in M. Nielsen & D. Simpson, ed.,'Proc. Application and Theory of Petri Nets 2000, 21st International Conference, ICATPN 2000, Aarhus, Denmark, June 2000.', Springer, , pp. 24--38.

[14] Fons, J.; Pelechano, V.; Albert, M. & Pastor, O. (2003), Development of Web Applications from

Web Enhanced Conceptual Schemas, in 'ER', pp. 232-245.
[15] Fowler, M. (2001), 'Separating User Interface Code', IEEE Software 18, 96-97.
[16] Frankel, D. (2002), Model Driven Architecture: Applying MDA to Enterprise Computing, John Wiley & Sons, Inc., New York, NY, USA.
[17] Griffiths, T.; McKirdy, J.; Paton, N. W.; Kennedy, J. B.; Cooper, R.; Barclay, P. J.; Goble, C. A.; Gray, P. D.; Smyth, M.; West, A. & Dinn, A. (1998), An Open-Model-Based Interface Development System: The Teallach Approach, in 'DSV-IS (2)', pp. 34-50.
[18] Haraty, M.; Nobarany, S.; DiPaola, S. & Fisher, B. (2009), AdWiL: adaptive windows layout manager, in 'CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems', ACM, New York, NY, USA, pp. 4177--4182.
[19] Jacobson, I.; Booch, G. & Rumbaugh, J. (1999), The unified software development process, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
[20] Karwaczyński, P. & Maciejewski, L. (2004), UML Proile for Analysis and Design of Jakarta Struts Framework Based Web Applications, in 'Proceedings of NWUML', pp. 185--196.
[21] Kirwan, B. & Ainsworth, L. K. (1992), A Guide to Task Analysis, Taylor & Francis.
[22] Knapp, A.; Koch, N.; Moser, F. & Zhang, G. (2003), ArgoUWE: A CASE Tool for Web Applications, in 'First Int. Workshop on Engineering Methods to Support Information Systems Evolution (EMSISE 2003)'.
[23] Koch, N. (2001), 'Software Engineering for Adaptive Hypermedia Applications', PhD thesis, Ludwig-Maximilians-Universit?t M?nchen.
[24] Koch, N. & Kraus, A. (2002), The expressive Power of UML-based Web Engineering, in 'Proceedings Second Internatioanl Workshop on Web-Oriented Software Technology (IWWOST'02)'.
[25] Koch, N. & Mandel, L. (1999), 'Extending UML for Modeling Navigation and Presentation in Web Applications', online.
[26] Kong, X.; Liu, L. & Lowe, D. (2005), 'Separation of concerns: a web application architecture framework', Journal of Digital Information 6.
[27] Kozaczynski, W. & Thario, J. (2002), Transforming User Experience Models To Presentation Layer Implementations, in 'Second Workshop on Domain Specific Visual Languages'.
[28] Kruchten, P.; Ahlqvist, S. & Bylund, S. (2001), 'User interface design in the rational unified process', Object modeling and user interface design: designing interactive systems, 161--196.
[29] Kryštof, J. (2009): Formální popis rozložení prvků grafického uživatelského rozhraní, in 'The 11th nternational Conference MEKON'.
[30] Kryštof, J. & Chalupová, N. (2008), Prerekvizity pro novou koncepci modelování GUI v modelovacích nástrojích, in 'Objekty 2008', pp. 127-–136.
[31] Kryštof, J. & Motyčka, A. (2009), Extrakce scénářů do modelu úloh a akcí., in 'Objekty 2009'.
[32] Kryštof, J. & Motyčka, A. (2008), Metamodel for presentation layer, in 'Information Society', pp. 270--273.
[33] Kryštof, J. & Procházka, D. (2009), Rozšíření UML modelovacích nástrojů pro potřeby vývoje grafických uživatelských rozhraní, in 'Objekty 2009', pp. 264-–272.
[34] Lutteroth, C. (2008), Automated reverse engineering of hard-coded GUI layouts, in 'AUIC '08: Proceedings of the ninth conference on Australasian user interface', Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 65--73.
[35] Martínez-Ruiz, F. J.; Vanderdonckt, J. & Arteaga, J. M. (2009), Web User Interface Generation for Multiple Platforms, in 'Proceedings of the 7th International Workshop on Web-Oriented Software Technologies (IWWOST'2008) in conjunction with the 8th International Conference on Web Engineering (ICWE'2008)', pp. 63--68.
[36] Mišovič, M. & Turčínek, J. (2008), 'Teoretický přístup k tvorbě uživatelského rozhraní softwarových systémů', Acta Universitatis agriculturae et silviculturae Mendelianae Brunensis : Acta of Mendel University of agriculture and forestry Brno 6, 180--189.
[37] Myers, B. A. & Rosson, M. B. (1992), Survey on user interface programming, in 'CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems', ACM, New York, NY, USA, pp. 195--202.
[38] Myers, B.; Hudson, S. E. & Pausch, R. (2000), 'Past, present, and future of user interface software tools', ACM Trans. Comput.-Hum. Interact. 7(1), 3--28.
[39] Nierstrasz, O. & Achermann, F. (2000), Separation of Concerns through Unification of Concepts, in 'In ECOOP 2000 Workshop on Aspects & Dimensions of Concerns'.
[40] OMG (2010), 'UML Infrastructure specification', http://www.omg.org/spec/UML/2.1.2/.
[41] OMG (2007), 'XMI specification', http://www.omg.org/spec/XMI/2.1.1/.
[42] Paterno, F. (1999), Model-Based Design and Evaluation of Interactive Applications, Springer-Verlag, London, UK.
[43] Phillips, C. & Kemp, E. (2002), In support of user interface design in the rational unified process, in 'AUIC '02: Proceedings of the Third Australasian conference on User interfaces', Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 21--27.
[44] Puerta, A. & Eisenstein, J. (1999), Towards a general computational framework for model-based interface development systems, in 'IUI '99: Proceedings of the 4th international conference on

Intelligent user interfaces', ACM, New York, NY, USA, pp. 171--178.

[45] Rehak, M.; Tozicka, J.; Pěchouček, M.; Zelezny, F. & Rollo, M. (2005), An Abstract Architecture for Computational Re.ection in Multi-Agent Systems, in 'IAT '05: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology', IEEE Computer Society, Washington, DC, USA, pp. 128--131.

[46] Riesco, D.; Martellotto, P. & Montejano, G. (2003), 'Extension to UML using stereotypes', UML and the unified process, 273--293.

[47] Ryder, B. G.; Soffa, M. L. & Burnett, M. (2005), 'The impact of software engineering research on modern progamming languages', ACM Trans. Softw. Eng. Methodol. 14(4), 431--477.

[48] Schlungbaum, E. (1996), 'Model-based User Interface Software Tools - Current state of declarative models', Technical report, Graphics, Visualization and Usability Centre, Georgia Institute of Technology, GVU Tech Report.

[49] Schwabe, D.; de Almeida Pontes, R. & Moura, I. (1999), 'OOHDM-Web: an environment for implementation of hypermedia applications in the WWW', SIGWEB Newsl. 8(2), 18--34.

[50] Schwabe, D. & Gustavo, R. (1998), 'An object oriented approach to Web-based applications design', Theor. Pract. Object Syst. 4(4), 207--225.

[51] da Silva, P. & Paton, N. W. (2003), 'User Interface Modeling in UMLi', IEEE Softw. 20(4), 62--69.

[52] da Silva, P.; Paulo; Griffiths; Tony & Paton, N. W. (2000), Generating user interface code in a model based user interface development environment, in 'AVI '00: Proceedings of the working conference on Advanced visual interfaces', ACM, New York, NY, USA, pp. 155--160.

[53] da Silva, P. P. (2000), User Interface Declarative Models and Development Environments: A Survey, *in* 'DSV-IS', pp. 207-226.

[54] Tolvanen, J.-P. & Kelly, S. (2005), Defining Domain-Specific Modeling Languages to Automate Product Derivation: Collected Experiences, in 'Proceedings of the 9th International Conference on Software Product Lines, SPLC 2005', Springer, , pp. 198-209.

[55] Troyer, O. M. F. D. & Leune, C. J. (1998), 'WSDM: a user centered design method for Web sites', Comput. Netw. ISDN Syst. 30(1-7), 85--94.

[56] Vanderdonckt, J.; Ouedraogo, M. & Ygueitengar, B. (1994), A comparison of placement strategies for effective visual design, in 'HCI '94: Proceedings of the conference on People and computers IX', Cambridge University Press, New York, NY, USA, pp. 125--143.

[57] Viana, W. & Andrade, R. M. C. (2008), 'XMobile: A MB-UID environment for semi-automatic generation of adaptive applications for mobile devices', *J. Syst. Softw.* **81**(3), 382--394.

[58] WebRatioGroup (2010), 'WebRatio', online, http://www.webratio.com.

[59] Wirfs-Brock, R. (1993), 'Designing Scenarios: Making the Case for a Use Case Framework', *The Smalltalk Report* **3**(3).

A Fast Convex Hull Algorithm for Binary Image

Xianquan Zhang and Zhenjun Tang*
Department of Computer Science, Guangxi Normal University, Guilin 541004, P.R. China
E-mail: {zxq6622, tangzj230}@163.com

Jinhui Yu
State Key Lab. of CAD&CG, Zhejiang University, Hangzhou 310027, P.R. China

Mingming Guo
Department of Computer Science, Guangxi Normal University, Guilin 541004, P.R. China

Keywords: convex hull, extreme point, point set, monotone segment, computational geometry

Received: May 28, 2009

Convex hull is widely used in computer graphic, image processing, CAD/CAM and pattern recognition. In this work, we derive some new convex hull properties and then propose a fast algorithm based on these new properties to extract convex hull of the object in binary image. It is achieved by computing the extreme points, dividing the binary image into several regions, scanning the regions existing vertices dynamically, calculating the monotone segments, and merging these calculated segments. Theoretical analyses show that the proposed algorithm has low complexities of time and space.

Povzetek: Predstavljen je nov algoritem za obdelavo binarnih slik.

1 Introduction

Convex hull is a central problem in various applications of computational geometry, such as Vornoi diagrams constructing, triangulation computing, etc. It is widely applied to computer graphic [1], image processing [2-3], CAD/CAM and pattern recognition [4-6]. Convex hull of a planar point set S is defined as the intersection of all the half-planes containing S. The shape of convex hull is a convex polygon whose vertices belong to S. For an edge pq, all other points lie on one side of the line running through p and q.

Many research efforts have been devoted to develop algorithms for 2-D convex hull computation. In 1970, Chand et al. [7] initially proposed a convex hull algorithm with $O(n^2)$ time by constructing the borders of convex hull according to the geometric properties of S. Another algorithm with $O(mn)$ time was give by Jarvis [8] ,where m is the number of convex hull vertices. Both of them have a high time complexity. Graham [9] provided a solution to compute the convex hull of a plane. Determine the point with minimal y-coordinate and calculate the angles between the horizontal line and the lines connecting the determined point and other points. According to the sorted angles, vertices are obtained. The divide-and-conquer method [10] was also applied to solve the problem. Point set was divided into two roughly equal-sized subsets. Their convex hulls were recursively computed, respectively. And the entire convex hull was determined by merging the two convex hulls. In another study, Chan [11] used point pairs to calculate the slopes of lines and determine the median values of these slopes, then divided the point set into two parts by median values and recursively computed the convex hull. He gave another algorithm which partitioned the point set and then computed the convex hull of each group, respectively. The entire convex hull was finally obtained by computing the union of the polygons. Exploiting the parallel computational model EREW PRAM, Chen et al. [12] proposed a parallel robust method for constructing convex hull. Brönnimann et al. [13] investigated the storage space of planar convex hull algorithms. As for dynamic planar convex hull, Overmars et al. [14] provided a solution that used $O(\log^2 n)$ time per update operation and maintained a leaf-linked balanced search tree of the vertices on the convex hull in clockwise order. Chan [15] gave a construction for the fully dynamic problem with $O(\log^{1+\varepsilon} n)$ amortized time for updates (for any constant $\varepsilon>0$), and $O(\log n)$ time for extreme point queries. In another work, Brodal and Jacob [16] presented a data structure that maintained a finite set of n points in the plane under insertion and deletion of points in amortized $O(\log n)$ time per operation. In [17], Ye considered the convex hull extraction in binary image and proposed a scheme with two procedures. In the first procedure, the image is scanned and a non-self-intersecting polygon is extracted; in the second procedure, the convex hull is

* Corresponding author. E-mail: tangzj230@163.com (Zhenjun Tang PhD). Address: Department of Computer Science, Guangxi Normal University, 15 Yucai Road, Guilin 541004, P. R. China

extracted from the polygon through checking the convexity of the polygon.

In this work, we firstly investigate the convex hull properties and then derive some new properties, e.g., monotonicity. Finally, we use these new properties to design convex hull algorithm for binary image. The proposed algorithm extracts eight extreme points on the boundary of binary image, and then partitions the image into 5 regions by using the extreme points. During the vertex computation, only these points in 4 regions need to be processed. By orderly scanning, the temporary convex hull is extracted. The entire convex hull is finally obtained by continuously updating the temporary convex hull. As the scanned areas are few and only the vertices of temporary convex hull require storage, the proposed algorithm has low complexities of time and space.

The rest of the paper is organized as follows. In Section 2, new convex hull properties are derived. The convex hull algorithm for binary image and the complexity analysis are then described in Section 3 and Section 4, respectively. Conclusions are drawn in Section 5.

2 Convex hull properties

Convex hull is a convex polygon having the following properties. For an edge pq, all other points lie on one side of the line running through p and q. Any line segment connecting two arbitrary nonadjacent points is in the interior of the polygon. And the interior angle is less than 180 degrees, etc. In this section, we will investigate the convex hull structure and then derive new convex hull properties, which will be applied to improve the efficiency of convex hull algorithm.

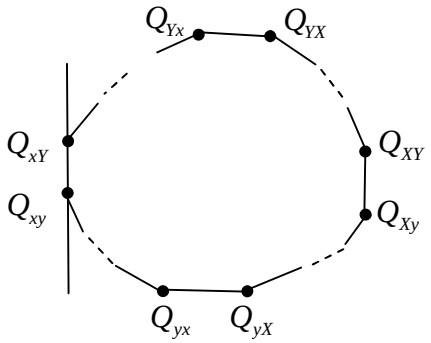
2.1 Extreme points

Let $Q = \{q_1, q_2, \ldots, q_M\}$ be a planar point set. In the subset whose points' x-coordinate are minimal among Q, Q_{xy} and Q_{xY} denote the points with minimal and maximal y-coordinate, respectively. In the subset whose points' x-coordinate are maximal among Q, Q_{Xy} and Q_{XY} represent the points with minimal and maximal y-coordinate, respectively. Likewise, in the subset whose points' y-coordinate are minimal among Q, Q_{yx} and Q_{yX} denote the points with minimal and maximal x-coordinate, respectively. In the subset whose points' y-coordinate are maximal among Q, Q_{Yx} and Q_{YX} represent the points with minimal and maximal x-coordinate, respectively. In the above variables, the first subscript denotes the extremum of coordinate and the second subscript denotes the extremum of the other coordinate under the first coordinate. Subscripts of capitalization and minuscule mean maximum and minimum, respectively, as shown in Fig.1. The definition of these points is given below.

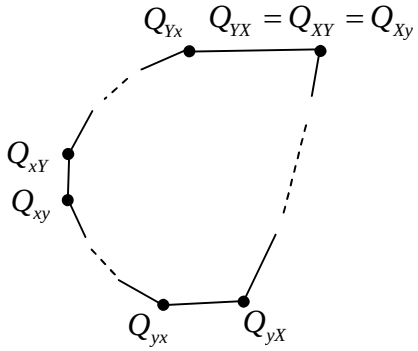
Definition 1. In the planar point set Q, Q_{xy}, Q_{xY}, Q_{Xy}, Q_{XY}, Q_{yx}, Q_{yX} ,Q_{Yx} and Q_{YX} are the extreme points of the convex hull, where Q_{xy} and Q_{xY}, Q_{Xy},and Q_{XY}, Q_{yx} and Q_{yX}, Q_{Yx} and Q_{YX} are the homogeneous extreme points, respectively.

Theorem 1. The extreme points in the planar point set Q are the convex hull vertices.

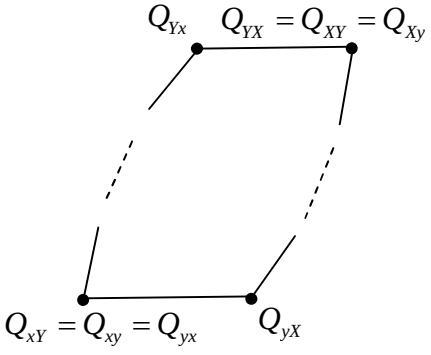
Proof. Assume that points $q_1, q_2, \ldots, q_M$ are convex hull vertices, and make a line l parallel to y-axis through Q_{xy} and Q_{xY}. Suppose Q_{xy} and Q_{xY} are not the convex hull vertices, as shown in Fig.1 (a). According to their definition, points are all on l or on the right side of l. For any vertex q_i, if q_i is on l, it must locate between Q_{xy} and Q_{xY}. This means that it can't be a convex hull vertex. So these vertices are all on the right side of l. Thus Q_{xy} and Q_{xY} fall in the left side of the convex hull instead of its interior. This contradicts the convex hull definition. Therefore Q_{xy} and Q_{xY} are vertices. Similar proofs can be given to other extreme points.



(a) Four monotone segments



(b) Three monotone segments



(c) Two monotone segments

Figure 1: Extreme points and monotone segments of the convex hull

2.2 Convex hull monotonicity and its construction

For segment $Q_{xY}Q_{Yx}$ of convex hull, let its points be numbered in a clockwise order, namely q_m, q_{m+1}, …, q_n ($n > m$), where q_i's coordinate is (x_i, y_i), $q_m = Q_{xY}$ and $q_n = Q_{Yx}$. Then q_m, q_{m+1}, …, q_{n-1} should be on the same side of straight line $\overline{q_m q_n}$ and the x-coordinate and y-coordinate of q_{m+1} should both increase. Suppose that both the x-coordinates and y-coordinates of q_{m+1}, q_{m+2}, …, q_i monotone increase while those of q_{i+1} decrease (either or both of them decrease). Since q_m, q_{i+1}, q_n are on the same side of straight line $\overline{q_{i-1}q_i}$, thus q_{i+1} should lie beneath the line $y = y_i$, as shown in Fig.2 (a). Hence q_{i-1} and q_n are on the different sides of the line $\overline{q_i q_{i+1}}$. This contradicts the fact that q_m, q_{m+1}, …, q_n ($n > m$) are all the convex hull vertices. Therefore both the x-coordinates and y-coordinates of points on segment $Q_{xY}Q_{Yx}$ monotone increase. Similarly, the x-coordinates of points on segment $Q_{YX}Q_{XY}$ monotonically increase while the y-coordinates of them decrease. The x-coordinates of points on segment $Q_{yx}Q_{xy}$ monotonically decrease while the y-coordinates of them increase. Both two coordinates of points on segment $Q_{Xy}Q_{yX}$ monotonically decrease. The monotonicity of these segments is defined as follows.

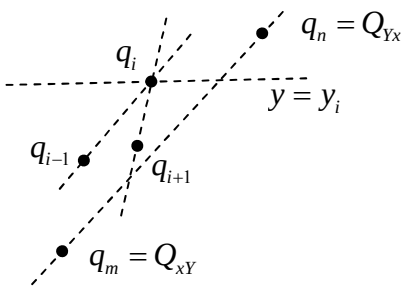


Figure 2: Convex hull monotonicity

Definition 2. If $Q_{xY} \neq Q_{Yx}$, the convex hull segment consisting of vertices from Q_{xY} to Q_{Yx} is called monotone increasing top segment. Likewise, if $Q_{YX} \neq Q_{XY}$, the convex hull segment consisting of vertices from Q_{YX} to Q_{XY} is named monotone decreasing top segment. If $Q_{Xy} \neq Q_{yX}$, the convex hull segment consisting of vertices from Q_{Xy} to Q_{yX} is called monotone decreasing bottom segment. If $Q_{yx} \neq Q_{xy}$, the convex hull segment consisting of vertices from Q_{yx} to Q_{xy} is named monotone increasing bottom segment.

Definition 3. All the monotone (both increasing and decreasing) top and bottom segments are called monotone segment.

If the monotone segments of a given convex hull are already determined, utilizing the definition 2 and the 8 extreme vertices can determine whether a specific monotone segment exists or not. The detailed theorem is as follow.

Theorem 2. The monotone increasing top segment exists if and only if $Q_{xY} \neq Q_{Yx}$. The monotone decreasing top segment exists if and only if $Q_{YX} \neq Q_{XY}$. The monotone increasing bottom segment exists if and only if $Q_{Xy} \neq Q_{yX}$. The monotone decreasing bottom segment exists if and only if $Q_{yx} \neq Q_{xy}$.

Similarly, according to the definition of convex hull monotonicity, the type of monotone segments can be determined by its vertices. Let f(P, A, B) = 0 represent the line equation, where the line runs through points A and B, P is a dynamic point on the line. There is a theorem about the type of monotone segments as follows.

Theorem 3. Let q_m, q_{m+1}, …, q_n ($n - m > 1$) be the vertices on a specific monotone segment of convex hull, and the coordinate of q_i be (x_i, y_i). For arbitrary i, j ($m \leq i < j$, $m \leq j < n$, $j \neq i$, $j \neq i + 1$), the sufficient and necessary conditions for that this monotone segment is a monotone increasing top segment are that $\begin{cases} \mathrm{f}(q_j, q_i, q_{i+1}) < 0 \\ y_i < y_{i+1} \end{cases}$. Likewise, as for the monotone decreasing top segment, the monotone decreasing bottom segment, monotone increasing bottom segment, their sufficient and necessary conditions are $\begin{cases} \mathrm{f}(q_j, q_i, q_{i+1}) < 0 \\ y_i > y_{i+1} \end{cases}$, $\begin{cases} \mathrm{f}(q_j, q_i, q_{i+1}) > 0 \\ y_i > y_{i+1} \end{cases}$, $\begin{cases} \mathrm{f}(q_j, q_i, q_{i+1}) > 0 \\ y_i < y_{i+1} \end{cases}$, respectively.

According to the convex hull definition, it has 4 monotone segments at most. Since convex hull is a closed shape, it has two monotone segments at least. The number of monotone segments can be determined by the extreme points. According to the number of monotone segments, convex hulls are classified into three types, as shown in Fig.1. Fig.1 (a) shows the convex hull with 4 monotone segments, Fig.1 (b) and Fig.1 (c) show the convex hull with 3 and 2 monotone segments, respectively.

3 Convex hull algorithm for binary image

3.1 Algorithm of monotone segment

Since the extreme points are convex hull vertices, the convex hull can be obtained by determining the vertices on the monotone segments between each pair of extreme points. In this work, a dynamic computation method is applied to determine the convex hull. Calculate the extreme points and determine the monotone segments. By dynamic scanning the boundary of image, the temporal convex hull of the scanned image is obtained. Scan the image boundary pixel by pixel until encountering the last boundary pixel. Thus, the monotone segments are obtained and the convex hull is extracted. The theorem of convex hull computation is as follows.

Theorem 4. Let $Q = \{q_m, q_{m+1}, \ldots, q_n\}$ $(n > m)$ be the vertices of a monotone segment of a specific convex hull, the coordinate of q_i and p be (x_i, y_i) and (x, y), respectively, $Q' = \{p\} \cup Q$ and $\min\{y_{n-1}, y_n\} < y < \max\{y_{n-1}, y_n\}$. If p and q_k $(k < n)$ are both the points in a specific monotone segment of Q', then $q_m, q_{m+1}, \ldots, q_k, p, q_n$ are all vertices on the monotone segment.

Proof: Suppose that $q_m, q_{m+1}, \ldots, q_n$ $(n > m)$ are the vertices of monotone increasing top segment of a specific convex hull. Since $p(x, y)$ belonging to $\{p\} \cup Q$ is a point on the monotone increasing top segment and $y_{n-1} < y < y_n$, then $f(p, q_{n-1}, q_n) > 0$ according to theorem 3. The location of p is shown in Fig.4. If q_k $(k < n)$ belonging to $\{p\} \cup Q$ is a vertex with maximum subscript on the monotone increasing top segment, then q_k and p are adjacent vertices on the new monotone increasing top segment. So for all q_i $(i \neq k)$, $f(p, q_{n-1}, q_n) < 0$. If q_j $(j < k)$ isn't the vertex of new convex hull, then $f(p, q_{j-1}, q_j) > 0$. But $f(q_k, q_{j-1}, q_j) < 0$. It means that q_k and p are on the different side of line $\overline{q_{j-1}q_j}$. So $f(q_k, p, q_j) < 0$. Therefore q_k isn't the vertex of new convex hull. This contradicts the precondition. Hence $q_m, q_{m+1}, \ldots, q_k, p, q_n$ are vertices on the monotone segment. Likewise, similar proofs of other monotone segments can be easily given.

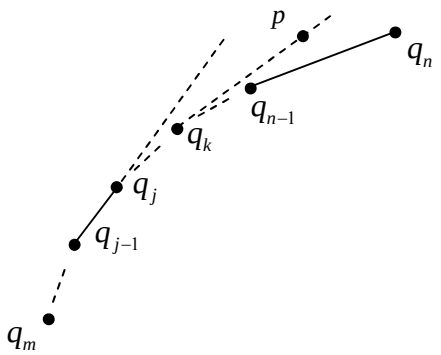


Figure 3: Monotone algorithm

Whether or not a pixel or a boundary point is a convex hull vertex depends on the relation about its position and the line. Take the monotone increasing top segment $q_m, q_{m+1}, \ldots, q_n$ $(n > m)$ for example. Theorem 3 shows that $f(q_j, q_i, q_{i+1}) < 0$ for arbitrary i, j $(m \leq i < j, m \leq j < n, j \neq i, j \neq i + 1)$. If a point p of image boundary satisfies $f(p, q_{n-1}, q_n) > 0$, then p is outside of the temporary convex hull. Thus p must be a new vertex of convex hull. Start from $k = n-1$ and decrease k by 1 each time. If $f(p, q_k, q_{k-1}) < 0$, then for arbitrary point A $(A \neq q_{k-1}, A \neq q_k)$, $f(A, q_k, q_{k-1}) < 0$. According to theorem 3, q_k is a new vertex of convex hull. By applying theorem 4, all vertices on this segment can be obtained.

3.2 Convex hull algorithm for binary image

Convex hull of binary image can be determined by its boundary pixel set. In fact, the convex hull of boundary pixel set is equal to the convex hull of binary image. So obtaining the boundary is an important step. Generally, boundary extraction by scanning the whole image requires storing all pixels. However, only few pixels are the convex hull vertices. Reducing the number of scanned pixels can both improve the time and space efficiency of algorithm. In this section, the method scanning from outside to inner is applied to extract the extreme points, as shown in Fig.4. The scanned regions are determined by the extreme points. By dynamic scanning the image boundary, temporary convex hull of the scanned boundary pixel set is computed. Finally, convex hull of binary image is available.

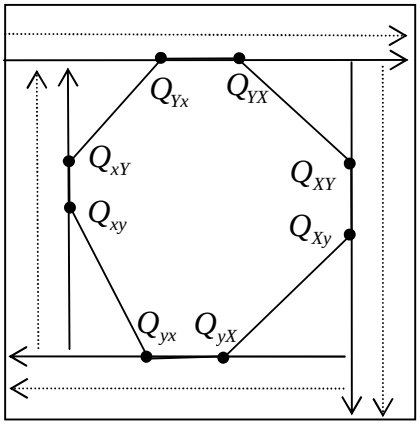


Figure 4: Extreme points of image convex hull

3.2.1 Collect the extreme points

In order to avoid repeated scanning, the method scanning from outside to inner is utilized to collect the extreme points on the image boundary. The detailed steps are as follows.

STEP 1: Begin at the top left of image and scan image from top to bottom until encountering the image boundary. Each row scan starts from left to right. If the scanned row has boundary points, Q_{Yx} and Q_{YX} represent the leftmost and rightmost boundary points, respectively. Thus two extreme points on the image boundary, Q_{Yx} and Q_{YX}, are obtained.

STEP 2: Begin at the line l_1 running through Q_{Yx} and Q_{YX} and scan image from right to left until encountering the image boundary. Each column scan starts from top to bottom. For the column having boundary pixels, let Q_{XY} and Q_{Xy} represent the topmost and bottommost boundary pixels, respectively. Thus two extreme points on the image boundary, Q_{XY} and Q_{Xy}, are obtained.

STEP 3: Begin at the line l_2 running through Q_{XY} and Q_{Xy} and scan image from bottom to top until encountering the image boundary. Each row scan starts from right to left. For the row having boundary pixels, let Q_{yX} and Q_{yx} represent the rightmost and leftmost boundary pixels, respectively. Thus two extreme points on the image boundary, Q_{yX} and Q_{yx}, are obtained.

STEP 4: Start from the line l_3 which is through Q_{yx} and Q_{yX} to the line l_1 and scan image from left to right until encountering the image boundary. Each column scan starts from bottom to top. For the column having boundary points, let Q_{xY} and Q_{xy} represent the topmost

and bottommost boundary pixels, respectively. Thus two extreme points on the image boundary, Q_{xY} and Q_{xy}, are obtained.

By the above steps, 8 extreme points of image convex hull are extracted.

3.2.2 Determine the scanned regions

Theorem 1 shows that the 8 extreme points are convex hull vertices. The lines connecting adjacent extreme points divide image into several regions, as shown in Fig.5. There are no boundary pixels outside of the rectangle formed by l1, l2, l3 and l4. If the boundary pixels are on the edges of the rectangle, then they aren't convex hull vertices. In the interior of the rectangle, pixels in region 0 aren't vertices either. Only those in regions 1, 2, 3 and 4 are likely to be vertices. Hence we just need to scan region 1~4 to obtain the boundary pixels and compute the convex hull by applying theorem 4. Since the vertices are numbered in a clockwise order, pixels extracted by scanning the four regions should satisfy theorem 4. The detailed method is as follows.

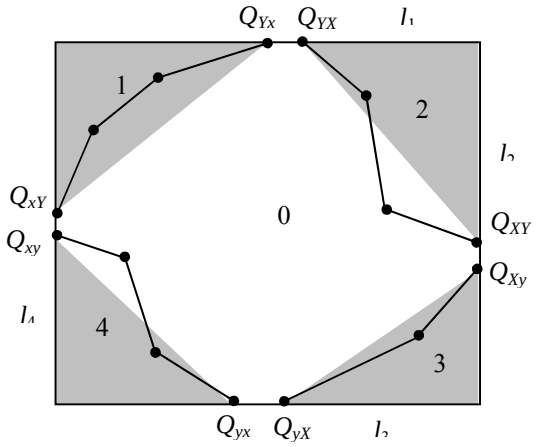


Figure 5: Scanned regions of image

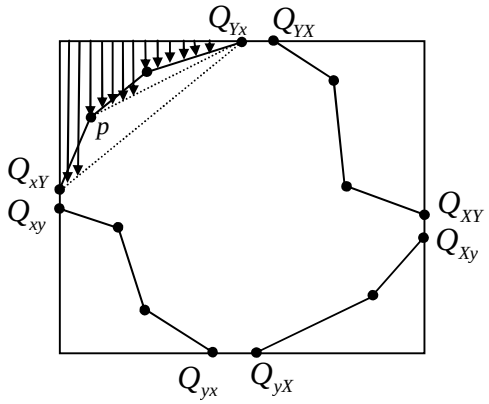
Region 1: Begin at the right side of l_4 and scan the region 1 horizontally from left to right. Each column in region 1 is scanned vertically from top to bottom. If there is no boundary pixel on the scanned line, then scan next column until encountering a boundary point p on the scanned line. Then p is a vertex of temporary convex hull in the scanned image. Compute the monotone increasing top segment of temporary convex hull by theorem 4. To improve the efficiency of algorithm and guarantee that the next scanned boundary pixels must be the vertices of temporary convex hull, the next column scan should stop once reaching the line pQ_{Yx}, as shown in Fig.6 (a). Continue to scan and compute the vertices of temporary convex hull until Q_{Yx} is encountered.

Region 2: Begin at the down side of l_1 and scan region 2 vertically from top to bottom. Each row in region 2 is scanned from right to left. Utilize the similar method introduced in region 1 to determine whether the boundary pixels are vertices or not. Continue to scan until Q_{XY} is encountered, as shown in Fig.6 (b).

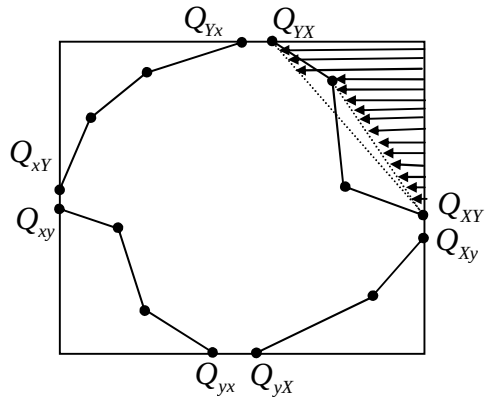
Region 3: Begin at the left side of l_2 and scan region 3 horizontally from right to left. Each column in region 3 is scanned from bottom to top. Utilize the similar method introduced in region 1 to determine whether the boundary pixels are vertices or not. Continue to scan until Q_{yX} is encountered, as shown in Fig.6 (c).

Region 4: Begin at the left side of l_3 and scan region 4 vertically from bottom to top. Each row in region 4 is scanned from left to right. Utilize the similar method introduced in region 1 to determine whether the boundary pixels are vertices or not. Continue to scan until Q_{yx} is encountered, as shown in Fig.6 (d).

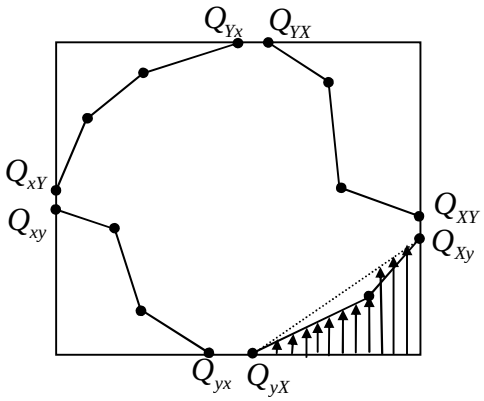
The extracted boundary pixels in the above steps both satisfy the monotone condition and the sequence required by theorem 4. Applying theorem 4 can extract the convex hull of image.



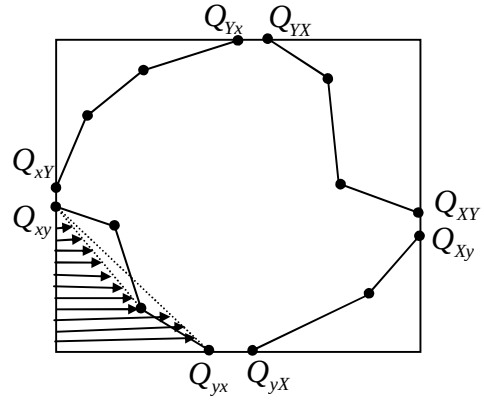
(a) Scan in region 1



(b) Scan in region 2



(c) Scan in region 3



(d) Scan in region 4

Figure 6: Scanned areas in each region

3.2.3 Compute convex hull vertices in scanned areas

Whether or not the pixel in scanned area is a convex hull vertex is just relative to other pixels in this region. By scanning the boundary pixels in each region, convex hull vertices in the corresponding region can be determined, respectively. Take region 1 for example. Let (x_{m1}, y_{m1}) and (x_{m2}, y_{m2}) be the coordinates of Q_{xY} and Q_{Yx}, respectively, f(p, A, B) = 0 be the equation of line running through A and B (p is a dynamic point), $v[i][j]$ and $c[i][j]$ be the pixel value and coordinate of p in the ith row and the jth column of the image, respectively. If $v[i][j] > 0$, p is a boundary pixel, or less a background pixel. Begin at $(x_{m1} + 1, y_{m1} + 1)$ and scan region 1 horizontally from left to right. In region 1, column is scanned from top to bottom. If there is no boundary pixel in the current column, scan next column in its right. If pixel p is a boundary pixel, then it must be a vertex of temporary convex hull. Apply theorem 4 to compute all vertices of temporary convex hull. Scan next column in the right. At this time, the scanned line is above the line pQ_{Yx}, as shown in Fig.6 (a). Stop scanning when the line $x = x_{m2}$ is encountered. Then the monotone segment of convex hull in region 1 is extracted. The detailed algorithm is as follows.

STEP 1: $i=x_{m1}+1$, $j = y_{m2} -1$, $q_1= Q_{xY}$, $A=Q_{xY}$, $n=2$;

STEP 2: IF ($i \geq x_{m2}$) goto **STEP 8**;

//no boundary pixel on the scanned line

IF (f(p, A, Q_{Yx}) ≤ 0) goto **STEP 3**;

//p is a vertex of temporary convex hull

IF ($v[i][j] > 0$) //p is the foreground pixel.

$k = n - 1$, $A = c[i][j]$, goto **STEP** 4;

ELSE goto **STEP** 5; //scan next pixel

STEP 3: $i = i +1$, $j = y_{m2}-1$, goto **STEP 2**;

//scan next vertical line in the right

STEP 4: IF ($k >1$) goto **STEP 6**;

ELSE $n = n +1$, goto **STEP 2**;

STEP 5: $j = j + 1$, goto **STEP 2**;

STEP 6: IF (f(p,q_{k-1},q_k) ≥ 0) // backtrack again.

goto **STEP 7**;

ELSE // finish backtracking

$n = k+1$, $q_n=c[i][j]$, $n =n +1$, goto **STEP 2**;

STEP 7: IF ($k>2$) //backtrack and process next pixel

$k = k - 1$, goto **STEP 6**;

ELSE //backtrack to the extreme point

$n = 2$, $q_n = c[i][j]$, $n = n + 1$, goto **STEP 2**;

STEP 8: $q_n= Q_{Yx},q_1, q_2,\ldots,q_n$ are convex hull vertices.

3.2.4 Convex hull algorithm for binary image

For the convex hull of binary image, compute the 8 extreme points Q_{xy}, Q_{xY}, Q_{Xy}, Q_{XY}, Q_{yx}, Q_{yX}, Q_{Yx} and Q_{YX}. According to these extreme points, determine the scanned regions of image, as shown in Fig.5. Then, convex hull vertices locate the regions 1~4, which are divided by the lines connecting the adjacent extreme points, as shown in Fig.5. Therefore, only the boundary pixels in these regions require computation. Utilize the monotone properties of convex hull and scan each region dynamically. Then, apply theorem 4 to compute each monotone segment of convex hull. The entire convex hull is obtained by merging these monotone segments. The detailed algorithm is as follows.

STEP 1: Scan the binary image and compute the 8 extreme points, Q_{xy}, Q_{xY}, Q_{Xy}, Q_{XY}, Q_{yx}, Q_{yX}, Q_{Yx} and Q_{YX}.

STEP 2: Utilize the 8 extreme points to determine the four regions where the convex hull vertices may exist.

STEP 3: Scan each region dynamically and obtain convex hull vertices on each monotone segment respectively.

STEP 4: Extract convex hull vertices on each monotone segment according to the following order, $Q_{xY}\rightarrow Q_{Yx}$, $Q_{YX}\rightarrow Q_{XY}$, $Q_{Xy}\rightarrow Q_{yX}$, $Q_{yx}\rightarrow Q_{xy}$. Each extreme point is extracted only one time. Then convex hull is obtained.

4 Complexity analysis

4.1 Time complexity

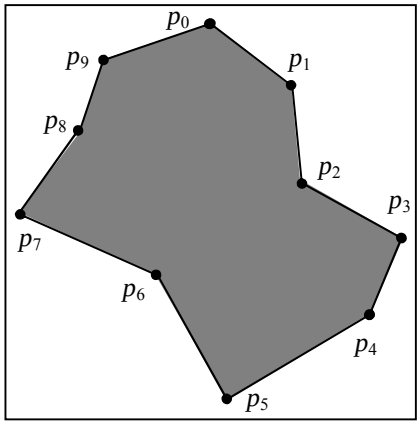
The time complexity is analyzed in the following ways. Suppose that the size of binary image is $N\times N$.

(1) If the image consists of a single pixel, then no convex hull exists. The time complexity is N^2.

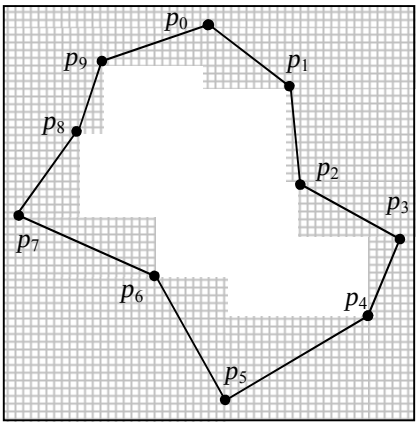
(2) If the image consists of two pixels or all pixels are on a line, then no convex hull exists, either. The time complexity is also N^2.

(3) The binary image has a convex hull if and only if three boundary pixels at least aren't on a line. Suppose that there are S pixels in the polygon whose vertices are the adjacent and inhomogeneous extreme points. The proposed method scans $N^2 - S$ pixels at most. And only $2N$ pixels at most should be computed when it determines whether or not a boundary pixel is a convex vertex. So the time complexity is $O(N^2 - S) + O(N)$.

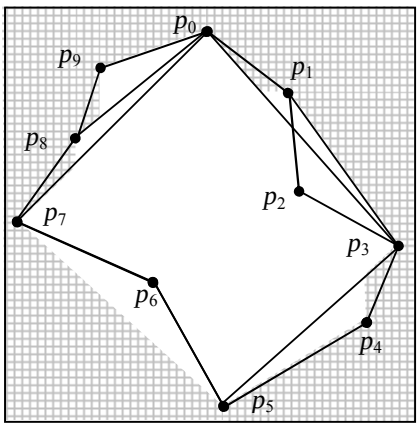
The above analyses show that the bigger the convex hull of binary image, the less the time complexity of the proposed algorithm. The time complexity of convex hull algorithm mainly depends on the size of scanned area. To show the efficiency of time complexity, we compare the proposed algorithm with the algorithm presented in [17]. A typical example is given in Fig.7. Fig.7 (a) is a binary image containing an object whose boundary has 10 vertices. Both algorithms are exploited to extract convex hull of the object in the binary image. Fig.7 (b) and Fig.7 (c) show the scanned areas of the algorithm [17] and the proposed algorithm respectively, where the gray grids denote their scanned areas. It is observed that our algorithm scans less area than the algorithm [17]. In general, if the object's boundary isn't a convex polygon, the scanned areas of the proposed algorithm are less than those of the algorithm [17]. Otherwise, the scanned areas of two algorithms are equivalent. Hence, the proposed algorithm needs less time than the algorithm [17] on average.



(a) Binary image containing an object with 10 vertices



(b) Scanned areas of the algorithm [17]



(c) Scanned areas of the proposed algorithm

Figure 7: A binary image and its scanned areas using different algorithms

4.2 Space complexity

The boundary pixels scanned by the proposed algorithm are the vertices of temporary convex hull. During the convex hull computation, only these vertices require storage. Therefore, the proposed algorithm has a low space complexity. Take Fig.7 for example. The algorithm [17] must store all 10 points from p_0 to p_9. Since p_2 and p_6 aren't scanned, the proposed algorithm doesn't need to compute and store them. So the space complexity of the proposed algorithm is lower than that of the algorithm [17].

5 Conclusions

In this paper, we derive some new convex hull properties, such as monotonicity, and use them to design algorithm for extracting convex hull of object in binary image. The proposed algorithm has a high efficiency by reducing computational cost in the following ways. (1) Divide the binary image into several regions by using the extreme points. Only those boundary pixels in a few regions require computation. (2) To determine a vertex in a given region doesn't need to compute those pixels in other regions. (3) Since the boundary pixels obtained by

scanning are computed dynamically, only these vertices of temporary convex hull require storage. Theoretical analyses show that the proposed algorithm has lower complexities of time and space than the algorithm [17] on average.

Acknowledgement

This work was partially supported by the Natural Science Foundation of China (60963008, 60763011), the Natural Science Foundation of Guangxi (0832104, 0447035), the project of the education administration of Guangxi (200911MS55, 200607MS135), and the Scientific and Technological Research Projects of Chongqing's Education Commission (KJ081309). The authors would like to thank the anonymous referees for their valuable comments and suggestions.

References

[1] Bhaniramka P., Wenger, R., and Crawfis, R. (2004) Isosurface construction in any dimension using convex hulls. *IEEE Transactions on Visualization and Computer Graphics*, vol.10, no.2, pp.130–141.

[2] Yuan B., and Tan C. L. (2007). Convex hull based skew estimation. *Pattern Recognition*, vol.40, no.2, pp.456-475.

[3] Nikolay M. Sirakov et al. (2004). Search space partitioning using convex hull and concavity features for fast medical image retrieval. In: *Proc. of the IEEE International Symposium on Biomedical Imaging*, Arlington, USA, pp.796–799.

[4] Yu X., Sun H., and Chen J. (2005). Points matching via iterative convex hull vertices paring. in: *Proc. of the fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, China, pp. 5350–5354.

[5] Gope C., and Kehtarnavaz N. (2007). Affine invariant comparison of point-sets using convex hulls and hausdorff distances. *Pattern Recognition*, vol.40, no.1, pp.309–320.

[6] Yu M. P., and Lo K. C. (2001). Object recognition by combining viewpoint invariant Fourier descriptor and convex hull. in: *Proc. of the 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, China, pp.401–404.

[7] Chand D. R., and Kapur S. S. (1970). An algorithm for convex polytopes. *JACM*, vol.17, no.1, pp.78–86.

[8] Jarvis R. A. (1973). On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, vol.2, no.1, pp.18–21.

[9] Graham R. L. (1972). An efficient algorithm for determine the convex hull of a finite linear set. *Information Processing Letters*, vol.1, no.1, pp.132–133.

[10] Preparata F. P. and Hong S. J. (1977). Convex hulls of finite sets of points in two and three dimensions. *CACM*, vol.20, no.2, pp.87–93.

[11] Chan T. (1996). Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete Comput. Geom*, vol.16, no.3, pp.361–368.

[12] Chen W., Wada K., and Kawaguchi K. (2002). Robust algorithms for constructing strongly convex hulls in parallel. *Theoretical Computer Science*, vol.289, no.1, pp. 277–295.

[13] Brönnimann H. et al. (2004). Space-efficient planar convex hull algorithms. *Theoretical Computer Science*, vol.321, no.1, pp.25–40.

[14] Overmars M. H., and Leeuwen J. V. (1981). Maintenance of configurations in the plane. *J. Comput. System Sci.*, vol.23, no.2, pp.166–204.

[15] Chan T. M. (2001). Dynamic planar convex hull operations in near-logarithmic amortized time. *Journal of the ACM*, vol.48, no.1, pp.1–12.

[16] Brodal, G. S., and Jacob R. (2002). Dynamic planar convex hull. in: *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, Vancouver, Canada, pp.617–626.

[17] Ye Q. (1995). A fast algorithm for convex hull extraction in 2D image. *Pattern Recognition Letters*, vol.16, no.5, pp.531–537.

Using Genetic Algorithms and Dominance Concepts for Generating Reduced Test Data

Ahmed S. Ghiduk
Department of Mathematics, Faculty of Science, Beni-Suef University, Egypt
E-mail: asaghiduk@yahoo.com

Moheb R. Girgis
Department of Computer Science, Faculty of Science, Minia University, Egypt
E-mail: mrgirgis@mailr.eun.eg

Keywords: genetic algorithms, software testing, automatic test-data generation, dominance. january

Received: March 6, 2008

Testing takes a considerable amount of the time and resources that are spent on producing software. Testing accounts for approximately 50% of the cost of the development of a software system. Therefore, techniques to reduce the cost of testing would be useful. This paper presents an automatic test-data generation technique that uses a genetic algorithm (GA). This technique applies the concepts of dominance relations between nodes to reduce the cost of software testing. These concepts are used to define a new fitness function to evaluate the generated test data. Finally, the paper presents the results of the experiments that have been conducted to evaluate the effectiveness of the proposed GA technique compared to the random testing (RT) technique. These experiments are used to evaluate the effectiveness of the new fitness function and the technique used to reduce the cost of software testing.

Povzetek: Predstavljen je genstski algoritem za zmanjšanje števila testnih podatkov.

1 Introduction

Software testing is the main technique used to improve the quality and increase the reliability of software. Software testing is a complex, labor-intensive, and time consuming task that accounts for approximately 50% of the cost of a software system development [1]. Increasing the degree of automation and the efficiency of software testing can reduce the cost of software design, decrease the time required for software development, and increase the quality of software.

One critical task in the automation of software testing is the automation of the generation of test data to satisfy a given adequacy criterion. Test-data generation is the process of identifying a set of program input data that satisfies a given testing criterion. Test-data generation has two main aspects: test generation technique and application of a test-data adequacy criterion. A test generation technique is an algorithm that generates test data, whereas an adequacy criterion is a predicate that determines whether the testing process is finished.

There has been much previous work in automatically generating test data. Perhaps the most commonly encountered are random test-data generation, symbolic test-data generation, dynamic test-data generation, and recently, test-data generation based on *GA*.

Random test-data generation consists of generating inputs at random until useful inputs are found (e.g., [2, 3, 4]). The problem with this approach is clear with complex programs or complex adequacy criteria, an adequate test input may have to satisfy very specific requirements. In such cases, the number of adequate inputs may be quite small compared to the total of inputs, so the probability of selecting an adequate input by chance may be low.

Symbolic test-data generation consists of assigning symbolic values to variables to create an abstract, mathematical characterization of the program's functionality. With this approach, test-data generation can be reduced to a problem of solving an algebraic expression. Many test-data generation methods that use symbolic execution to find inputs that satisfy a test requirement have been proposed (e.g., [5, 6, 7, 8, 9, 10]). A number of problems are encountered in practice when symbolic execution is used. One of such problems arises in indefinite loops, where the number of iterations depends on a non-constant expression, and the index of array, where data is referenced indirectly. Pointer references also present a problem because of the potential for aliasing.

Dynamic test-data generation is based on the idea that if some desired test requirement is not satisfied, data collected during execution can be used to determine which tests come closest to satisfying the requirement [11] and [12]. With the help of this feedback, test inputs are incrementally modified until one of them satisfies the requirement. Two limitations are commonly found in dynamic test-data generation systems. First many systems make it difficult to generate tests for large programs because they work only on simplified

programming languages. Second, many systems use gradient descent techniques to perform function minimization and, therefore, they can stall when they encounter local minima.

Several search based test-data generation techniques have been developed (e.g., [13, 14, 15, 16, 17, 18, 19, 20]). These techniques had focused on finding test data to satisfy a number of control-flow and data-flow testing criteria. Genetic algorithms have been the most widely employed search-based optimization technique in software testing [21]. The new features of GAs make them capable of finding the nearly global optimum solution. Test-data generation methods based on genetic algorithms have many problems due to the use of fitness functions that depend on control dependences or branch-distance in its calculations. The fitness function that takes control dependencies into account faces a problem to find an input to traverse a target node within loops. A further problem is the assignment of approximation levels for some classes of program with unstructured control flow. A branch-distance-related problem can occur with nested branch predicates. Once input data is found for one or more of the predicates, the chances of finding input data that also fits subsequent predicates decreases, because a solution for subsequent conditions must be found without violating any of the earlier conditions [22, 23, 24]).

To solve the problem of reducing the cost of software testing, we have developed a new GA-based technique with a new fitness function that reduces the test requirements and overcomes the problems of the previous GA-based test-data generation methods.

This paper presents an automatic test-data generation technique that uses a GA for white-box testing. This technique applies the concepts of dominance relations between nodes to reduce the cost of software testing. These concepts are used to define a new fitness function to evaluate the generated test data.

The paper is organized as follows: Section 2 gives some important definitions. Section 3 describes the proposed technique, which is used to reduce the cost of software testing. Section 4 describes the proposed GA technique for automatic test-data generation, and the results of applying this algorithm to an example program. Section 5 presents the results of the experiments that are conducted to evaluate the effectiveness of the proposed GA compared to the random testing technique, to evaluate the effectiveness of the new fitness function and the technique used to reduce the cost of software testing. Section 6 presents the conclusions and future work.

2 Background

We introduce here some basic concepts that will be used through this work.

2.1 The principles of genetic algorithms

The basic concepts of GAs were developed by Holland [25]. GAs are commonly applied to a variety of problems involving search and optimization. GAs search methods are rooted in the mechanisms of evolution and natural genetics. GAs draw inspiration from the natural search and selection processes leading to the survival of the fittest individuals. GAs generate a sequence of populations by using a selection mechanism, and use crossover and mutation as search mechanisms [26].

The principle behind GAs is that they create and maintain a population of individuals represented by chromosomes (essentially a character string analogous to the chromosomes appearing in DNA). These chromosomes are typically encoded solutions to a problem. The chromosomes then undergo a process of evolution according to rules of selection, mutation and reproduction. Each individual in the environment (represented by a chromosome) receives a measure of its fitness in the environment. Reproduction selects individuals with high fitness values in the population, and through crossover and mutation of such individuals, a new population is derived in which individuals may be even better fitted to their environment. The process of crossover involves two chromosomes swapping chunks of data (genetic information) and is analogous to the process of sexual reproduction. Mutation introduces slight changes into a small proportion of the population and is representative of an evolutionary step. The structure of a simple GA is given below.

```
Simple Genetic Algorithm ()
{
   initialize population;
   evaluate population;
   while termination criterion not reached {
     select solutions for next population;
      perform crossover and mutation;
     evaluate population; }
  }
```

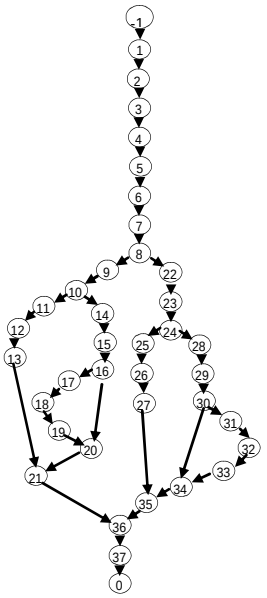
The algorithm will iterate until the population has evolved to form a solution to the problem, or until a maximum number of iterations have occurred (suggesting that a solution is not going to be found given the resources available).

2.2 The control flow graph

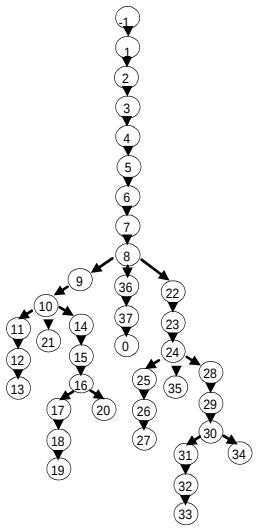
A program's structure is conveniently analyzed by means of a directed graph, called control flow graph that gives a graphical representation of the program's control flow. A directed graph or digraph $G = (V, E)$ consists of a set V of nodes or vertices, where each node represents a statement, and a set E of directed edges or arcs, where a directed edge $e = (n, m)$ is an ordered pair of adjacent nodes, called *Tail* and *Head* of e, respectively. For a node n in V, indegree(n) is the number of arcs entering and outdegree(n) the number of arcs leaving it. Figure 1.b shows the control flow graph G of the example program, which is shown in Figure 1.a. We are augmented the control flow graph by the unique entry node (-1) and the unique exit node (0).

```
1. #include <iostream.h>
2. void main ( )
3. {
4.    int x, y, z;
5.    int mid;
6.    cin>>x>>y>>z;
7.    mid = z;
8.    if(y<z)
9.    {
10.       if(x<y)
11.       {
12.          mid = y;
13.       }
14.       else
15.       {
16.         if(x<z)
17.         {
18.           mid = x;
19.         }
20.       }
21.     }
22.     else
23.     {
24.       if(x>=y)
25.       {
26.         mid = y;
27.       }
28.       else
29.       {
30.         if(x>z)
31.         {
32.           mid = x;
33.         }
34.       }
35.     }
36.     cout<<"Middle value="<<mid;
37.  }
```

(a)



(b)



(c)

Figure 1: (a) Example program, (b) its Control Flow Graph *G*, (c) its Dominator Tree *DT(G)*

2.3 Dominance

Let $G = (V, E)$ be a digraph with two distinguished nodes n_0 and n_k. A node n dominates a node m if every path P from the entry node n_0 to m contains n.

Several algorithms are given in the literature to find the dominator nodes in a digraph (e.g., [27] and [28]).

By applying the dominance relations between the nodes of a digraph G, we can obtain a tree (whose nodes represent the digraph nodes) rooted at n_0. This tree is called the dominator tree; we denote it by $DT(G)$. A (rooted) tree $DT(G) = (V, E)$ is a digraph in which one distinguished node n_0, called the root, is the Head of no arcs; every node n except the root n_0 is a Head of just one arc and there exists a (unique) path (dominance path) from the root n_0 to each node n; we denote this path by $dom(n)$. Tree nodes of outdegree zero are called leaves.

For example, Figure1.c shows the dominator tree of the flow graph G (Figure 1.b) of the example program (Figure 1.a). The dominance path of node 21 in $DT(G)$ is $dom(21)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 21.

3 Reducing the cost of testing

This section describes our proposed technique for reducing the cost of software testing that fulfils the all-statements coverage criterion. The proposed technique is based on the concepts of the dominance relations between nodes of the program's control flow graph. This technique aims to cover a subset of statements (nodes of the program's control flow graph) that guarantees the coverage of all statements of the tested program.

The set of leaves of the dominator tree is an essential set (i.e., every set of paths that covers it, covers all nodes in the tree). To illustrate the effectiveness of this technique, we apply it to the example program given in Figure 1. The set of leaves of the example program is L = {0, 13, 19, 20, 21, 27, 33, 34, 35}. The dominance paths of the elements of this set are:

$dom(0)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 36, 37, 0.
$dom(13)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13.
$dom(19)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 15, 16, 17, 18, 19.
$dom(20)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 15, 16, 20.
$dom(21)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 21.
$dom(27)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 22, 23, 24, 25, 26.
$dom(33)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 22, 23, 24, 28, 29, 30, 31, 32, 33.
$dom(34)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 22, 23, 24, 28, 29, 30, 34.
$dom(35)$ = -1, 1, 2, 3, 4, 5, 6, 7, 8, 22, 23, 24, 35.

Covering an element of the set L guarantees the coverage of its dominance path. It is clear that, the union of nodes of this set of dominance paths is the set of all nodes of the program's control flow graph (i.e., all statements of the tested program).

So, our goal of covering all nodes of the program's control flow graph can be reduced to covering only the set of leaves of the dominator tree. Thus, by applying the proposed technique the cost of testing of the example program is reduced by 75.5%.

4 GA-based test-data generation

This section describes the proposed GA for automatic test-data generation, which uses a new fitness function to evaluate the generated test data. This new fitness function depends on the concepts of the dominance relations between nodes of the program's control flow graph. The algorithm searches for test cases that satisfy the all-statements criterion. The major components of this GA are discussed below.

4.1 Representation

The proposed GA uses a binary vector as a chromosome to represent values of the program input variables. The length of the vector depends on the required precision and the domain length for each input variable.

Suppose we wish to generate test cases for a program of k input variables $x_1,\ldots, x_k$ where each variable x_i can take values from a domain $D_i = [a_i, b_i]$. Suppose further that d_i decimal places are desirable for the values of each variable x_i. To achieve such precision, each domain D_i should be divided into $(b_i - a_i)\times 10^{d_i}$ equal size ranges. Let us denote by m_i the smallest integer such that $(b_i - a_i)\times 10^{d_i} \le 2^{m_i} - 1$. Then, a representation having each variable x_i coded as a binary string $string_i$ of length m_i clearly satisfies the precision requirement. The mapping from the binary string $string_i$ to a real number x_i from the range $[a_i, b_i]$ is performed by the following formula:

$$x_i = a_i + x_i' \times \frac{b_i - a_i}{2^{m_i} - 1} \qquad (4.1)$$

Where x_i' represents the decimal value of the binary string $string_i$ [Michalewicz, 1999].

It should be noted that the above method can be applied for representing values of integer input variables by setting d_i to 0, and using the following formula instead of formula (4.1):

$$x_i = a_i + \mathrm{int}(x_i' \times \frac{b_i - a_i}{2^{m_i} - 1}) \qquad (4.2)$$

Now, each chromosome (as a test case) is represented by a binary string of length $m = \sum_{i=1}^{k} m_i$; the first m_1 bits map into a value from the range $[a_1, b_1]$ of variable x_1, the next group of m_2 bits map into a value from the range $[a_2, b_2]$ of variable x_2, and so on; the last group of m_k bits map into a value from the range $[a_k, b_k]$ of variable x_k.

For example, suppose a program has 2 input variables x and y, where $-3.0 \le x \le 12.1$ and $4.1 \le y \le 5.8$, and the required precision is four decimal places for each variable. The domain of variable x has length 15.1; the precision requirement implies that the range [-3.0, 12.1] should be divided into at least 15.1×10000 equal size ranges. This means that 18 bits are required as the first part of the chromosome: $2^{17} < 151000 \le 2^{18}$. The domain of variable y has length 1.7; the precision requirement implies that the range [4.1, 5.8] should be divided into at least 1.7×10000 equal size ranges. This means that 15 bits are required as the second part of the chromosome: $2^{14} < 17000 \le 2^{15}$. The total length of a chromosome (test case) is then m = 18+15=33 bits; the first 18 bits code x and remaining 15 bits code y. Let us consider an example chromosome:

010001001011010000111110010100010.

By using formula (4.1), the first 18 bits, 010001001011010000, represents x = 1.0524, and the next 15 bits, 111110010100010, represents y = 5.7553. So the given chromosome corresponds to the data values 1.0524 and 5.7553 for the variables x and y, respectively [19].

4.2 Initial population

As mentioned above, each chromosome (as a test case) is represented by a binary string of length m. We randomly generate *pop_size* m-bit strings to represent the initial population, where *pop_size* is the population size. The appropriate value of *pop_size* is experimentally determined. Each chromosome is converted to k decimal numbers representing values of k input variables $x_1,\ldots, x_k$ (i.e. a test case) by using formula (4.1) or (4.2).

4.3 Evaluation function

The algorithm uses a new evaluation (fitness) function to evaluate the generated test data. This new fitness function depends on the concepts of the dominance relations between nodes of the program's control flow graph. The algorithm uses this new fitness function to evaluate each test case by executing the program with it as input, and recording the traversed nodes in the program that are covered by this test case. We denote to the set of traversed nodes by *exePath*. Also, it finds the dominance path *dom*(n) of the target node n. The fitness function is the ratio of the number of covered nodes of the dominance path of the target node to the total number of nodes of the dominance path of the target node. The fitness value $ft(v_i)$ for each chromosome v_i (i = 1, …, *pop_size*) is calculated as follows:

1. Find *exePath*: the set of the traversed nodes in the program that are covered by a test case.
2. Find *dom*(n): dominance path of the target node n (the set of dominator nodes from the entry of the dominator tree to n).
3. Determine $(dom(n) - exePath)$: uncovered nodes of the dominance path (the difference between the dominance path and the traversed nodes).
4. Determine $(dom(n) - exePath)'$: covered nodes of the dominance path (the complement set of the difference set between the dominance path and the traversed nodes).
5. Calculate $\left|(dom(n) - exePath)'\right|$: number of covered nodes of the dominance path (cardinality of the complement set).

6. Calculate $|dom(n)|$: number of nodes of the dominance path of the target node *n* (cardinality of the dominance set).

Then, $ft(v_i) = \dfrac{\left|(dom\ (n) - exePath\ )'\right|}{|dom\ (n)|}$

The fitness value is the only feedback from the problem for the GA. A test case that is represented by the chromosome v_i is optimal if its fitness value $ft(v_i) = 1$.

4.4 Selection

After computing the fitness of each test case in the current population, the algorithm selects test cases from all the members of the current population that will be parents of the new population. In the selection process, the GA uses the roulette wheel method [29]. This method is described below.

For the selection of a new population with respect to the probability distribution based on fitness values, a roulette wheel with slots sized according to fitness is used. Such roulette wheel is constructed as follows:

- Calculate the fitness value $ft(v_i)$ for each chromosome v_i ($i = 1,\ldots,pop\_size$).
- Find the total fitness of the population $F = \sum_{i=1}^{pop\_size} ft(v_i)$.
- Calculate the relative fitness value *rft* for each chromosome $rft(v_i) = \dfrac{ft(v_i)}{F}$.
- Calculate the cumulative fitness value *cft* for each chromosome

$$cft(v_i) = \begin{cases} rft(v_i) & \text{i=1} \\ cft(v_{i-1}) + rft(v_i) & \text{i=2,...,pop\_size} \end{cases}.$$

The selection process is based on spinning the roulette wheel *pop_size* times; each time we select a single chromosome for a new population in the following way:

- Generate a random (float) number *r* from the range [0..1].
- If $r < cft(v_1)$ then select the first chromosome v_1; otherwise select the i-th chromosome v_i ($2 \le i \le pop\_size$) such that $cft(v_i) \le r < cft(v_{i+1})$.

Obviously, some chromosomes would be selected more than once.

4.5 Recombination

In the recombination phase, we use two operators, crossover and mutation, which are the key to the power of GAs. These operators create new individuals from the selected parents to form a new population.

Crossover: It operates at the individual level. During crossover, two parents (chromosomes) exchange substring information (genetic material) at a random position in the chromosome to produce two new strings (offspring). The objective here is to create better population over time by combining material from pairs of (fitter) members from the parent population. Crossover occurs according to a crossover probability. The probability of crossover *PXOVER* gives us the expected number $PXOVER \times pop\_size$ of chromosomes, which undergo the crossover operation. We proceed in the following way:

For each chromosome in the parent population:

- Generate a random (float) number r from the range [0..1];
- If $r < PXOVER$ then select given chromosome for crossover.

Now we mate selected chromosomes randomly: For each pair of coupled chromosomes we generate a random integer number *pos* from the range [1..*m*-1] (*m* is the number of bits in a chromosome). The number *pos* indicates the position of the crossing point. Two chromosomes $(b_1\ldots b_{pos}b_{pos+1}\ldots b_m)$ and $(c_1\ldots c_{pos}c_{pos+1}\ldots c_m)$ are replaced by a pair of their offspring $(b_1\ldots b_{pos}c_{pos+1}\ldots c_m)$ and $(c_1\ldots c_{pos}b_{pos+1}\ldots b_m)$.

Mutation: It is performed on a bit-by-bit basis. Mutation always operates after the crossover operator, and flips each bit with the pre-determined probability. The probability of mutation *PMUTATION*, gives us the expected number of mutated bits $PMUTATION \times m \times pop\_size$. Every bit (in all chromosomes in the whole population) has an equal chance to undergo mutation (i.e., change from 0 to 1 or vice versa). So we proceed in the following way:

For each chromosome in the current (i.e., after crossover) population and for each bit within the chromosome:

- Generate a random (float) number r from the range [0..1];
- If $r < PMUTATION$ then mutate the bit.

In the traditional GA approach the population would evolve until one individual from the whole set which represents the solution is found. In our case, this condition would correspond to finding groups of data items achieving the test requirements (i.e., covering the set of leaves of the dominator tree) of the tested program. We let the population evolves until a combined subset of the population achieves the desired test requirement. The evolution stops when a set of individuals has traversed the dominance path of the test requirement and its fitness value $ft(v_i) = 1$. The solution is this set.

4.6 Elitist

The elitist function enhances the current population by storing the best member of the previous population. If the best member of the current population is worse than the best member of the previous population it exchanges them, and the best member of the current population would replace the worst member of the current population. After that, it stores the best member of the current population.

4.7 Example

To illustrate the operations of the above genetic algorithm, a part of the result of applying the system, which implements it, to the example program, is presented below. The final report (Figure 2) of the result

contains a table that shows the run number and the test requirement to be covered in this run and the number of the generation in which the test requirement is covered and the status whether it is covered or not. The final statistics shows that we needed 36 generations to obtain 100% coverage of the nine test requirements.

Appendix A shows the part of the result of applying the system to test requirement number 5 (statement 27). This part of the result shows the execution of the steps of the genetic algorithm and operations of our proposed technique.

```
** ---------------------------------------------- Final Report ----------------------------------
*************************************************************************
** ------------------ GA completed successfully ------------------- **
** Final Statistics:-
** -------------------
** Total number of Req.-----------------:  9
** No. of Covered Req.------------------:  9
** The Covered Req. are-----------------: 13, 19, 20, 21, 27, 33, 34, 35, 0
** No. of Uncovered Req.----------------:  0
** The Uncovered Req. are---------------:
** Coverage Ratio-----------------------: 100.0%
** No. of Runs--------------------------:  9
** ------------------------------------------------------------------------------
** | Run No. |    Test Req to be Covered   |    Generation No. |  Covered  |
** ------------------------------------------------------------------------------
**|    1     |            13               |         1         |     Y     |
**|    2     |            19               |         1         |     Y     |
**|    3     |            20               |         6         |     Y     |
**|    4     |            21               |         1         |     Y     |
**|    5     |            27               |         2         |     Y     |
**|    6     |            33               |        22         |     Y     |
**|    7     |            34               |         1         |     Y     |
**|    8     |            35               |         1         |     Y     |
**|    9     |             0               |         1         |     Y     |
** ------------------------------------------------------------------------------
** Total no. of Generations-------------: 36
** Total no. of Test Cases---------------:144
** No. of Successful Test Cases----------: 16
** No. of Distinct Successful Test Cases-: 13
** The Distinct Successful Test Cases are:
**                          1)[ 2 , 3 , 4 ]                    2)[ 1 , 2 , 3 ]
**                          3)[ 2 , 1 , 4 ]                    4)[ 3 , 0 , 3 ]
**                          5)[ 2 , 2 , 2 ]                    6)[ 2 , 3 , 1 ]
**                          7)[ 3 , 4 , 1 ]                    8)[ 4 , 2 , 1 ]
**                          9)[ 4 , 2 , 2 ]                   10)[ 2 , 4 , 3 ]
**                         11)[ 4 , 3 , 4 ]                   12)[ 4 , 2 , 3 ]
**                         13)[ 1 , 2 , 4 ]
** No. of Covering Test Cases------------:  9
** The Covering Test Cases are----------:
**                          1)[ 2 , 3 , 4 ]                    2)[ 3 , 2 , 1 ]
**                          3)[ 3 , 0 , 3 ]                    4)[ 1 , 2 , 1 ]
**                          5)[ 2 , 2 , 2 ]                    6)[ 2 , 3 , 1 ]
**                          7)[ 4 , 1 , 1 ]                    8)[ 4 , 2 , 1 ]
**                          9)[ 2 , 4 , 3 ]
** ---------------------------------------------- The end of Report ----------------------------- **
*************************************************************************
```

Figure 2: The Final Report.

4.8 Overall algorithm

The proposed GA-based technique accepts as input the program to be tested, the number of input variables, and the domain and precision of each input variable. Also, it accepts the GA parameters: population size, maximum number of generations, and probabilities of the crossover and mutation operators. The algorithm produces a set of test cases, the set of nodes covered by these test cases, and the list of uncovered nodes, if any.

The algorithm selects, one at a time, an uncovered node of the set of leaves nodes of the dominator tree and evolves the initial test data until the required test data are obtained or the maximum number of generations is exceeded. Whenever a node is covered, the test case that caused this coverage is stored in a score board. The technique checks the coverage of remaining uncovered nodes by the generated test data that cover the current node. The overall algorithm is presented in Figure 3.

5 Empirical evaluation

This section presents the results of the experiments that have been carried out to evaluate the effectiveness of the proposed GA compared to the random testing (RT) technique, and to evaluate the effectiveness of the proposed fitness function. A set of nine C++ programs is used in the experiments. To achieve a fair comparison, the random test-data generator was designed to randomly generate sets of *pop_size* test cases in each iteration. The used GA parameters were as follows: Maximum Number of Generations *MAXGENS* = 100, *PXOVER* = 0.8 and *PMUTATION* = 0.15.

```
/* A GA algorithm to automatically generate test cases for a given program */
Input:
      The program to be tested P;
      Number of program input variables;
      Domain and precision of input data;
      Population size;
      Maximum no. of generations (Max_Gen);
      Probability of crossover;
      Probability of mutation;
Output:
      Set of test cases for P, and the set of nodes covered by each test case;
      List of uncovered nodes, if any;
Begin
      Step 0: Setup (Analysis P to find prerequisites)
          1. Classify the program's statements.
          2. Build the program's control flow graph CFG.
          3. Build the program's dominator tree DT.
          4. Find the set of leaves L of the dominator tree.
          5. Instrument P to obtain P'.
      Step 1: Initialization
          Initialize the score board to zero;
          nRun ← 0;
          Set of test cases for P ← φ;
          nCases ← 0;
       Step 2: Generate test cases
          For each uncovered node and not selected before in the set of nodes to be tested (L)
              Begin
                nRun ← nRun + 1;
                Create Initial_Population;
                Current_population ← Initial_Population;
                No_Of_Generations ← 0;
                For each member of current population do
                  Begin
                      Convert the current chromosome to the corresponding set of decimal values;
                      Execute P' with this data set as input;
                      Evaluate the current test case;
                      If (the current node is covered) then
                          Mark the current node as covered;
                      End If
                End For;
                Keep the best member of the current population;
                While (current node is not covered and No_Of_Generations ≤ Max_Gen) do
                    Begin
                      Select set of parents of new population from members of
                      current population using roulette wheel method;
                      Create New_Population using crossover and mutation operators;
                      Current_Population ← New_Population;
                      For each member of Current_Population do
                      Begin
                        Convert current chromosome to the corresponding set of decimal values;
                        Execute P' with this data set as input;
                        Evaluate the current test case;
                        If (the current node is covered) then
                            Mark the current node as covered;
                        End If
                      End For;
                      Elitist function: If the best member of the current population is worse than the
                      best member of the previous population then exchange them, and the best member
                      of the current population would replace the worst member of the current
                      population.
                      Increment No_Of_Generations;
                End While;
                If (the current node is covered) then
                    nCases ← nCases + 1;
                    Add this test cases to set of test cases for P;
                    Update the score board;
                    Check all uncovered nodes by this test case.
                End If
          End For;
      Step 3: Produce output
          Return set of test cases for P, and set of nodes covered by each test case;
          Report on uncovered nodes, if any;
End.
```

Figure 3: The overall algorithm.

Table 1 shows the reduction percentage of the test requirements. Column#2 shows the total number of test requirements which are demanded by the all-statements criterion and column#4 gives the number of the reduced test requirements. The reduction percentage is 83.3% for prog# 6 and prog# 9 and 75.6% for prog#2. It is clear that the reduction percentage isn't less than 75%. These results show the effectiveness of the proposed technique to reduce the cost of all-statements testing by reducing the number of the test requirements.

Table 1: The reduction percentage of the cost of software testing

Prog#	Program Size ProgSize)	No. Of Variable	No. of Test Requirements (nTestReq)	Reduction percentage= 100*(1 - nTestReq/ProgSize)%
1	42	3	8	80.9%
2	37	3	9	75.6%
3	27	2	5	81.4%
4	41	2	9	78%
5	38	2	7	81.5%
6	36	2	6	83.3%
7	33	2	7	78.7%
8	19	1	4	78.9%
9	18	2	3	83.3%

Table 2 shows the results of applying the proposed GA technique and the RT technique to nine C++ programs. These results show the effectiveness of the proposed GA technique over the random testing technique where the GA covers 100% of the set of test requirements in 8 programs while random testing covers 100% of the set of test requirements in 2 programs. In program 3, the GA needed only 9 generations and 90 test cases to reach 100% coverage while RT needed 203 generations and 2030 test cases to reach 60% coverage. In program 4, the GA needed 231 generations and 2310 test cases to reach 77.8% coverage while RT needed 504 generations and 5040 test cases to reach 44.4% coverage.

6 Conclusions and future work

This paper presented an automatic test-data generation technique that uses a genetic algorithm. This technique applies the concepts of dominance relations between nodes to reduce the cost of software testing. These concepts used to define a new fitness function to evaluate the generated test data.

Experiments have been carried out to evaluate the effectiveness of the proposed GA technique compared to the RT technique, and to evaluate the effectiveness of the new fitness function and the technique used to reduce the cost of software testing. The results of these experiments showed that the proposed GA technique outperformed the RT technique in 7 out of the 9 programs used in the experiments. In the other two programs, the proposed GA reached the same coverage percentage as the RT technique. The experiments also showed that the proposed technique reduced the cost of software testing by more than 75%. Also, the results of the experiments showed that the new fitness function is quite suitable to evaluate the generated test-data and showed the usefulness of the concepts of dominance relations between nodes of the program's control flow graph in reducing the number of test requirements.

This technique is being modified to generate test data for data flow testing. The concepts of dominance relations between nodes of the program's control flow graph will be used to define a new fitness function to evaluate the generated test data for data flow testing.

Table 2: A comparison between the proposed GA technique and the RT technique.

Prog#	Pop. Size	Method	Total no. of Generations	Total no. of Test Cases	No. of successful Test Cases	Total no. of test Req.	No. of Covered Req.	Coverage Ratio %
1	9	GA	19	171	34	8	8	100%
		RT	109	981	35	8	7	87.5%
2	10	GA	9	90	36	9	9	100%
		RT	9	90	36	9	9	100%
3	10	GA	9	90	32	5	5	100%
		RT	203	2030	30	5	3	60%
4	10	GA	231	2310	34	9	7	77.8%
		RT	504	5040	40	9	4	44.4%
5	10	GA	9	90	56	7	7	100%
		RT	106	1060	54	7	6	85.7%
6	9	GA	26	234	37	6	6	100%
		RT	105	945	36	6	5	83.3%
7	10	GA	35	350	48	7	7	100%
		RT	106	1060	47	7	6	85.7%
8	10	GA	10	100	27	4	4	100%
		RT	103	1030	26	4	3	75%
9	10	GA	3	30	25	3	3	100%
		RT	3	30	25	3	3	100%

References

[1] B. Beizer (1990). Software Testing Techniques. Second Edition, Van Nostrand Reinhold, New York.

[2] H. D. Mills, M. D. Dyer, and R. C. Linger (1987). Cleanroom Software Engineering. IEEE Software 4(5), pp. 19-25.

[3] J. M. Voas, L. Morell, and K. W. Miller (1991). Predicting where Faults Can Hide From Testing. IEEE, 8(2), pp. 41-48.

[4] P. Thévenod-Fosse, H. Waeselynck (1993). STATEMATE: Applied to Statistical Software Testing. ACM SIGSOFT Proceedings of the 1993 International Symposium on Software Testing and Analysis, Software Engineering Notes 23(2), pp. 78-8.

[5] R. S. Boyer, B. Elspas, and K. N. Levitt (1975). SELECT - a Formal System for Testing and Debugging Programs by Symbolic Execution. Proceedings of the International Conference on Reliable software, pp. 234-24.

[6] L. A. Clarke (1976). A System to Generate Test Data and Symbolically Execute Programs. IEEE Transactions on Software Engineering, 2(3), pp. 215-222.

[7] J. C. King (1976). Symbolic Execution and Program Testing. Communications of the ACM, 19 (7), pp. 385-394.

[8] W. E. Howden (1977). Symbolic Testing and the DISSECT Symbolic Evaluation System. IEEE Transactions on Software Engineering, 3(4), pp. 266-278.

[9] T. E. Lindquist, and J. R. Jenkins (1988). Test-Case Generation with IOGen. IEEE Software, 5 (1), pp. 72-79.

[10] M. R. Girgis (1993). Using Symbolic Execution and Data Flow Criteria to Aid Test Data Selection. The Journal of Software Testing, Verification and Reliability, 3(2), pp. 101-112.

[11] B. Korel (1990). Automated Software Test Data Generation." IEEE Transactions on Software Engineering, 16(8), pp. 870-879.

[12] R. Ferguson and B. Korel (1996). The Chaining Approach for Software Test Data Generation." ACM TOSEM, vol. 5, no. 1, pp. 63-86.

[13] Min Pei, E. D. Goodman, Zongyi Gao, and Kaixiang Zhong (1994). Automated Software Test Data Generation Using a Genetic Algorithm" Technical Report GARAGe of Michigan State University.

[14] M. Roper, I. Maclean, A. Brooks, J. Miller, and M. Wood (1995). Genetic Algorithms and the Automatic Generation of Test Data. Technical report RR/95/195[EFoCS-19-95].

[15] R. P. Pargas, M. J. Harrold, R. R. Peck (1999). Test Data Generation Using Genetic Algorithms" Journal of Software Testing, Verifications, and Reliability, vol. 9, pp. 263-282.

[16] Jin-Cherng Lin and Pu-Lin Yeh (2000). Using Genetic Algorithms for Test Case Generation in Path Testing. Proceedings of the 9th Asian Test Symposium (ATS'00).

[17] C. C. Michael, G. E. McGraw, M. A. Schatz (2001). Generating Software Test Data by Evolution. IEEE Transactions on Software Engineering, vol.27, no.12, pp. 1085-1110.

[18] Paulo Marcos Siqueira Bueno and Mario Jino (2002). Automatic Test Data Generation for Program Paths Using Genetic Algorithms" International Journal of Software Engineering and Knowledge Engineering, vol. 12, no. 6, pp 691-709.

[19] M. R. Girgis (2005). Automatic Test Data Generation for Data Flow Testing Using a Genetic Algorithm. Journal of Universal computer Science, vol. 11, no. 5, pp. 898-915.

[20] A. S. Ghiduk, M. J. Harrold, M. R. Girgis (2007). "Using genetic algorithms to aid test-data generation for data flow coverage," Proc. of 14th Asia-Pacific Software Engineering Conference (APSEC 07), pp. 41-48. IEEE Press.

[21] M. Harman (2007). The current state and future of search based software engineering. Proc. of the International Conference on Future of Software Engineering (FOSE'07), pp. 342-357. IEEE Press.

[22] Baresel A, Sthamer H, Schmidt M. Fitness (2002). Function Design to Improve Evolutionary Structural Testing. In proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), pp. 1329-1336, New York, USA.

[23] P. McMinn (2004). Search-based Software Test Data Generation: A Survey. Journal of Software Testing Verification and Reliability, vol.14, no.2, pp.105-156.

[24] A S. Ghiduk (2009). Search-Based Testing Guidance Using Dominances vs. Control Dependencies. 16th Asia-Pacific Software Engineering Conference apsec2009, pp.145-151. IEEE Press.

[25] J. Holland (1975). Adaptation in Natural and Artificial Systems, ISBN 0 472 08460 7. University of Michigan Press, Ann Arbor, MI.

[26] M. Srinivas, L. M. Patnaik, Genetic Algorithms: a Survey, IEEE Computer, 27 (6), 17-26, 1994.

[27] M. S. Hecht (1977). Flow Analysis of Computer Programs, Elsevier North Holland, New York.

[28] T. Lengauer and R. E. Trajan (1979). A Fast Algorithm for Finding Dominators in a Flowgraph. ACM Transactions on programming Languages and Systems, vol. 1, pp. 121-141.

[29] Z. Michalewicz (1999). Genetic Algorithms + Data Structures = Evolution Programs, 3rd Edition, Springer.

Appendix A

A part of the result of applying the system to test requirement number 5 (statement 27).

```
Population Size:  4
Maximum Number of Generation:  100
Crossover Probability:  0.80
Mutation Probability:  0.15
Number of Input Variables:  3
Domain and Precession of Input Variables:
1..5, 0; 1..5, 0;          1..5, 0
** GA Started **
 ----------------------------------------------------------------
 Test Requirement No. 5 is Statement: 27
 Its Dominance Path is: -1 1 2 3 4 5 6 7 8 22 23 24 25 26 27
 ----------------------------------------------------------------
*** Generation 1
 * ------------
*** Initial Population
 * Individual   1 =   2,  2,  3 = 001000100011
 * Individual   2 =   1,  1,  3 = 000100010011
 * Individual   3 =   1,  1,  2 = 000100010010
 * Individual   4 =   1,  3,  4 = 000100110100
 *
*** Evaluation of the Population
 *
 * Individual 1:
 * Traversed Path: -1 1 2 3 4 5 6 7 8 9 10 14 15 16 17 18 19 20 21 36 37 0
 * Uncovered Dominator Nodes: 22 23 24 25 26 27
 * Fitness Value: 0.600
 * Individual 2:
 * Traversed Path: -1 1 2 3 4 5 6 7 8 9 10 14 15 16 17 18 19 20 21 36 37 0
 * Uncovered Dominator Nodes: 22 23 24 25 26 27
 * Fitness Value: 0.600
 * Individual 3:
 * Traversed Path: -1 1 2 3 4 5 6 7 8 9 10 14 15 16 17 18 19 20 21 36 37 0
 * Uncovered Dominator Nodes: 22 23 24 25 26 27
 * Fitness Value: 0.600
 * Individual 4:
 * Traversed Path: -1 1 2 3 4 5 6 7 8 9 10 11 12 13 21 36 37 0
 * Uncovered Dominator Nodes: 22 23 24 25 26 27
 * Fitness Value: 0.600
 * ------------------------------------------------
*** Generation 2
 * ------------
*** 1- Selection
 *
 * The Selection Preformed using Roulette Wheel depended on Cumulative Fitness
 * The Selected Cases to be Parents of New Population are:
 * Parent   1 = Individual   1 =   2,  2,  3 = 001000100011
 * Parent   2 = Individual   3 =   1,  1,  2 = 000100010010
 * Parent   3 = Individual   2 =   1,  1,  3 = 000100010011
 * Parent   4 = Individual   3 =   1,  1,  2 = 000100010010
 *
*** 2- Recombination
 *
 * 2.1- Crossover
 * The Crossover Operation (Single Point Crossover) ***
 * Selected Parents  Crossover Position          Offsprings
 *    1 , 2                10                    000100010011    001000100010
 *    3 , 4                10                    000100010011    000100010010
 *
*** 2.2- Mutation
 * The Mutation Operation (Simple Mutation) ***
 * Selected Chromosome  Mutation Position   Mutated Chromosome
 *        1                      2                 010100010011
 *
*** The New Population is:
 * Individual   1 =   5,  1,  3 = 010100010011
 * Individual   2 =   2,  2,  2 = 001000100010
 * Individual   3 =   1,  1,  3 = 000100010011
 * Individual   4 =   1,  1,  2 = 000100010010
 *
*** Pre_Evaluation of the Population before adaptation to check is one of the out of range individuals
 * satisfies the test requirement or not, and keep the optimal
 *   2,  2,  2 is a test case covers the test requirement.
 *
*** Check Range
 * Is the generated data locate in the specified range?
 * Yes, all generated data locates in the specified range.
 *
*** 3- Evaluation of the Population
 *
 * Individual 1:
 * Traversed Path: -1 1 2 3 4 5 6 7 8 9 10 14 15 16 20 21 36 37 0
 * Uncovered Dominator Nodes: 22 23 24 25 26 27
 * Fitness Value: 0.600
 * Individual 2:
 * Traversed Path: -1 1 2 3 4 5 6 7 8 22 23 24 25 26 27 35 36 37 0
 * Uncovered Dominator Nodes:
 * Fitness Value: 1.000
 * Individual 3:
 * Traversed Path: -1 1 2 3 4 5 6 7 8 9 10 14 15 16 17 18 19 20 21 36 37 0
 * Uncovered Dominator Nodes: 22 23 24 25 26 27
 * Fitness Value: 0.600
 * Individual 4:
 * Traversed Path: -1 1 2 3 4 5 6 7 8 9 10 14 15 16 17 18 19 20 21 36 37 0
 * Uncovered Dominator Nodes: 22 23 24 25 26 27
 * Fitness Value: 0.600
 *
*** Elitist: If the best member of the current generation is worse than the best member of the previous generation we exchange them, and the best
 * member of the current generation would replace the worst member of the current population.
 *
*** The New Population is:
 * Individual   1 =   5,  1,  3 = 010100010011
 * Individual   2 =   2,  2,  2 = 001000100010
 * Individual   3 =   1,  1,  3 = 000100010011
 * Individual   4 =   1,  1,  2 = 000100010010
*********************************Report ****************************
** Best Fitness is:  1.000                ** Average Fitness is:  0.700
** Standard deviation is:  0.200          ** No. of Generations =   2
** The Test Requirement is satisfied and The Generated Test Case is:   2,  2,  2
** see individual 2 and its evaluation.
*****************************************************************
```

Using Meta-Structures in Database Design

Hui Ma
Victoria University of Wellington, School of Engineering and Computer Science
Wellington, New Zealand
E-mail: hui.ma@ecs.vuw.ac.nz

René Noack
Christian-Albrechts-University Kiel, Department of Computer Science
Kiel, Germany
E-mail: noack@is.informatik.uni-kiel.de

Klaus-Dieter Schewe
Software Competence Center Hagenberg, Hagenberg, Austria
E-mail: kd.schewe@scch.at

Bernhard Thalheim
Christian-Albrechts-University Kiel, Department of Computer Science
Kiel, Germany
E-mail: thalheim@is.informatik.uni-kiel.de

Keywords: database design, schema algebra, meta-structures, components, graph rewriting

Received: November 4, 2010

Practical experience shows that the design of very large database schemata causes severe problems, and no systematic support is provided. In this paper we address this problem. We define an Entity-Relationship schema algebra, which permits the representation of very large database schemata by algebraic expressions involving smaller schemata. Similar to abstraction mechanisms found in semantic data models the schema constructors can be classified into three groups for building associations and collections of subschemata, and for folding subschemata. Furthermore, based on the analysis of a large number of very large database schemata we identify twelve frequently recurring meta-structures in three categories associated with schema construction, lifespan and context. In combination with the schema algebra the meta-structures permit a component-based approach to database schema design, which can further be formalised by graph-rewriting.

Povzetek: Predstavljena je nova shema entitet in relacij za velike podatkovne baze.

1 Introduction

While data modellers learn about data modelling by means of small "toy" examples, the database schemata that are developed in practical projects tend to become very large. For instance, the relational SAP/R3 schema contains more than 21,000 tables. Moody discovered that as soon as ER schemata exceed 20 entity- and relationship types, they already become hard to read and comprehend for many developers [10].

Therefore, the common observation that very large database schemata are error-prone, hard to read and consequently difficult to maintain is not surprising at all. Common problems comprise repeated components as e.g. in the LH Cargo database schema with respect to transport data or in the SAP/R3 schema with respect to addresses.

Some remedies to the problem have already been discussed in previous work of some of the authors, and applied in some database development projects. For instance, modular techniques such as *design by units* [18] allow schemata to be drastically simplified by exploiting principles of hiding and encapsulation that are known from Software Engineering. Different subschemata are connected by bridge types. *Component engineering* [12] extends this approach by means of view-centered components with well-defined composition operators, and *hierarchy abstraction* [20] permits to model objects on various levels of detail.

In order to contribute to a systematic development of very large schemata the *co-design* approach, which integrates structure, functionality and interactivity modelling, emphasises the initial modelling of skeletons of components, which is then subject to further refinement

[21]. Thus, components representing subschemata form the building blocks, and they are integrated in skeleton schemata by means of connector types, which commonly are modelled by relationship types.

In this article we further develop the method for systematic schema development focussing on very large schemata. In Section 2 we first present an algebra for higher-order Entity-Relationship schemata [18], which permits the representation of very large schemata as algebraic expressions involving smaller and thus easier tractable schemata. Similar to abstraction mechanisms found in semantic data models [17] only three main groups of constructors are needed: *association* constructors that are used to combine schemata in a way that allows the original schemata to be regained, *folding* constructors that integrate schemata into a compact form, and *collection* constructors that deal with recurring similar subschemata. This extends our previous conference publication [7]. In particular, we permit handling schemata with constraints, and extend the description of the semantics of the operations.

In an extended theoretical study in [8] we develop a formal notion of schema morphisms, show that the corresponding category of schemata with these morphisms is finitely complete and co-complete, and also show that the algebra in this paper is well-defined and complete in the sense that all operators give rise to canonical morphisms, and all finite limits and co-limits can be expressed by the algebra. This complements our work reported in this article, which is devoted to the practical usage of the algebra for dealing with meta structures in the design of huge database schemata.

In Section 3, based on the analysis of more than 8500 database schemata, of which around 3500 should be considered very large we identify twelve frequently recurring meta-structures. These meta-structures are classified into three categories addressing schema construction, lifespan and context. This presentation polishes and extends another previous conference publication on the subject [9].

Finally, in Section 4 we address how meta-structures in combination with the schema algebra can be exploited for systematic, component-based database schema design. We analyse skeletons and subschemata more deeply and identify distinguishing dimensions [3]. Then we sketch how graph-rewriting can be used to support the design process extending and formalising existing approaches such as *design-by-units* [18], *string-bag modelling* [22], and *incremental structuring* [11].

2 An Entity-Relationship Schema Algebra

In the following we first present the gist of the Entity-Relationship model as our basis for schema design following [18]. On this basis we then describe three groups of schema constructors dealing with associations, folding, and collections of schemata. This defines a (partial) schema algebra, as constructors are only applicable, if certain preconditions are satisfied. The composition operators presented in this section will permit the construction of any schema of interest, as they mimique all set operations similar to the structural approach in [1].

2.1 Entity-Relationship Schemata

Let us briefly review the key definitions of Entity-Relationship schemata following [18]. We adopt the possibility to have higher-order relationship types and clusters, but for simplicity we disregard complex attributes, as attributes will be preserved by the schema constructors.

Thus, let $\mathcal{U}$ be a set of *attributes*. Each attribute $A \in \mathcal{U}$ is associated with a set $dom(A)$ of values called the *domain* of A.

An *entity type* (or *type of level* 0) E is defined by a finite set $attr(E) \subseteq \mathcal{U}$ of attributes and a key $k(E) \subseteq attr(E)$. The definition of an *entity* of type E is straighforward. It can be represented as a tuple $(A_1 : v_1, \ldots, A_n : v_n)$ for $attr(E) = \{A_1, \ldots, A_n\}$ and $v_i \in dom(A_i)$ for all $i = 1, \ldots, n$. An *entity set* of type E is a finite set $\{e_1, \ldots, e_m\}$ of entities of type E, such that whenever the projections $e_i[k(E)]$ and $e_j[k(E)]$ on the key coincide, then $e_i = e_j$ holds.

An *entity cluster* (or *cluster of level* 0) C is defined by a finite set $\{\ell_1 : E_1, \ldots, \ell_k : E_k\}$ with pairwise different labels ℓ_i and entity types $E_1, \ldots, E_k$ (not necessarily different). A *cluster set* of type C is defined is a labelled disjoint union $\{\ell_i : v_i \mid v_i \in \mathcal{S}(E_i)\}$ with entity sets $\mathcal{S}(E_i)$ of type E_i $(i = 1, \ldots, k)$.

A *relationship type* R *of order* $k + 1$ (or simply a *type of level* $k + 1$ with $k \geq 0$) is defined by a finite set $comp(R) = \{r_1 : R_1, \ldots, r_k : R_k\}$ with pairwise distinct role labels r_i and types or clusters R_i of level at most k, such that at least one R_i has level exactly k, a finite set $attr(R) \subseteq \mathcal{U}$ of attributes and a key $k(R) \subseteq comp(R) \cup attr(R)$. A *relationship* of type R can be represented as a tuple $(r_1 : e_1, \ldots, r_k : e_k, A_1 : v_1, \ldots, A_n : v_n)$ for $attr(R) = \{A_1, \ldots, A_n\}$ with entities or relationships e_i of type R_i, respectively, and $v_i \in dom(A_i)$.

A *cluster of level* k is defined analogously to an entity cluster with the only difference that the participating types must be of level at most k, and one of them must have level exactly k.

As an entity type E can be identified with a relationship type with an empty set of components, i.e. $comp(E) = \emptyset$, we will dispense with the separation and simply talk about types. Types of level 0 are entity types, and types of level $k > 0$ are relationship types.

An *Entity-Relationship schema* $\mathcal{S}$ (*ER-schema* for short) is a finite set of types and clusters such that, whenever $R \in \mathcal{S}$ is a relationship type, i.e. a type of level $k > 0$, and $E \in comp(R)$ is one of its components, then we must have also $E \in \mathcal{S}$, and whenever $C \in \mathcal{S}$ is a cluster, then also all types participating in C must be in $\mathcal{S}$.

If $\mathcal{S}$ is an ER-schema, a *database* over $\mathcal{S}$ is defined by

entity, relationship, and cluster sets $\mathcal{S}(R)$, respectively, for all $R \in \mathcal{S}$ such that, whenever $R \in \mathcal{S}$ is a relationship type, $r_i : R_i \in comp(R)$ is one of its components, and $(r_1 : e_1, \ldots, r_k : e_k, A_1 : v_1, \ldots, A_n : v_n) \in \mathcal{S}(R)$, then $e_i \in \mathcal{S}(R_i)$ holds, and similarly for a cluster $C = \{\ell_1 : R_1, \ldots, \ell_k : R_k\}$ we must have $\mathcal{S}(C) = \{\ell_i : e_i \mid e_i \in \mathcal{S}(R_i)\}$. Furthermore, whenever the projections $t_1[k(R)]$ and $t_2[k(R)]$ of relationships $t_1, t_2 \in \mathcal{S}(R)$ to the key $k(R)$ for a relationship type $R \in \mathcal{S}$ coincide, then already $t_1 = t_2$ holds.

In addition to the structural information that is provided by an ER-schema, a schema is usually extended by a set Σ of *integrity constraints*. These are first order formulae defined over the types and clusters in $\mathcal{S}$. In case of an extended schema $(\mathcal{S}, \Sigma)$ a database must further satisfy the constraints in Σ. In our presentation of the schema algebra constructors we will mainly deal with the schema, and the handling of integrity constraints will only mentioned briefly. If the applicability of a constructor depends on the presence of some constraints, we will mention this separately.

In the following we will commonly use the graphical representation of an ER-schema $\mathcal{S}$ by a directed graph with vertices defined by $\mathcal{S}$ and directed edges from a relationship type to all its components (labelled by the roles if necessary), as well as directed edges from a cluster type to its participating types (also labelled by the labels, if necessary). For convenience, entity types are represented by rectangles, relationship types by diamonds, and clusters by circles marked with a $+$. Attributes are usually attached to types or omitted, and keys are emphasized in some way, e.g. underlining attributes in the key and marking components in the key. We usually refer to the graphical representation of an ER-schema as an *ER-diagram*. Constraints are not indicated in ER-diagrams.

Sometimes we like to emphasize a distinguished *root* in an ER-schema. In case there is a type (or cluster) from which all other types and cluster can be reached by following the edges in the ER-diagram, this type is of course a natural choice for the root. In general, however, such a type does not exist, but there may be several types (or clusters) that cannot be reached from any other type or cluster by following component edges. Each of these types/clusters can be used as root of the schema.

2.2 Renaming

As the names of types and clusters in ER-schemata must be unique, we must avoid name clashes when applying the schema constructors. Therefore, we have to provide a renaming constructor. For this, if $R_1, \ldots, R_k$ and $R'_1, \ldots, R'_k$ are pairwise distinct sequences of names, a *renaming* is a mapping $\{R_1 \mapsto R'_1, \ldots, R_k \mapsto R'_k\}$. If $(\mathcal{S}, \Sigma)$ is an ER-schema, then replacing each occurrence of R_i in $\mathcal{S}$ and Σ by R'_i results in the schema

$$\varrho_{R_1 \mapsto R'_1, \ldots, R_k \mapsto R'_k}(\mathcal{S}, \Sigma).$$

2.3 Association Constructors

We disitinguish two kinds of association constructors: constructors that lead to schemata, into which the original schemata can be embedded as subschemata, and constructors that lead to schemata that can be projected onto the original schemata.

2.3.1 Sum and Join

The simplest form of a composition through association is by means of a direct sum, i.e. disjoint union constructor. More generally, we consider joins of two schema along input- and output-views [12]. For this let $(\mathcal{S}_i, \Sigma_i)$ be a schema with two subschemata $\mathcal{I}_i \subseteq \mathcal{S}_i$ called *input view*, and $\mathcal{O}_i \subseteq \mathcal{S}_i$ called *output-view* ($i = 1, 2$). We request that $\mathcal{I}_i$ and $\mathcal{O}_j$ for $i = 1, j = 2$ or $i = 2, j = 1$ are isomorphic in a purely graph-theoretic sense (not as in [8]), i.e. there exists a graph-isomorphism $\sigma : \mathcal{I}_i \to \mathcal{O}_j$.

The *join schema*

$$\mathcal{S} = \mathcal{S}_1 \bowtie_{\mathcal{I}_1 := \mathcal{O}_2 \| \mathcal{I}_2 := \mathcal{O}_1} \mathcal{S}_2$$

results from the two given schemata by identifying in $\mathcal{S}_1 \cup \mathcal{S}_2$ the input-view of first schema with the output-view of the second one and vice versa. That is, we rename $\mathcal{S}_i$ in a way that the subschemata $\mathcal{I}_1$ and $\mathcal{O}_2$ (and likewise $\mathcal{I}_2$ and $\mathcal{O}_1$) become identical, while all other types are different, and then build the union. Attribute sets for types that are identified are merged by using set union.

Furthermore, if a type the subschemata $\mathcal{I}_1$ and $\mathcal{O}_2$ has components outside the subschema, these components will be preserved in the join. This applies analogously to $\mathcal{I}_2$ and $\mathcal{O}_1$. This may have the effect that an entity-type in one of the views becomes a relationship type in the join schema. The set Σ of constraints on $\mathcal{S}$ is defined by the union $\Sigma_1 \cup \Sigma_2$ after the renaming. In this way the original schemata $\mathcal{S}_i$ become subschemata of the join schema $\mathcal{S}$, and consequently, each database over $\mathcal{S}$ can be mapped to a database over $\mathcal{S}_i$.

The join with empty input-and output views is the *direct sum* $\mathcal{S}_1 \oplus \mathcal{S}_2$. The join of the schemata $\mathcal{S}_1$ and $\mathcal{S}_2$ along the input- and output-views shown in Figure 1 is the schema $\mathcal{S}$ shown in the same figure. We omitted all attributes, as these are preserved by the join.

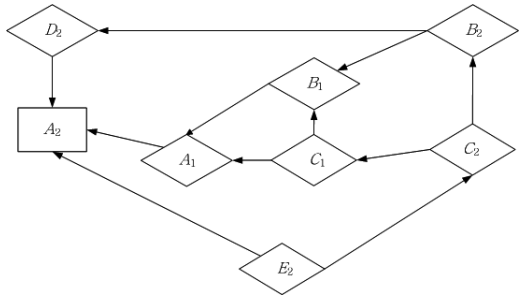


Figure 2: The reference-join on two schemata.

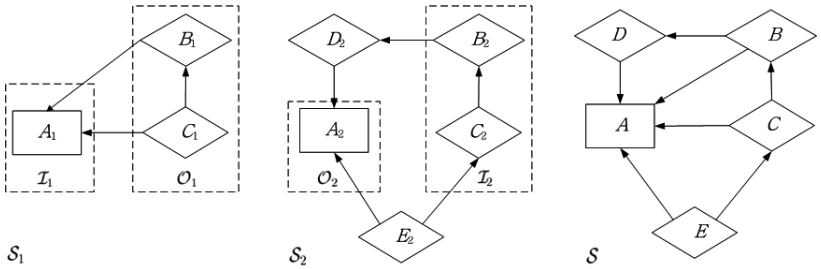


Figure 1: The join operator on two schemata.

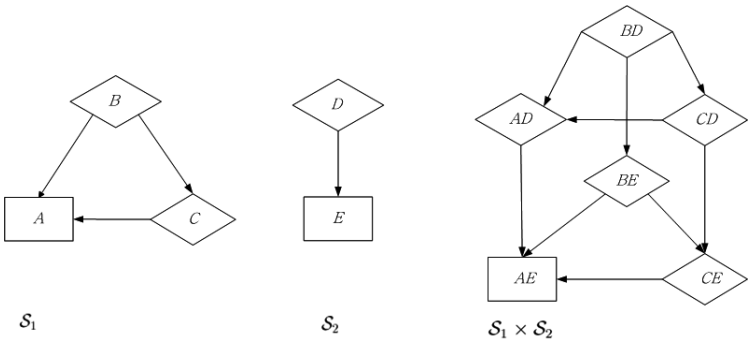


Figure 3: The product operator on two schemata.

A variant of the join operator is provided by means of a *reference-join*. The prerequisites are the same as for the join operator, only that we need that the set of type names of both schemata are disjoint. In this case, however, the output-views $\mathcal{O}_i$ $(i = 1, 2)$ of the original schemata are preserved within the resulting schema

$$\mathcal{S} = \mathcal{S}_1 \bowtie_{\mathcal{I}_1 \to \mathcal{O}_2 \| \mathcal{I}_2 \to \mathcal{O}_1} \mathcal{S}_2$$

and references from the types in $\mathcal{I}_i$ to those in $\mathcal{O}_j$ for $(i, j) = (1, 2)$ or $(2, 1)$ are added. This requires that entity-types in the input-views be turned into relationship types. The schema shown in Figure 2 shows the result of the reference-join of the schemata $\mathcal{S}_1$ and $\mathcal{S}_2$ from Figure 1. In this case, the set of constraints Σ associated with $\mathcal{S}$ is simply defined by the union $\Sigma = \Sigma_1 \cup \Sigma_2$.

Another variant can be obtained, when cooperating views [18] are employed instead of merging input- and output-views or letting the former ones reference the latter ones. In this case the data exchange has to be specified explicitly by means of operations. As we neglected operations in our model, we have to discard this alternative for the presentation here.

2.3.2 Product and Meet

Dual to the sum constructor we can define a product constructor. In this case let $(\mathcal{S}_i, \Sigma_i)$ be schemata with disjoint name sets $(i = 1, 2)$. For types $R_i \in \mathcal{S}_i$ defined as $(comp(R_i), attr(R_i), k(R_i))$ $(i = 1, 2)$ define their product $R_1 \times R_2$ by the type

$$R_{1,2} = (comp(R_1) \times \{R_2\} \cup \{R_1\} \times comp(R_2), \\ attr(R_1) \cup attr(R_2), k(R_{12})),$$

i.e. if $comp(R_i) = \{r_{i1} : R_{i1}, \ldots, r_{ik_i} : R_{ik_i}\}$, we obtain $comp(R_{12}) = \{r_{11} : R_{11,2}, \ldots, r_{1k_1} : R_{1k_1,2}, r_{21} : R_{1,21}, \ldots, r_{2k_2} : R_{1,2k_2}\}$, and the key $k(R_{12})$ is defined as $\{r_{1j} : R_{1j,2} \mid r_{1j} : R_{1j} \in k(R_1)\} \cup \{r_{2j} : R_{1,2j} \mid r_{2j} : R_{2j} \in k(R_2)\} \cup \{A \mid A \in attr(R_1) \cap k(R_1)\} \cup \{A \mid A \in attr(R_2) \cap k(R_2)\}$.

If R_1 is a cluster, say $R_1 = \{\ell_1 : R_{11}, \ldots, \ell_{k_1} : R_{1k_1}\}$, and R_2 is a type as before, then their product is the cluster

$$R_1 \times R_2 = \{\ell_1 : R_{11,2}, \ldots, \ell_{k_1} : R_{1k_1,2}\}.$$

The product of a type R_1 and a cluster R_2 is defined analogously. Finally, if both R_1 and R_2 are clusters, say $R_i = \{\ell_{i1} : R_{i1}, \ldots, \ell_{ik_i} : R_{ik_i}\}$ for $i = 1, 2$, then their product is the cluster

$$R_1 \times R_2 = \\ \{\ell_{1j_1,2j2} : R_{1j_1,2j_2} \mid 1 \leq j_1 \leq k_1,\ 1 \leq j_2 \leq k_2\}.$$

The *product schema* is defined as

$$\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 = \{R_1 \times R_2 \mid R_1 \in \mathcal{S}_1,\ R_2 \in \mathcal{S}_2\}.$$

Of course, in all cases we have to create new names for the new types (or clusters) $R_{i,j} = R_i \times R_j$, and also new names for labels in the clusters and roles in the components. Figure 3 shows the product $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2$ of the schemata in the same figure. We omitted all attributes.

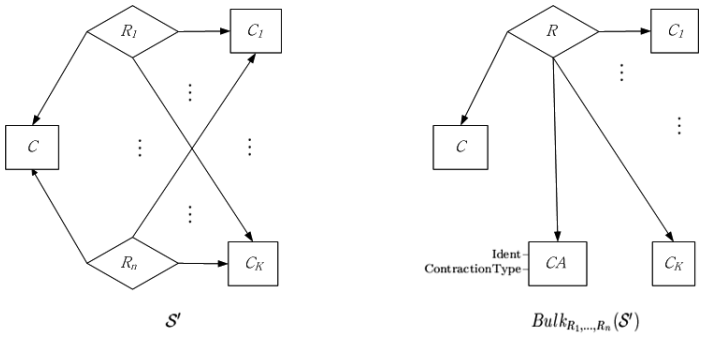


Figure 4: The bulk operator on a database schema.

Each product type (or cluster) $R_1 \times R_2 \in \mathcal{S}_1 \times \mathcal{S}_2$ contains the roles and attributes from R_1 and R_2, and thus defines projections $R_1 \times R_2[R_i]$ for $i = 1, 2$. Thus, whenever a constraint in Σ_i refers to a type R, this type may be replaced by any projection $R \times R'[R]$ (or $R' \times R[R]$, respectively). Taking all the constraints defined in this way defines the set of constraints $\hat{\Sigma}_i$, and the set of constraints on $\mathcal{S}$ is defined by the union $\Sigma = \hat{\Sigma}_1 \cup \hat{\Sigma}_2$.

In this way, similar to the case of the join-operator, a database over a product schema $\mathcal{S}_1 \times \mathcal{S}_2$ can be projected to a database over the original schemata $\mathcal{S}_i$ $(i = 1, 2)$.

We can also define a dual *meet constructor* $\bullet_\varphi$ for the join constructor. In this case we need an additional *matching condition* φ, and we define the meet schema as

$$\mathcal{S} = \mathcal{S}_1 \bullet_\varphi \mathcal{S}_2 = \{R_1 \times R_2 \mid R_1 \in \mathcal{S}_1,\ R_2 \in \mathcal{S}_2 \text{ with } \varphi(R_1, R_2)\}.$$

Matching conditions can express requirements such as common attributes or inclusion constraints.

2.4 Folding and Unfolding of Schemata

As observed in [9] similar subchemata can be integrated by replacing a number of relationship types by a new relationship type plus an additional entity type. For this assume we have a schema $\mathcal{S}'$ with a central entity (or relationship) type C, and n relationship types $R_1, \ldots, R_n$ that all relate C to a number $C_1, \ldots, C_k$ of entity or relationship types as shown in the left hand part of Figure 4.

Then we can replace $R_1, \ldots, R_n$ by a new relationship type R with a new additional component CA. This type must have an attribute "ContractionType" with domain $\{1, \ldots, n\}$ that will be used to identify the original relation. It may further be advisable to add an identifying attribute "Ident".

The schema $\mathcal{S} = Bulk_{R_1,\ldots,R_n}(\mathcal{S}')$ resulting from applying this *bulk* operator is illustrated in the right hand part of Figure 4. With respect to integrity constraints in Σ each occurrence of a type R_i has to be replaced by the projection $R[C, C_1, \ldots, C_k]$ and the condition $R.CA.\text{ContractionType} = R_i$ has to be added.

The bulk constructor $Bulk_{R_1,\ldots,R_n}$ can be refined to better handle attributes that are not common to all types $R_1, \ldots, R_n$. Such an attribute A becomes an "optional" attribute of the type CA, i.e. its domain will be defined as $dom_{\mathcal{S}}(A) = dom_{\mathcal{S}'}(A) \cup \{\mathit{undef}\}$. If A is not an attribute of the type R_i, the constraint

$$CA.\text{ContractionType} = R_i \Rightarrow CA.A = \mathit{undef}$$

has to be added to Σ.

Semantically, it is easy to see how databases over $\mathcal{S}'$ are mapped onto databases over $\mathcal{S} = Bulk_{R_1,\ldots,R_n}(\mathcal{S}')$. We get $\mathcal{S}(C) = \mathcal{S}'(C)$ and $\mathcal{S}(C_i) = \mathcal{S}'(C_i)$ for $i = 1, \ldots, k$, $\mathcal{S}(CA) = \{(\text{Ident} : i, \text{ContractionType} : R_i \mid i = 1, \ldots, n\}$, and $\mathcal{S}(R) = \bigcup_{i=1}^{n} \{\hat{t}_i \mid t_i \in \mathcal{S}'(R_i)\}$, where the tuple $\hat{t}_i$ results from t_i by adding the role $(CA : i)$.

The *expansion* constructor $Expand_{E:A}$ is inverse to the bulk constructor. In this case we need an entity type E with $k(E) = \{ident\}$, and an attribute $A \in attr(E) - k(E)$ with a finite enumeration domain $dom(A) = \{v_1, \ldots, v_n\}$. Furthermore, there must be a unique relationship type $R \in \mathcal{S}$ with a component E occurring once, i.e. $r : E \in comp(R)$, and for all r' and all R' with $r' : E \in comp(R')$ we must have $R' = R$ and $r' = r$.

In the resulting schema $Expand_{E:A}(\mathcal{S})$ the type R will be replaced by n types $R_1, \ldots, R_n$ corresponding to the values $v_1, \ldots, v_n$ of the attribute A. For each of these types we have $comp(R_i) = comp(R) - \{r : E\}$. Each attribute of R becomes an attribute of R_i, and each attribute $B \in attr(E) - \{ident,\ A\}$ is added as an attribute of R_i, unless Σ contains a constraint of the form above.

The mapping of databases over $\mathcal{S}$ to databases over $Expand_{E:A}(\mathcal{S})$ is just the inverse of the mapping for the bulk operator: "forget" $\mathcal{S}(CA)$ and split $\mathcal{S}(R)$ into n sets according to the value of the CA role.

Component nesting can be applied to a schema $\mathcal{S}_1$ to replace a component C of a type R by a complete subschema $\mathcal{S}_2$ that is rooted at a type T. Attributes, identifying components $I_1, \ldots, I_k$ and other components $C_1, \ldots, C_\ell$ of C will become components of the root type T of $\mathcal{S}_2$ within the new schema. We denote the schema $\mathcal{S}$ resulting from the application of the nesting operator by $nest_{C:\mathcal{S}_2(T)}(\mathcal{S}_1)$. Figure 5 illustrates the application of the nesting operator.

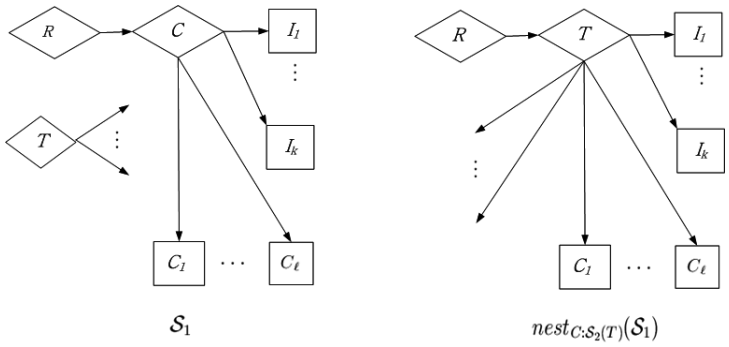


Figure 5: The nesting operator on a database schema.

If we define an input-view $\mathcal{I}_1 = \{C\}$ for $\mathcal{S}_1$, an output-view $\mathcal{O}_2 = \{T\}$ for $\mathcal{S}_2$, and let $\mathcal{O}_1 = \emptyset = \mathcal{I}_2$, then component nesting is actually a special case of a join. Component nesting is usually applied with a type C that has not yet been developed, i.e. it is an entity type in $\mathcal{S}_1$. It generalises entity model clustering, entity clustering, entity and relationship clustering, entity tree clustering in the design-by-units method [18].

2.5 Collection Constructors for Schemata

While all operators discussed so far have arity 1 or 2, the collection constructions apply to any number k of schemata. If $\mathcal{S}_1, \ldots, \mathcal{S}_k$ are schemata, we can build the *set schema* $\{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$, provided the element schemata are pairwise distinct, the *multiset schema* $\langle \mathcal{S}_1, \ldots, \mathcal{S}_k \rangle$, the *list schema* $[\mathcal{S}_1, \ldots, \mathcal{S}_k]$, and the *tree schema* $\langle\langle \mathcal{S}_1, \ldots, \mathcal{S}_k \rangle\rangle$.

As schemata, the result of the first three constructions can be identified with the sum, i.e. the join with empty views, of the element schemata, while a tree schema contains an additional relationship type with k components that are root types of the element schemata. Renaming has to be applied in all cases to avoid name clashes, and for constraint sets the union operator is used. As such, the collection constructions are only a mild extension.

However, they unfold their power by means of collection operators that can be applied to a set, multiset, list or tree schema $\mathcal{S}'$:

- $all\_of(\mathcal{S}')$ denotes the schemata that contains all schemata in the collection as subschemata. The construction can be used to specify that all $\mathcal{S}_1, \ldots, \mathcal{S}_k$ (or their root types, respectively) must appear as components in some other construction, e.g. in the bulk or nesting construction we discussed above.

- Similarly, $any\_of(\mathcal{S}')$ denotes one arbitrary element schema, and $n\_of(\mathcal{S}')$ denotes an arbitrary selection of n of the element schemata. Semantically, this leads to the disjoint union of databases, i.e. the original databases are embedded in the resultung databases after applying the operator.

- The selection of subschemata in the collection using any of the constructors all_of, any_of or n_of can be refined by adding selection criteria in form of a `where`-clause. For instance, $n\_of(\{\mathcal{S}_1, \ldots, \mathcal{S}_k\})$ `where` φ would select n of the element schemata among those satisfying the condition φ.

- $n\_th(\mathcal{S}')$ for a list or tree schema denotes the n'th element schema, provided $1 \leq n \leq k$ is satisfied.

As an example consider again the schema $\mathcal{S}'$ in Figure 4. If we define schemata $\mathcal{S}_i$ for $i = 0, \ldots, k$ to contain only one type – C_i for $i \neq 0$ and C for $i = 0$ – then we could define the types R_i as

$$R_i = (all\_of(\{\mathcal{S}_0, \ldots, \mathcal{S}_k\}), \mathcal{S}, \mathcal{K}),$$

i.e. the components are the (root) types in $\mathcal{S}_i$, while the set of attributes $\mathcal{A}$ and the keys $\mathcal{K}$ are specified elsewhere. Alternatively, if $\mathcal{A} = \emptyset$, we could define R_i as the root type in the schema $Tree(\mathcal{S}_0, \ldots, \mathcal{S}_k)$.

Similarly, the type R in the schema $\mathcal{S} = Bulk_{R_1,\ldots,R_n}(\mathcal{S}')$ can be defined as

$$R = \{CA\} \cup all\_of(\{\mathcal{S}_0, \ldots, \mathcal{S}_k\}), \mathcal{S}, \mathcal{K})$$

with the entity type $CA = (\{\text{Ident, ContractionType}\}, \{\text{Ident}\})$.

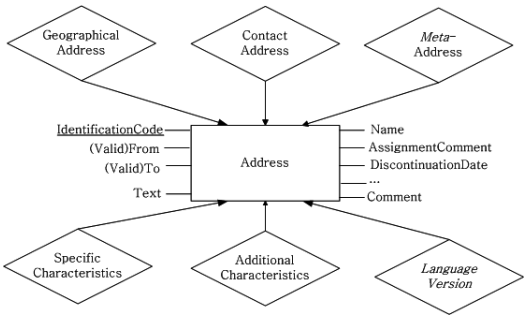


Figure 6: The General Structure of Addresses.

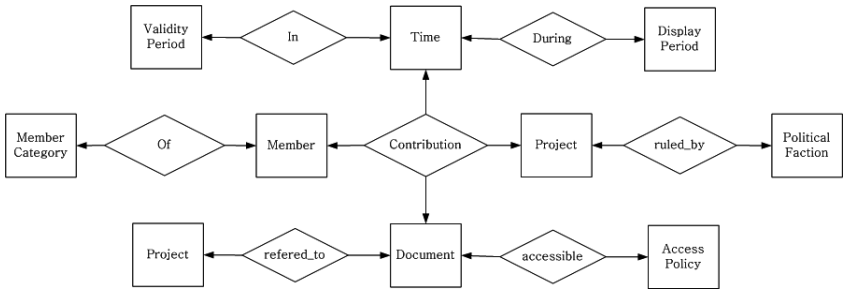


Figure 7: Snowflake Schema on Contributions.

3 Meta-Structures in Very Large Database Schemata

Based on an extensive study of a large number of conceptual database schemata – we analysed more than 8500 database schemata, of which around 3500 should be considered being very large – we identify frequently occurring meta-structures and classify them in three categories according to construction, lifespan and context. In the following we describe these meta-structures.

3.1 Construction Meta-Structures

Structures are based on building blocks such as attributes, entity types and relationship types. In order to capture also versions, variations, specialisations, application restrictions, etc. structures can become rather complex. As observed in [12, 14] complex structures can be primarily described on the basis of *star* and *snowflake meta-structures*. In addition, *bulk meta-structures* describing the similarity between things and thus enable generalisation and combination, and *architecture meta-structures* describe the internal construction by building blocks and the interfaces between them.

3.1.1 Star and Snowflake Meta-Structures

Star typing has been used already for a long time outside the database community. The star constructor permits to construct associations within systems that are characterized by complex branching, diversification and distribution alternatives. Such structures appear in a number of situations such as composition and consolidation, complex branching analysis and decision support systems.

A *star meta-structure* is characterized either by a (core) entity type E and a number of (peripheral) subtypes, i.e. unary relationship types R_i with $comp(R_i) = \{E\}$ $(i = 1, \ldots, n)$, or by a core (level 1) relationship type R together with its components, which are of course entity types. In the former case the core type is usually used for storing basic data, and the subtypes are used to capture additional properties [20]. Such a star structure is shown in Figure 6 with the entity type `Address` as its core. Taking the relationship type `Contribution` in Figure 7 as core type of a star schema, the subschema containing `Contribution`, `Member`, `Document`, `Project`, and `Time` defines another star structure.

We consider star structures as the simplest schemata, which naturally appear as subschemata of any conceptual schema. However, if the core type is an entity type, even a simple star schema can be written as the join of several schemata (in any order). For instance, the star schema in Figure 6 can be composed out of six small schemata, each consisting of the entity type `Address` and a single subtype such as `GeographicalAddress` or `ContactAddress`. For building the joins we always have to take the subschema $\{$`Address`$\}$ as input- and output-schemata, respectively.

A slighly more complicated meta-structure arises, if we take a star schema $\mathcal{S}_1$ and apply the nesting operator $nest_{C:\mathcal{S}_2(T)}$ with a type T in another star schema $\mathcal{S}_2$ to one of its peripheral types C. More generally, as nesting is a special case of the join-operator, we could apply the join $\bowtie_{\mathcal{I}_1:=\mathcal{O}_2 \| \mathcal{I}_2:=\mathcal{O}_1}$ with $\mathcal{I}_1$ containing several peripheral types of the star schema $\mathcal{S}_1$, $\mathcal{O}_2$ containing several types of another star schema $\mathcal{S}_2$, and $\mathcal{I}_2 = \mathcal{O}_1 = \emptyset$. This procedure may be applied repeatedly. In all these cases the result is called a *snowflake schema*.

For example, the snowflake schema in Figure 7 – for simplicity, attributes have been omitted – represents the information structure of documented contributions of members of working groups during certain time periods. In this case the schema result from extending the original star schema with core relationship type `Contribution` by means of nesting and join with six star schemata centred around the relationship types `In`, `During`, `Of`, `ruled_by`, `referred_to`, and `accessible`, respectively.

Star and snowflake schemata are common in data warehouses and OLAP systems [6].

3.1.2 Bulk Meta-Structures

A *bulk meta-structure* is represented by a schema that results from the application of the bulk-operator, i.e. $\mathcal{S} = Bulk_{R_1,\ldots,R_n}(\mathcal{S}')$. Thus, in a bulk structure types that are used in a very similar way are clustered together. Apply-

ing the expand-operator $Expand_{E:A}$ to the bulk structure $\mathcal{S}$ returns the original schema $\mathcal{S}'$. Thus, a bulk structure is merely a compacted representation for sructurally similar information.

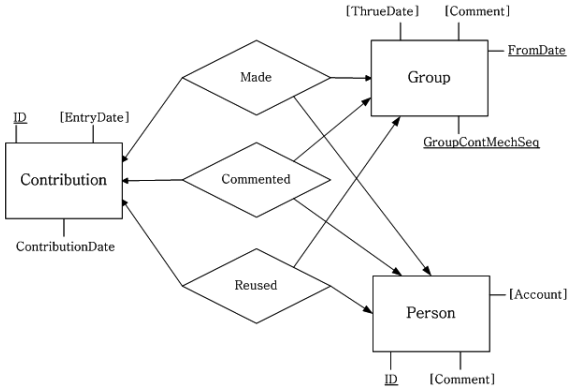


Figure 8: E-Community Application.

Let us exemplify this approach for the commenting process in an e-community application. The relationship types `Made`, `Commented`, and `Reused` in Figure 8 are all similar. They associate contributions with both `Group` and `Person`. They are used together and at the same objects, i.e. each contribution object is at the same time associated with one group and one person.

We can combine the three relationship types into the type `ContributionAssociation` as shown in Figure 8. The type `ContributionAssociationClassifier` and the domain {Made, Commented, Reused} for the attribute `ContractionDomain` can be used to reconstruct the three original relationship types. The handling of classes that are bound by the same behaviour and occurrence can be simplified by this construction.

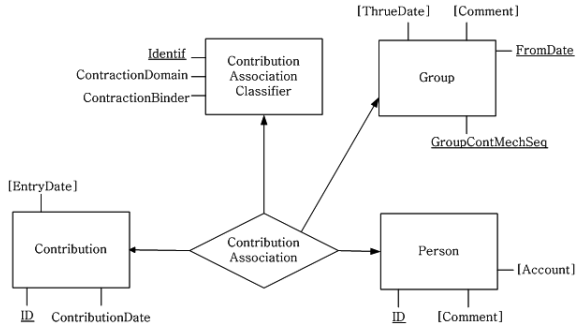


Figure 9: Bulk Meta-Structure for E-Community.

If $\mathcal{S}'$ denotes the schema in Figure 8, and $\mathcal{S}$ the one in Figure 9, we have

$$\mathcal{S} = Bulk_{\text{Made, Commented, Reused}}(\mathcal{S}') \qquad \text{and}$$
$$\mathcal{S}' = Expand_{\text{ContributionAssociationClassifier: ContractionDomain}}(\mathcal{S}) \ .$$

3.1.3 Architecture and Constructor-based Meta-Structures

Categorisation and compartment building have been widely used for modelling complex structures. For instance, the architecture of SAP R/3 has often been displayed in form of a waffle. That is, the schema is constructed out of several subschemata that are integrated by means of *bridge* or *binding schemata*. Technically, this integration is performed by means of joins involving two subschemata and their bridge schema.

We illustrate the building of a waffle structure in Figure 10. All subschemata are sketched by hexagons, and thebinding schemata are sketched as ovals.

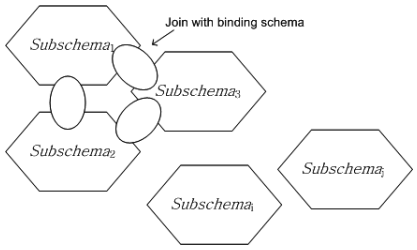


Figure 10: Waffle Meta-Structure.

Therefore, we adopt the term *waffle meta-structure* or *architecture meta-structure* for structures that arise this way. These meta-structures are especially useful for the modelling of distributed systems with local components and behaviour. They provide solutions for interface management, replication, encapsulation and inheritance, and are predominant in component-based development and data warehouse modelling.

3.2 Lifespan Meta-Structures

The evolution of an application over its lifetime is orthogonal to the construction. This leads to a number of *lifespan meta-structures*, which we describe next. *Evolution meta-structures* record life stages similar to workflows, *circulation* or *loop meta-structures* display the phases in the lifespan of objects, e.g. chaining and scaling to different perspectives of objects, *incremental meta-structures* permit the recording of the development, enhancement and ageing of objects, and *network meta-structures* permit the flexible treatment of objects during their evolution by supporting to pass objects in a variety of evolution paths and enable multi-object collaboration.

All these lifespan structures are determined by three dimensions: *expansion*, *seed*, and *feedback*. The expansion dimension captures the development of objects using a starting (entity) type that is stepwise expanded by relationship types as shown in Figure 11. Besides the added new relationship type in the i'th expansion step having the added type of the $(i-1)$'th expansion step as one of its components other types may be added to the schema and

identified with existing types. If the expansion dimension is the only one used, we obtain an incremental lifespan meta-structure as discussed below.

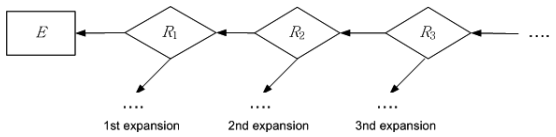


Figure 11: Expansion Dimension in Lifespan Meta-Structures.

The seed dimension captures the spreading of an objects into several related objects, thus producing a tree of types as illustrated in Figure 12. For instance, the entity type E may be `book`, and the relationship type R_1 may be `book_copy`. The technical difference to the expansion dimension is by means of participation cardinality constraints – these have been omitted in Figure 12. For expansion we have to request $card(R_i, R_{i-1}) = (0, 1)$ (with $R_0 = E$), i.e. for each entity of type E appears as a component of at most one relationship of type R_i ($i > 0$), which means that we deal with different lifespan versions of the same object. For seed the corresponding participation cardinality constraints are $card(R_i, R_{i-1}) = (1, \infty)$, i.e. each entity of type E spreads out into many relationships of type R_i ($i > 0$), which means that we do not deal with the same object, but with different levels of abstraction as book, book edition and book copy. If the seed dimension is the only one used, we obtain an evolution lifespan meta-structure as discussed below.

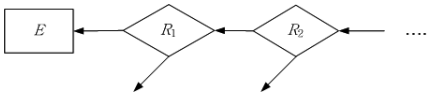


Figure 12: Seed Dimension in Lifespan Meta-Structures.

The feedback dimension captures the case of cyclic development as illustrated in Figure 13. In this case we will need relationship types linking the different stages. Alternatively, star or snowflake schemata could be used with a core entity type representing the developing object and the peripheral types modelling the various stages. If the feedback dimension is the only one used, we obtain a loop or circular lifespan meta-structure as discussed below.

For all three dimensions the basic meta-structure can be formalised by using the join-operator. This also applies, if the lifespan meta-structures is to be combined with a structural meta-structure, in which the type to be developed appears. Similarly, if several lifespan meta-structures appear together, this is reflected by the use of the product- and meet-operators. As circular (feedback) and incremental (expansion) cannot be combined, the only reasonable combined lifespan meta-structure is the network meta-structure, in which seed comes together with either incremental or loop development.

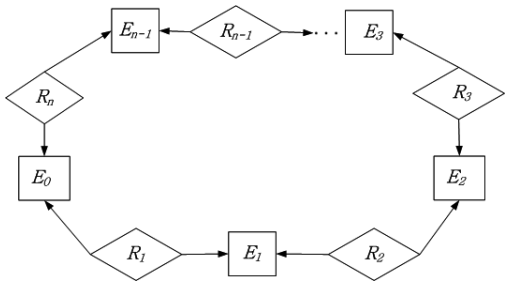


Figure 13: Feedback Dimension in Lifespan Meta-Structures.

3.2.1 Incremental Meta-Structures

Incremental meta-structures enable the production of new associations based on a core object. They employ containment, sharing of common properties or resources, and alternatives. Typical examples are found in applications, in which processes collect a range of inputs, generate multiple outcomes, or create multiple designs.

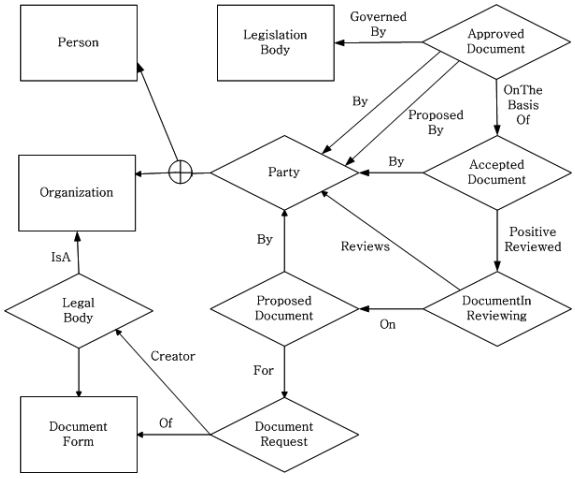


Figure 14: Incremental Meta-Structure.

Incremental development builds layers of an application with a focus on the transport of data and cooperation, thereby enabling the management of systems complexity. It is quite common that this leads to a multi-tier architecture and object versioning. Typical incremental constructions appear in areas such as facility management [4]. A special *layer constructor* is widely used in frameworks, e.g. the OSI framework for communicating processes.

As an example take the schema in Figure 14, which expands the single entity type E = DocumentForm in five steps. In the first step the relationship type R_1 = `DocumentRequest` is added, but for this also the types `LegalBody` and `Organization` are required. In the second step the relationship type

R_2 = ProposedDocument is added, which requires in addition the type Party, which is a cluster of Organization and the new type Person. In step three the added relationship type is R_3 = DocumentInReviewing, which links again to Party; in step four we have to add R_4 = AcceptedDocument, again linking to Party. In the final step the added relationship type is R_5 = ApprovedDocument with two additional roles to Party, and LegislationBody as additional fourth component.

The sequence $E, R_1, \ldots, R_5$ reflects the incremental development of a legal document in the e-governance application SeSAM. It uses a specific composition frame, i.e. the type DocumentInReviewing is based on the type ProposedDocument. Legal documents typically employ particular document patterns, which are represented by the type DocumentForm. Actors in this applications are of type Party, which generalises Person and Organisation.

Formally, an incremental lifespan meta-structures results from a sequence of join operations.

3.2.2 Evolution Meta-Structures

Objects in a database may have a number of stages. Evolution meta-structures are characterised by repetition and evolution cycles for self-correction and self-reinforcement. The core of such meta-structures is the repetition of stages of objects. We may differentiate between linear evolution models and cyclic evolution models. The former one uses non-repeatable, non-iterative specialisation schemata.

By using a *flow* constructor evolution meta-structures permit the construction of a well-communicating set of types with a P2P data exchange among the associated types. Such associations often appear in workflow applications, business processes, customer scenarios, and when when identifying variances. Evolution is based on the treatment of *stages* of objects. Objects are passed to handling agents (teams), which maintain and update their specific properties.

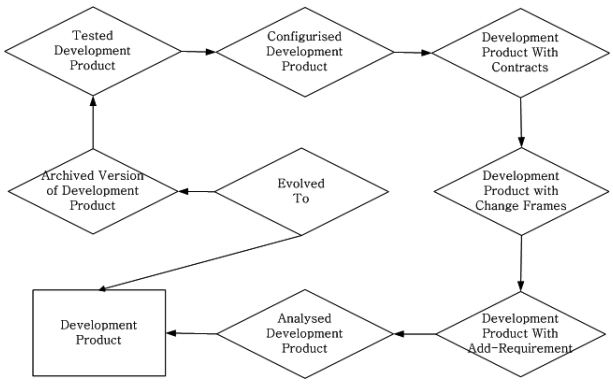


Figure 15: Evolution Meta-Structure for Software Project Management.

As an example consider the schema in Figure 15, which illustrates cyclic evolution for the support of software project management. Software development processes involve a number of actors or stakeholders, which are typically repeatable, defined, managed and optimised. Processes follow an internal work organisation, and products are analysed before requirements for development are applied. The company has developed frames or templates for changes within a product, and the requested changes are contracted to sub-divisions and sub-contractors. Finally, the next product is stored after testing and integration has been conducted. The relationship type EvolvedTo has been introduced for an explicit separation of generations or versions of development products.

3.2.3 Loop or Circulation Meta-Structures

These meta-structures appear whenever the lifespan of objects contains cycles. They are used for the representation of objects that store chains of events, people, devices, products, etc. Similar to the circulation meta-structure it employs non-directional, non-hierarchical associations with different modes of connectivity being applicable. In this way temporal assignment and sharing of resources, association and integration, rights and responsibilities can be neatly represented and scaled.

In circulation meta-structures objects may be related to each other by life-cycle stages such as repetition, self-reinforcement and self-correction. Typical examples are objects representing iterative processes, recurring phenomena or time-dependent activities. A circulation meta-structure supports primarily iterative processes.

Circulation meta-structures permit to display objects in different phases. For instance, legal document handling in the SeSAM e-government system is based on such phases, and a loop meta-structure provides an alternative to the incremental meta-structure in Figure 14.

As an example consider the schema sketch in Figure 16 dealing with document handling in a very general way. Though document handling may vary in various ways, we may assume an *inductive construction*, i.e. each document is constructed on the basis of simpler documents and base documents. The lower part of the snowflake schema addresses aspects of raw documents such as legal aspects, format, encoding, associations and contract involvement. These capture static aspects of a document. The upper part of the schema captures dynamic aspects that evolve over time. In particular, the operational document captures data entry into the document in relation to a rather complex workflow with several stages, different associated actors, various responsibilities and different stages of preparations. This leads to the almost completed blueprint and the completed submission document. Documents that are no longer subject to change are stored in an archive together with a summary or docket [13].

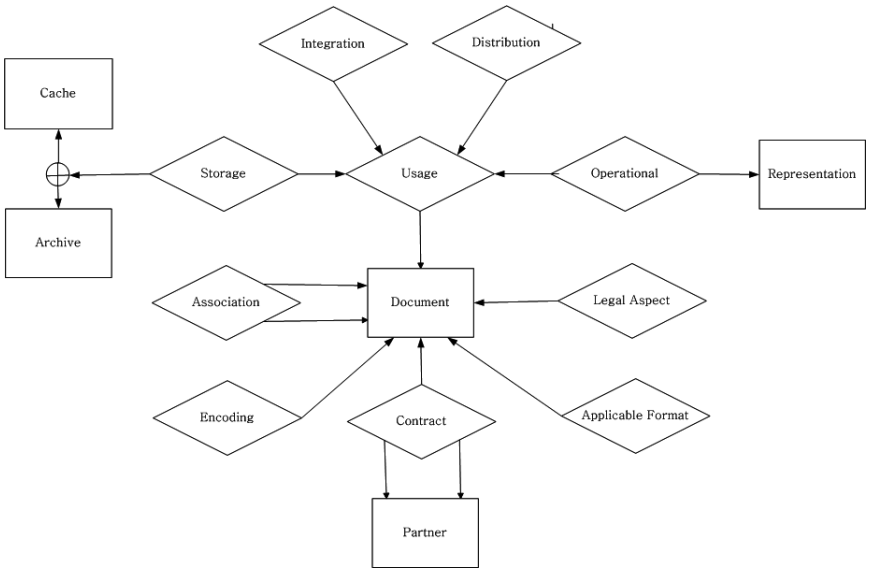


Figure 16: Loop or Circulation Meta-Structure.

3.2.4 Network Meta-Structures

Network or web meta-structures enable the collection of a network of associated types, and the creation of a multi-point web of associated types with specific control and data association strategies. The web has a specific data update mechanism, a specific data routing mechanism, and a number of communities of users building their views on the web.

Network meta-structures offer a unique opportunity to overcome the exploding type number problems in many applications, where relationships among objects are flexible, constantly changing, and reflect partial views or variants of other relationships. System configuration and configuration database management applications may however use the separation of types into categorised associations.

Network meta-structures are used for modern web services. Online resources for learning communities, special interest groups, and other shared information sources use a large number of associations among objects.

For instance, financial services are based on portfolios, combined on the fly, and provided, supported and used by institutions. Instead of representing the complex web of portfolio management we may split portfolios into basic portfolios, which relate portfolio providers and users. Such basic portfolios are combined and provided to customers or partner institutions. Whether a combination is considered to be a service depends on the customer's point of view.

As another example, railway management systems may be tightly bind to the application. The terminology varies from country to country. For instance, the notions of tunnel, path, segment, track, train or movement is different for most railway companies in Europe. Since trains also run between different regions, their scheduling, logging and reporting must combine all different systems. Moreover, identities for tracks and trains are derived from local databases and are not integrated resulting in a variety of schemata based on geographical separation. Thus, the development of a network meta-structure schema must be based on a common understanding of basic units and their disparate utilisation in the applications.

Another typical network meta-structure application is the support of legal documents as illustrated in Figure 17. They constitute a network of constantly renewed and deconstructed links among objects. In addition, local variations and specific portfolio for treatment and support are derived. Classical modelling approaches typically lead to schemata with document classes that either use chaotic sets of integrity constraints or use a confusing set of relationship types among the types in a schema. Figure 17 also illustrates the transformation of network meta-structures to abstract multi-layer structures, in which documents are interwoven with a large variety of links. This variety reflects the hierarchical structuring, usage and the evolution during the document lifespan.

As networks evolve quickly and irregularly, i.e. they grow fast and then are rebuilt and renewed, a network meta-structure must take care of a large number of variations to enable growth control and change management. Usually, they are supported by a multi-point center of connections, controlled routing and replication, change protocols, controlled assignment and transfer, scoping and localisation abstraction, and trader architectures. Furthermore, export/import converters and wrappers are supported. The database farm architecture [20] with check-in and check-out facilities supports flexible network extension.

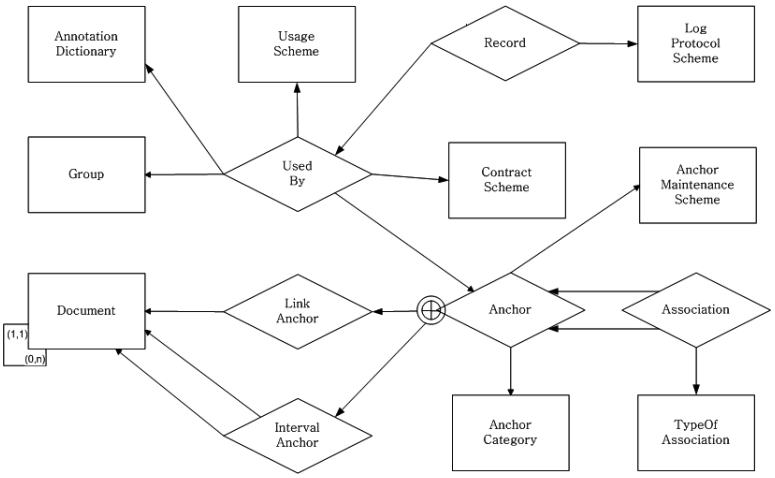


Figure 17: Network Meta-Structure.

3.3 Context Meta-Structures

According to [23] we distinguish between the *intext* and the *context* of things that are represented as objects. Intext reflects the internal structuring, associations among types and subschemata, the storage structuring, and the representation options. Context reflects general characterisations, categorisation, utilisation, and general descriptions such as quality. Therefore, we distinguish between *meta-characterisation meta-structures* that are usually orthogonal to the intext structuring and can be added to each of the intext types, *utilisation-recording meta-structures* that are used to trace the running, resetting and reasoning of the database engine, and *quality meta-structures* that permit to reason on the quality of the data provided and to apply summarisation and aggregation functions in a form that is consistent with the quality of the data. The dimensionality of a schema permits the extraction of other context meta-structures [3].

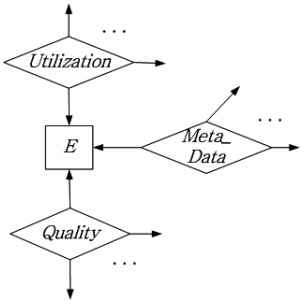


Figure 18: Context Meta-Structures.

Context meta-structures arise from joining in additional schemata – or using nesting – that capture meta information, e.g. for a document how it is used, by whom for which purpose, etc. (utilization meta-data), how accurate, complete or consistent it is (quality meta-data), or any other meta-data including technical and formating restrictions. This is illustrated in Figure 18. The three different classes of context meta-structures refer to a classification of context information.

3.3.1 Meta-Characterisation Meta-Structures

Meta-characterisation is orthogonal to the structuring dimension that may have led to a schema as displayed in Figure 6. They may refer to insertion/update/deletion time, keyword characterisation, utilisation pattern, format descriptions, utilisation restrictions and rights such as copyright and costs, and technical restrictions.

Meta-characterisations apply to a large number of types and should therefore be factored out. For instance, in an e-learning application learning objects, elements and scenes are commonly characterised by educational information such as interactivity type, learning resource type, interactivity level, age restrictions, semantic density, intended end user role, context, difficulty, utilisation interval restrictions, and pedagogical and didactical parameters.

3.3.2 Utilisation-Recording Meta-Structures

Logging, usage and history information is commonly used for recording the lifespan of the database. Therefore, we can distinguish between *history meta-structures* that are used for storing and recording the computation history within a small time slice, *usage-scene meta-structures* that are used to associate data to their use in a business process at a certain stage, a workflow step, or a scene in an application story, and record the actual usage.

Such meta-structures are related to one or more aspects of time, e.g. transaction time, user-defined time, validity time, or availability time, and associated with concepts such as temporal data types (instants, intervals, periods), and temporal statements such as current (now), sequenced (at each instant of time) and nonsequenced (ignoring time).

3.3.3 Quality Meta-Structures

Data quality is modelled by a variety of meta-structures capturing the sources (data source, responsible user, business process, source restrictions, etc.), intrinsic quality parameters (accuracy, objectivity, trustability, reputation, etc.), accessibility and security, contextual quality (relevance, value, timelineness, completeness, amount of information, etc.), and representation quality (ambiguity, ease of understanding, concise representation, consistent representation, ease of manipulation). Data quality is essential whenever versions of data have to be distinguished according to their quality and reliability.

4 Component-Based Schema Design

In this section we want to show how meta-structures and the associated schema algebra can be exploited to support component-based engineering as proposed in [12, 20]. We briefly review the rationale behind component-based development leading to the guiding principle of skeleton schemata that combine several components. We then extend the amalgamation approach from [12] in the light of the meta-structures discussed in the previous section and the new constructs introduced in this paper. In particular, we will emphasise that amalgamation and thus schema design can be based on graph rewriting.

4.1 Rationale for Component-Driven Development

Large database schemata can be drastically simplified, if techniques of modular design such as *design by units* [18] are used. Modular design is an abstraction technique based on principles of hiding and encapsulation that are known from Software Engineering. Different subschemata are connected by bridge types. *Component engineering* [12] extends this approach by means of view-centered components with well-defined composition operators, exploiting the observation that large subschemata often have the structure of star- or snowflake-schemata known from data warehousing. *Hierarchy abstraction* [20] permits to model objects on various levels of detail.

The co-design approach to database applications [18] aims at a consistent development of all facets of database applications: structuring of the database by schema types that are controlled by static integrity constraints, behaviour modelling by specification of functionality and dynamic integrity constraints, and interactivity modelling by assigning views to activities of actors in corresponding dialogue steps. Thus, co-design integrates the specification of the static database schema, functions, views and dialogues, which is facilitated by the use of view-extended schemata. At the same time, various abstraction layers are separated such as the conceptual layer, requirements acquisition layer and implementation layer, which has now become popular under the "model-driven architecture" theme.

Understandably, co-design is a rather complex procedure. However, if combined with the component-based approach it becomes simpler. In doing so first a skeleton of components is developed. This skeleton is then subject to stepwise refinement during further development of the view-extended schema. In particular, each component is refined thereby taking care of component interaction. In summary, co-design can be based on two principles:

Use of components: Components are the main building blocks for structuring the core data. In order to capture functionality components are modelled by view-extended schemata, in which each view contains also dialogue operations [12].

Skeleton-based construction: Components are assembled and amalgamated by applying connector types, which are usually relationship types.

4.2 Dimensions of Skeletons and Subschemata

A *component* – formally defined in [12, 20] – is a database schema together with import and export interfaces for connecting it to other components by standardised interface techniques. *Schema skeletons* [19] provide a framework for the general architecture of an application, to which details such as types are to be added. They are composed of *units*, which are defined by sets of components provided this set can be semantically separated from all other components without losing application information. Units may contain entity, relationship and cluster types, and the types in it should have a certain affinity or adhesion to each other.

In addition, units may be associated with each other in a variety of ways reflecting the general associations within an application. Associations group the relation of units by their meaning. Therefore, different associations may exist between the same units. Associations can also relate associations with each other. Therefore, structuring mechanisms as provided by the higher-order entity-relationship model [18] may be used to describe skeletons.

The usage of types in a database schema differs in many aspects. In order to support the maintenance of very large schemata this diversity of usage should be made explicit. Following an analysis of usage patterns [12] leads to a number of dimensions including the following important ones:

- Types may be specialized on the basis of roles objects play or categories into which objects are separated. This *specialization dimension* usually leads to subtype, role, and categorisation hierarchies, and to versions for development, representation or measures.

- As objects in the application domain hardly ever occur in isolation, we are interested in representing their associations by bridging related types, and adding meta-characterisation on data quality. This *association dimension* often addresses specific facets of an appli-

cation such as points of view, application areas, and workflows that can be separated from each other.

- Data may be integrated into complex objects at runtime, and links to business steps and rules as well as log, history and usage information may be stored. Furthermore, meta-properties may be associated with objects such as category, source and quality information. This defines the *usage, meta-characterisation* or *log dimension*. Dockets [13] may be used for tracking processing information, superimposed schemata for explicit log of the treatment of the objects, and provenance schemata for the injection of meta-schemata.
- As data usage is often restricted to some user roles, there is a *rights and obligations dimension*, which entails that the characterisation of user activities is often enfolded into the schema.
- As data varies over time and different facets are needed at different moments, there is a *data quality, lifespan and history dimension* for modelling data history and quality , e.g. source data, and data referring to the business process, source restrictions, quality parameters etc. With respect to time the dimension distinguishes between transaction time, user-defined time, validity time, and availability time.
- The *meta-data dimension* refers to temporal, spatial, ownership, representation or context data that is often associated with core data. These meta-data are typically added after the core data has been obtained.

We often observe that very large database schemata incorporate some or all of these dimensions, which explains the difficulty for reading and comprehension. For instance, various architectures such as technical and application architecture may co-appear within a schema [15].

Furthermore, during its lifetime a database schema, which may originally have captured just the normalised structure of the application domain, is subjected to performance considerations and extended in various ways by views. A typical example for a complete schema full of derived data is given by OLAP applications [5]. Thus, at each stage the full schema is in fact the result of folding extensions by means of a so-called *grounding schema* into the core database schema.

4.3 Graph-Grammar Composition for Schemata

As emphasised in [9], a structural approach to schema construction as in [1] is possible. All constructors known for database schemata may also be applied to meta-structures. Therefore, we can base a theory of schema composition on constructors for generalised Entity-Relationship schemata as in [18].

A general composition theory for such schemata can be based on the theory of graph grammars [2, 16], which has been already exploited for the CASE tool RADD [18]. The composition of graphs can be formalised by two pushouts in the category of directed graphs. However, we will avoid using category-theoretical terminology. Furthermore, instead of general graph homomorphisms we only consider subgraphs.

A *graph production rule* takes the form

$$\varrho\colon\ L\ \supseteq\ K\ \subseteq\ R$$

with graphs L, R called the left-hand side and the right-hand side of the production rule ϱ, respectively, and a common subgraph K with $L \cap R = K$, which is called the *gluing graph* of ϱ.

The intuitively clear meaning of a graph production rule is to replace the left-hand side L by the right-hand side R, whenever L appears as a subgraph of any graph G. Naturally, as the *gluing graph* K of the rule is the intersection of the left- and right-hand sides, it will be invariant under the replacement.

However, the context of L within the graph G has to be taken into account as well, i.e. it has to be specified how edges connecting vertices in $G - L$ to vertices in L are handled. This leads to the exact definition of a rule application. Such an application of a graph production rule must be conflict-free in the sense that no name clashes occur between the graphs R and $G - L$. In order to avoid name clashes, vertices and edges in $R - L$ are to be renamed.

Let $\varrho : L \supset K \subset R$ be a graph production rule, and let G be a graph. Furthermore, in order to apply ϱ to G, we assume to be given

- a *renaming function* m defined on $L \cup R$ such that $m(L)$ becomes a subgraph of G and $m(L \cup R) \cap G = m(L)$ holds, and
- a subgraph C of G called *context graph* with $C \cap m(L) = m(K)$ and $G = C \cup m(L)$.

The graph H resulting from *applying the graph production rule ϱ to G* is defined by

$$H = (G - m(L)) \cup m(R).$$

We denote the graph transformation defined by ϱ and m – C is defined implicitly – by $G \overset{\varrho,m}{\Rightarrow} H$.

In graph rewriting the used set of graph production rules must satisfy the substitution rule, i.e. none of the transformations may have side effects. If vertices and edges outside $L - K$ are not affected, graph production rules can be composed to form derived graph production rules.

If $\mathcal{S}_1, \ldots, \mathcal{S}_n$ are schemata and $\mathcal{O}$ is an n-ary operator applicable to them, the resulting schema $\mathcal{S}$ defines the equation $\mathcal{S} = \mathcal{O}(\mathcal{S}_1, \ldots, \mathcal{S}_n)$. Using these equations in a directed way defines a graph-rewriting system GRS. In view of the previous section, the graph production rules in GRS are rather simple, but more complex and presumably more convenient rules can be derived by rule composition.

Table 1: Rewrite rules for component amalgamation

join:	$\varrho_{\bowtie_{:=}} : \mathcal{S}_1 \cup \mathcal{S}_2 \supseteq \mathcal{S}_1 \cap \mathcal{S}_2 \subseteq \mathcal{S}_1 \bowtie_{\mathcal{I}_1:=\mathcal{O}_2 \| \mathcal{I}_2:=\mathcal{O}_1} \mathcal{S}_2$
sum:	$\varrho_{\oplus} : \mathcal{S}_1 \cup \mathcal{S}_2 \supseteq \emptyset \subseteq \mathcal{S}_1 \oplus \mathcal{S}_2$
reference-join:	$\varrho_{\bowtie_{\rightarrow}} : \mathcal{S}_1 \cup \mathcal{S}_2 \supseteq \mathcal{S}_1 \cup \mathcal{S}_2 \subseteq \mathcal{S}_1 \bowtie_{\mathcal{I}_1 \rightarrow \mathcal{O}_2 \| \mathcal{I}_2 \rightarrow \mathcal{O}_1} \mathcal{S}_2$
product:	$\varrho_{\times} : \mathcal{S}_1 \cup \mathcal{S}_2 \supseteq \emptyset \subseteq \mathcal{S}_1 \times \mathcal{S}_2$
meet:	$\varrho_{\bullet_\varphi} : \mathcal{S}_1 \cup \mathcal{S}_2 \supseteq \emptyset \subseteq \mathcal{S}_1 \bullet_\varphi \mathcal{S}_2$
nesting:	$\varrho_{Nest} : \mathcal{S}_1 \cup \mathcal{S}_2 \supseteq \mathcal{S}_1 - \{C\} \cup \mathcal{S}_2 \subseteq Nest_{C:\mathcal{S}_2(T)}(\mathcal{S}_1)$

4.4 Rewriting-Based Component Amalgamation

According to the decomposition theorem in [12] each behaviour-extended schema is the amalgamation of snowflake components. This naturally extends to all twelve types of meta-structures identified in [9]. Formally, this means that component sub-schemata can be written as algebraic expressions involving

- the renaming operator $\varrho_{R-1 \mapsto R'_1,\ldots,R_k \mapsto R'_k}$,
- the join operator $\bowtie_{\mathcal{I}_1:=\mathcal{O}_2 \| \mathcal{I}_2:=\mathcal{O}_1}$, and as a special case the direct sum operator $\oplus$,
- the reference-join operator $\bowtie_{\mathcal{I}_1 \rightarrow \mathcal{O}_2 \| \mathcal{I}_2 \rightarrow \mathcal{O}_1}$,
- the product operator $\times$ and more generally, the meet operator $\bullet_\varphi$,
- the bulk operator $Bulk_{R_1,\ldots,R_n}$ and its inverse expand operator $Expand_{E:A}$,
- the nesting operator $Nest_{C:\mathcal{S}'(T)}$, and
- the collection operators $\{\cdot\}$, $[\cdot]$, $\langle\cdot\rangle$, and $\langle\!\langle\cdot\rangle\!\rangle$, and the related operators all_of, $any\_of\ n\_of$, and n_th.

This defines the formal underpinnings for the following pragmatic steps in view-extended schema design:

1. We start from behaviour-extended schemata for certain tasks of the application, as they may arise from cutting up a development project and then working independently. These schemata may not be snowflake components. However, they can be represented as amalgams. Then the definition of views that connect these components defines an amalgam for the whole application.
2. Each component resulting from step one can be decomposed into snowflake components by the decomposition theorem. So we need algorithms for detecting components and checking, whether they are almost hierarchical or not. Together with phase one this amounts to an amalgam with a larger number of components, but these components are now snowflakes.
3. In the third phase we consider the overlap between components aiming at minimising them as much as possible. The result will still be an amalgam with snowflake components, but these components do not overlap excessively any more.
4. Finally, we reconsider the components resulting from phase three and recombine some of them, if this the result is still a snowflake component and the application considers the initial components as belonging together to one task of the application.

Naturally, amalgamation itself will exploit the algebra operators above. Thus the pragmatic approach can be supported by formal graph rewriting. If two component schemata $\mathcal{S}_1$ and $\mathcal{S}_2$ are given, we may define the amalgam by exploiting one of the rewrite rules in Table 1.

When applying these rules, suitable views $\mathcal{I}_j$ and $\mathcal{O}_j$, and types C and T have to be selected.

5 Conclusions

In this article we addressed the unsatisfactory situation that the design of very large database schemata is not well supported. Such schemata with hundreds or thousands of types are usually developed over years, and then require sophisticated skills to read and comprehend them. However, lots of similarities, repetitions, and similar structuring elements appear in such schemata. In this paper we highlighted the frequently occurring meta-structures in such schemata, and classified them according to structure, lifespan and context. Furthermore, we presented an algebra for handling these meta-structures, which permits large schemata to be composed out of smaller ones. In this way a component-based approach to schema design is enabled, in which the application of the schema algebra constructors can be formalised by graph rewriting.

Practically speaking, meta-structures can be exploited to modularise schemata, which would ease querying, searching, reconfiguration, maintenance, integration and extension. From a development perspective different aspects dealing with structures, lifespan and context could be separated. Thus, an easier integration of development subpro-

jects would be possible. Also reengineering and reuse are enabled.

In this way data modelling using meta-structures enables systematic schema development, extension and implementation, and thus contributes to overcome the maintenance problems arising in practice from very large schemata. Furthermore, the use of meta-structures also enables component-based schema development, in which schemata are developed step-by-step on the basis of the skeleton of the meta-structure, and thus contributes to the development of industrial-scale database applications.

However, in our presentation in this article we concentrated on schemata with constraints, thus ignoring additional aspects such as views and operations. In the component-model in [12] these were also considered as part of component-based information systems engineering. Consequently, our approach requires additional investigation of the interaction aspect. The question is, whether frequently occurring patterns can also be discovered for views and operations. For the classical application area of decision support this question has already been addressed and answered positively by means of standard OLAP operations [6].

References

[1] Brown, L. *Integration Models – Templates for Business Transformation*. SAMS Publishing, 2000.

[2] Ehrig, H., Engels, G., Kreowski, H.-J., and Rozenberg, G., Eds. *Handbook of Graph Grammars and Computing by Graph Transformations – Vol. 2: Applications, Languages and Tools*. World Scientific, 1999.

[3] Feyer, T., and Thalheim, B. Many-dimensional schema modeling. In *Advances in Databases and Information Systems – Proc. ADBIS 2002*, Y. Manolopoulos and P. Návrat, Eds., vol. 2435 of *LNCS*. Springer-Verlag, 2002, pp. 305–318.

[4] Kahlen, H. *Integrales Facility Management – Management des ganzheitlichen Bauens*. Werner Verlag, 1999.

[5] Lenz, H.-J., and Thalheim, B. OLAP schemata for correct applications. In *Trends in Enterprise Application Architecture*, vol. 3888 of *LNCS*. Springer-Verlag, 2005, pp. 99–113.

[6] Lenz, H.-J., and Thalheim, B. A formal framework of aggregation for the OLAP-OLTP model. *Journal of Universal Computer Science 15*, 1 (2009), 273–303.

[7] Ma, H., Noack, R., and Schewe, K.-D. Algebraic meta-structure handling of huge database schemata. In *Advances in Conceptual Modeling – Challenging Perspectives*, C. Heuser and G. Pernul, Eds., vol. 5833 of *LNCS*. Springer-Verlag, 2009, pp. 23–32.

[8] Ma, H., Noack, R., Schewe, K.-D., Thalheim, B., and Wang, Q. Complete conceptual schema algebras. submitted for publication, 2009.

[9] Ma, H., Schewe, K.-D., and Thalheim, B. Modelling and maintenance of very large database schemata using meta-structures. In *Information Systems and e-Business Technologies – 3rd International Conference UNISCON 2009, Proceedings*, J. Yang et al., Eds., vol. 20 of *LNBIP*. Springer-Verlag, 2009, pp. 17–28.

[10] Moody, D. *Dealing with Complexity: A Practical Method for Representing Large Entity-Relationship Models*. PhD thesis, University of Melbourne, 2001.

[11] Raak, T. Database systems architecture for facility management systems. Master's thesis, Fachhochschule Lausitz, 2002.

[12] Schewe, K.-D., and Thalheim, B. Component-driven engineering of database applications. In *Conceptual Modelling – Proc. APCCM 2006*, vol. 53 of *CRPIT*. Australian Computer Society, 2006, pp. 105–114.

[13] Schmidt, J. W., and Sehring, H.-W. Dockets: A model for adding value to content. In *Conceptual Modeling – ER '99*, vol. 1728 of *LNCS*. Springer-Verlag, 1999, pp. 248–262.

[14] Shoval, P., Danoch, R., and Balaban, M. Hierarchical ER diagrams (HERD) – the method and experimental evaluation. In *Advanced Conceptual Modeling Techniques*, vol. 2784 of *LNCS*. Springer-Verlag, 2002, pp. 264–274.

[15] Siedersleben, J. *Moderne Softwarearchitektur*. dpunkt-Verlag, 2004.

[16] Sleep, M. R., Plasmeijer, M. J., and van Eekelen, M. C. J. D., Eds. *Term Graph Rewriting – Theory and Practice*. John Wiley and Sons, 1993.

[17] Smith, J. M., and Smith, D. C. P. Database abstractions: Aggregation and generalization. *ACM ToDS 2*, 2 (1977), 105–133.

[18] Thalheim, B. *Entity Relationship Modeling – Foundations of Database Technology*. Springer-Verlag, 2000.

[19] Thalheim, B. Component construction of database schemes. In *Conceptual Modeling – ER 2002*, vol. 2503 of *LNCS*. Springer-Verlag, 2002, pp. 20–34.

[20] Thalheim, B. Component development and construction for database design. *Data and Knowledge Engineering 54* (2005), 77–95.

[21] Thalheim, B. Engineering database component ware. In *Trends in Enterprise Application Architecture*, vol. 4473 of *LNCS*. Springer-Verlag, 2007, pp. 1–15.

[22] Thalheim, B., and Kobienia, T. Generating database queries for web natural language requests using schema information and database content. In *Applications of Natural Language to Information Systems – NLDB 2001*, vol. 3 of *LNI*. GI, 2001, pp. 205–209.

[23] Wisse, P. *Metapattern – Context and Time in Information Models*. Addison-Wesley, 2001.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 800 staff, has 600 researchers, about 250 of whom are postgraduates, nearly 400 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of S**love**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit an email with the manuscript to one of the editors from the Editorial Board or to the Managing Editor. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica LaTeX format and figures in .eps format. Style and examples of papers can be obtained from http://www.informatica.si. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

QUESTIONNAIRE

☐ Send Informatica free of charge

☐ Yes, we subscribe

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: drago.torkar@ijs.si

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than sixteen years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering the European computer science and informatics community - scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name: .....

Title and Profession (optional): .....

.....

Home Address and Telephone (optional): .....

.....

Office Address and Telephone (optional): .....

.....

E-mail Address (optional): .....

Signature and Date: .....

Informatica WWW:

http://www.informatica.si/

Referees from 2008 on:

Ajith Abraham, Siby Abraham, Renato Accornero, Hameed Al-Qaheri, Gonzalo Alvarez, Wolfram Amme, Nicolas Anciaux, Rajan Arora, Costin Badica, Zoltán Balogh, Andrea Baruzzo, Norman Beaulieu, Paolo Bellavista, Zbigniew Bonikowski, Marco Botta, Pavel Brazdil, Andrej Brodnik, Ivan Bruha, Wray Buntine, Yunlong Cai, Juan Carlos Cano, Tianyu Cao, Norman Carver, Marc Cavazza, Jianwen Chen, LM Cheng, Chou, Cheng-Fu, Girija Chetty, G. Chiola, Yu-Chiun Chiou, Ivan Chorbev, Shauvik Roy Choudhary, Lawrence Chung, Jean-Noël Colin, Jinsong Cui, Alfredo Cuzzocrea, Gunetti Daniele, Grégoire Danoy, Manoranjan Dash, Paul Debevec, Fathi Debili, Carl James Debono, Joze Dedic, Abdelkader Dekdouk, Bart Demoen, Sareewan Dendamrongvit, Tingquan Deng, Gaël Dias, Ivica Dimitrovski, Jana Dittmann, Simon Dobrišek, Quansheng Dou, Jeroen Doumen, Dejan Drajic, Jozo Dujmovic, Umut Riza Ertürk, Ling Feng, YiXiong Feng, Andres Flores, Vladimir A. Fomichov, Stefano Forli, Massimo Franceschet, Alberto Freitas, Chong Fu, Gabriel Fung, Andrea Gambarara, Matjaž Gams, Juan Garbajosa, David S. Goodsell, Jaydeep Gore, Zhi-Hong Guan, Donatella Gubiani, Bidyut Gupta, Marjan Gusev, Zhu Haiping, Juha Hyvärinen, Dino Ienco, Natarajan Jaisankar, Imad Jawhar, Yue Jia, Ivan Jureta, Džani Juričić, Zdravko Kačič, Boštjan Kaluža, Dimitris Kanellopoulos, Rishi Kapoor, Daniel S. Katz, Mustafa Khattak, Ivan Kitanovski, Tomaž Klobučar, Ján Kollár,, Peter Korošec, Agnes Koschmider, Miroslav Kubat, Chi-Sung Laih, Niels Landwehr, Andreas Lang, Yung-Chuan Lee, John Leggett, Aleš Leonardis, Guohui Li, Guo-Zheng Li, Jen Li, Xiang Li, Xue Li, Yinsheng Li, Yuanping Li, Lejian Liao, Huan Liu, Xin Liu, Hongen Lu, Mitja Luštrek, Inga V. Lyustig, Matt Mahoney, Dirk Marwede, Andrew McPherson, Zuqiang Meng, France Mihelič, Nasro Min-Allah, Vojislav Misic, Mihai L. Mocanu, Jesper Mosegaard, Marta Mrak, Yi Mu, Josef Mula, Phivos Mylonas, Marco Di Natale, Pavol Navrat, Nadia Nedjah, R. Nejabati, Wilfred Ng, Zhicheng Ni, Fred Niederman, Omar Nouali, Franc Novak, Petteri Nurmi, Barbara Oliboni, Matjaž Pančur, Gregor Papa, Marcin Paprzycki, Marek Paralič, Byung-Kwon Park, Gert Schmeltz Pedersen, Torben Bach Pedersen, Zhiyong Peng, Ruggero G. Pensa, Dana Petcu, Macario Polo, Victor Pomponiu, Božidar Potočnik, S. R. M. Prasanna, HaiFeng Qian, Lin Qiao, Jean-Jacques Quisquater, Jean Ramaekers, Jan Ramon, Wilfried Reimche, Juan Antonio Rodriguez-Aguilar, Pankaj Rohatgi, Wilhelm Rossak, Sattar B. Sadkhan, Khalid Saeed, Motoshi Saeki, Evangelos Sakkopoulos, M. H. Samadzadeh, MariaLuisa Sapino, Piervito Scaglioso, Walter Schempp, Barabara Koroušić Seljak, Mehrdad Senobari, Subramaniam Shamala, Heung-Yeung Shum, Tian Song, Andrea Soppera, Alessandro Sorniotti, Liana Stanescu, Martin Steinebach, Xinghua Sun, Marko Robnik,vSikonja, Jurij,vSilc, Carolyn Talcott, Camillo J. Taylor, Drago Torkar, Christos Tranoris, Denis Trček, Katarina Trojacanec, Mike Tschierschke, Filip De Turck, Aleš Ude, Alessia Visconti, Petar Vračar, Valentino Vranić, Chih-Hung Wang, Huaqing Wang, Hao Wang, YunHong Wang, Sigrid Wenzel, Woldemar Wolynski, Allan Wong, Stefan Wrobel, Konrad Wrona, Bin Wu, Xindong Wu, Li Xiang, Yan Xiang, Di Xiao, Fei Xie, Yuandong Yang, Chen Yong-Sheng, Jane Jia You, Ge Yu, Mansour Zand, Dong Zheng, Jinhua Zheng, Albrecht Zimmermann, Blaz Zupan, Meng Zuqiang

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Vožarski pot 12, 1000 Ljubljana, Slovenia.
The subscription rate for 2010 (Volume 34) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar.
Printing: Dikplast Kregar Ivan s.p., Kotna ulica 5, 3000 Celje.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Franjo Pernuš)
Slovenian Artificial Intelligence Society; Cognitive Science Society (Matjaž Gams)
Slovenian Society of Mathematicians, Physicists and Astronomers (Bojan Mohar)
Automatic Control Society of Slovenia (Borut Zupančič)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Igor Grabec)
ACM Slovenia (Dunja Mladenič)

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

The issuing of the Informatica journal is financially supported by the Ministry of Higher Education, Science and Technology, Trg OF 13, 1000 Ljubljana, Slovenia.

Volume 34 Number 3 October 2010 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Editor's Introduction to the Special Issue on Semantic Informational Technologies	V.A. Fomichov	**267**
Obtaining Status Descriptions via Automatic Analysis of Hospital Patient Records	S. Boytcheva, I. Nikolova, E. Paskaleva, G. Angelova, D. Tcharaktchiev, N. Dimitrova	**269**
Corpus and Web: Two Allies in Building and Automatically Expanding Conceptual Classes	N. Béchet, J. Chauché, V. Prince, M. Roche	**279**
Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web	V.A. Fomichov	**287**
Wikipedia2Onto — Building Concept Ontology Automatically, Experimenting with Web Image Retrieval	H. Wang, X. Jiang, L.-T. Chia, A.-H. Tan	**297**
A Service Oriented Framework for Natural Language Text Enrichment	T. Štajner, D. Rusu, L. Dali, B. Fortuna, D. Mladenič, M. Grobelnik	**307**
Applications of Semantics in Agent-Based Manufacturing Systems	M. Obitko, P. Vrba, V. Mařík, M. Radakovič, P. Kadera	**315**
The Role of the Semantic Web for Knowledge Management in the Construction Industry	I. Svetel , M. Pejanović	**331**
End of Special Issue / Start of normal papers		
Cryptanalysis of a Simple Three-party Key Exchange Protocol	H. Debiao, C. Jianhua, H. Jin	**337**
KP-Lab System for the Support of Collaborative Learning and Working Practices, Based on Trialogical Learning	J. Paralič, F. Babič, J. Wagner, P. Bednár, M. Paralič	**341**
An LPGM method: Platform Independent Modeling and Development of Graphical User Interface	J. Kryštof	**353**
A Fast Convex Hull Algorithm for Binary Image	X. Zhang, Z. Tang, J. Yu, M. Guo	**369**
Using Genetic Algorithms and Dominance Concepts for Generating Reduced Test Data	A.S. Ghiduk, M.R. Girgis	**377**
Using Meta-Structures in Database Design	H. Ma, R. Noack, K.-D. Schewe, B. Thalheim	**387**

Informatica **34** (2010) Number 3, pp. 267–405