ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

TEZE K DISERTAČNÍ PRÁCI

České vyskoké učení technické v Praze
Fakulta elektrotechnická
Katedra kybernetiky

Petr Aubrecht

ONTOLOGY TRANSFORMATION BETWEEN FORMALISMS

Studijní obor: Umělá inteligence a biokybernetika

Teze disertace k získání akademického titulu „doktor," ve zkratce „Ph.D."

Praha, únor 2005

Disertační práce byla vypracována v kombinované formě doktorského studia na katedře kybernetiky fakulty elektrotechnické ČVUT v Praze.

Uchazeč: Ing. Petr Aubrecht
Katedra kybernetiky
Fakulta elektrotechnická ČVUT
Technická 2, 166 27 Praha 6

Školitel: Prof. RNDr. Olga Štěpánková, CSc.
Katedra kybernetiky
Fakulta elektrotechnická ČVUT
Technická 2, 166 27 Praha 6

Školitel-specialista: Doc. Ing. Zdeněk Kouba, CSc.
Katedra kybernetiky
Fakulta elektrotechnická ČVUT
Technická 2, 166 27 Praha 6

Oponenti: .....

.....

.....

Teze byly rozeslány dne: .....

Obhajoba disertace se koná dne ..... v ..... hod. před komisí pro obhajobu disertační práce ve studijním oboru Umělá inteligence a biokybernetika v zasedací místnosti č. ..... fakulty elektrotechnické ČVUT v Praze.

S disertací se je možno seznámit na děkanátu fakulty elektrotechnické ČVUT v Praze, na oddělení pro vědeckou a výzkumnou činnost, Technická 2, 166 27, Praha 6.

Prof. Ing. Vladimír Mařík DrSc.
.....
předseda komise pro obhajobu disertační práce
ve studijním oboru
Umělá inteligence a biokybernetika
Fakulta elektrotechnickă ČVUT
Technická 2, 166 27, Praha 6

1 STATE OF THE ART

1.1 Ontologies

As internet was growing, intelligent processing of the available information and searching relevant data within it became necessary. A need of computer programs with "common sense" also meets the problems of limited use of expert systems. Programs have to know the context of their work in order to provide meaningful results.

Ontologies are used for description of the world. Ontology is regarded as a common dictionary for communication between intelligent systems. The first attempt to define such dictionary was carried out by Aristotle, who started with constitution of terminology in many areas and these terms are used up to today (e.g. category, quality, quantity, metaphor, hypothesis). He also defined a hierarchy of ten basic categories. Wilhelm Leibniz (1646–1716) tried to establish mathematical foundation of mental processes. His formalism was able to express only conjunction, but he never found a way how to represent all the rules of inference and logical operators. A widely accepted hierarchy (even though not called ontology) introduced by Swedish scientist Carl Linnaeus (1707–1778, later Carl von Linné) for classification of plants. The Linnaean taxonomy is a base for taxonomies used by biologists up to today.

Currently, the best-known definition of the term *ontology* is the definitions of Gruber from 1993 [5]: An ontology is a explicit specification of a conceptualisation. It has been later modified, e.g. by Borst in 1997 [2], who emphasised the formal and public side of ontology: An ontology is a formal specification of a shared conceptualisation. An important person in knowledge domain domain, John F. Sowa, stated [10]: Ontology defines the kinds of things that exist in the application domain. The idea of a shared dictionary leads to *upper ontologies* (SUMO, WordNet), describing the most common concepts.

Ontology also serves as a background knowledge. It often contains a procedural part, which allows stating restrictions and actions carried out within the knowledge base. This kind of utilisation is the basic idea of the famous and long-term Douglas Lenat's project Cyc, aiming to describe all the human background knowledge and allow computers to think "with human-level breadth and depth of knowledge."

In the second half of twentieth century many formalisms have been developed for artificial intelligence domain. In 1956, Richard H. Richens of the Cambridge Language Research Unit defined "semantic nets" for machine translation of natural languages. Ross Quillian's introduced in his Ph.D. thesis (1968) a term "semantic network" as a way of talking about the organisation of human semantic memory, or memory for word concepts.

Marvin Minsky proposed *frames* to represent facts and structure of some

object as a record in [7] in 1975. The usage of frames were demonstrated on spatial imagery and linguistic understanding. Frames are organised into a network of nodes and relations, where "top levels" are fixed, represent things that are always true about the supposed situation. The lower levels have many terminals – *slots*, that must be filled by specific instances or data. Together with the slots there can be defined conditions an assignment must meet; they can require a terminal assignment to be of a particular type or an object of sufficient value. Formalisms based on frames are typically implemented in Lisp and partially depend on LISP expressions. An example of such formalism is SUO-KIF, based on KIF and used for SUMO upper ontology. Ontolingua with its *Frame Ontology* [5] was an attempt to unify capabilities of formalisms and to translate knowledge bases between formalisms. The constructs introduced by Ontolingua were supplied with inference mechanism in OCML [8].

Description logics (DL, previously called terminological logics) started with a motivation of providing a formal foundation for the semantic networks. The first DL implementation KL-ONE grew out of Ron Brachman's thesis in 1977. In 1983, Ron Brachman introduced T-box for defining terms and A-box for making assertions.

An important role in ontology formalisms development plays an idea of interconnected intelligent machines – semantic web. Initial simple attempts like SHOE or XOL did not much attention. Well known are activities of World Wide Web Consortium (w3c.org). Their standards are based mainly on description logics. The first contribution to the semantic web activities was standard RDF. A line of standards has been developed on top of RDF: RDF Schema, DAML-ONT, DAML+OIL, OWL. The RDF Schema adds basic concepts like *Class* and *Property*. DAML-ONT adds restriction on properties. It has been soon replaced by DAML+OIL, which was a joint activity of DARPA (DAML) and European Union (OIL). The join language uses RDFS backward compatibility. It lead to problems known from the beginning – for example *reification* introduced in RDF disallows to create inference engine. Therefore in the latest standard, OWL, several ideas were inspired by OIL, e.g. split language to parts with increasing capability and complexity. The simplest sublanguage OWL-Lite restricts the constructs in order to provide a minimal useful subset, which can be easily implemented. Only simple class hierarchies can be built, there can be used property constraints and characterisations, and classes can be constructed only though intersection or property constraints.

1.2 Ontology Transformations

A frequently solved problem is sharing ontologies between system with one common formalism. Pieces of ontologies (usually data, instances) within one application migrate between two particular ontologies, i.e. information exchange be-

tween two knowledge bases. This problem is often addressed by multiagent systems with independent agents (MAS) collecting knowledge and creating their own structured view of the surrounding world. At the moment two agents have to communicate, they have to consolidate their ontologies.

Nowadays, there exist several tools supporting (semi)automatic ontology sharing or conversion. They can be classified according to different features. The tools provide ontology merging, building of semantic bridges, and also reasoning done simultaneously on a number of ontologies. A list of such tools contains Chimaera, OntoMerge, FCA-Merge, ONION, GLUE, PROMPT, and others. Although the tools can support multiple formalisms, they do not solve the problem of incompatible capabilities of the formalisms.

A conversion of ontologies between formalisms is a rather difficult problem. Formalisms are used being mutually incompatible in their constructs and often include procedural constructs to express features, dependencies, or restrictions in the knowledge base. According to [11] there are three main approaches to transformation between different knowledge representation formalisms:

Mapping Approach – This approach leads to the lowest loss of information. A mapping is created which transforms expressions in the source formalism to expressions in the target formalism. Such mapping has to be defined for every pair of formalisms. Therefore it can be well adapted to the two specific formalisms. However the number of transformations that have to be designed increases sharply with the number of formalisms involved. It is also necessary to check properties of every transformation individually. That is why this approach is feasible only for systems working with a relatively small and fixed set of formalisms. An example of this approach is the OntoMorph system described in [3].

Pivot Approach – To avoid the necessity to create a large number of transformations one formalism is chosen as the pivot formalism. It has to be a formalism that is the most expressive of all the considered formalisms. For each of the other formalisms a mapping is designed that transforms expressions between the particular formalism and the pivot formalism. A transformation between two different formalisms is then done via the pivot formalism. The pivot formalism has to be very expressive to enable lossless transformation of all other formalisms into it. It has to be extended almost every time a new formalism is added to the system. Especially in case the system involves formalisms that are unlike each other e.g. formalisms based on description logic, formalisms based on frames, UML etc., the pivot formalism would have to be quite complex. It is also difficult to design the pivot formalism so that it would not be biased towards one type of formalisms.

Layered Approach – The third approach uses a layered architecture containing languages with increasing expressiveness [11]. There has been an attempt by W3C to provide a standard group of languages that would be layered on top

of each other, using RDFS as the layer. However other requirements on properties of the higher ontology languages, especially their decidability needed for reasoning, were more significant for their design than full backwards compatibility with the RDFS. The higher ontology languages such as DAML+OIL and OWL only use terms defined by RDFS as their basis. Except for OWL-Full the ontological languages do not cover RDFS completely. Some expressions valid in RDFS are not allowed in the other languages.

Family of Languages – In addition to the approaches described above a new approach called the Family of languages approach is proposed in [4]. It is a generalisation of layered and pivot language approach. A set of languages form a lattice with respect to a coverage relation. The coverage relation can be defined in a number of ways. It depends on the properties, which the transformation should preserve. There are four types of coverage relation: language-based coverage (one language is a subset of the other), interpretation-based coverage (there exists an interpretation-preserving transformation between the languages), consequence preserving and consistency-preserving. The last two of them imply a loss of information which is inherent in transformation from a more expressive language to a less expressive one.

2 THESIS STATEMENT

The main objective of this thesis is to provide a framework for transformation of ontologies between various formalisms. In order to achieve this major objective, the following goals had to be accomplished:

1. Currently available definitions of *ontology* do not provide sufficient formal basis for making analyses, comparisons, and conversions of ontologies. Hence, a more formal definitions of ontology from syntactical point of view and definitions of further formal terms like *formalism* are necessary to be introduced.

2. Formal definition of ontology transformation has to be defined making use of the developed formal definition of ontology.

3. Demands on properties of such transformations need to be specified.

4. A generalised formalism, which will allow expressing meta-models of all common ontology formalisms, needs to be designed and the respective meta-models of individual ontology formalisms shall be expressed by means of this generalised one.

5. Individual transformations making use of the generalised ontology formalism and respective ontology formalism meta-models shall be designed.

6. As the existing ontology transformations have different expression power, it is not always possible to achieve conversion without losing some information. An analysis of ontology transformations needs to be done with respect to the natural requirement to lose as few information as possible.

7. The whole designed framework needs to be implemented and verified on existing publicly available ontologies. Proper candidates are large upper ontologies SUMO and Cyc.

8. Finally, the developed methodology needs to be evaluated.

As the term "language" is used in several meanings, "language" will be called the way of encoding for persistence purposes. Language at this level is an ordered set of symbols. A description at a higher level will be denoted as "formalism." Several formalisms (e.g. RDF) allow encoding in more than one language (XML or N3).

Current definitions of the term *ontology* are rather vague. All three classic definitions – Gruber, 1993 (*explicit* specification of a conceptualisation); Borst, 1997 (*formal* specification of a *shared* conceptualisation); Sowa, 2000 (ontology defines the kinds of things that exist in the application domain) – emphasise only the requirement to specify a set of concepts and differ only in further requests (on formality, sharing).

For purposes of ontology transformations between formalisms and description of required or desired results,

3 SUGGESTED METHODS

The current ontology definitions are first replaced with a formal definition. It has been observed, that the term *formalism* is commonly used for both a description of available constructs and a set of ontologies. This ambiguity has been removed by introducing term *ontology grammar*, inspired by similarity with formal languages. The grammar is defined as 5-tuple $(\mathcal{C}_\mathcal{F}, \mathcal{R}_\mathcal{F}, \mathcal{S}_\mathcal{F}, \mathbb{S}_\mathcal{F}, \mathbb{A}_\mathcal{F})$, where $\mathcal{C}_\mathcal{F}$ is a set of concepts of the formalism (e.g. class, slot, literal), $\mathcal{R}_\mathcal{F}$ is a set of relations (SubclassOf, Has-Slot), $\mathcal{S}_\mathcal{F}$ is a set of structural restrictions (class HasSlot slot), and $\mathbb{S}_\mathcal{F}$ and $\mathbb{A}_\mathcal{F}$ are languages for further restrictions and actions (for frame-base formalisms often Lisp). Formalism $\mathcal{F}$ is then a set of ontologies and it is possible to generate it by the grammar.

Analogously, ontology is defined as a 6-tuple $(\mathcal{C}, \mathcal{R}, \phi_\mathcal{C}, \phi_\mathcal{R}, \mathcal{S}, \mathcal{A})$, where $\mathcal{C}$ is a set of concepts (e.g. tree, car, wheel), $\mathcal{R}$ is a set of relations between concepts (Has-wheel), $\phi_\mathcal{C}$ is a function $\phi_\mathcal{C} : \mathcal{C} \rightarrow \mathcal{C}_\mathcal{F}$ (car $\rightarrow$ class, wheel $\rightarrow$ class), $\phi_\mathcal{R}$ is a function $\phi_\mathcal{R} : \mathcal{R} \rightarrow \mathcal{R}_\mathcal{F}$ (Has-wheel $\rightarrow$ HasSlot), $\mathcal{S}$ is a set of restrictions (e.g. *car has up to 4 wheels*), and $\mathcal{A}$ is set of actions.

In the thesis, a list of common formalisms is described together with definitions of their grammars. To ease a basic comparison, a running example is developed for the most of the formalisms.

The approach to ontology transformation presented in this thesis is a syntactical transformation of ontologies using an internal model, consisting of concepts and binary relations between them. The most important feature is a tendency to define an abstract formalism with only few symbols. This formalism is called *Generalised Ontology Formalism* (GOF). Other attempts to solve ontology migration between multiple formalisms tend to using rich languages covering all features of desired formalisms. Such language become obsolete, whenever a new formalism with a new construct appears. On the contrary, GOF lowers the limitations of the formalism in order to be able to describe a wide range of possible situations, even not yet investigated. This concept does not require extension of the model language for every new supported formalism.

GOF consists of one kind of concepts and six relations: instanceOf ($\multimap$), subclassOf ($\blacktriangleright$), has-domain ($\blacktriangleleft$), has-range ($\square$), propertyOf ($\lessdot$), and has-value ($\rightarrow$). All well-known concepts (class, property etc.) are expressed as a combination of the relations. In the thesis the basic constructs are described.

Every supported formalism is represented by a *gate*, which converts ontology in its native formalism to GOF and vice versa. Each gate has also defined a metamodel of the formalism in GOF; ontology describing concepts used in the particular formalism (Formalism Specific Ontology, FSO).

There are two kinds of a transformation. The basic one translates ontology from the source formalism to GOF and then to the target one. Structures in GOF are recognised by the target gate and the corresponding native constructs are formed. The GOF framework offers a universal library called *mapping engine*, which makes use of a set of valid rules in form (class $\blacktriangleright$ class). The rules are evaluated and concepts in GOF are mapped to concepts in FSO: skoda $\blacktriangleright$ car, both skoda and car are mapped to class. If there exists no such mapping covering all concepts and satisfying all rules, relations are ignored and a mapping with least information loss is found. Although the algorithm is non-polynomial, the mapping is very fast thanks to a very small variable part.

For a pair of formalism, between which a transformation is done frequently, a mapping between metamodels can be specified. Such information helps to achieve more precise results and omits the mapping part of the transformation. This method is called *informed transformation*; the prior one is then called *uninformed*.

Quality of ontology transformation $\tau_{s,t}$ between formalisms $\mathcal{F}_s, \mathcal{F}_t$ has been investigated. Ontologies can be compared only in the same formalism and thus only double transformation (source $\rightarrow$ target $\rightarrow$ source) is considered. To be able to compare ontologies a $\prec$ relation has been defined between ontologies (analogous to $\subset$ between sets). If a transformation is *purely lossy*, e.g. no infor-

mation is added, it keeps the $\prec$ relation. The aspect, which forbids more more restrictive conclusions is an approximation of constructs, which are present in the source formalism and not in the target one.

Similarly, the relation $\prec$ is defined between ontology grammars. It has been shown that a lossless transformation between formalisms s, t can be carried out only if $\Psi_s \prec \Psi_t$, i.e. $\tau_{t,s}(\tau_{s,t}(\Omega_s)) \prec \Omega_s$.

GOF allows performing set operations. Union and difference have been defined; join or change detection between two ontologies become trivial tasks.

As an implementation platform has been selected SumatraTT, developed at the department of cybernetics under leadership of author of this thesis from year 2001. An automatic generating of SumatraTT modules from gates has been added to GOF.

4 THESIS EVALUATION

The individual requirements of the thesis statement were successfully fulfilled. The results and research contribution of the author expressed in this thesis comprise:

1. **ontology definition** A formal definition of *ontology* Ω and *ontology formalism* $\mathcal{F}$ have been provided. A missing concept has been found and called *ontology grammar* Ψ. These three terms form a theoretical framework for the thesis. The definitions clearly separate a structural part of ontology (and grammar) from the procedural one.

2. **ontology transformation definition** A transformation $\tau_{s,t}$ has been defined together with explanation how to compare the results. Both ontology and ontology grammar transformations were described.

3. **demands on properties of transformation** In order to investigate properties of transformation, a relation $\prec$ was defined for both ontology and grammar and conclusions were drawn about quality of results. Conditions for lossless informations have been described.

4. **generalised formalism** The generalised formalism have been developed and defined in the previously mentioned theoretical framework. A simple formalism was chosen with one kind of concepts and six relations. Required information is encoded in a combination of the used relations.

5. **individual transformations** Gates of GOF were developed for the most important formalisms. For the most important formalisms were developed their metamodel. This metamodels show differences between formalisms and are used in informed transformation.

6. **analysis of transformations** A special attention was dedicated to a transformation from OWL or DAML to OCML and a set of problems with incompatibility was solved, including subproperty and a chain of instanceOf relations. From OWL to OCML has been implemented an informed transformation.

7. **implementation** GOF has been implemented and its gates were incorporated into set of SumatraTT modules in order to provide user-friendly general interface. GOF has been tested on ontologies ranging from several concepts to upper ontologies SUMO and Cyc with tens of thousands of concepts. The tests show successful transformation of all the structural information.

8. **evaluation** The objectives of this thesis were accomplished. Based on developed theoretical framework for ontologies and transformations, generalised ontology formalism has been designed and implemented. The original ideas have been verified in experiments, where GOF successfully transformed all structural information in all the trial ontologies. A difference exists in approximation of restriction, because the informed transformation can more precisely describe the meaning of the construct.

The most important and original research contributions are definitions of ontology and accompanying terms, the generalised ontology formalism, meta-models of the most important formalisms, and various ideas incorporated into the SumatarTT system.

5 CONCLUSIONS

This thesis presents a new ontology formalism for ontology sharing, called Generalised Ontology Formalism (GOF), consisting of six relations between concepts. Its purpose is supporting migration of ontologies between formalisms with as small as possible information loss. Among others, GOF provides a means for defining meta-models of formalisms and thus it makes possible to study and compare their expressive power.

The theoretical part of the thesis introduces definitions of *ontology*, *ontology grammar*, and *formalism*. These definitions allow distinguishing between the set of expressing constructs (ontology grammar), which make possible to encode ontology, and the set of all ontologies utterable by means of this set of expressing constructs (formalism). Such a theoretical basis provides means for an explicit separation of the structural part of ontology from the procedural one. All common ontology formalisms have been described by means of this theoretical framework.

A framework for converting the structural part of ontology between different formalisms has been designed, implemented, and verified. The main idea consists in expressing the ontology by means of GOF. Thus, the ontology is transformed between respective formalisms in two steps – first from the source formalism into GOF and then to the target formalism. This allows migration of the structural part between any supported formalisms.

The transformation have been analysed from the point of view of information loss during the conversion. The conditions for achieving lossless transformation were stated. Two different methods of ontology transformations – called *informed* and *non-informed* ones – were proposed.

For a pair of formalisms, between which the migration is expected to be carried out frequently, specific (informed) transformation can be defined. The informed transformation makes use of the knowledge of both the source and the target formalism meta-models. Thus, a set of transformation rules, which is specific for the given pair of formalisms, can be prepared and utilised in the course of the particular conversion. Because of its specificity, the informed transformation shall provide better accuracy of ontology conversion than the non-informed one, which does not use the knowledge of mapping between the source and target formalism meta-models.

The non-informed transformation makes possible to quickly include a new formalism into the framework. The reason is that only $2n$ transformations are necessary for migrating between any two of n formalisms. On the other hand, $2\binom{n}{2}$ transformations are necessary in the case of the informed one.

Particular formalisms are processed by so called *gates*, which transform ontologies from the particular formalism to GOF and vice-versa. For each gate, the respective Formalism-Specific Ontology (FSO) has been developed in terms of GOF, which defines the set of meta-level concepts used by the formalism (concepts like *Class*, *Instance*, etc.).

It has been shown, that GOF can handle all the structures, which occur in all common ontology formalisms, using various combinations of GOF relations. An advantage of this simple formalism is the ability to ignore relations, which are not recognised by a particular gate, i.e. they have no corresponding analogy in the respective formalism. In this way, ontologies can migrate between formalisms of very different expressive capabilities without a need of a purposefully written converter.

The generic data processing system SumatraTT, which was designed by the author of this thesis few years ago, has been chosen as the implementation basis for the theoretical framework introduced by this thesis. SumatraTT has been equipped with an algorithm making possible to automatically generate SumatraTT modules from descriptions of respective gates. The respective gates' FSOs are included for purposes of the informed transformation.

The whole framework was verified on a number of ontologies of various size

including SUMO and Cyc, which is the largest publicly available ontology.

GOF is planned also as a platform for annotating an archive of the *Best Patterns* of using SumatraTT modules. Thus, SumatraTT is not only a means for implementing knowledge management applications, but it becomes an application domain itself.

6 SELECTED REFERENCES

[1] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, May 2001.

[2] Pim Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, Tweente University, 1997.

[3] Hans Chalupsky. OntoMorph: A Translation System for Symbolic Knowledge. In *Principles of Knowledge Representation and Reasoning*, pages 471–482, 2000.

[4] Jerome Euzenat and Heiner Stuckenschmidt. Family of Languages' Approach to Semantic Interoperability, 2001.

[5] Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowl. Acquis.*, 5(2):199–220, 1993.

[6] Thomas R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.

[7] Marvin Minsky. A Framework for Representing Knowledge. *The Psychology of Computer Vision, McGraw-Hill*, 1975.

[8] Enrico Motta. *Reusable Components for Knowledge Modelling: Case Studies in Parametric Design Problem Solving*. IOS Press, 1999.

[9] Marek Obitko. Ontologies Description and Applications. Technical Report GL–126/01, Czech Technical University, Department of Cybernetics, Technická 2, 166 27 Prague 6, 2001.

[10] John F. Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole Publishing Co., 2000.

[11] Heiner Stuckenschmidt. Ontology-Based Information Sharing in Weakly Structured Environments, 2002.

[12] Monika Žáková. Semantic Annotations. Master's thesis, The Czech Technical University in Prague, 2005.

7 AUTHOR'S PUBLICATIONS RELATED TO THE THESIS

7.1 Chapter in Book

[13] Dunja Mladenic, Nada Lavrac, Marko Bohanec, and Steve Moyle, editors. *Data Mining and Decision Support: Integration and Collaboration*. Kluwer Academic Publishers, 2003. ISBN 1-4020-7388-7. Chapter PREPROCESSING FOR DATA MINING AND DECISION SUPPORT – **50 %**.

8 Conference Paper

[14] Petr Aubrecht. MINDANAO — ETL proces řízený meta daty. In Olga Štěpánková Jan Rauch, editor, *Znalosti 2001*. Vysoká Škola Ekonomická v Praze, 2001. **100 %**.

[15] Petr Aubrecht and Zdeněk Kouba. Metadata Driven Data Transformation. In *SCI 2001*, volume I, pages 332–336. International Institute of Informatics and Systemics and IEEE Computer Society, 2001. **50 %**.

[16] Petr Aubrecht and Zdeněk Kouba. A Universal Data Pre-processing System. In Luboš Popelínský, editor, *DATAKON 2003*. Masaryk University in Brno, 2003. **50 %**.

[17] Petr Aubrecht and Luboš Král. Ontology Formalism Transformation. In Fernando Galindo, Makoto Takizawa, and Roland Traunmüller, editors, *Database and Expert Systems Applications – DEXA 2004*, pages 95–99. IEEE Computer Society, September 2004. **50 %**.

[18] Petr Aubrecht, Petr Mikšovský, and Zdeněk Kouba. Metadata Driven Data Pre-processing for Data Mining. In *Proceedings of DATESO 2003*, pages 54–61, Desná-Černá ŘíčkaX, 2003. VŠB–Technical University of Ostrava. **33 %**.

[19] Petr Aubrecht, Petr Mikšovský, and Luboš Král. SumatraTT: a Generic Data Pre-processing System. In *Database and Expert Systems Applications*, pages 120–124, Heidelberg, 2003. Springer. **33 %**.

[20] Petr Aubrecht, Olga Štěpánková, and Zdeněk Kouba. Data Pre-processing with SumatraTT. In *Intelligent and Adaptive Systems in Medicine*, pages 61–76, Praha, 2003. CVUT FEL Praha. **33 %**.

[21] Petr Aubrecht and Monika Žáková. Ontology Formalism Transformation. In Karel Ježek, editor, *DATAKON 2004*, pages 191–200. Masaryk University in Brno, 2004. **50 %**.

[22] Petr Aubrecht, Monika Žáková, and Zdeněk Kouba. Ontology Transformation Using Generalised Formalism. In *Znalosti 2005 – sborník příspěvků 4. ročníku konference*, pages p. 154–161. VŠB-TUO, 2005. (in Czech) **33 %**.

[23] Petr Aubrecht, Filip Železný, Petr Mikšovský, and Olga Štěpánková. Progress in SumatraTT: ILP Connectivity and More New Features. In *Data Mining and Decision Support in Action! (subconference of Information Society IS 2001)*, volume 1, pages 107–110. Ljubljana : Institut Jozef Stefan, 2001. **25 %**.

[24] Petr Aubrecht, Filip Železný, Petr Mikšovský, and Olga Štěpánková. SumatraTT: Towards a Universal Data Preprocessor. In *Cybernetics and Systems 2002*, volume II, pages 818–823, Vienna, 2002. Austrian Society for Cybernetics Studies. **25 %**.

[25] Olga Štěpánková, Petr Aubrecht, Zdeněk Kouba, and Petr Mikšovský. Preprocessing for Data Mining and Decision Support. In *Data Mining and Decision Support: Integration and Collaboration*, pages p. 107–117, Dordrecht, 2003. Kluwer Academic Publishers. **25 %**.

[26] Olga Štěpánková, Zdeněk Kouba, Petr Mikšovský, and Petr Aubrecht. Předzpracování dat pro dobývání znalostí: metody a nástroje. In *Znalosti 2003 – sborník příspěvků 2. ročníku konference*, pages p. 21–22. VŠB-TUO, 2003. (in Czech) **25 %**.

[27] Olga Štěpánková, Štěpán Lauryn, Petr Aubrecht, Jiří Klema, Petr Mikšovský, and Lenka Nováková. Data Mining for Resource Allocation: A Case Study. In *Intelligent Methods for Quality Improvement in Industrial Practice*, volume 1, pages 94–10, Prague, 2002. CTU FEE, Department of Cybernetics, The Gerstner Laboratory. **17 %**.

[28] Filip Železný, Petr Aubrecht, and Petr Mikšovský. Connecting Sumatra to Aleph. In Christophe Giraud-Carrier, Nada Lavrač, Steve Moyle, and Branko Kavšek, editors, *Integrating Aspects of Data Mining, Decision Support and Meta-Learning: Internal SolEuNet Session*, pages 43–52. ECML/PKDD'01 workshop notes, September 2001. **33 %**.

9 Research Reports

[29] Petr Aubrecht. Sumatra Basics. Technical Report GL–121/00, Czech Technical University, Department of Cybernetics, Technická 2, 166 27 Prague 6, December 2000. **100 %**.

[30] Petr Aubrecht. Tutorial of Sumatra Embedding. Technical Report GL–101/00, Czech Technical University, Department of Cybernetics, Technická 2, 166 27 Prague 6, October 2000. **100 %**.

[31] Petr Aubrecht. Specification of SumatraTT. Technical Report K333–2/01, Czech Technical University, Department of Cybernetics, Technická 2, 166 27 Prague 6, 2001. **100 %**.

[32] Petr Aubrecht. SumatraTT — předzpracování dat. Research Report K333–1/01, Czech Technical University, Department of Cybernetics, Technická 2, 166 27 Prague 6, July 2001. **100 %**.

[33] Petr Aubrecht. Data Mining – Automatic Data Preproceesing. In *Proceedings of Workshop 2003*, volume vol. A, pages 278–279, Prague, 2003. CTU. **100 %**.

[34] Petr Aubrecht, Filip Železný, Petr Mikšovský, and Olga Štěpánková. SumatraTT: ILP Connectivity and Additional Features. Research Report K333–10/01, Czech Technical University, Department of Cybernetics, Technická 2, 166 27 Prague 6, 2001. **25 %**.

[35] Petr Mikšovský, Petr Aubrecht, and Olga Štěpánková. Spa Dataset: First Data Exploration Review. Technical report, Czech Technical University, Department of Cybernetics, Technická 2, 166 27 Prague 6, 2001. **33 %**.

[36] Filip Železný, Petr Aubrecht, and Petr Mikšovský. Connecting Sumatra to Aleph and Other ILP Systems. Research Report K333–4/01, Czech Technical University, Department of Cybernetics, Technická 2, 166 27 Prague 6, May 2001. **33 %**.

[37] Bernard Ženko and Petr Aubrecht. Experiments with the Pilot Version of SumatraTT—A Case Study. Research Report K333–6/01, Czech Technical University, Department of Cybernetics, Technická 2, 166 27 Prague 6, May 2001. **50 %**.

10 Invited Lecture

[38] Petr Aubrecht, Olga Štěpánková, and Lenka Nováková. SumatraTT – Data Transformation Tool for DM. Invited lecture in *6th IFIP International Conference on Information Technology for BALANCED AUTOMATION SYSTEMS in Manufacturing and Services BASYS '04*, Vienna, Austria, 2004. **33 %**.

[39] Olga Štěpánková, Zdeněk Kouba, Petr Mikšovský, and Petr Aubrecht. Předzpracování dat pro data mining: metody a nástroje. Invited lecture in *Znalosti 2003*, VŠB - Technická univerzita Ostrava, 2003. **25 %**.

11 CITATIONS

- Petr Aubrecht and Zdeněk Kouba. Metadata Driven Data Transformation. In *SCI 2001*, volume I, pages 332–336. International Institute of Informatics and Systemics and IEEE Computer Society, 2001

- Olga Štěpánková, Štěpán Lauryn, Petr Aubrecht, Jiří Klema, Petr Mikšovský, and Lenka Nováková. Data Mining for Resource Allocation: A Case Study. In *Intelligent Methods for Quality Improvement in Industrial Practice*, volume 1, pages 94–10, Prague, 2002. CTU FEE, Department of Cybernetics, The Gerstner Laboratory.

  – Hendrik Blockeel and Steve Moyle. Centralized Model Evaluation For Collaborative Data Mining. In M. Grobelnik, D. Mladenic, M. Bohanec, and M. Gams, editors, *Proceedings A of the 5th International Multi-Conference Information Society 2002 – Data Mining and Data Warehousing/Intelligent Systems*, pages 100–103. Jozef Stefan Institute, Ljubljana, Slovenia, 2002.

- Petr Aubrecht and Filip Železný and Petr Mikšovský, and Olga Štěpánková. SumatraTT: Towards a Universal Data Preprocessor. In: *Cybernetics and Systems 2002, Proceedings of 16th European Meeting on Cybernetics and Systems Research*, volume II, pages 818–824, Vienna, 2002. Austrian Society for Cybernetics Studies.

  – Lenka Lhotská and Marcela Fejtová and Jan Macek, and Daniel Novák. Biological Data Preprocessing: A Case Study. In *Intelligent and Adaptive Systems in Medicine*, pages 77–99, Prague, 2003. CTU FEE.

12 SUMMARY

This dissertation thesis is focused on ontologies and transformations of ontologies between formalisms. The current vague ontology definitions (Gruber, Borst, Sowa) are replaced by a formal definition. The term *formalism* is commonly used for both a description of available constructs and a set of ontologies. This ambiguity has been removed by introducing term *ontology grammar*, inspired by similarity with formal languages. The grammar is defined as 5-tuple $(\mathcal{C}_\mathcal{F}, \mathcal{R}_\mathcal{F}, \mathcal{S}_\mathcal{F}, \mathbb{S}_\mathcal{F}, \mathbb{A}_\mathcal{F})$, where $\mathcal{C}_\mathcal{F}$ is a set of concepts of the formalism, $\mathcal{R}_\mathcal{F}$ is a set of relations, $\mathcal{S}_\mathcal{F}$ is a set of structural restrictions, and $\mathbb{S}_\mathcal{F}$ and $\mathbb{A}_\mathcal{F}$ are languages for further restrictions and actions. Formalism is then a set of ontologies and it is possible to generate it by the grammar. In the thesis, a list of common formalisms is described together with definitions of their grammars. To ease a basic comparison, a running example is developed for the most of the formalisms.

The main topic of the thesis is a transformation of ontologies between formalisms. For this objective there is important the clear separation of structural part of grammar from the procedural one. The procedural part has usually expression power of the Turing machine (often LISP) and thus it is unable to translate. Also possibility of lossless transformation has been investigated. It is feasible only if there exists a injective function between the grammars. In the other cases the unsupported features are either approximated or ignored.

For ontology transformation between very different formalisms there has been designed *Generalised Ontology Formalism* (GOF). GOF consists of one kind of concepts and six relations: instanceOf ($\multimap$), subclassOf ($\rightarrow\!\!\blacktriangleright$), has-domain ($\rightarrow\!\!\blacktriangleleft$), has-range ($\rightarrow\!\!\square$), propertyOf ($\rightarrow\!\!\lessdot$), and has-value ($\rightarrow$). Ontologies are translated from their source form into GOF and then to the target formalism. For every supported formalism so called *gate* has been developed. A gate transforms ontologies from native formalisms to GOF and vice versa. This method allows fast adding of a new formalism.

As an implementation platform has been selected SumatraTT, developed at the department of cybernetics under leadership of author of this thesis from year 2001. An automatic generating of SumatraTT modules from gates has been added to GOF.

GOF has been also used for describing of metamodels of the described formalisms. These metamodels allow comparing of the formalisms. For a pair of formalisms, it is possible to define a mapping between the corresponding metamodels. It make possible transforming with less information loss.

Both methods – with and without mapping between metamodels – have been tested on ontologies ranging from several concepts to upper ontologies SUMO and Cyc with tens of thousands of concepts. All structural information has been successfully translated.

13 SHRNUTÍ

Předkládaná disertační práce se zabývá ontologiemi a převodem ontologií mezi formalismy. Stávající definice ontologie, které jsou velmi vágní (Gruber, Borst, Sowa; mluví pouze o konceptualizaci), byly nahrazeny formálním popisem. Přitom se zjistilo, že termín *formalismus* je používán jak pro popis přípustných konstrukcí, tak zároveň pro vyjádření množiny ontologií. Tato nejasnost byla odstaněna zavedením pojmu *ontologická gramatika*, inspirovaná podobností s formálními jazyky. Gramatika je zavedena jako pětice $(\mathcal{C}_\mathcal{F}, \mathcal{R}_\mathcal{F}, \mathcal{S}_\mathcal{F}, \mathbb{S}_\mathcal{F}, \mathbb{A}_\mathcal{F})$, kde $\mathcal{C}_\mathcal{F}$ je množina konceptů daného formalismu, $\mathcal{R}_\mathcal{F}$ množina relací, $\mathcal{S}_\mathcal{F}$ je množina strukturálních omezení a $\mathbb{S}_\mathcal{F}$ a $\mathbb{A}_\mathcal{F}$ jsou jazyky pro popis omezení (restrictions) a akcí. Formalismus je potom množina ontologií a je jí možno generovat gramatikou. V disertaci je uveden výčet používaných formalismů včetně definic jejich gramatik. Pro základní porovnání byl vypracován ukázkový příklad ve většině těchto formalismech.

Nosným tématem práce je transformace ontologií mezi formalismy. Tato transformace využívá jasného oddělení strukturální části ontologií od procedurální. Procedurální část má obvykle vyjadřovací sílu Turingova stroje (často Lisp) a nelze tedy převádět. Předmětem transformace tedy může být pouze stukturální část. Byla zkoumána možnost bezeztrátově transformace. To je možné tehdy, jestliže existuje zobrazení mezi gramatikami. V ostatních případech jsou nepodporované konstrukce aproximovány nebo ignorovány.

Pro účely transformace ontologií mezi velmi odlišnými formalismy byl vyvinut speciální transportní formalismus, nazvaný *Generalised Ontology Formalism* (GOF). GOF má jediný typ konceptu a šest relací: instanceOf ($\multimap$), subclassOf ($\blacktriangleright$), has-domain ($\blacktriangleleft$), has-range ($\square$), propertyOf ($\lessdot$) a has-value ($\rightarrow$). Ontologie ve zdrojovém formalismu jsou převedeny do interní formy (GOF) a posléze do cílového formalismu.

Pro každý podporovaný formalismus byla vytvořena tzv. brána, která nativní ontologii převede do vnitří reprezentace a naopak. Tento způsob dovoluje rychlé přidání nového formalismu – stačí vytvořit odpovídající bránu. Jako prostředí pro implementaci byl zvolen systém SumatraTT. Tento systém je vyvíjen na katedře kybernetiky pod vedením autora této práce od roku 2001. GOF byl doplněn o automatické generování modulů pro SumatraTT z dostupných bran.

GOF byl použit také pro vytvoření metamodelu podporovaných formalismů. Tyto metamodely dovolují porovnání struktury formalismů. Pro dvojici formalismů, mezi kterými se často převádí, lze definovat mapování mezi odpovídajícími metamodely. To dovolí spolehlivější transformaci s menší ztrátou informace.

Obě uvedené metody byly testovány na řadě ontologií různých velikostí, od jednotek konceptů až po tzv. upper ontologie SUMO a Cyc řádově o desítkách tisíc konceptů. Veškerá strukturální informace byla úspěšně převedena.