Elena Bellodi, Alessio Bonfietti (Eds.)

AI*IA 2015 DC

Proceedings of the Doctoral Consortium (DC) co-located with the 14th Conference of the Italian Association for Artificial Intelligence (AI*IA 2015)

Ferrara, Italy, September 23-24, 2015

Copyright ©2015 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. Re-publication of material from this volume requires permission by the copyright owners. This volume is published and copyrighted by its editors.

Editors' address:
University of Ferrara
Department of Engineering (ENDIF)
Via Saragat, 1
44122 Ferrara, Italy
elena.bellodi@unife.it

University of Bologna
Department of Computer Science and Engineering (DISI)
Via Zamboni, 33
40126 Bologna, Italy
alessio.bonfietti@unibo.it

Preface

This CEUR volume contains the research papers accepted for presentation at the Doctoral Consortium (DC) held in Ferrara, Italy, the 23rd and 24th of September 2015, as an event included in the 14th Conference of the Italian Association for Artificial Intelligence (AI*IA 2015).

The AI*IA 2015 Doctoral Consortium was designed for students enrolled in a Ph.D. program and for students in a Master's program and interested in doctoral studies.

The Consortium provided doctoral students with the opportunity to present their research directions in the field of Artificial Intelligence and being involved into state-of-the-art research, through discussion sessions.

This year eleven research papers were submitted and nine selected for presentation. Each paper was subject to reviews by three different members of the Program Committee. Each paper was assigned 25 minutes, 20 for presentation and 5 for discussion.

The Doctoral Consortium conferred the "Doctoral Consortium Best Paper Award" to the author of the Doctoral Consortium best paper (Ontology Based Semantic Image Interpretation by Ivan Donadello), based on both the reviewers' votes and the Ph.D. students' votes the day of the presentations.

We would like to thank all the people involved in the organization of the event: the AI*IA Conference Chairs, Evelina Lamma, Fabrizio Riguzzi and Marco Gavanelli, who provided support and advice throughout the event organization; the Program Committee (see next page for the listing), who took care of the reviews and participated in the event; all the Ph.D. students who submitted their work to the Doctoral Consortium.

Ferrara, October 2015

Elena Bellodi
Alessio Bonfietti

Programme Chairs

Elena Bellodi, University of Ferrara, Italy
Alessio Bonfietti, University of Bologna, Italy

Programme Committee

Stefania Bandini, CSAI - Complex Systems & Artificial Intelligence Research Center
Nicola Basilico, University of Milano
Federico Bergenti, University of Parma
Francesco Buccafurri, University of Reggio Calabria
Stefano Cagnoni, University of Parma
Luigia Carlucci Aiello, University of Roma La Sapienza
Antonio Chella, University of Palermo
Stefania Costantini, University of L'Aquila
Nicola Di Mauro, University of Bari
Francesco Donini, Tuscia University
Floriana Esposito, University of Bari
Stefano Ferilli, University of Bari
Nicola Guarino, ISTC-CNR
Luca Iocchi, University of Roma La Sapienza
Nicola Leone, University of Calabria
Chendong Li, Dell
Sara Manzoni, CSAI - Complex Systems & Artificial Intelligence Research Center
Paola Mello, University of Bologna
Alfredo Milani, University of Perugia
Michela Milano, University of Bologna
Stefania Montani, University of Piemonte Orientale
Alessandro Moschitti, University of Trento
Angelo Oddi, ISTC-CNR
Andrea Omicini, University of Bologna
Maria Teresa Pazienza, University of Roma Tor Vergata
Roberto Pirrone, University of Palermo
Gian Luca Pozzato, University of Torino
Luca Pulina, University of Sassari
Daniele P. Radicioni, University of Torino
Francesco Ricca, University of Calabria
Andrea Roli, University of Bologna
Salvatore Ruggieri, University of Pisa
Ken Satoh, National Institute of Informatics, Sokendai
Andrea Schaerf, University of Udine
Floriano Scioscia, Politecnico of Bari
Giovanni Semeraro, University of Bari
Eloisa Vargiu, Barcelona Digital Technology Center
Marco Villani, University of Modena and Reggio Emilia

Contents

Tableau Reasoners for Probabilistic Ontologies Exploiting Logic Programming Techniques

Riccardo Zese[1], Elena Bellodi[1], Fabrizio Riguzzi[2], and Evelina Lamma[1]

[1] Dipartimento di Ingegneria – University of Ferrara
Via Saragat 1, I-44122, Ferrara, Italy
[2] Dipartimento di Matematica e Informatica – University of Ferrara
Via Saragat 1, I-44122, Ferrara, Italy
{riccardo.zese,elena.bellodi,evelina.lamma,fabrizio.riguzzi}@unife.it

Abstract. The adoption of Description Logics for modeling real world domains within the Semantic Web is exponentially increased in the last years, also due to the availability of a large number of reasoning algorithms. Most of them exploit the tableau algorithm which has to manage non-determinism, a feature that is not easy to handle using procedural languages such as Java or C++. Reasoning on real world domains also requires the capability of managing probabilistic and uncertain information. We thus present TRILL, for "Tableau Reasoner for descrIption Logics in proLog" and TRILL^P, for "TRILL powered by Pinpointing formulas", which implement the tableau algorithm and return the probability of queries. TRILL^P, instead of the set of explanations for a query, computes a Boolean formula representing them, speeding up the computation.

Introduction

The Semantic Web aims at making information regarding real world domains available in a form that is understandable by machines [7]. The World Wide Web Consortium is working for realizing this vision by supporting the development of the Web Ontology Language (OWL), a family of knowledge representation formalisms for defining ontologies based on Description Logics (DLs). Moreover, uncertain information is intrinsic to real world domains, thus the combination of probability and logic theories becomes of foremost importance.

Efficient DL reasoners, such us Pellet, RacerPro and HermiT, are able to execute inference on the modeled ontologies, but only a few number of reasoners are able to manage probabilistic information. One of the most common approaches for reasoning is the tableau algorithm that exploits some non-deterministic expansion rules. This requires the developers to implement a search strategy in an or-branching search space able to explore all the non-deterministic choices done during the inference in order to find all the possible explanations for the query.

In this paper, we present two systems which implement a tableau reasoner in Prolog: TRILL for "Tableau Reasoner for descrIption Logics in proLog" and TRILL^P for "TRILL powered by Pinpointing formulas", able to reason on

$\mathcal{SHOIQ}$ DL and on $\mathcal{ALC}$ DL respectively. Prolog search strategy is exploited for taking into account the non-determinism of the tableau rules. They use the Thea2 library [18] for translating OWL files into a Prolog representation in which each axiom is mapped into a fact. TRILL and TRILL^P can check the consistency of a concept and the entailment of an axiom from an ontology, and can compute the probability of the entailment following DISPONTE [13], a semantics for probabilistic ontologies based on the Distribution Semantics [15], one of the most widespread approaches in probabilistic logic programming. The availability of a Prolog implementation of a DL reasoner which follows DISPONTE will also facilitate the development of probabilistic reasoners that can integrate probabilistic logic programming with probabilistic DLs.

Description Logics

DLs are knowledge representation formalisms that are at the basis of the Semantic Web [1, 2] and are used for modeling ontologies. They possess nice computational properties such as decidability and/or low complexity.

Usually, DLs' syntax is based on concepts and roles which correspond respectively to sets of individuals and sets of pairs of individuals of the domain. We now describe $\mathcal{ALC}$ and we refer to [11] for a description of $\mathcal{SHOIQ}$.

Let $\mathbf{C}$, $\mathbf{R}$ and $\mathbf{I}$ be sets of *atomic concepts*, *atomic roles* and *individuals*, respectively. *Concepts* are defined by induction as follows. Each $C \in \mathbf{C}$ is a concept, $\bot$ and $\top$ are concepts. If C, C_1 and C_2 are concepts and $R \in \mathbf{R}$, then $(C_1 \sqcap C_2)$, $(C_1 \sqcup C_2)$ and $\neg C$ are concepts, as well as $\exists R.C$ and $\forall R.C$. A *TBox* $\mathcal{T}$ is a finite set of *concept inclusion axioms* $C \sqsubseteq D$, where C and D are concepts. We use $C \equiv D$ to abbreviate the conjunction of $C \sqsubseteq D$ and $D \sqsubseteq C$. An *ABox* $\mathcal{A}$ is a finite set of *concept membership axioms* $a : C$, *role membership axioms* $(a, b) : R$, *equality axioms* $a = b$ and *inequality axioms* $a \neq b$, where $C \in \mathbf{C}$, $R \in \mathbf{R}$ and $a, b \in \mathbf{I}$. A *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ and is usually assigned a semantics in terms of interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty *domain* and $\cdot^{\mathcal{I}}$ is the *interpretation function* that assigns an element in $\Delta^{\mathcal{I}}$ to each $a \in \mathbf{I}$, a subset of $\Delta^{\mathcal{I}}$ to each $C \in \mathbf{A}$ and a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each $R \in \mathbf{R}$.

A query Q over a KB $\mathcal{K}$ is usually an axiom for which we want to test the entailment from the KB, written $\mathcal{K} \models Q$. The entailment test may be reduced to checking the unsatisfiability of a concept in the knowledge base, i.e., the emptiness of the concept. For example, the entailment of the axiom $C \sqsubseteq D$ may be tested by checking the unsatisfiability of the concept $C \sqcap \neg D$ while the entailment of the axiom $a : C$ may be tested by checking the unsatisfiability of $a : \neg C$.

Querying Description Logics KBs

The problem of finding explanations for a query has been investigated by various authors [16, 8, 6, 9]. It was called *axiom pinpointing* in [16] and considered as a non-standard reasoning service useful for tracing derivations and debugging ontologies. In particular, *minimal axiom sets* or *MinAs* for short, also called *explanations*, are introduced in [16].

Definition 1 (MinA). *Let $\mathcal{K}$ be a knowledge base and Q an axiom that follows from it, i.e., $\mathcal{K} \models Q$. We call a set $M \subseteq \mathcal{K}$ a* minimal axiom set *or* MinA *for Q in $\mathcal{K}$ if $M \models Q$ and it is minimal w.r.t. set inclusion.*

The problem of enumerating all MinAs is called MIN-A-ENUM in [16]. ALL-MINAS$(Q, \mathcal{K})$ is the set of all MinAs for query Q in the knowledge base $\mathcal{K}$.

The tableau algorithm is able to returns a single MinA. To solve MIN-A-ENUM, reasoners written in imperative languages, like Pellet [17], have to implement a search strategy in order to explore the entire search space of the possible explanations. A *tableau* is an ABox represented as a graph G where each node corresponds to an individual a and is labeled with the set of concepts to which a belongs. Each edge $\langle a, b\rangle$ in the graph is labeled with the set of roles to which the couple (a, b) belongs. A *tableau algorithm* proves an axiom by refutation, starting from a tableau that contains the negation of the axiom and repeatedly applying a set of consistency preserving *tableau expansion rules* until a clash (i.e., a contradiction) is detected or a clash-free graph is found to which no more rules are applicable. If no clashes are found, the tableau represents a model for the negation of the query, which is thus not entailed. The Prolog language allows developers of reasoning algorithms to exploit Prolog's backtracking facilities instead of implementing a search strategy from scratch.

In [3] the authors consider the problem of finding a *pinpointing formula* instead of ALL-MINAS$(Q, \mathcal{K})$. The pinpointing formula is a monotone Boolean formula in which each Boolean variable corresponds to an axiom of the KB. This formula is built using the variables and the conjunction and disjunction connectives. It compactly encodes the set of all MinAs. Let assume that each axiom E of a KB $\mathcal{K}$ is associated with a propositional variable, indicated with $var(E)$. The set of all propositional variables is indicated with $var(\mathcal{K})$. A valuation ν of a monotone Boolean formula is the set of propositional variables that are true. For a valuation $\nu \subseteq var(\mathcal{K})$, let $\mathcal{K}_\nu := \{t \in \mathcal{K} | var(t) \in \nu\}$.

Definition 2 (Pinpointing formula). *Given a query Q and a KB $\mathcal{K}$, a monotone Boolean formula ϕ over $var(\mathcal{K})$ is called a* pinpointing formula *for Q if for every valuation $\nu \subseteq var(\mathcal{K})$ it holds that $\mathcal{K}_\nu \models Q$ iff ν satisfies ϕ.*

In Lemma 2.4 of [3] the authors proved that the set of all MinAs can be obtained by transforming the pinpoiting formula into DNF and removing disjuncts implying other disjuncts. The example below illustrates axiom pinpointing and the pinpointing formula.

Example 1 (Pinpointing formula). The following KB is inspired by the ontology `people+pets` [12] and show also the Boolean variables associated to axioms: $F_1 = \exists hasAnimal.Pet \sqsubseteq NatureLover$, $F_2 = (kevin, fluffy) : hasAnimal$, $F_3 = (kevin, tom) : hasAnimal$, $F_4 = fluffy : Cat$, $F_5 = tom : Cat$, $F_6 = Cat \sqsubseteq Pet$. It states that individuals that own an animal which is a pet are nature lovers and that *kevin* owns the animals *fluffy* and *tom*, which are cats. Moreover, cats are pets. Let $Q = kevin : NatureLover$ be the query, then $\textsc{All-MinAs}(Q, \mathcal{K})$ $= \{\{F_2, F_4, F_6, F_1\}, \{F_3, F_5, F_6, F_1\}\}$, while the pinpointing formula is $((F_2 \wedge F_4) \vee (F_3 \wedge F_5)) \wedge F_6 \wedge F_1$.

TRILL and TRILL^P

Both TRILL and TRILL^P implement a tableau algorithm, the first solves MIN-A-ENUM while the second computes the pinpointing formula representing the set of MinAs. They can answer concept and role membership queries, subsumption queries and can test the unsatifiability of a concept of the KB or the inconsistency of the entire KB. TRILL and TRILL^P are implemented in Prolog, so the management of the non-determinism of the rules is delegated to the language.

We use the Thea2 library [18] for converting OWL DL KBs into Prolog. Thea2 performs a direct translation of the OWL axioms into Prolog facts. For example, a simple subclass axiom between two named classes $Cat \sqsubseteq Pet$ is written using the `subClassOf/2` predicate as `subClassOf('Cat','Pet')`.

Deterministic and non-deterministic tableau expansion rules are treated differently. Non-deterministic rules are implemented by a predicate that, given the current tableau Tab, returns the list of tableaux created by the application of the rule to Tab. Deterministic rules are implemented by a predicate that, given the current tableau Tab, returns the tableau obtained by the application of the rule to Tab. The computation of $\textsc{All-MinAs}(Q, \mathcal{K})$ is performed by simply calling `findall/3` over the tableau predicate.

The code of TRILL and TRILL^P is available at `https://sites.google.com/a/unife.it/ml/trill`. Experiments presented in [19] show that Prolog is a viable language for implementing DL reasoning algorithms and that their performances are comparable with those of state-of-art reasoners. In order to popularize DISPONTE, we developed a Web application called "TRILL-on-SWISH" and available at `http://trill.lamping.unife.it`. We exploited SWISH [10], a recently proposed Web framework for logic programming that is based on various features and packages of SWI-Prolog. TRILL-on-SWISH is based on SWISH [10] and allows users to write a KB directly in the web page or load it from a URL, and run TRILL to execute queries.

Computing the Probability

The aim of our work is to implement algorithms which can compute the probability of queries to KBs following DISPONTE [13]. DISPONTE applies the distribution semantics [15] of probabilistic logic programming to DLs. A program following this semantics defines a probability distribution over normal logic programs called *worlds*. Then the distribution is extended to a joint distribution over worlds and queries from which the probability of a query is obtained by marginalization.

In DISPONTE, a *probabilistic knowledge base* $\mathcal{K}$ contains a set of *probabilistic axioms* which take the form $p :: E$ where p is a real number in $[0, 1]$ and E is a DL axiom. The probability p can be interpreted as an *epistemic probability*, i.e., as the degree of our belief in the truth of axiom E. For example, a probabilistic concept membership axiom $p :: a : C$ means that we have degree of belief p in $C(a)$. The idea of DISPONTE is to associate independent Boolean random variables to the axioms. To obtain a *world* w we decide whether to include each axiom or not in w. A world therefore is a non probabilistic KB that can be assigned a semantics in the usual way. A query is entailed by a world if it is true in every model of the world.

To compute the probability of queries to KBs following the DISPONTE semantics, we can first perform MIN-A-ENUM. Then the explanations must be made mutually exclusive, so that the probabilities of individual explanations are computed and summed. This can be done by assigning independent Boolean random variables to the axioms contained in the explanations and define the Disjunctive Normal Form Boolean formula f_K which models the set of explanations K. Thus $f_K(\mathbf{X}) = \bigvee_{Ex \in \text{ALL-MINAS(Q,K)}} \bigwedge_{E_i \in Ex} X_i$ where $\mathbf{X} = \{X_i | (E_i) \in \text{ALL-MINAS(Q,K)}\}$ is the set of Boolean random variables. TRILL^P, instead, computes directly a pinpointing formula which is a monotone Boolean formula that represents the set of all MinAs.

Irrespective of which representation of the explanations we choose, a DNF or a general pinpointing formula, we can apply knowledge compilation and transform it into a Binary Decision Diagram (BDD), from which we can compute the probability of the query with a dynamic programming algorithm that is linear in the size of the BDD.

A BDD for a function of Boolean variables is a rooted graph that has one level for each Boolean variable. A node n in a BDD has two children: one corresponding to the 1 value of the variable associated with the level of n, indicated with $child_1(n)$, and one corresponding to the 0 value of the variable, indicated with $child_0(n)$. The leaves store either 0 or 1. A BDD performs a Shannon expansion of the Boolean formula $f(\mathbf{X})$, so that, if X is the variable associated with the root level of a BDD, the formula $f(\mathbf{X})$ can be represented as $f(\mathbf{X}) = X \wedge f^X(\mathbf{X}) \vee \overline{X} \wedge f^{\overline{X}}(\mathbf{X})$ where $f^X(\mathbf{X})$ $(f^{\overline{X}}(\mathbf{X}))$ is the formula obtained by $f(\mathbf{X})$ by setting X to 1 (0). Now the two disjuncts are pairwise exclusive and the probability of $f(\mathbf{X})$ being true can be computed as $P(f(\mathbf{X})) = P(X)P(f^X(\mathbf{X})) + (1-P(X))P(f^{\overline{X}}(\mathbf{X}))$ by knowing the probabilities of the Boolean variables of being true.

Conclusions

In this paper we have presented the algorithm TRILL for reasoning on $\mathcal{SHOIQ}$ KBs and the algorithm TRILL^P for reasoning on $\mathcal{ALC}$ KBs.

In the future we plan to apply various optimizations to our systems in order to better manage the expansion of the tableau. In particular, we plan to carefully choose the rule and node application order. We are also studying an extension of our systems for managing KBs integrating rules and DL axioms. Moreover, we plan to exploit TRILL for implementing algorithms for learning the parameters of probabilistic DISPONTE KBs, along the lines of [4, 5, 14].

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Baader, F., Horrocks, I., Sattler, U.: Description logics. In: Handbook of knowledge representation, chap. 3, pp. 135–179. Elsevier (2008)
3. Baader, F., Peñaloza, R.: Axiom pinpointing in general tableaux. J. Log. Comput. 20(1), 5–34 (2010)
4. Bellodi, E., Riguzzi, F.: Learning the structure of probabilistic logic programs. In: Muggleton, S.H., Tamaddoni-Nezhad, A., Lisi, F.A. (eds.) ILP 2011. LNCS, vol. 7207, pp. 61–75. Springer (2012)
5. Bellodi, E., Riguzzi, F.: Expectation Maximization over binary decision diagrams for probabilistic logic programs. Intel. Data Anal. 17(2), 343–363 (2013)
6. Halaschek-Wiener, C., Kalyanpur, A., Parsia, B.: Extending tableau tracing for ABox updates. Tech. rep., University of Maryland (2006)
7. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. CRCPress (2009)
8. Kalyanpur, A.: Debugging and Repair of OWL Ontologies. Ph.D. thesis, The Graduate School of the University of Maryland (2006)
9. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 267–280. Springer (2007)
10. Lager, T., Wielemaker, J.: Pengines: Web logic programming made easy. TPLP 14(4-5), 539–552 (2014), `http://dx.doi.org/10.1017/S1471068414000192`
11. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. J. Web Sem. 6(4), 291–308 (2008)
12. Patel-Schneider, P, F., Horrocks, I., Bechhofer, S.: Tutorial on OWL (2003)
13. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R.: Epistemic and statistical probabilistic ontologies. In: Bobillo, F., et al. (eds.) URSW 2012. CEUR Workshop Proceedings, vol. 900, pp. 3–14. Sun SITE Central Europe (2012)
14. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R., Cota, G.: Learning probabilistic description logics. In: URSW III. LNCS, vol. 8816, pp. 63–78. Springer (2014)
15. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: ICLP 1995. pp. 715–729. MIT Press (1995)
16. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Gottlob, G., Walsh, T. (eds.) IJCAI 2003. pp. 355–362. Morgan Kaufmann (2003)
17. Sirin, E., Parsia, B., Cuenca-Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. Web Sem. 5(2), 51–53 (2007)
18. Vassiliadis, V., Wielemaker, J., Mungall, C.: Processing OWL2 ontologies using thea: An application of logic programming. In: OWLED 2009. CEUR Workshop Proceedings, vol. 529. CEUR-WS.org (2009)
19. Zese, R., Bellodi, E., Lamma, E., Riguzzi, F., Aguiari, F.: Semantics and inference for probabilistic description logics. In: URSW III. LNCS, vol. 8816, pp. 79–99. Springer (2014)

Learning Probabilistic Ontologies with Distributed Parameter Learning

Giuseppe Cota[1], Riccardo Zese[1], Elena Bellodi[1], Fabrizio Riguzzi[2], and Evelina Lamma[1]

[1] Dipartimento di Ingegneria – University of Ferrara
Via Saragat 1, I-44122, Ferrara, Italy
[2] Dipartimento di Matematica e Informatica – University of Ferrara
Via Saragat 1, I-44122, Ferrara, Italy
[giuseppe.cota,riccardo.zese,elena.bellodi,evelina.lamma,
fabrizio.riguzzi]@unife.it

Abstract. We consider the problem of learning both the structure and the parameters of Probabilistic Description Logics under DISPONTE. DISPONTE ("DIstribution Semantics for Probabilistic ONTologiEs") adapts the distribution semantics for Probabilistic Logic Programming to Description Logics. The system LEAP for "LEArning Probabilistic description logics" learns both the structure and the parameters of DISPONTE knowledge bases (KBs) by exploiting the algorithms CELOE and EDGE. The former stands for "Class Expression Learning for Ontology Engineering" and it is used to generate good candidate axioms to add to the KB, while the latter learns the probabilistic parameters and evaluates the KB. EDGE for "Em over bDds for description loGics paramEter learning" is an algorithm for learning the parameters of probabilistic ontologies from data. In order to contain the computational cost, a distributed version of EDGE called EDGE$^{\mathrm{MR}}$ was developed. EDGE$^{\mathrm{MR}}$ exploits the MapReduce (MR) strategy by means of the Message Passing Interface. In this paper we propose the system LEAP$^{\mathrm{MR}}$. It is a re-engineered version of LEAP which is able to use distributed parallel parameter learning algorithms such as EDGE$^{\mathrm{MR}}$.

Keywords: Probabilistic Description Logics, Structure Learning, Parameter Learning, MapReduce, Message Passing Interface.

1 Introduction

In real world domains the information is often uncertain, hence it is of foremost importance to model uncertainty in representations of the world, including Description Logics (DLs).

In [10, 16, 6] the authors studied the use of probabilistic DLs and various approaches for representing uncertainty in DLs.

Moreover, some works have started to appear about learning the probabilistic parameters or the whole structure of probabilistic ontologies. These are motivated, on one hand, from the fact that specifying the values of the probabilities is a difficult task for humans and data is usually available that could be leveraged for tuning them, and, on the other hand, from the fact that in some domains there exist poor-structured knowledge bases which could be improved [10, 9].

In Probabilistic Logic Programming (PLP) various proposals for representing uncertainty have been presented. One of the most successful approaches is the distribution semantics [15]. In [3, 14, 11] the authors proposed an approach to represent probabilistic axioms in DLs called DISPONTE ("DIstribution Semantics for Probabilistic ONTologiEs"), which adapts the distribution semantics for Probabilistic Logic Programming to DLs.

LEAP [13] for "LEArning Probabilistic description logics" is an algorithm for learning the structure and the parameters of probabilistic DLs following DISPONTE. It combines the learning system CELOE [8] with EDGE [12]. The former, CELOE ("Class Expression Learning for Ontology Engineering"), provides a method to build new (subsumption) axioms that can be added to the KB, while the latter is used to learn the parameters of these probabilistic axioms.

EDGE stands for "Em over bDds for description loGics paramEter learning" and learns the parameters of a probabilistic theory. This algorithm is rather expensive from a computational point of view. Therefore, in order to reduce EDGE running time, we developed EDGE$^{\text{MR}}$ [5]. It represents a distributed implementation of EDGE and uses a simple MapReduce approach based on the Message Passing Interface (MPI).

In this paper we present an evolution of LEAP called LEAP$^{\text{MR}}$ which adapts the LEAP algorithm to use EDGE$^{\text{MR}}$. In addition, due to a software re-engineering effort, it was possible to remove the RMI module used by LEAP. To the best of our knowledge there are no other algorithms that perform distributed structure learning of probabilistic DLs.

The paper is structured as follows. Section 2 introduces Description Logics and summarizes DISPONTE. Sections 3 and 4 briefly describe the EDGE and EDGE$^{\text{MR}}$ algorithms. Section 5 presents LEAP$^{\text{MR}}$. Finally, Section 7 draws conclusions.

2 Description Logics and DISPONTE

Description Logics (DLs) are a family of logic based knowledge representation formalisms which are of particular interest for representing ontologies and for the Semantic Web. For an extensive introduction to DLs we refer to [1, 2].

While DLs are a fragment of first order logic, they are usually represented using a syntax based on concepts and roles. A concept corresponds to a set of individuals while a role corresponds to a set of couples of individuals of the domain.

A query over a KB is usually an axiom for which we want to test the entailment from the KB. The entailment test may be reduced to checking the unsatisfiability of a concept in the KB, i.e., the emptiness of the concept.

DISPONTE [3] ("DIstribution Semantics for Probabilistic ONTologiEs") applies the distribution semantics to probabilistic ontologies [15]. In DISPONTE a *probabilistic knowledge base* $\mathcal{K}$ is a set of certain and probabilistic axioms. *Certain axioms* take the form of regular DL axioms. *Probabilistic axioms* take the form $p :: E$, where p is a real number in $[0,1]$ and E is a DL axiom. A DISPONTE KB defines a distribution over DL KBs called *worlds* assuming that the axioms are independent. Each world w is obtained by including every certain axiom plus a subset of chosen probabilistic axioms.

3 Parameter Learning for Probabilistic DLs

EDGE [12] is a parameter learning algorithm which adapts the algorithm EMBLEM [4], developed for learning the parameters for probabilistic logic programs, to the case of probabilistic DLs under DISPONTE. Inspired by [7], it performs an Expectation-Maximization cycle over Binary Decision Diagrams (BDDs).

EDGE performs supervised parameter learning. It takes as input a DISPONTE KB and a number of positive and negative examples that represent the queries in the form of concept membership axioms, i.e., in the form $a : C$ for an individual a and a class C.

First, EDGE generates, for each query, the BDD encoding its explanations using BUNDLE [14]. Then, EDGE starts the EM cycle in which the steps of Expectation and Maximization are iterated until a local maximum of the log-likelihood (LL) of the examples is reached. The LL of the examples is guaranteed to increase at each iteration. EDGE stops when the difference between the LL of the current iteration and that of the previous one drops below a threshold ϵ or when this difference is below a fraction δ of the previous LL. Finally, EDGE returns the reached LL and the new probabilities for the probabilistic axioms.

EDGE is written in Java, hence it is highly portable. For further information about EDGE please refer to [12].

4 Distributed Parameter Learning for Probabilistic DLs

In this section we briefly describe a parallel version of EDGE that exploits the MapReduce approach in order to compute the parameters. We called this algorithm EDGE$^{\mathrm{MR}}$ [5].

Like most MapReduce frameworks, EDGE$^{\mathrm{MR}}$'s architecture follows a master-slave model. The communication between the master and the slaves is done by means of the Message Passing Interface (MPI).

In a distributed context, the performances depend on the scheduling strategy. In order to evaluate different methods, we developed two scheduling strategies: *single-step scheduling* and *dynamic scheduling*. These are used during the queries computation phase.

Single-step Scheduling if N is the number of the slaves, the master divides the total number of queries into $N+1$ chunks, i.e. the number of slaves plus the master. Then the master begins to compute its queries while, for the other chunks of queries, the master starts a thread for sending each chunk to the corresponding slave. After the master has terminated dealing with its queries, it waits for the results from the slaves. When the slowest slave returns its results to the master, EDGE$^{\text{MR}}$ proceeds to the EM cycle.

Dynamic Scheduling is more flexible and adaptive than single-step scheduling. At first, each machine is assigned a fixed-size chunk of queries in order. Then, if the master ends handling its chunk it just takes the next one, instead, if a slave ends handling its chunk, it asks the master for another one and the master replies by sending a new chunk of queries to the slave. During this phase the master runs a thread listener that waits for the slaves' requests of new chunks and for each request the listener starts a new thread that sends a chunk to the slave which has done the request (to improve the performances this is done through a thread pool). When all the queries are evaluated, EDGE$^{\text{MR}}$ starts the EM cycle.

Experimental results conducted in [5] show that dynamic scheduling has usually better performances than single-step.

5 Structure Learning with Distributed Parameter Learning

LEAP$^{\text{MR}}$ is an evolution of the LEAP system [13]. While the latter exploits EDGE, the first was adapted to be able to perform EDGE$^{\text{MR}}$. Moreover, after a process of software re-engineering it was possible to remove the RMI communication module used by LEAP and therefore reduce some communication overhead.

It performs structure and parameter learning of probabilistic ontologies under DISPONTE by exploiting: (1) CELOE [8] for the structure, and (2) EDGE$^{\text{MR}}$ (Section 4) for the parameters.

CELOE [8] was implemented in Java and belongs to the open-source framework DL-Learner[3]. Let us consider a knowledge base $\mathcal{K}$ and a concept name `Target` whose formal description, i.e. class description, we want to learn. It learns a set of n class expressions C_i $(1 \leq i \leq n)$ from a set of positive and negative examples. Let $\mathcal{K}' = \mathcal{K} \cup \{C\}$ where $\mathcal{K}$ is the background knowledge, we say that a concept C *covers* an example e if $\mathcal{K}' \models e$. The class expressions found are sorted according to a heuristic. Such expressions can be used to generate candidate axioms of the form $C_i \sqsubseteq \texttt{Target}$.

In order to learn an ontology, LEAP$^{\text{MR}}$ first searches for good candidate probabilistic subsumption axioms by means of CELOE, then it performs a greedy search in the space of theories using EDGE$^{\text{MR}}$ to evaluate the theories using the log-likelihood as heuristic.

LEAP$^{\text{MR}}$ takes as input the knowledge base $\mathcal{K}$ and a set of examples, then generates a set of candidate axioms by exploiting CELOE. A first execution of EDGE$^{\text{MR}}$ is applied to $\mathcal{K}$ to compute the initial value of the parameters and of the LL. Then LEAP$^{\text{MR}}$ adds to $\mathcal{K}$ one probabilistic subsumption axiom

[3] `http://dl-learner.org/`

generated from CELOE. After each addition, EDGE$^{\mathrm{MR}}$ is performed on the extended KB to compute the LL of the data and the parameters. If the LL is better than the current best, the new axiom is kept in the knowledge base and the parameter of the probabilistic axiom are updated, otherwise the learned axiom is removed from the ontology and the previous parameters are restored. The final theory is obtained from the union of the initial ontology and the probabilistic axioms learned.

6 Experiments

In order to test how much the exploitation of EDGE$^{\mathrm{MR}}$ can improve the performances of LEAP$^{\mathrm{MR}}$, we did a preliminary test where we considered the Moral[4] KB which qualitatively simulates moral reasoning. It contains 202 individuals and 4710 axioms (22 axioms are probabilistic).

We performed the experiments on a cluster of 64-bit Linux machines with 8-cores Intel Haswell 2.40 GHz CPUs and 2 GB (max) memory allotted to Java per node. We allotted 1, 3, 5, 9 and 17 nodes, where the execution with 1 node corresponds to the execution of LEAP, while for the other configurations we used the dynamic scheduling with chunks containing 3 queries. For each experiment 2 candidate probabilistic axioms are generated by using CELOE and a maximum of 3 explanations per query was set for EDGE$^{\mathrm{MR}}$. Table 1 shows the speedup obtained as a function of the number of machines (nodes). The speedup is the ratio of the running time of 1 worker to the one of n workers. We can note that the speedup is significant even if it is sublinear, showing that a certain amount of overhead (the resources, and thereby the time, spent for the MPI communications) is present.

Dataset	**N. of Nodes**			
	3	**5**	**9**	**9**
Moral	2.3	3.6	6.5	11.0

Table 1. Speedup of LEAP$^{\mathrm{MR}}$ relative to LEAP for Moral KB.

7 Conclusions

The paper presents the algorithm LEAP$^{\mathrm{MR}}$ for learning the structure of probabilistic description logics under DISPONTE. LEAP$^{\mathrm{MR}}$ performs EDGE$^{\mathrm{MR}}$ which is a MapReduce implementation of EDGE, exploiting modern computing infrastructures for performing distributed parameter learning.

We are currently working for distributing both the structure and the parameter learning of probabilistic knowledge bases by exploiting EDGE$^{\mathrm{MR}}$ also during

[4] `https://archive.ics.uci.edu/ml/datasets/Moral+Reasoner`

the building of the class expressions. In particular we would like to distribute the scoring function used to evaluate the obtained refinements.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York, NY, USA (2003)
2. Baader, F., Horrocks, I., Sattler, U.: Description Logics, chap. 3, pp. 135–179. Elsevier Science (2008)
3. Bellodi, E., Lamma, E., Riguzzi, F., Albani, S.: A distribution semantics for probabilistic ontologies. In: International Workshop on Uncertainty Reasoning for the Semantic Web. CEUR Workshop Proceedings, vol. 778, pp. 75–86. Sun SITE Central Europe (2011)
4. Bellodi, E., Riguzzi, F.: Expectation Maximization over Binary Decision Diagrams for probabilistic logic programs. Intell. Data Anal. 17(2), 343–363 (2013)
5. Cota, G., Zese, R., Bellodi, E., Lamma, E., Riguzzi, F.: Distributed parameter learning for probabilistic ontologies (2015), to appear
6. Fleischhacker, D., Völker, J.: Inductive learning of disjointness axioms. In: On the Move to Meaningful Internet Systems: OTM 2011, pp. 680–697. Springer (2011)
7. Ishihata, M., Kameya, Y., Sato, T., Minato, S.: Propositionalizing the EM algorithm by BDDs. In: Late Breaking Papers of the International Conference on Inductive Logic Programming. pp. 44–49 (2008)
8. Lehmann, J., Auer, S., Bühmann, L., Tramp, S.: Class expression learning for ontology engineering. J. Web Semant. 9(1), 71–81 (2011)
9. Minervini, P., d'Amato, C., Fanizzi, N.: Learning probabilistic description logic concepts: Under different assumptions on missing knowledge. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 378–383. ACM (2012)
10. Ochoa-Luna, J.E., Revoredo, K., Cozman, F.G.: Learning probabilistic description logics: A framework and algorithms. In: Advances in Artificial Intelligence, pp. 28–39. Springer (2011)
11. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R.: Epistemic and statistical probabilistic ontologies. In: Uncertainty Reasoning for the Semantic Web. CEUR Workshop Proceedings, vol. 900, pp. 3–14. Sun SITE Central Europe (2012)
12. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R.: Parameter learning for probabilistic ontologies. In: RR 2013. LNCS, vol. 7994, pp. 265–270. Springer Berlin Heidelberg (2013)
13. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R., Cota, G.: Learning probabilistic description logics. In: Uncertainty Reasoning for the Semantic Web III, pp. 63–78. LNCS, Springer International Publishing (2014)
14. Riguzzi, F., Lamma, E., Bellodi, E., Zese, R.: BUNDLE: A reasoner for probabilistic ontologies. In: RR 2013. LNCS, vol. 7994, pp. 183–197. Springer Berlin Heidelberg (2013)
15. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: Proceedings of the 12th International Conference on Logic Programming. pp. 715–729. MIT Press (1995)
16. Völker, J., Niepert, M.: Statistical schema induction. In: The Semantic Web: Research and Applications, pp. 124–138. Springer Berlin Heidelberg (2011)

STOLE: A Reference Ontology for Historical Research Documents

Laura Pandolfo

DIBRIS, Università di Genova, Via Opera Pia, 13 – 16145 Genova – Italy
laura.pandolfo@edu.unige.it

Abstract. Historical documents are a relevant part of cultural heritage. It is well-established that this domain is very complex: data are often heterogeneous, semantically rich, and highly interlinked. For this reason, searching and linking them with related contents represents a challenging task. The use of Semantic Web technologies can provide innovative methods for more effective search and retrieval operations. In this paper we present STOLE, a reference ontology that provides a vocabulary of terms and relations that model the history of Italian public administration' domain. STOLE can be considered a first step towards the development of a knowledge exchange standard on this specific domain.

1 Context and Motivation

Historical texts and documents are considered an important component of cultural heritage. Among other sources, historical newspapers and journals archived in libraries represent a valuable source of information for historians.

In the last years, several portals and digital libraries in the cultural heritage field have been enhanced with Semantic Web (SW) technologies in order to implement methods for more effective search and retrieval operations. In fact, they can offer effective solutions about design and implementation of user-friendly ways to access and query content and meta-data – see [11] for a survey in the historical research domain. The use of SW technologies can improve the productivity of research in historical domain, since they help to identify implicit and explicit knowledge included in the documents, e.g., reference to a historical person contained in a historical source can be discovered and related to other entities, for example events in which that person has been involved in.

One of the main elements of the SW infrastructure are ontologies. They provide a shared and clear representation for a specific domain and they may play a major role in supporting knowledge extraction, knowledge discovery, and data integration processes. An ontology is usually defined as a formal specification of domain knowledge conceptualization. A *Knowledge Base* (KB) is composed of two elements, i.e., a *Terminological Box* (TBox) and an *Assertional Box* (ABox). The TBox represents the intensional knowledge of the KB and it is made up of classes of data and relations among them. In the following, we use the terms TBox and ontology interchangeably to denote the conceptual part of a KB. The

ABox contains extensional knowledge, which is specific to the individuals of the domain.

In this paper we present STOLE[1], a reference ontology that provides a vocabulary of terms and relations with which it is possible to model the history of Italian public administration' domain. STOLE represents a first step towards the development of a knowledge exchange standard on this specific domain. To the best of our knowledge, this is the first time that ontological modeling has been undertaken for this specific domain.

The paper is organized as follows: in Section 2 we describe the specific case study that motivated this research. In Section 3 we present the STOLE ontology and its design process. This ontology is built on an extended version of STOLE [1]. We conclude the paper in Section 4 by identifying stimulating directions and challenging issues for continued and future research in this application domain.

2 Case Study

In the last years the way of doing historical research has deeply changed. From traditional research in archives and libraries, which necessarily required the presence on site and had precise limits and constraints, the digitization of historical documents allows new perspectives and ways to conduct research. The use of SW technologies has upset times, costs, methods of historical research, and is also changing the so-called culture of the document.

The heritage of the history of public administration represents a fundamental element to understand the history of the Italian institutions as well as the history of the country in general. One of the main sources used in this field of research is represented by historical text and documents, including journals and newspapers of the age. Recently, several web sites concerning specific domain databases on Italian institutions, e.g., the Bibliography of Italian Parliament [2] and Internet Archive [3], offer to the scholars the opportunity to easily access the sources, which are sometimes totally unknown.

The development of the STOLE ontology responds to the needs of some researchers of Department of History of the University of Sassari which, since the 1980s, have been involved in a project designed to collect and digitalize historical journals regarding the origin and the evolution of institutions, customs, usages, and rules in the Italian public administration. As a result, the ARAP[4] digital archive of the University of Sassari was created and it actually collects a large amount of information about some of the most relevant journal articles published between 1848 and 1946 concerning the legislative history of public administration in Italy. The main goal of ARAP was to offer to the scientific community

[1] STOLE is the acronym for the Italian "STOria LEgislativa della pubblica amministrazione italiana", that means "Legislative History of Italian Public Administration".
[2] `http://bpr.camera.it`
[3] `https://archive.org`
[4] ARAP is the acronym for the Italian "Archivio di Riviste sull'Amministrazione Pubblica", that means "Archive of Journals on Italian Public Administration".

interested in those documents, such as historians, lawyers, political scientists, and sociologists, a repository of important sources, otherwise hardly accessible.

The journal articles included in ARAP have a remarkably value for the wealth of information they contain. In fact, starting the research from the journals could make a positive and significant contribution to the current knowledge. For example, through the study of these documents it is possible to establish connections between authors, institutions, persons and historical events mentioned in an article. Specifically, the link between an author and the people cited can reveal a lot, such as the political and cultural reference of the author. Clearly, the relationship between an institution and an historical event can offer support to understand the evolution of public administration. The use of certain concepts or the recurring of names in relation to some historical events can be an indicator of a trend.

In this context, STOLE represent the core element of the ARAP digital archive and its main goal is to to model historical concepts and gain insights into this specific field in order to support historians in their research tasks.

3 The STOLE Ontology

The main steps of the STOLE ontology design process concerned the identification both of the key concepts for this specific domain, and the proper language for the TBox implementation. Moreover, we populated the ontology, i.e., filling the ABox with semantic annotations. A team of domain experts was involved during the whole ontology development process. In particular, they contributed at the early stage in order to define the key issues related to the application domain.

In the first phase, we detected the main categories of data expressed in the considered historical documents. The results of this process enabled us to detect three categories of elements: 1) Data concerning the authors of the articles, e.g., name, surname and biography; 2) Data concerning the journal and the article, e.g., article title, journal name, date and topics raised in the article; 3) Data concerning some relevant facts and people cited in the article, e.g, people, historical events, institutions. In this specific domain, historical analysis is based on these categories of information and focused on the interrelations among these data. For instance, the relation between an author and the people mentioned in an article could provide valuable information to historians, e.g., if an author has often referred to Giuseppe Mazzini then it could easily be interpreted that this author was favourable to the republic.

During the second phase, the TBox of the STOLE ontology has been designed building on some existing standards and meta-data vocabularies, such as Bibliographic Ontology (BIBO) [5], Bio Vocabulary (BIO) [6], Dublin Core (DC) [7], Friend

[5] http://bibliontology.com

[6] http://vocab.org/bio/0.1/.html

[7] http://dublincore.org

of A Friend (FOAF) [8], and Ontology of the Chamber of Deputies (OCD) [9]. In details, BIBO has been used to describe information about documents, for instance `bibo:volume`, `bibo:issue`, `bibo:pageStart` and `bibo:pageEnd`, denoting values of volume, issue, page start and page end of an article, respectively. Both BIO and FOAF have been used in order to describe information about people. In the case of FOAF, we reused terms such as `foaf:person`, `foaf:firstName`, `foaf:surname`, `foaf:gender`. Other information about people, their relationships and the events in their lives have been described using BIO concepts, such as `bio:birth`, `bio:death`, `bio:event`, `bio:place`, `bio:biography`. From DC we reused concepts related to the structure and characteristics of a document, e.g, `dc:title`, `dc:isPartOf`, `dc:publisher`, `dc:description`. Considering OCD, we noticed that it is a relevant source for STOLE ontology since, for example, most part of the authors of articles related to the history of the public administration topics during their lives were also involved in government activities. In particular, OCD provided us valuable concepts in order to give detailed information about people involved in political offices. Finally, the usage of these core ontologies allows both extensibility and interoperability of STOLE with other resources and applications.

In dealing with the modeling language, we decide to use OWL2 DL [8] since it allows to properly model the knowledge for our application domain by means of constructs like cardinality restrictions and other role constraints, e.g., functional properties. The TBox is composed of 268 axioms, 19 classes, 34 object properties, and 33 data properties.

In the following, we describe main classes of the STOLE ontology.

`Article` represents our library, namely the collection of historical journal articles. Every instance of this class has data properties such as `articleTitle`, `articleDate`, `pageStart`, and `pageEnd`.

`Institution` is used to represent the public institutions cited in the articles. This class contains four subclasses, namely `Central`, `Local`, `PoliticalInstitution` and `EconomicInstitution`. Notice that `Central` and `Local` are disjoint classes. The same holds for `PoliticalInstitution` and `EconomicInstitution`. In this way, for example, an institution can be both a local institution and a political institution at the same time.

`Jurisprudence` is a subclass of `Article` and contains a series of verdicts which are entirely written in the articles. Every individual of this subclass has the following data properties: `verdictDate`, `verdictTitle`, and `byCourt`.

`Law` is also a subclass of `Article`, and it contains a set of principles, rules, and regulations set up by a government or other authority which are entirely written in the articles. This subclass has data properties such as `lawDate` and `lawTitle`.

`Event` denotes relevant events. It contains five subclasses modeling different kinds of events: `Birth` and `Death` are subclasses related to a person's life;

[8] `http://www.foaf-project.org`
[9] `http://data.camera.it/data`

`BeginPublication` and `EndPublication` represent the publication period of a journal; `HistoricalEvent` contains the most relevant events that have marked the Italian history.

`Journal` denotes the collection of historical journals. This class has data properties such as `journalArticle`, `publisher`, and `issn`.

`Person` is the class representing people involved in the Italian legislative and public administration history. This class contains one subclass, `Author`, that includes the contributors of the articles. Every instance of this class has some data properties as `firstName`, `surname`, and `biography`.

`Place` represents cities and countries related to people and events.

`Subject` is a class representing topics tackled in the historical journals.

In conclusion, there are other examples of ontologies designed for the cultural heritage domain, e.g., CIDOC CRM [5], however these are rather generic to be applied in our context where a detailed modeling of the singular domain is needed. Despite its specific nature, STOLE could be used in several fields of research, e.g., admnistrative law, political science, history of institutions.

4 Current Work and Open Problems

Currently, we are extending and improving in many ways the implementation of this work. First, we are dealing with a key issue for historians, namely how to disambiguate individuals and how to manage changing names, e.g., different people with the same name or institutions that changed name retaining the same functions. This point still represents an open challenge in this application domain– see [11].

Another open problem relates the ontology population process. For the current version, STOLE has been populated leveraging a set of annotated historical documents comprised into the ARAP archive. Semantic annotations were provided by a team of domain experts and individuals were added to the ontology by means of a JAVA program built on top of the OWL APIs [9]. This activity requires specialized expertise, it is time consuming and resource-intensive. Given these reasons, we are studying to find solutions for its automatization on the basis of some recent contributions – see, e.g.,[7, 10, 13, 6]. Most approaches for automatic or semi-automatic ontology population process from texts are based on the following techniques: Natural Language Processing [4], Machine Learning [2], and Information Extraction [12]. Once the ontology will be fully populated, we are planning to perform an experimental analysis on the STOLE ontology involving state of the art DL reasoners on both classification and query answering tasks.

Acknowledgments I would to thank the anonymous reviewers for their valuable suggestions, which were helpful in improving the final version of the paper. Moreover, I would also like to thank my supervisors, Giovanni Adorni and Luca Pulina, for their support, and Salvatore Mura and Prof. Francesco Soddu for the valuable discussions about the application domain.

References

1. Adorni G., Maratea M., Pandolfo L. Pulina L. An Ontology for Historical Research Documents. Web reasoning and Rule Systems. Springer (2015) 11–18
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - A Crystallization Point for the Web of Data. Web Semantics: Science, Services and Agents on the World Wide Web **7**(3) (2009) 154–165
4. Dale, R., Moisl, H., Somers, H.L.: Handbook of Natural Language Processing. CRC (2000)
5. Doerr, M.: The CIDOC Conceptual Reference Module: An Ontological Approach to Semantic Interoperability of Metadata. AI Mag. **24** (3) (2003) 75–92
6. Faria, C., Serra I., Girardi, R.: A Domain-Independent Process for Automatic Ontology Population from Text. Journal of Science of Computer Programming **95** (2014) 26–43
7. Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., Motta, E.: Semantically Enhanced Information Retrieval: An Ontology-based Approach. Web Semantics: Science, Services and Agents on the World Wide Web **9**(4) (2011) 434–452
8. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: Owl 2: The next step for owl. Web Semantics: Science, Services and Agents on the World Wide Web **6**(4) (2008) 309–322
9. Horridge, M., Bechhofer, S.: The OWL API: A Java Api for OWL Ontologies. Semantic Web **2**(1) (2011) 11–21
10. Kara, S., Alan, Ö., Sabuncu, O., Akpınar, S., Cicekli, N.K., Alpaslan, F.N.: An Ontology-based Retrieval System Using Semantic Indexing. Information Systems **37**(4) (2012) 294–305
11. Meroño-Peñuela, A., Ashkpour, A., van Erp, M., Mandemakers, K., Breure, L., Scharnhorst, A., Schlobach, S., van Harmelen, F.: Semantic Technologies for Historical Research: A survey. Semantic Web Journal (2014)
12. Moens, M.F.: Information Extraction: Algorithms and Prospects in a Retrieval Context. Springer Netherlands (2006)
13. Sánchez, D., Batet, M., Isern, D., Valls, A.: Ontology-based Semantic Similarity: A New Feature-based Approach. Expert Systems with Applications **39**(9) (2012) 7718–7728

Ontology Based Semantic Image Interpretation

Ivan Donadello[1,2]*

[1] Fondazione Bruno Kessler, Via Sommarive 18, I-38123, Trento, Italy
[2] DISI, University of Trento, Via Sommarive 9, I-38123, Trento, Italy
donadello@fbk.eu

Abstract. Semantic image interpretation (SII) leverages Semantic Web ontologies for generating a mathematical structure that describes the content of images. SII algorithms consider the ontologies only in a late phase of the SII process to enrich these structures. In this research proposal we study a well-founded framework that combines logical knowledge with low-level image features in the early phase of SII. The image content is represented with a partial model of an ontology. Each element of the partial model is grounded to a set of segments of the image. Moreover, we propose an approximate algorithm that searches for the most plausible partial model. The comparison of our method with a knowledge-blind baseline shows that the use of ontologies significantly improves the results.

1 Introduction

Semantic image interpretation (SII) is the task of generating a semantically rich structure that describes the content of an image [8]. This structure is both human and machine understandable and can be encoded by using the Semantic Web (SW) language RDF. The first advantage is that RDF enables the enrichment of the semantic content of images with SW resources, the second one is that an RDF based description of images enables content-based image retrieval via query languages like SPARQL.

The main challenge in SII is bridging the so called *semantic gap* [3], which is the complex correlation between low-level image features and high-level semantic concepts. High-level knowledge plays a key role in bridging the semantic gap [17,18]. This knowledge can be found in the ontologies provided by the SW.

Most of the current approaches to SII exploit ontologies at a later stage when some hypothesis (a geometric description of the objects and their spatial relations) of the image content have already been formulated by a bottom-up approach (see for instance [13,15,17,11,12,3,6,1]). In these cases background knowledge is exploited to check the consistency of the output and/or to infer new facts. These works do not consider uncertainty coming from the low-level image analysis or require a set of DL rules for defining what is abducible, which need to be manually crafted.

In this research proposal we study a general framework for SII that allows the integration of ontologies with low-level image features. The framework takes as input the ontology and exploits it in the process of image interpretation. The output is a description of the content of an image in terms of a (most plausible) partial logical model

* I thank my advisor Luciano Serafini for his precious help, suggestions and patience.

of the ontology [15]. Instead of lifting up low-level features into a logical form using concrete domain (as in [11]) we proceed in the opposite direction, by compiling down the background knowledge into low-level features. This allows us a more flexible inference in processing numeric information and to use simpler, and more efficient, logical reasoners for the semantic part. This partial model is generated by using optimisation methods (e.g. clustering) that integrate numeric and logical information. Our contribution is a formal framework for SII that integrates low-level features and logical axioms. Moreover, we developed an early prototype and we evaluated it, with promising results, on the task of detecting complex objects starting from the presence of their parts [5].

2 Theoretical framework

The proposed framework takes as input a *labelled picture* that is a picture partitioned into segments (regions of pixels) using a semantic segmentation algorithm [4,7]. Each segment has a set of weighted labels that represent the level of confidence of the semantic segmentation. Labels are taken from the signature Σ which is the alphabet of the ontology. A labelled picture is a pair $\mathcal{P} = \langle S, L \rangle$ where $S = \{s_1 \ldots, s_n\}$ is a set of segments of the picture $\mathcal{P}$, and L is a function that associates to each segment $s \in S$ a set $L(s)$ of weighted labels $\langle l, w \rangle \in \Sigma \times (0, 1]$.

In this research proposal we study a method for discovering new objects (e.g., composite objects) and relations between objects by exploiting low-level image features and a Description Logic (DL) [2] ontology. The ontology has the classical signature $\Sigma = \Sigma_C \uplus \Sigma_R \uplus \Sigma_I$ of symbols for concepts, relations and individuals respectively. We adopt the standard definitions for syntax and semantics of DL[3] . An ontology $\mathcal{O}$ on Σ is a set of DL axioms. An interpretation of a DL signature Σ is a pair $\mathcal{I} = \left\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \right\rangle$, where $\Delta^{\mathcal{I}}$ is a non empty set and $\cdot^{\mathcal{I}}$ is a function that interprets the symbols of Σ in $\Delta^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ if it satisfies all the axioms in $\mathcal{O}$. The axioms of the ontology are constraints on the states of the world. A picture, however, provides only a partial view of the state of the world, indeed, it could show a person with only one (visible) leg. Therefore, the content of a picture is not isomorphic to a model, as a model could contain objects not appearing in the picture (the invisible leg). The content of a picture should instead be represented as a *partial model*[4].

Definition 1 (Partial model). *Let $\mathcal{I}$ and $\mathcal{I}'$ be two interpretations of the signatures Σ and Σ' respectively, with $\Sigma \subseteq \Sigma'$; $\mathcal{I}'$ is an* extension *of $\mathcal{I}$, or equivalently $\mathcal{I}'$* extends *$\mathcal{I}$, if $\Delta^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}'}$, $a^{\mathcal{I}} = a^{\mathcal{I}'}$, $C^{\mathcal{I}} = C^{\mathcal{I}'} \cap \Delta^{\mathcal{I}}$ and $R^{\mathcal{I}} = R^{\mathcal{I}'} \cap \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, for all $a \in \Sigma_I$, $C \in \Sigma_C$ and $R \in \Sigma_R$. $\mathcal{I}_p$ is a* partial model *for a ontology $\mathcal{O}$, in symbols $\mathcal{I}_p \models_p \mathcal{O}$, if there is a model $\mathcal{I}$ of $\mathcal{O}$ that extends $\mathcal{I}_p$.*

In this framework the use of DL ontologies is twofold: first they are a terminological source for labelled pictures, second the DL inference services are exploited to check if an interpretation is a partial model and thus inferring new facts. The semantic interpretation of a picture is a partial model plus an alignment, called *grounding*, of every element of $\Delta^{\mathcal{I}_p}$ with the segments of the picture.

[3] In this paper we use the $\mathcal{SHIQ}$ DL.

[4] This intuition was introduced in [15], our formalization however is slightly different.

Definition 2 (Semantically interpreted picture). *Given an ontology $\mathcal{O}$ with signature Σ and a labelled picture $\mathcal{P} = \langle S, L \rangle$, a* semantically interpreted picture *is a triple $\mathbb{S} = (\mathcal{P}, \mathcal{I}_p, \mathcal{G})_{\mathcal{O}}$ where:*

- *$\mathcal{I}_p = \langle \Delta^{\mathcal{I}_p}, \cdot^{\mathcal{I}_p} \rangle$ is a partial model of $\mathcal{O}$;*
- *$\mathcal{G} \subseteq \Delta^{\mathcal{I}_p} \times S$ is a left-total and surjective relation called* grounding relation*: if $\langle d, s \rangle \in \mathcal{G}$ then there exists an $l \in L(s)$ such that:*
  1. *if $l \in \Sigma_C$ then $d \in l^{\mathcal{I}_p}$;*
  2. *if $l \in \Sigma_I$ then $d = l^{\mathcal{I}_p}$;*
  3. *if $l \in \Sigma_R$ then $\langle d, d' \rangle \in R^{\mathcal{I}}$ or $\langle d', d \rangle \in R^{\mathcal{I}}$ for some $d' \in \Delta^{\mathcal{I}}$.*

The grounding *of every $d \in \Delta^{\mathcal{I}_p}$, denoted by $\mathcal{G}(d)$, is the set $\{s \in S \mid \langle d, s \rangle \in \mathcal{G}\}$.*

There are many possible explanations of the picture content, thus there are many partial models describing a picture via a grounding relation. We define a cost function $\mathcal{S}$ that assigns a cost to a partial model based on its adherence to the image content: the higher the adherence the lower the cost. The *most plausible partial model* $\mathcal{I}_p^*$ is the partial model that minimizes $\mathcal{S}$, in symbols:

$$\mathcal{I}_p^* = \operatorname*{argmin}_{\substack{\mathcal{I}_p \models_p \mathcal{O} \\ \mathcal{G} \subseteq \Delta^{\mathcal{I}_p} \times S}} \mathcal{S}(\mathcal{P}, \mathcal{I}_p, \mathcal{G})_{\mathcal{O}} \tag{1}$$

The definition of $\mathcal{S}$ has to take into account low-level features of the segments and high-level semantic features of the partial model derivable form the ontology. Intuitively the cost function measures the semantic gap between the two types of features.

Definition 3 (Semantic image interpretation problem). *Given an ontology $\mathcal{O}$ and a labelled picture $\mathcal{P}$, a cost function $\mathcal{S}$, the* semantic image interpretation problem *is the construction of a semantically interpreted picture $\mathbb{S} = (\mathcal{P}, \mathcal{I}_p, \mathcal{G})_{\mathcal{O}}$ that minimizes $\mathcal{S}$.*

3 Method

In this proposal we restrict to the recognition of complex objects from their parts. For example, given a labelled picture where only some parts of a man (the legs, one arm and the head) and of a horse (the legs, the muzzle and the tail) are labelled we want to infer the presence of some logical individuals with their classes (man and horse respectively). These individuals are linked with their parts through the partOf relation. This can be seen as a clustering problem and we specify the cost function in terms of clustering optimisation. The parts (simple objects) are the input of the clustering problem whereas a single cluster contains the parts of a composite object. In addition, the parts to cluster are the individuals $d \in \Delta^{\mathcal{I}_p}$ with the following features:

- a set of low-level image features extracted from $\mathcal{G}(d)$, the grounding of d;
- a set of semantic features corresponding the most specific concepts extracted from the set $\{C \in \Sigma_C \mid d \in C^{\mathcal{I}_p}\}$ assigned to d by $\mathcal{I}_p$.

We use the centroid of $\mathcal{G}(d)$ as a numeric feature but the approach can be generalised to other features. Clustering algorithms are based on some distance between the input elements defined in terms of their features. Let $\delta_{\mathcal{G}}(d, d')$ be the Euclidean distance of the centroids of $\mathcal{G}(d)$ and $\mathcal{G}(d')$, $\delta^s_{\mathcal{O}}(d, d')$ a semantic distance between simple objects and $\delta^c_{\mathcal{O}}(d, d')$ a semantic distance between a simple object and its corresponding composite object. We define the cost function as the quality measure of the clustering:

$$\mathcal{S}(\langle \mathcal{P}, \mathcal{I}_p, \mathcal{G} \rangle_{\mathcal{O}}) = \alpha \left(\sum_{d,d' \in (\exists \mathsf{hasPart}.\top)^{\mathcal{I}_p}} \delta_{\mathcal{G}}(d, d') \right) +$$

$$(1-\alpha) \left(\sum_{\substack{\langle d',d \rangle \in \mathsf{partOf}^{\mathcal{I}_p} \\ \langle d'',d \rangle \in \mathsf{partOf}^{\mathcal{I}_p}}} (\delta_{\mathcal{G}}(d', d'') + \delta^s_{\mathcal{O}}(d', d'')) + \sum_{\langle d',d \rangle \in \mathsf{partOf}^{\mathcal{I}_p}} (\delta_{\mathcal{G}}(d', d) + \delta^c_{\mathcal{O}}(d', d)) \right).$$

Following [9], the first component of the above equation measures the centroid distance between the composite objects (inter-cluster distance). The second component estimates the distance between the elements of each single cluster (intra-cluster distance).

Minimising analytically the above equation is rather complex, thus we developed an iterative algorithm that at each loop groups the several parts of a composite object approximating the cost function. If the grouping is not a partial model the algorithm enters in the next loop and selects another clustering. In the first step our algorithm generates an initial partial model $\mathcal{I}_p$ from $\mathcal{P} = \langle S, L \rangle$ where $\Delta^{\mathcal{I}_p}$ contains an element d_s for every segment $s \in S$ and any concept C in the labelled picture is interpreted as $C^{\mathcal{I}_p} = \{d_s | C \in L(s)\}$. The grounding $\mathcal{G}$ is the set of pair $\langle d_s, s \rangle$. Then, the algorithm enters in a loop where a non-parametric clustering procedure [10] clusters the input elements $d \in \Delta^{\mathcal{I}_p}$ by using their numeric and semantic features according to $\delta_{\mathcal{G}}$ and $\delta^s_{\mathcal{O}}$. Each cluster cl corresponds to a composite object d_{cl} which is introduced in $\mathcal{I}_p$ and is connected via the $\mathsf{hasPart}$ relation to the elements of cl. We predict the type of this new individual via abductive reasoning: the type is the ontology concept that shares the maximum number of parts with the elements of the cluster. For example, if we cluster some elements of type Tail, Muzzle and Arm an abducted ontology concept will be Horse. These new facts are introduced in $\mathcal{I}_p$ and the algorithm checks if $\mathcal{I}_p$ is a partial model of $\mathcal{O}$ by using a DL reasoner (Pellet [16]). If true the algorithm returns $\mathcal{I}_p$ otherwise it extends the input elements with a set of consistency features that encode information about the inconsistency of $\mathcal{I}_p$. These features tend to separate (resp. join) the segments that have been joined (resp. separated) in the previous clustering. The cluster of our example is inconsistent because a horse does not have arms. Then the algorithm returns at the beginning of the loop.

4 Evaluation

To evaluate our approach we created, by using LabelMe [14], a dataset of 204 labelled pictures. For each picture we manually annotated simple objects, composite objects and

Table 1. Performance of the proposed algorithm for SII and comparison with the baseline. The reported data are the average of the three measures on each single picture.

	$prec_{\mathrm{GRP}}$	rec_{GRP}	$F1_{\mathrm{GRP}}$	$prec_{\mathrm{COP}}$	rec_{COP}	$F1_{\mathrm{COP}}$
SII	**0.61**	**0.89**	**0.67**	**0.73**	**0.75**	**0.74**
Baseline	0.45	0.71	0.48	0.66	0.69	0.66

their part-whole relations[5]. We also created a simple ontology[6] with a basic formalisation of meronymy in the domains of: houses, trees, people, and street vehicles. We built a ground truth by associating every single labelled picture $\mathcal{P}$ to its partial model encoded in an ABox $A_{\mathcal{P}}$. The partial model returned by our algorithm is encoded in the $A^*_{\mathcal{P}}$ ABox, in order to compare $A_{\mathcal{P}}$ with $A^*_{\mathcal{P}}$ we define the following two measures.

Grouping (GRP): this measure expresses how good is our algorithm at grouping parts of the same composite object. We define precision, recall and F1 measure on the set of siblings (the parts of the same composite object): $\mathsf{sibl}(\mathcal{A}) = \{\langle d, d' \rangle \mid \exists d'' : \mathsf{partOf}(d, d''), \mathsf{partOf}(d', d'') \in \mathcal{A}\}$. Thus:

$$prec_{\mathrm{GRP}}(\mathcal{P}) = \frac{|\mathsf{sibl}(\mathcal{A}_{\mathcal{P}}) \cap \mathsf{sibl}(\mathcal{A}^*_{\mathcal{P}})|}{|\mathsf{sibl}(\mathcal{A}^*_{\mathcal{P}})|} \qquad rec_{\mathrm{GRP}}(\mathcal{P}) = \frac{|\mathsf{sibl}(\mathcal{A}_{\mathcal{P}}) \cap \mathsf{sibl}(\mathcal{A}^*_{\mathcal{P}})|}{|\mathsf{sibl}(\mathcal{A}_{\mathcal{P}})|}$$

Complex-object prediction (COP): this measure expresses how good is our algorithm at predicting the type of the composite object. We define precision, recall and F1 measure on the types of the composite object each part is assigned to: $\mathsf{ptype}(\mathcal{A}) = \{\langle d, \mathsf{C} \rangle \mid \exists d' : \{\mathsf{partOf}(d, d'), \mathsf{C}(d')\} \subset \mathcal{A}\}$. Thus:

$$prec_{\mathrm{COP}}(\mathcal{P}) = \frac{|\mathsf{ptype}(\mathcal{A}_{\mathcal{P}}) \cap \mathsf{ptype}(\mathcal{A}^*_{\mathcal{P}})|}{|\mathsf{ptype}(\mathcal{A}^*_{\mathcal{P}})|} \quad rec_{\mathrm{COP}}(\mathcal{P}) = \frac{|\mathsf{ptype}(\mathcal{A}_{\mathcal{P}}) \cap \mathsf{ptype}(\mathcal{A}^*_{\mathcal{P}})|}{|\mathsf{ptype}(\mathcal{A}_{\mathcal{P}})|}$$

To measure how the semantics improves the recognition of composite objects from their parts we implemented a baseline that clusters without semantic features, see Table 1. We can see that the explicit use of semantic knowledge via semantic distance, abductive and deductive reasoning improves the baseline that relies only on numeric features.

5 Conclusions

We proposed a well-founded and general framework for SII that integrates symbolic information of an ontology with low-level numeric features of a picture. An image is interpreted as a (most plausible) partial model of an ontology that allows the query about the semantic content. We applied the framework to the specific task of recognizing composite objects from their parts. The evaluation shows good results and the injection of semantic knowledge improves the performance with respect to a semantically-blind baseline. As future work, we want to extend our evaluation by using more low-level features, by studying other relations and by using a semantic segmentation algorithm as source of labelled pictures.

[5] An example of labelled picture is available at `http://bit.ly/1DXZxic`

[6] The ontology is available at `http://bit.ly/1AruGh0`

References

1. Atif, J., Hudelot, C., Bloch, I.: Explanatory reasoning for image understanding using formal concept analysis and description logics. Systems, Man, and Cybernetics: Systems, IEEE Transactions on 44(5), 552–570 (May 2014)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York, NY, USA (2003)
3. Bannour, H., Hudelot, C.: Towards ontologies for image interpretation and annotation. In: Martinez, J.M. (ed.) 9th International Workshop on Content-Based Multimedia Indexing, CBMI 2011, Madrid, Spain, June 13-15, 2011. pp. 211–216. IEEE (2011)
4. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic segmentation with second-order pooling. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision – ECCV 2012. LNCS, Springer Berlin Heidelberg (2012)
5. Donadello, I., Serafini, L.: Mixing low-level and semantic features for image interpretation. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) Computer Vision - ECCV 2014 Workshops. LNCS, Springer International Publishing (2014), best paper award.
6. Espinosa, S., Kaya, A., Möller, R.: Logical formalization of multimedia interpretation. In: Paliouras, G., Spyropoulos, C., Tsatsaronis, G. (eds.) Knowledge-Driven Multimedia Information Extraction and Ontology Evolution, Lecture Notes in Computer Science, vol. 6050, pp. 110–133. Springer Berlin Heidelberg (2011)
7. Gould, S., Zhao, J., He, X., Zhang, Y.: Superpixel graph label transfer with learned distance metric. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Springer International Publishing (2014)
8. Hudelot, C., Maillot, N., Thonnat, M.: Symbol grounding for semantic image interpretation: From image data to semantics. In: Proc. of the 10th IEEE Intl. Conf. on Computer Vision Workshops. ICCVW '05, IEEE Computer Society (2005)
9. Jung, Y., Park, H., Du, D.Z., Drake, B.L.: A decision criterion for the optimal number of clusters in hierarchical clustering. Journal of Global Optimization 25(1), 91–111 (2003)
10. Kohonen, T.: The self-organizing map. Proc. of the IEEE 78(9), 1464–1480 (Sep 1990)
11. Neumann, B., Möller, R.: On scene interpretation with description logics. Image and Vision Computing 26(1), 82 – 101 (2008), cognitive Vision-Special Issue
12. Peraldi, I.S.E., Kaya, A., Möller, R.: Formalizing multimedia interpretation based on abduction over description logic aboxes. In: Proc. of the 22nd Intl. Workshop on Description Logics (DL 2009). CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009)
13. Reiter, R., Mackworth, A.K.: A logical framework for depiction and image interpretation. Artificial Intelligence 41(2), 125–155 (1989)
14. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: A database and web-based tool for image annotation. Int. J. Comput. Vision 77(1-3), 157–173 (May 2008)
15. Schroder, C., Neumann, B.: On the logics of image interpretation: model-construction in a formal knowledge-representation framework. In: Image Processing, 1996. Proceedings., Int. Conf. on. vol. 1, pp. 785–788 (Sep 1996)
16. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Web Semant. 5(2), 51–53 (Jun 2007)
17. Town, C.: Ontological inference for image and video analysis. Mach. Vision Appl. 17(2), 94–115 (Apr 2006)
18. Yuille, A., Oliva, A.: Frontiers in computer vision: Nsf white paper (November 2010), `http://www.frontiersincomputervision.com/WhitePaperInvite.pdf`

Labelled Variables in Logic Programming: A First Prototype in tuProlog

Roberta Calegari, Enrico Denti, and Andrea Omicini

Dipartimento di Informatica, Scienza e Ingegneria (DISI)
Alma Mater Studiorum–Università di Bologna, Italy
{roberta.calegari, enrico.denti, andrea.omicini}@unibo.it

Abstract. We present the first prototype of Labelled tuProlog, an extension of tuProlog exploiting labelled variables to enable a sort of multi-paradigm / multi-language programming aimed at pervasive systems.

1 Context & Motivation

To face the challenges of today pervasive system, which are inherently *complex*, *distributed*, *situated* [7] and *intelligent*, suitable models and technologies are required to effectively support *distributed situated intelligence*. To this end, in this paper we investigate a logic programming (LP) extension based on *labelled variables*, and test it by means of a prototype rooted in the tuProlog system [3, 9], called *Labelled* tuProlog. Our general aim is to define a unique blend of LP and labelled variables – a sort of multi-paradigm and multi-language programming framework for a distributed environment – where diverse computational models can be tailored to the *local* needs of situated systems by means of suitable labelled variables systems, and work together against the common LP background according to the original tuProlog aim [3].

Our work builds upon the general notion of *label* defined by Gabbay [4], and adopts the techniques introduced by Holzbaur [6] to develop a generalisation of LP where labels are exploited to define computations in domain-specific contexts, while retaining the conventional syntax of LP. More generally, our work moves from the observation that pervasiveness of today systems requires awareness of the environment as well as the ability of reacting to changes. This mandates for models promoting system intelligence, and for technologies making it possible to spread intelligence wherever needed. While logic-based approaches are natural candidates for intelligent systems, a pure LP approach seems not to fit the needs of situated systems. Instead, a hybrid approach would make it possible to exploit LP for what it is most suited – symbolic computation –, while possibly delegating other aspects (such as situated computations) to other languages, or to other levels of computation. This is precisely where the notion of labelled variables can be of help: by enabling some parts of the computation to be expressed at a separate level, while retaining the general coherence of the LP approach.

Being light-weight, intentionally designed around a minimal core, and Java-based, tuProlog is an ideal candidate to support the above goal—distributing situated intelligence while supporting labelled-variable-based computations.

While several works exists in this area – such as [8, 1] – they mostly focus onto specific scenarios and sorts of systems—e.g. modal logic, deductive systems, fuzzy systems, etc. Instead, our model aims at providing a general-purpose mechanism that could fit several relevant context-specific systems, while seamlessly rooted in the LP framework.

While in our first prototype labels are only applied to variables – and not generally to formulas like in Gabbay [4] – the proposed notion of label is already general enough to capture Holzbaur's attribute variables [6], opening the way towards the expressiveness and the computational power of Constraint Logic Programming (CLP) [10, 5]—which is why the two case studies presented below assume CLP as the reference label domain.

2 The Labelled Variables Model in Logic Programming

A *Labelled Variables Model for Logic Programming* is defined by:

- a set of *basic labels* $b_1, \ldots, b_n$, each with the form of a logic term, which represent entities in the *domain of labels*;
- a set of *labels*, each defined as a set of basic labels—i.e., $l_i = \{b_{1i}, .., b_{ni}\}$;
- a *labelling association* $\langle v, l \rangle$, associating label l to variable v: as a convenient shortcut, such association can be written as v^l;
- a *combining function* f_L, synthesising a new label from two given ones, combining the two labels according to some scenario-specific criteria.

The unification of two labelled variables is represented by the extended tuple *(true/false, θ, label)* where *true/false* represents the existence of an answer, θ represents the most general substitution, and *label* represents the new label associated to the unified variables defined by the *combining function* f_L. Figure 1 reports the unification rules for two generic terms T_1 and T_2: to lighten the notation, undefined elements in the tuple are omitted—i.e., labels or substitutions that do not exist/do not apply in a particular situation. Since, by design, only variables can be labelled here, the only case to be added to the standard unification table is represented by labelled variables.

Fig. 1. Unification rules summary

T_1 / T_2	**constant C_2**	**variable X_2**	**labelled variable X_2^{l2}**	**compund term S_2**
constant C_1	true if $C_1 = C_2$	true - $\{X_2/C_1\}$	false	false
variable X_1	true - $\{X_1/C_2\}$	true - $\{X_1/X_2\}$	true - $\{X_1/X_2\} - l_2$	true - $\{X_1/S_2\}$
labelled variable X_1^{l1}	false	true - $\{X_1/X_2\}$ - l_1	true if not $\{\ \}$ - $\{X_1/X_2\} - f_L(l_1, l_2)$	false
compund term S_1	false	true - $\{X_2/S_1\}$	false	true if S_1 and S_2 unify

While this choice clearly confines the impact of labelling, keeping the label computational model well separate from the LP one, it is also too restrictive in practice: in fact, application scenarios often need to express some relevant properties via by suitable terms, so that they can influence the label computation. For this purpose, we introduce the notion of *label-interpreted terms*, as a set of semantically-relevant terms in the domain of labels, along with a *pre-processing phase*, where each label-interpreted term is intercepted and replaced with *an anonymous variable labelled with that term.* In this way, any special term in the domain of label can be treated normally by the f_L combining function, with no need to change the basic model above.

3 Prototype in tuProlog

In this Section we apply our model in the context of the tuProlog system, designing a tuProlog extension that enables users to define their own labelled applications for their specific domains of interest.

3.1 System architecture

Since tuProlog is Java-based, a language extension can be provided both as a Prolog meta-level or library, or via suitable Java methods [3]. We support both ways, defining first a Prolog-language level to denote labelled variables, then two language extensions (Prolog-based and Java-based) to be used in alterative to denote the labelled application (Subsection 3.2).

Moreover, practical reasons suggest to avoid the proliferation of anonymous variables in the pre-processing phase, since they would pollute the name space and make the code less readable: for this reason, the prototype adopts a language shortcut to specify the label of a term *directly*, with no need to actually replace terms with unbound variables explicitly. However, this is just a linguistic extension – not a model extension –, thus leaving the model properties untouched.

3.2 The language extension

In order to enable users to define their labelled application, a suitable set of Prolog predicates / Java methods is introduced to let users define: *(i)* the label/-variable association; *(ii)* the function f_L that specifies under which conditions two labels unify in the selected domain, returning the new label associated with the unified term; *(iii)* a shortcut for the pre-processing transformation, moving *label-interpreted* terms to the label world, making them labels of undefined variables. For the sake of brevity, we show here how to express the three entities just on the Prolog side only—the Java side being conceptually identical.

- the label/variable association is expressed by the `label_associate(+Var, +Term)` predicate, which associates variable *Var* to label *Term*; as a further convenience, the `°/2` infix operator is also provided for the same purpose.

- the function f_L is expressed by the `label_generate(+Label1, +Label2, -Label3)` predicate, which encodes how to build a new label from two given ones.
- the pre-processing phase is embedded in the `label_interpret(+Label, +Term)` helper predicate, which succeeds if *Term* can potentially be unified with the label *Label* in the label world, or fails otherwise. Only if the check succeeds, the labelled variable is actually unified with the term.

4 Case studies

In the following we present two application examples: Interval CLP [2] and Dress Selection based on colour-match rules. There, figures show both the syntax (left), and the prototype screenshot (right), which sometimes differs due to the current implementation shortcuts, as discussed below.

4.1 Interval CLP

In this application scenario, variables are labelled with their admissible numeric interval—that is, `X°[A,B]` means that `X` can span over the range $[A..B]$. Unification succeeds if two numeric intervals overlap, in which case the intersected interval is the newly-computed label. Being specific of the application scenario, the behaviour is defined by the f_L function.

In the code in Figure 2 (top), labelled variable `Y` is unified with labelled variable `X`: the operation succeeds (`X/Y`), with both variables receiving the new label `[2,3]`. Conversely, in the subsequent case the unification fails, because the intersected interval is empty.

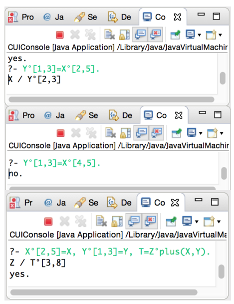
In the third case, three aspects are worth highlighting. First of all, an explicit unification is needed to link an unlabelled logic variable (`X`) to a labelled variable (`X°[2,5]`). The second aspect concerns the last term: because the shortcut notation `Z°plus(X,Y)` is not yet supported by the prototype, the unification with a new (possibly anonymous) variable must be made explicit (`Z=_°plus(X,Y)`).

Fig. 2. Interval CLP in Labelled tuProlog

```
?- Y°[1,3]=X°[2,5].
X / Y °[2,3]
yes.
```

```
?- Y°[1,3]=X°[4,5].
no.
```

```
?- X°[2,5], Y°[1,3], Z°plus(X,Y).
Z °[3..8]
yes.
```



This is why an auxiliary variable (`T`) is introduced—but `Z` could have been reused in alternative, as above. Finally, `plus(X,Y)` is an example of a label-interpreted term (whose meaning is obvious) requiring pre-processing.

4.2 Dress Selection

Here, the goal is to select from the wardrobe all the shirts that respect some given colour constraints: the domain of labels includes then shirts and their colours.

The predicate `shirt(`*`Colour`*`,`*`Description`*`)` represents a shirt with of colour *`Colour`*, expressed in terms of its RGB composition – a triple like `rgb(`*`Red,Green,Blue`*`)` –, synthetically described by *`Description`*. This means that in standard LP the knowledge representation would include facts like `shirt(rgb(255,255,255), amy_white_blouse)`, and alike.

In the labelled context, however, the objective is to move relevant information to the domain of labels. Since labels can only be attached to variables, a different knowledge representation is adopted, where *`Colour`* is seen as a labelled variable, having the `rgb/3` term above as its label. Accordingly, the wardrobe content representation is as follows:

$\texttt{shirt(A}^{\texttt{rgb(255,240,245)}}\texttt{, pink\_blouse).}$
$\texttt{shirt(B}^{\texttt{rgb(255,222,173)}}\texttt{, yellow\_tshirt).}$
$\texttt{shirt(C}^{\texttt{rgb(119,136,153)}}\texttt{, army\_tshirt).}$
$\texttt{shirt(D}^{\texttt{rgb(188,143,143)}}\texttt{, periwinkle\_blouse).}$
$\texttt{shirt(E}^{\texttt{rgb(255,245,238)}}\texttt{, cream\_blouse).}$

To obtain all the shirts in the wardrobe, a query such as

```
?- shirt(Colour, Description)
```

would obviously generate all the possible solutions in backtracking (Figure 3, top). By suitable exploiting labelled variables, however, the query can be refined by defining a *target colour* in the goal, and exploiting the combining function to get only the dresses whose *dress_colour* is "similar enough" to the target.

To this end, two RGB colours $c_1 = rgb(r_1, g_1, b_1)$ and $c_2 = rgb(r_2, g_2, b_2)$ are considered similar – so, the corresponding labels unify – if their distance is below a given threshold. For the sake of concreteness, we assume the threshold to be ≤ 100, and the colour distance to be normalised and computed as a distance in a 3D Euclidean space.

The f_L function, checking whether two given labels (c_1=*target colour* and c_2=*dress colour*) are to be considered as similar, can be defined as:

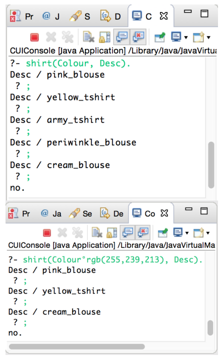
$$f_L(c_1, c_2) = \begin{cases} c_2 & \text{if } distance(c_1, c_2) \leq threshold \in [0..100] \\ \{\} & \text{if } distance(c_1, c_2) > threshold \in [0..100] \end{cases}$$

During the unification of the two labelled variables whose label represent the *target colour* and the *dress colour*, the following steps are performed: if the *dress_colour* is similar to the *target_colour*, the returned label is *dress_colour*—that is, the colour of the selected shirt; if, instead, the two colours are not similar, the empty label is returned, causing the unification process to fail.

Fig. 3. Colour Constraints in Labelled tuProlog

```
?- shirt(Colour, Desc).
Desc / pink_blouse ;
Desc / yellow_tshirt ;
Desc / army_tshirt ;
Desc / periwinkle_blouse ;
Desc / cream_blouse ;
no.
```

```
?-shirt(Colour°rgb(255,239,213), Desc).
Desc / pink_blouse ;
Desc / yellow_tshirt ;
Desc / cream_blouse ;
no.
```



Now let us look for all the shirts similar to the papaya colour: assuming a normalised similarity threshold of 30 (max is 100), the system returns just three solutions (Figure 3, bottom) instead of the five available in un unconstrained world (Figure 3, top), thus actually pruning the solution tree.

References

1. Alsinet, T., Chesñevar, C.I., Godo, L., Simari, G.R.: A logic programming framework for possibilistic argumentation: Formalization and logical properties. Fuzzy Sets and Systems 159(10), 1208–1228 (2008)
2. Benhamou, F.: Interval constraint logic programming. In: Podelski, A. (ed.) Constraint Programming: Basics and Trends, LNCS, vol. 910, pp. 1–21. Springer (1995)
3. Denti, E., Omicini, A., Ricci, A.: Multi-paradigm Java-Prolog integration in tuProlog. Science of Computer Programming 57(2), 217–250 (Aug 2005)
4. Gabbay, D.M.: Labelled Deductive Systems, Volume 1. Clarendon Press, Oxford Logic Guides 33 (Sep 1996)
5. Gavanelli, M., Rossi, F.: Constraint logic programming. In: Dovier, A., Pontelli, E. (eds.) A 25-year Perspective on Logic Programming, LNCS, vol. 6125, pp. 64–86. Springer (2010)
6. Holzbaur, C.: Metastructures vs. attributed variables in the context of extensible unification. In: Bruynooghe, M., Wirsing, M. (eds.) Programming Language Implementation and Logic Programming, LNCS, vol. 631, pp. 260–268. Springer (1992)
7. Mariani, S., Omicini, A.: Coordinating activities and change: An event-driven architecture for situated MAS. Engineering Applications of Artificial Intelligence 41, 298–309 (May 2015)
8. Russo, A.: Generalising propositional modal logic using labelled deductive systems. In: Baader, F., Schulz, K.U. (eds.) Frontiers of Combining Systems, Applied Logic Series, vol. 3, pp. 57–73. Springer (1996)
9. tuProlog: Home page. `http://tuprolog.unibo.it/`
10. Van Hentenryck, P.: Constraint logic programming. The Knowledge Engineering Review 6, 151–194 (Sep 1991)

Learning Static Constraints for Domain Modeling from Training Plans

Rabia Jilani

University of Huddersfield, Huddersfield UK
rabia.jilani@hud.ac.uk

Abstract. Intelligent agents solving problems in the real-world require domain models containing widespread knowledge of the world. Synthesising operator descriptions and domain specific constraints by hand for AI planning domain models is time-intense, error-prone and challenging. To alleviate this, automatic domain model acquisition techniques have been introduced. For example, the LOCM system requires as input some plan traces only, and is effectively able to automatically encode the dynamic part of the domain model. However, the static part of the domain, i.e., the underlying structure of the domain that can not be dynamically changed, but that affects the way in which actions can be performed is usually missed, since it can hardly be derived by observing transitions only.

In this paper we introduce ASCoL, a tool that exploits graph analysis for automatically identifying static relations, in order to enhance planning domain models. ASCoL has been evaluated on domain models generated by LOCM for the international planning competition, and has been shown to be effective.

Keywords: Automated Planning, Knowledge Engineering, Domain Model Acquisition, Domain Constraints.

1 Introduction

Intelligent agents that solve problems in the real-world use models containing vast knowledge of that world to plan their actions. Designing complete domain models and gathering application knowledge is crucial not only for the efficiency of intelligent planning systems, but also for the correctness of the resulting action plans generated by intelligent agents. Creating such models is a difficult task, that is usually done manually; it requires planning experts and it is time-consuming and error-prone.

In this paper we present ASCoL, (Automated Static Constraint Learner), a tool that can effectively identify static knowledge by considering plan traces as the only source of information.

In the following sections, we first provide a general background of the areas and concepts included in this research which is followed by the research question, problem, contribution, the developed system and system evaluation. The paper is concluded with a discussion of some future goals.

2 Background

2.1 AI Planning

The term Planning refers to the process of reasoning about actions and then organizing those actions to accomplish a desired goal. It gives a full scheme for doing or accomplishing something. In Artificial Intelligence (AI) terms this scheme is known as a Plan. Plan generation is considered an important property of intelligent behaviour. AI Planning has been a part of study in artificial intelligence for over two decades.

Classical planning [2] deals with finding a (partially or totally ordered) sequence of actions transforming the static, deterministic and fully observable environment from some initial state to a desired goal state. In the classical representation *atoms* are predicates. *States* are defined as sets of ground (positive) atoms. A *planning operator* $op = (name(o), pre(o), eff^-(o), eff^+(o))$ is specified such that $name(o) = op\_name(x_1, \ldots, x_k)$ (op_name is a unique operator name and $x_1, \ldots x_k$ are variable symbols (arguments of certain types) appearing in the operator), $pre(o)$ is a set of predicates representing the operator's preconditions, $eff^-(o)$ and $eff^+(o)$ are sets of predicates representing the operator's negative and positive effects. *Actions* are ground instances of planning operators. An action $A = (pre(A), eff^-(A), eff^+(A))$ is *applicable* in a state s if and only if $pre(A) \subseteq s$. Application of A in s (if possible) results in a state $(s \setminus eff^-(A)) \cup eff^+(A)$.

2.2 Knowledge Engineering for AI Planning

Knowledge Acquisition and Knowledge Engineering (KE) have been significant to Artificial Intelligence (AI) research since the fields inception. Acquiring domain knowledge from input training data has attracted much interest in research.

The domain model acquisition problem has mainly been tackled by exploiting two approaches. On the one hand, knowledge engineering tools for planning have been introduced over time, for supporting human experts in modelling the knowledge. Two particular examples are itSIMPLE [10] and PDDL studio [8]. Recently, also crowd-sourcing has been exploited for acquiring planning domain models [13]. On the other hand, a number of techniques are currently available for automatic domain model acquisition; they rely on example data for deriving domain models. For example ARMS [12], SLAF [9] and many more. Significant differences can be found in terms of the quantity and quality of the required inputs. [6] presents a brief overview of nine different knowledge engineering tools used in planning and compares these systems on a set of criteria.

2.3 The LOCM System

One such knowledge acquisition tool is LOCM (Learning Object-Centred Models) [1] that carries out automated generation of a planning domain model from a set of example training plans. Each plan is a sequence of actions, where each action in the plan is stated as a name and a list of objects that are affected or are needed during the actions execution. Different plan traces are generated by observing the behaviour of an agent in its environment.

The LOCM method is unique in that it requires no prior knowledge and can learn action schema without requiring any additional information about predicates or initial, goal or intermediate state descriptions of objects for the example action sequences. The exception to this is that LOCM effectively determines the dynamic part of the domain model of objects but not the static part i.e., the underlying structure of the domain that can not be dynamically changed, but that affects the way in which actions can be performed.

1. Dynamic Knowledge: a set of parametrised operator schema representing generic actions and resulting changes in the domain under study.
2. Static Knowledge: relationships/constraints that are implicit in the set of operators and are not directly expressed but essentially are present while defining a domain model. These can be seen to appear in the preconditions of operators only and not in the effects. In simple words static facts never change in the world. According to Wickler [11], let $O = \{O_1, O_2, \ldots, O_n\}$ be a set of operators and let $Pr = \{Pr_1, Pr_2, \ldots, Pr_n\}$ be a set of all the predicate symbols that occur in these operators. A predicate $Pr_i \in Pr$ is fluent *iff* there is an operator $O_j \in O$ that has an effect that changes the truth of the predicate Pr_i. Otherwise the predicate is static.

In many domains, there are static relationships or constraints which restrict the values of variables in domain modelling. For example learning the map of roads in transport domains or the fixed card stacking rules between specific cards in card-games domains, that never change with the execution of actions.

3 The Research

Our work is aimed at automating the acquisition of static constraints. We aim to enhance the LOCM system to learn complete domain models including the knowledge of static constraints by using sequences of plans as the only input training data. The static knowledge is not explicitly captured in the plan traces and so it is a big challenge to learn such static constraints from them.

We introduced ASCoL, an efficient and effective tool for identifying static knowledge missed by domain models automatically acquired.

The proposed approach generates a directed graph for each pair of same-type arguments of operators and, by analysing linearity properties of the graphs, identifies relevant relations between arguments. Remarkably, the contributions of ASCoL, as demonstrated by our large experimental analysis, are: (i) the ability to identify different types of static relations, by exploiting graph analysis; (ii) ASCoL can work with both optimal and suboptimal plan traces; (iii) considering pairs of same-typed objects allows the identification of all the static relations considered in the benchmark models, and (iv) it can be a useful debugging tool for improving existing models, which can indicate hidden static relations helpful for pruning the search space.

A preliminary version of ASCoL has been presented in [5]; this version was able to identify inequality constraints only. After further development and a large experimental analysis, in the recent AI*IA publication [7], we demonstrate the ability of ASCoL in finding static relations for enhancing domain models automatically acquired by LOCM.

3.1 The Learning Problem

We define the learning problem that ASCoL addresses as follows. Given the knowledge about object types, operators and predicates, and a set of plan traces, how can we automatically identify the static relation predicates that are needed by operators' preconditions? We base our methodology on the assumption that plan traces contain tacit knowledge about constraints validation/acquisition.

Specifically, a *learning problem description* is a tuple *(P, T)*, where P is a set of plan traces and T is a set of types of action arguments in P (taken from the LOCM learnt domain model). The *output* for a learning problem is a *constraint repository* R that stores all admissible constraints on the arguments of each action A in plan traces P.

4 The ASCoL Tool

We now briefly present the ASCoL tool that has been developed for identifying useful static relations. The process steps can be summarised as follows:

1. Read the partial domain model (induced by LOCM) and the plan traces.
2. Identify, for all operators, all the pairs of arguments involving the same object types.
3. For each of the pairs, generate a directed graph by considering the objects involved in the matching actions from the plan traces.
4. Analyse the directed graphs for linearity and extract hidden static relations between arguments.
5. Run inequality check.
6. Return the extended domain model that includes the identified static relations.

The main information available for ASCoL comes from the input plan traces. As a first control, we remove from the plan traces all the actions that refer to operators that do not contain at least two arguments of the same type.

Even though, theoretically, static relations can hold between objects of different types, they mostly arise between same-typed objects. This is the case in transport domains, where static relations define connections between locations. Moreover, considering only same-typed object pairs can reduce the computational time required for identifying relations. It is also worth noting that, in

Table 1. Overall results on considered domains. For each original domain, the number of operators (# Operators), and the total number of static relations (# SR) are presented. ASCoL results are shown in terms of the number of identified static relationships (Learnt SR) and number of additional static relations provided (Additional SR) that were not included in the original domain model. The seventh and eighth columns indicate respectively the number of plans provided in input to ASCoL, that allows it to converge, and the average number of actions per plan (A/P). The last column shows the CPU-time in milliseconds

Domain	# Operators	# SR	Learnt SR	Add. SR	# Plans	Avg. A/P	CPU-time
TPP	4	7	7	0	7	28	171
Zenotravel	5	4	6	2	4	24	109
Miconic	4	2	2	0	1	177	143
Storage	5	5	5	0	24	15	175
Freecell	10	19	13	0	20	60	320
Hanoi	1	0	1	1	1	60	140
Logistics	6	0	1	1	3	12	98
Driverlog	6	2	2	0	3	12	35
Mprime	4	7	7	0	10	30	190
Spanner	3	1	1	0	1	8	144
Gripper	3	0	1	1	1	14	10
Ferry	3	1	2	1	1	18	130
Barman	12	3	3	0	1	150	158
Gold-miner	7	3	1	0	13	20	128
Trucks	4	3	3	0	6	25	158

most of the cases where static relations involve objects of different types, this is due to a non-optimal modelling process. Furthermore, such relations can be easily identified by naively checking the objects involved in actions; whenever some objects of different type always appear together, they are likely to be statically related.

5 Experimental Evaluation

Remarkable results have been achieved in complex domains, with regards to the number of static relations. We considered fifteen different domain models, taken either from IPCs[1] or from the FF domain collection (FFd)[2].

We selected domains that are encoded using different modelling strategies, and their operators include more than one argument per object type. Table 1 shows the results of the experimental analysis. A detailed interpretation of results can be found in the recent AI*IA publication. All domains but Gripper, Logistics and Hanoi, exploit static relations. Input plans of these domains have been generated by using the Metric-FF planner [4] on randomly generated problems, sharing the same objects. ASCoL has been implemented in Java, and run on a Core 2 Duo/8GB processor. CPU-time usage of the ASCoL is in the range of 35-320 (ms) for each domain.

Interestingly, we observe that ASCoL is usually able to identify all the static relations of the considered domains. Moreover, in some domains it is providing

[1] http://ipc.icaps-conference.org/
[2] https://fai.cs.uni-saarland.de/hoffmann/ff-domains.html

additional static relations, which are not included in the original domain model. Remarkably, such additional relations do not reduce the solvability of problems, but reduce the size of the search space by pruning useless instantiations of operators.

6 Conclusion and Future Goals

We are considering several paths for future work. Grant, in [3], discusses the limitations of using plan traces as the source of input information. ASCoL faces similar difficulties as the only input source to verify constraints are sequences of plans. We are also interested in extending our approach for considering static relations that involve more than two arguments In particular, we aim to extend the approach for merging graphs of different couples of arguments. Finally, we plan to identify heuristics for extracting useful information also from acyclic graphs.

References

1. Cresswell, S.N., McCluskey, T.L., West, M.M.: Acquiring planning domain models using LOCM. The Knowledge Engineering Review 28(02), 195–213 (2013)
2. Ghallab, M., Nau, D., Traverso, P.: Automated planning: theory & practice (2004)
3. Grant, T.: Identifying Domain Invariants from an Object-Relationship Model. PlanSIG2010 p. 57 (2010)
4. Hoffmann, J.: The Metric-FF Planning System: Translating "Ignoring Delete Lists" to Numeric State Variables 20, 291–341 (2003)
5. Jilani, R., Crampton, A., Kitchin, D.E., Vallati, M.: ASCoL: Automated acquisition of domain specific static constraints from plan traces. In: The UK Planning and Scheduling Special Interest Group (UK PlanSIG) 2014 (2014)
6. Jilani, R., Crampton, A., Kitchin, D.E., Vallati, M.: Automated Knowledge Engineering Tools in Planning: State-of-the-art and Future Challenges. In: The Knowledge Engineering for Planning and Scheduling workshop (KEPS) (2014)
7. Jilani, R., Crampton, A., Kitchin, D.E., Vallati, M.: ASCoL: a tool for improving automatic planning domain model acquisition. In: Proceedings of the 14th Conference of the Italian Association for Artificial Intelligence (AI*IA 2015) (2015)
8. Plch, T., Chomut, M., Brom, C., Barták, R.: Inspect, edit and debug PDDL documents: Simply and efficiently with PDDL studio. System Demonstrations and Exhibits at ICAPS pp. 15–18 (2012)
9. Shahaf, D., Amir, E.: Learning partially observable action schemas. In: Proceedings of the national conference on artificial intelligence (AAAI) (2006)
10. Vaquero, T.S., Romero, V., Tonidandel, F., Silva, J.R.: itSIMPLE 2.0: An Integrated Tool for Designing Planning Domains. In: Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS). pp. 336–343 (2007)
11. Wickler, G.: Using planning domain features to facilitate knowledge engineering. KEPS 2011 (2011)
12. Yang, Q., Wu, K., Jiang, Y.: Learning action models from plan examples using weighted max-sat. Artificial Intelligence 171(2), 107–143 (2007)
13. Zhuo, H.H.: Crowdsourced Action-Model Acquisition for Planning. In: Proceedings of the AAAI Conference on Artificial Intelligence (2015)

A CP Scheduler for High-Performance Computers

Thomas Bridi, Michele Lombardi, Andrea Bartolini, Luca Benini, and Michela Milano

{thomas.bridi,michele.lombardi2,a.bartolini,luca.benini,michela.milano}@unibo.it
Università di Bologna

Abstract. Scheduling and dispatching tools for High-Performance Computing (HPC) machines have the role of mapping incoming jobs to the available resources, trying to maximize equipment utilization and user satisfaction. Optimal Job Scheduling is a well-known NP-hard problem, forcing commercial schedulers to adopt greedy approaches based on rules. Constraint Programming (CP) is a well-known combinatorial optimization approach that has been shown to be very effective in optimally solving scheduling problems. We present the first CP-based job scheduler for HPC machines, working in a real-life production environment. We evaluate our solution both on a cluster of virtual machines and on the Eurora Supercomputer with production workloads. Results show significant improvements in terms of user fair-waiting without degradation in overall machine utilization w.r.t state-of-the-art rule-based dispatchers.

Keywords: Constraint Programming, Scheduling, Supercomputer

1 Introduction

Today high-performance computing (HPC) centers are investment-intensive facilities with short depreciation cycles. An average supercomputer reaches full depreciation in just three years, hence its utilization has to be aggressively managed to produce and acceptable return on investment. A key role in this challenge is played by scheduling software that decides where and when a job submitted by a user has to be started. The scheduling software orders the set of jobs and the set of nodes and then tries to allocate each job to nodes taking into account the amount of currently available computing resources. When a new job is submitted, the software usually applies a back-filling algorithm that tries to place the job on unused resources without delaying the start of highest priority jobs in queues. These priority-rule-based algorithms are simple and reasonably fast, but they usually do not find the best solution of the scheduling problem. One of the most widespread queue-based scheduling software in HPC facilities is PBS Professional [5]. In this work, we present a complete CP model for solving the optimal scheduling problem in HPC machines. The model significantly extends the one in [1] to account for multiple classes of jobs and their temporal constraints.

In addition, the solution space exploration strategies have been optimized for on-line use, taking into account the impact of the schedule computation time on machine utilization. The CP solver based on the new model has been embedded as a plug-in module within the software framework of a well-known commercial HPC scheduler [5] replacing it's scheduling engine. By linking our solver with a state-of-the-art HPC scheduling tool, we have been able to validate our approach on a real-life HPC machine, Eurora from CINECA (Consorzio Interuniversitario Calcolo Automatico). Experiments on Eurora demonstrate that the new scheduler achieves significant improvements in job waiting time with respect to the commercial scheduler used in CINECA, while at the same time maintaining high machine utilization. An experimental campaign on a wide range of synthetic workloads proves that the approach is flexible, robust and well-suited for integration in a portfolio of scheduling strategies to cover different levels of average machine utilizations. In section 3 we show an overview of the scheduling software running on the Eurora HPC machine. In section 4 we formally describe the problem of scheduling. In section 5 we describe the optimization techniques used to model the problem. In section 6 we present our optimization model and all the features implemented to make it desirable for a real HPC center. In section 7 we show results from simulations and from the Eurora supercomputer and we report statistics on the computational overhead. Finally in section 8 we show our conclusions.

2 Related work

The problem of batch scheduling is well-known and widely investigated. The interested reader can refer to [4] for a good survey of the scheduling algorithms used in HPC and computing clusters. Most of the algorithms described in [4] can be implemented withincommercial scheduling software by defining appropriate "scheduling rules". To the best of our knowledge, the only examples that apply optimization techniques to a scheduler in a production context is [2]. In this paper, the author present an optimization technique applied as an extension to the TORQUE scheduler. This extension replaces the scheduling core of the framework with a backfilling-like algorithm that inserts one job at a time into the schedule starting from a previous solution and then applies a Tabu Search to optimize the solution. This approach considers a job as a set of resources. This assumption drastically decreases the flexibility of the scheduler by avoiding the possibility for a job to request more than one node. In our work, instead, we consider a job as a set of nodes, each requiring a set of resources. In this way we maintain the flexibility of commercial schedulers (like TORQUE and PBS Professional) but we deal with a more complex settings w.r.t. [2].

3 Eurora, Heterogeneous Supercomputer

Eurora is a heterogeneous HPC machine of CINECA. Eurora is composed of 65 nodes, one login node with 12 cores, 32 nodes with 16 cores at 3.1GHz and

2 GPU Kepler K20 each and 32 nodes with 16 cores at 2.1GHz and 2 Intel Xeon phi (MIC) each. Users from this HPC machine can submit a job that specifies the amount of resources, nodes and walltime to a queue; a queue is the place where job waits to be executed. Each queue has a name and a priority, after the submission. The scheduling software decides the start time and nodes where to execute the job. The scheduling and dispatching software currently used in Eurora is PBS Professional 12 from Altair; PBS Professional is a Portable Batch System [5] that schedules jobs based on rules. The original scheduler *PBS_sched* can be disabled and replaced by with ad-hoc scheduling algorithms. We take advantage of this functionality to implement in a plug-and-play fashion our optimized scheduling policy.

4 The Scheduling problem

n this section, we formally describe the problem of on-line scheduling and dispatching of a supercomputer. The scheduling problem considers a set of jobs J and a set of queues Q. Every job_i, defined on the set J, is submitted in a specific queue q_i defined on the set Q. Each job, when submitted, has to specify its maximal duration d_i, the number of jobs units u_i (the number of virtual nodes required for the execution) and the amount of required resource r_{ijkl} (cores, memory, GPUs, MICs) for each job unit $k \in [1..u_i]$ and for each $l \in R$, where R is the set of resource. Each node n_j of the system has a limit rl_{jl} for each resource l, with $j \in N$ where N is the set of nodes. We have to assign the starting execution time s_i and for each job unit ju_{ik} of the job job_i, the node n_j where it has to be executed. Given the current time t, the set of running jobs cannot be changed (migration is not supported), while the set of waiting jobs has to be allocated and scheduled on resources without exceeding their capacity at any point in time.

5 Constraint Programming

The technique used in this work to model the problem is Constraint Programming. A Constraint Program is defined on a set of variables, each defined on a discrete domain, and a set of constraints. Differently from Convex Optimization (like LP, ILP, etc...), with this paradigm we are not forced to have a convex polytope as solution set and a convex objective function. The global constraint we will use are:

- $alternative(a, [b], C)$: the constraint holds iff at least C activities from the vector $[b]$ has the same start time and duration of the activity a.
- $cumulative([s], [d], [r], L)$: the constraint holds iff all the activities i defined by a starting at time s_i, a duration s_i and a resource requirement r_i never exceed the resource capacity L at any point in time.
- $noOverlap(a, t, [s], [d])$: the constraint holds iff all the activities i with start time s_i and duration d_i do not overlap an activity with start time a and duration t (for each i we can have $a \geq s_i + di \; OR \; a + t \leq s_i$).

– $synchronize(a, [b])$: the constraint holds iff the start time a is synchronized with each start time i of the vector $[b]$.

6 CP Model

Starting from the work in [1] we create a model that contains all the requirements and services needed by supercomputers in production. For every job i we have a Conditional Interval Variables (CVI, see [3]) job_i. A CVI represents an interval variable. The domain of a CVI is a subset of $\{\perp\} \bigcup \{[s,e)|s,e \in Z, s \leq e\}$. If a CVI has domain $\perp$, this variable does not belong to the model and it is not considered in the solution process. A job's CVI contains the job walltime d_i specified by the user. For every job we have also a matrix UN_i of MxP_{ij} of CVIs, where M is the number of nodes in the system and P_{ij} is the maximum number of job units dispatchable in the jth node. These elements assume the value $s(i)$ if the ith job uses the node j, the value bottom otherwise. R is the set of resources of the node, A the set of jobs in a queue and B the set of running jobs. The base model created is described in 1. With the alternative constraint, we introduce the possibility for every job unit to be displaced in a node partially used by another job unit of the same job. The cumulative constrain the set of jobs start times.

$$
\begin{aligned}
& job_i \geq t \quad \forall i \in A \\
& job_i = s(b) \quad \forall i \in B \\
& alternative(job_i, UN_{ijk}, u_i) \ \forall i = 1..N \\
& cumulative(UN_{ijk}, d_i^{P_{ij}}, r_{ijkl}^{P_{ij}}, rl_{jl}) \ \forall k = 1..M, l \in R
\end{aligned}
\tag{1}
$$

Equation 2 represents the objective function used in this model. This function takes the job waiting-time weighted on the expected waiting time (ewt_i) of the queue where the job is submitted. This objective function is designed to optimize the jobs waitings paying attention to the fairness of these. This mean that waitings have to be distributed taking into account the priority of the jobs.

$$
\min z = \sum_{i=1}^{n} \frac{s_i - q_i}{ewt_i} \tag{2}
$$

7 Experimental Results

We have evaluated the performance of our scheduler in two distinct experimental setups, namely (1) in a simulated environment on Virtual Machines (VM); and (2) on the actual Eurora HPC machine. The PBS software can be configured in different modes to suit the purpose of the system administrator. the following experiments consider two different PBS setups:

1. The CINECA PBS configuration (referred to as PBSFifo): this setup uses a FIFO job ordering, no preemption, and backfilling limited to the first 10 jobs in the queue.

	Test 1			Test 2		
	PBSFifo	PBSWalltime	CP Sched.	PBSFifo	PBSWalltime	CP Sched.
WQT	152,94	137,74	119,77	1034,2	853,681	2441,3
NL	65	60	46	234	200	376
TR	1298810	1223690	1003970	16798300	13693000	16774800
AO	0,47	3,14	11,45	1,02	15,47	34,82

Table 1: Test 1 and Test 2 results

2. A PBS configuration (referred to as PBSWalltime) designed to get the best trade-off between waiting time and computational overhead: this setup employs a strict job ordering (by increasing walltime), no preemption and backfilling limited to the first 400 jobs.

7.1 Simulation-based tests

We have designed the simulation so as to evaluate the performance of our CP scheduler w.r.t. PBS. The experiments differ under a wide range of conditions with respect to number of jobs, job units, and platform nodes. The goal is to assess the scalability of both approaches and their ability to deal with workloads having different resource requirements and processing times. The quality of the schedules was measured according to a number of metrics. Specifically, we have defined:

- *Weighted queue time (WQT)*: sum of job waiting-times, each divided (for fairness) by the maximum wait-time of the job queue.
- *Number of late jobs (NL)*: the number of jobs exceeding the maximum wait-time of their queue.
- *Tardiness (TR)*: sum of job delays, where the delay of a job is the amount of time by which the maximum wait-time of its queue is exceeded.
- *Average overhead (AO)*: average computation time of the scheduler.

In test 1 we simulate all the 65 Eurora nodes: the results are in Table 1. Our model manages to outperform considerably PBSFifo and PBSWalltime in terms of all the metrics related to waiting time and delay. In test 2 tested a 65 nodes configuration with a larger number of jobs (namely 700): the results are reported in Table 1. Due to the large number of jobs and (more importantly) job units, in this case, our framework was forced to employ the overhead reduction techniques. Such techniques are indeed effective in limiting the overhead, but they also have an adverse effect on the quality of the model solutions. As it can be seen in the table, our model yields a small improvement in tardiness w.r.t. PBSFifo, a small increase in the total time in queue, and a considerable increase of the number of late jobs, the WQT, and the weighted tardiness.

7.2 Execution on Eurora

Thanks to our modeling and design from Section 6, we have managed to obtain a scheduling system that is mature enough to be deployed and evaluated on the actual Eurora HPC machine. In detail, we have compared the performance of our approach and the PBSFifo configuration over five weeks of regular utilization of the HPC machine. Since the comparison is performed in a production environment, it is impossible to guarantee that the two approaches process the same sequence of jobs. For this reason, we chose to compare the CP approach and PBSFifo in terms of: (1) the average WQT per job, and (2) the average number of used cores over time (i.e. the average core utilization). Our CP system performed consistently better with an average WQT per job of $\sim 2.50 * 10^{-6}$, against the $\sim 3.93 * 10^{-6}$ of PBSFifo. The standard deviation for the two approaches is very similar. The average core utilization obtained by both approaches during each week, show that the two approach have similar performance, which ranges between 520 and 599 for PBSFifo and between 510 and 573 for CP.

8 Conclusion

In this paper we presented a scheduler, based on Constraint Programming techniques, that can improve the results obtained from commercial schedulers highly tuned for a production environment and we implemented all the features for made it usable on a real-life HPC setting. The scheduler has been tested both in a simulated environment and on a real HPC machine with promising results. We have seen that in the medium hardness range we can improve results obtained by the commercial scheduler by a significant amount (21% on 152 points of WQT) and in the high hardness range we did not get an improvement due to the computational time and the too aggressive technique we had to implement. The experimental results on the Eurora HPC system shown an improvement on the weighted queue time while the system utilization. Future work will focus on the following directions: improving the integration between the scheduling management framework and the optimizer, and developing incremental strategies to hot-start the optimization engine.

References

1. Bartolini, A., Borghesi, A., Bridi, T., Lombardi, M., Milano, M.: Proactive workload dispatching on the eurora supercomputer. In: OSullivan, B. (ed.) Principles and Practice of Constraint Programming, Lecture Notes in Computer Science, vol. 8656, pp. 765–780. Springer International Publishing (2014)
2. Chlumsky, V., Klusácek, D., Ruda, M.: The extension of torque scheduler allowing the use of planning and optimization in grids. Computer Science 13, 5–19 (2012)
3. Laborie, P., Rogerie, J.: Reasoning with conditional time-intervals. In: Proc. of FLAIRS. pp. 555–560 (2008)
4. Salot, P.: A survey of various scheduling algorithm in cloud computing environment. International Journal of research and engineering Technology (IJRET), ISSN pp. 2319–1163 (2013)
5. Works, P.: Pbs professional 12.2, administrators guide, november 2013

Recommender Systems supporting Decision Making through Analysis of User Emotions and Personality

Marco Polignano, Marco de Gemmis

Dept. of Computer Science, University of Bari Aldo Moro, Italy
{name.surname}@uniba.it

Abstract. The influence of emotions in decision making is a popular research topic in psychology and cognitive studies. A person facing a choosing problem has to consider different solutions and take a decision. During this process several elements influence the reasoning, some of them are rational, others are irrational, such as emotions. Recommender Systems can be used to support decision making by narrowing the space of options. Typically they do not consider irrational elements during the computational process, but recent studies show that accuracy of suggestions improves whether user's emotional state is included in the recommendation process. In this paper we propose the idea of defining a framework for an Emotion-Aware Recommender System. The user emotions will be formalized in an affective user profile which can act as an emotional computational model. The Recommender System will use the affective profile integrated with case base reasoning to compute recommendations.

Keywords: Emotions, Recommender Systems, Human Decion Making

1 Background and Motivation

Emotions are an important aspect of our life. They have a regulatory affect on everyday task and heavy influence each decision that we take. Low intensity emotions have a positive advice role [5], while high intensity emotions can be a potential source of biases for a clear and logical reasoning [4]. When users do not have a complete knowledge of the domain, and the decision can produce risky consequences, negative and intensive emotions like fear and sadness can narrow attention and generate less consciousness decision [7]. A system that supports decision making has to adapt its behaviour according to the roles and the types of the emotions that users feel during the decision task. Recommender Systems (RSs) [11] are tools which implements information filtering strategies to deal with information overload and to support users into choosing tasks, by taking into account their preferences and contexts of choosing. RSs can adopt different filtering algorithms based on: the item content (description), the user activity, the knowledge of context, but usually they do not consider emotions and other irrational factors as features involved in the process that computes recommendations.

The research on emotion-aware RS is still at an early stage. Relevance of affect in RSs was discussed in the work by Zheng and Burke [21], that showed an increment of recommendation performance using emotions as a context in a context-aware RS. In this work, emotions play a minor role, because they are considered is the same way as other "rational" contextual features. Our work aims at defining a framework in which emotions play a more relevant role because they are *embedded* in the reasoning process. Another important work on this topic demonstrates how affective labelling of items can increases the performance of content-based RSs [20]. The work face with the problem of affective item modelling, while we would like to focus on modelling affective features of users. Anyway, literature on RS gives a positive valence to the possibility of including emotions in recommendation processes supporting decision making.
In our work, we focus on the following research questions:

1. Which techniques are most suitable to identify users' emotions from their behaviour?
2. How to define a computational model of personality and emotions for improving the user experience with recommender systems?
3. How to include the emotional computational model in a recommendation process?

For each research question we will propose some possible solutions, which are currently at a preliminary stage because the main author is in the first year of his doctoral program. As for emotion detection, we propose to use both explicit and implicit feedback strategies. Sentiment analysis techniques [14] will be adapted to infer the user affective state (emotions, mood) from the analysis of social network posts. Other implicit feedback techniques for the analysis of voice and biometric parameters will be also considered, as well as explicit feedback from questionnaires, like the Big Five Inventory questionnaire [12] to infer the user personality traits.

The idea is to collapse all the information acquired about the emotional state of the user and her personality into an affective profile which stores both rational and irrational data about user decisions. In other words, the affective profile is a knowledge base that allows the RS to reason about past user's choices, emotions felt during the decision process, contexts in which decisions were taken. The final goal is to define an Emotion-aware RS that will exploit case base reasoning [1] to solve new problems by adapting past solutions in similar context, and taking into account the emotional state of the user, as well as personality traits.

2 Emotion Detection Strategies

In the decision making literature, the decision task is influenced by expected emotions and immediate emotions. Expected emotions are affects that the user suppose to prove as a consequence of the decision. Immediate emotions are consequence of an external event that has recently affected the user. Emotion-aware RSs have to identify immediate emotions and forecast expected emotions. The reasoning process should provide recommendations that generate positive expected emotions for a specific user in a defined immediate emotional state. Emotions during the decision process can be detected using implicit or explicit strategies. Explicit strategies are based on method that interacts directly with the user asking her which emotion is felt at the decision time. Questionnaires can be used to identify both user personality traits and user emotions. The 44-items Big Five Inventory questionnaire [12] can be used to infer the user personality traits among the dimensions: Openness to experience, Conscientiousness, Extroversion, Agreeableness, Neuroticism. Often people are not able to explicate correctly emotions, and explicit strategies could not be enough to correct identify immediate emotions. For this reason, implicit strategies can be adopted. Tkalcic[19] shows that emotions can be detected from videos, but with limited accuracy. Poria [15] present a multi-modal framework that uses audio, video and text sources to identify user emotions and to map them into the Ekman's six emotions [6]. The results show that hight precision can be achieved in the emotion detection task by combining different signals. According to this work, an useful implicit source that can be used to obtain immediate emotions informations is the text, and particularly posts gathered from user's Social Network activities. Research on this topic showed that both user personality traits [10, 8] and user emotional state [17, 13] can be inferred by adopting NLP techniques. Machine learning techniques have been also used for this purpose: one of the most useful framework adopted is SNoW, a general purpose multi-class classifier [2]. Strategies based on emotion lexicon are also popular. They usually identify key terms in sentences and, then check the emotions associated with each word in an emotion-based lexicon [3]. In our proposal, we will evaluate different strategies for acquiring both emotions and personality traits.

3 Affective Profile and Recomendation Method

The user affective profile is an extension of the standard user profile used by RSs, usually a list of item with corresponding feedback given by the user. It will be used by the RS to adapt its computational process and to generate recommendations according to emotions. It integrates both rational and irrational elements: user personality traits (PT), historical decision cases (HC), contexts and user expertise (CE).

$$AP = PT \times HC \times CE \tag{1}$$

Personality Traits. Personality traits are formalized as a distribution of percent values among the dimensions: Openness to experience, Conscientiousness, Extroversion, Agreeableness, Neuroticism according to the Big Five model [9]. These elements are the distinctive traits of the user behaviour which allow to predict user common preferences and decisions. These are also important to define the affective features of the user.

Historical decision case. An historical decision case describes accurately the decision making task and emotions felt by the users. A case contains emotions felt during the decision process and the description of the decision task. The decision task can be divided in three stages [18]: early, consuming and exit. These emotions are formalized as a distribution of percent values among the six emotions of Ekman model[6]. During all the decision, strategies of emotion identification from video, audio source will be used. The process will be supported from strategies of emotion extracted from Social Network posts, while an additional emotional value could be gathered from user asking her the emotion felt at decision time. The description of the task is defined by: context of decision, problem, elements among which choosing must be performed, decision taken, feedback in a discrete scale from 1 to 10 to describe the utility of suggestions (1 means not useful, 10 means extremely useful). Future enrichment of these descriptions, including other emotional source are under evaluation.

Context and expertise. This is the rational part of the profile. The context is characterized by explicit features that describe user preferences in the domain. The expertise of a user in a specific domain is defined in terms of the number of decisions taken in that context.

The affective profile stores all the useful informations that RSs can use to adapt their behaviour according to the user affective description. Common RSs are based on an information filtering algorithm that do not consider user's irrational features such as emotions. An Emotion-Aware Recommender System takes as input information about context, immediate emotions and affective profile and generates a list of possible solutions of the problem, influenced by emotionally attributes. The reasoning strategy adopted is based on the emotional historical cases of the user collected in her affective profile. Case-based reasoning is the most appropriate strategy for considering user's rational preferences in a specific context, user's immediate emotions and user's past decisions. This is one of the most commonly adopted machine learning method, that exploits a knowledge-based representation of the context [1]. Case-Based Recommender Systems (CBR-RS) are specific RSs that adopt a representation of user preferences, historical cases and domain of decision to suggest solutions for a new problem, according to similarity to past cases.

The Emotion-Aware RS works similarly to a CBR-RS. In the first step, the recommender has to identify users similar to the active user (the one for which suggestion must be provided). Similarity measures, like cosine similarity, are used on vectors obtained combining personality features and preference features in the specific context. This set of users, including the active user, is used to identify decisions taken in the past. Each historical case have to match the problem, the active user immediate emotional state, and must have positive exit stage emotions (anticipated emotions) or positive user feedback. The matching of context and emotions will be computed using similarity strategies on their descriptive features. From the historical cases detected, candidate solutions are extracted and filtered or ranked according to the context of the problem. Then, an important task is to consider the influence of emotional features on the user based on the risk that the decision involves. Schlosser [16] describes the role of emotions in risky and uncertain domains. Immediate emotions influence the ability to consider all the relevant aspect of the decision, therefore they influence the quantity and quality of informations interpreted. Anticipated emotions influence the utility function of the decision. When users evaluate the consequences of the possible options, they will consider positive and negative anticipated emotions associated with them. For instance, a RS for financial investments must provide understandable and explicable solutions. It has to propose actions that will maximize positive expected emotions and minimize the diversification of the options. Emotions influence decisions also in a low-risk domains. As decisions in these domains are easy to revert, it is possible for the RS to suggest new and uncommon items, by diversifying recommendations according to preferences and emotional state of the user. An application that fall in this category is a music recommender system which can propose playlists according to the user mood and her tendency to maintain or change it based on her personality traits.

If poor data are available for the proposed computation pipeline, an inference from personality traits in the specific domain could be done to choose possible candidate solutions. For example, for people with an high value of "Openness", uncommon solutions can be selected as a candidate recommendations.

4 Final Remarks and Ongoing Work

Emotions are important elements of people's life. In each decision making task, emotions influence the choosing process. In those contexts where decisions lead to risky consequences, emotions need to be mitigated, while in others, such as music recommendation, they could be amplified and used to generate useful suggestions. Systems that support the decision making task, currently take into account emotions in a limited way, while we have proposed a solution able to embed emotions and personality traits into the recommendation process. The ideas proposed in this paper are currently developed within the doctoral program of the main author, therefore they are still at a preliminary stage.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications 7.1, 39-59, (1994)
2. Alm, C. O., Roth, D., Sproat, R.: Emotions from text: machine learning for text-based emotion prediction. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, 579-586, (2005)

3. Balahur, A., Hermida, J. M., Montoyo, A.: Building and exploiting emotinet, a knowledge base for emotion detection based on the appraisal theory model. Affective Computing, IEEE Transactions on 3.1, 88-101, (2012)
4. Baumeister, R. F., Heatherton, T. F., Tice, D. M.: Losing control: How and why people fail at self-regulation. Academic Press, (1994)
5. Damasio, A. R.: Descartes Error: Emotion, reason and the human brain. Penguin Group (USA), (1994)
6. Paul Ekman: "Facial expression and emotion." American Psychologist, 1993
7. Forgas, J. P.: Mood and judgment: the affect infusion model (AIM). Psychological bulletin 117.1, 39, (1995)
8. Golbeck, J., Robles, C., Edmondson, M., Turner, K.: Predicting Personality from Twitter. SocialCom/PASSAT, 149-156, (2011)
9. Goldberg, L. R.: An alternative "Description of personality": the big-five factor structure. Journal of personality and social psychology 59.6, 1216, (1990)
10. Gosling, S. D., Gaddis, S., Vazire, S.: Personality Impressions Based on Facebook Profiles. ICWSM 7, 1-4, (2007)
11. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender systems: an introduction. Cambridge University Press, (2010)
12. John, O. P., Srivastava, S.: The Big Five trait taxonomy: History, measurement, and theoretical perspectives. Handbook of personality: Theory and research, 2, 102-138, (1999)
13. Pak, A., Paroubek, P.: Twitter as a Corpus for Sentiment Analysis and Opinion Mining. LREC. Vol. 10, (2010)
14. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and trends in information retrieval 2.1-2, 1-135, (2008)
15. Poria, S., Cambria, E., Hussain, A., Huang, G. B.: Towards an intelligent framework for multimodal affective data analysis. Neural Networks 63, 104-116, (2015)
16. Schlosser, T., Dunning, D., Fetchenhauer, D.: What a feeling: the role of immediate and anticipated emotions in risky decisions. Journal of Behavioral Decision Making 26.1, 13-30, (2013)
17. Strapparava, C., Mihalcea, R.: Learning to identify emotions in text. SAC, 1556-1560, (2008)
18. Tkalcic, M., Kosir, A., Tasic, J.: Affective recommender systems: the role of emotions in recommender systems. In Proc. The RecSys 2011 Workshop on Human Decision Making in Recommender Systems, 9-13, (2011)
19. Tkalcic, M., Odic, A., Kosir, A.: The impact of weak ground truth and facial expressiveness on affect detection accuracy from time-continuous videos of facial expressions. Information Sciences 249, 13-23, (2013)
20. Tkalcic, M., Odic, A., Kosir, A., Tasic, J.: Affective labeling in a content-based recommender system for images. Multimedia, IEEE Transactions on 15(2), 391-400, (2013)
21. Zheng, Y., Mobasher, B., Burke, R. D.: The Role of Emotions in Context-aware Recommendation. Decisions@ RecSys, 21-28, (2013)

Interaction with a personalised smart space for enhancing everyday life

Assunta Matassa

Dip. Informatica, Università di Torino
Corso Svizzera 185 - 10149 Torino (Italy)
matassa@di.unito.it

Abstract. In the last years the interest for designing and implementing smart spaces grew significantly. Many researchers adopted a top-down approach, focusing on embedding smartness in buildings, objects and everyday artefacts. In my research work I propose the adoption of a user-centred design approach to reach a new definition of smart spaces based on people's needs and requirements. The main goal will be the definition of a new interaction paradigm supporting natural and spontaneous ways of exchanging information between people and their surroundings.

1 Introduction

The concept of smart space characterizes a physical place where people and technologies cohabit and continuously exchange information in order to create an interactive space where people's needs and requests are satisfied in an intelligent way. Focusing on the importance of space in people's everyday life, there are several studies that highlight how human beings establish a deep relation with the physical environment in which they live and how, in turn, the environment influences the creation of their own identities and personalities [20]. Being part of a space stimulates the creation of an emotional bond and a relationship based on the exchange of information and feelings with what is occurring in there.

Introducing technologies in everyday environments makes it harder to maintain these relationships. However, there are many authors interested in introducing technologies in spaces in order to make them smart. In this view, they adopted a top-down approach to characterize a smart space. However, this approach emphasizes the distance between people and space and makes it impossible to maintain an authentic relation between people and space. I propose the adoption of a user-centred design approach to achieve a new characterization of smart spaces based on people's needs and requirements and in which smartness is related to the capability of creating a personalised space that enables a deeper and emotional bond between an individual and the space itself.

The main aim of this work is to characterise a personalized smart space (PSMA) as a an entity where human bodies, the space itself and the smart objects in it create a real interactional network which can increase the quality of everyday life according to the needs, preferences and requirements of each human being that lives inside it.

2 Pillars

A PSMA is the setting where a plethora of different intelligent components with various capabilities and levels of smartness live and cooperate, a complex system created by the interrelation of people, places and objects.

Body. Traditionally, the body can be considered as a referent to construct adapted spaces and buildings [15]. Designing the Vitruvian Man, Leonardo da Vinci suggested to adopt the body as a measure of everything (e.g., inch and feet units in the imperial system), using the proportions of the body as instruments to identify a space that is best suited for humans.

Moreover, the body already presents a set of *special tools* for interacting with the environment: the senses [14]. Human experience starts from them: touching, smelling, hearing, tasting, seeing, exploring the environment with the senses, building knowledge about it. According to the theories of embodied cognition and embodied space, knowledge derives from the coupling of *action* and *perception*, exploiting the experiences generated by the continuous interactions between the body and the environment [4]. Nowadays, there is an increasing interest in rediscovering bodies [5], senses [18] and gestures[7] in order to achieve new interaction models to experience the surrounding environment.

Space. I would like to consider a space as a composite place, where people, objects and physical space cohabit.

Currently, a smart space (SMA) is defined as a place enhanced with digital capabilities. Bringing one step further the considerations about body and space in Section 2, we can analyse the role that the body plays in defining the space. The human represents the main component in the process to describe and organise the surrounding environment. Stressing the importance of the bond that people can establish with the space and exploiting body as the main element to keep in contact with the environment, I intend to reach an innovative definition of SMA based on the perceptions of space from a user-centred perspective, taking into account the potentiality that humans have to interact in a SMA. We will refer to this notion as Personalised Smart Space (PSMA).

Objects. Objects represent instruments designed to accomplish a task. First, objects extend human capabilities, improving and/or augmenting her common abilities [19]. Second, objects embody what people can do with it [6] [16], as an interface that enables actions and usages according to the affordances that it offers. Going towards the *Ubiquitous Computing paradigm* [21] and Internet of Things (IoT) era, smart physical objects (SPOs) are able to act in the environment and to connect to the Internet. SPOs are the combination of two main components: a physical layer including a controller and a set of sensors, actuators and communication capabilities and a digital layer that enables to manage their behaviour in the context of use [13]. SPOs can be characterised by different levels of smartness: from the ability to exchange information with people and other smart objects, to the ability of managing knowledge about themselves, their role, scope and relation in the surrounding environment [1] to the ability of learning from experience [17].

3 Goal and methodological approach

The main goal of my work is to define the novel concept of PSMA as a complex system where body and SPOs cohabit in a shared experiential space with a continuous exchange of information, collaboration and negotiation between them according to the needs of each individual. Using a user-centred approach, I intend to support the idea that not only is a PSMA defined by the intelligence embedded in it, but it is especially characterized by the capability to adapt in order to accomplish the individual needs, preferences, requirements of each single user, becoming her personal PSMA, able to reflect her personal experiences.

In order to characterize a PSMA and to introduce new paradigms of interaction with it, I will take two different methodological steps: first, I will characterise a PSMA as an SPO, or better as a composite SPO, in which the combination of SPOs with different levels of intelligence gives it an higher level of intelligence and responsiveness to users' needs; second, adopting user-centered design, my focus would be on people who will play a central role in the process of defining the intelligence of the space. Body will be used as an interface and senses would be the principal instruments for exploration of the space.

3.1 First step

Defining smartness. In order to characterize smartness in a space, we need a classification of smartness in objects. An SPO is a combination of a physical and a digital layer; the latter can be described as a set of computational functionalities that enhance its abilities yet preserving its physical aspect. Many dimensions can be taken into account in order to characterise intelligence in an SPO. First of all, smartness can be regarded as the awareness about its roles and goals according to different contexts of use. Second, it is related to the ability to interact with humans and other SPOs that constitute the surroundings. Third it can be related to the ability of making inferences and of learning from experience. In summary it can be related to the interactional and problem solving capabilities of an SPO. The highest level of intelligence can be reached with the ability to change its behaviour according to contextual situation, supporting a continuous and active exchange of information and states between the SPO and its surrounding [8]. The final goal is to introduce a strong characterization of SPOs describing their abilities and their problem solving capabilities in a contextual situations.

Coding the smart space corpus. Given the characterization of SPOs, I will characterize a SMA as a composite SPO, whose intelligence derives from the aggregation of the level of intelligence of the composing objects. I expect that the level of intelligence of the SPO is more than the aggregation of its components. I estimate the following main results. First, I will provide a framework following designers to describe how a SMA can be obtained taking into account each component that could be inside it and the minimum level of smartness that it should have in order to be active and proactive with the surroundings. Second, I will provide an innovative corpus supporting people's interaction with a smart space. The main idea is to create a coding of space components (objects, space, people) in order to define a framework to represent SMAs and to support the communication between the components in it and the negotiation of their actions.

3.2 Second step

In the second step, I will focus on human beings and their interaction with the surroundings. As already explained in Section 2, a PSMA can be characterized as complex customized system capable of reshaping and modifying itself for responding to the personal demands of each single individual.

In order to achieve this goal, my plan is: first, to take into account user's needs in order to transform a SMA into a PSMA able to respond to them; second, to design a new interaction paradigm that allow users to exchange needs and information with the environment in a more natural and spontaneous way.

Mapping user's needs. In order to map the user's needs, I will borrow techniques from user adaptive ubiquitous systems that are able to adapt their behaviour and interaction based on user's features and the context. These systems rely on representations of users (User Models) [2] that can provide a complete picture of each user with her features, habits, preferences, behaviours and activities. Exploiting these models, they can support the selection of a set of appropriate services adapted to the user's features. A PSMA can result from the combination of user modeling and adaptation technologies with the functionalities offered by a SMA in order to provide services personalised to user's features, place features and SPOs.

Designing a new interaction paradigm between people and spaces. The emergence of a PSMA able to know, understand and predict user's needs, preferences and requests, will allow a user to interact with it without additional effort.

Bypassing the traditional interaction model (Fig. 1), the increasing adoption of wearable technologies opens new opportunities, offering an interaction with surrounding spaces that needs a minimal effort from the users. In fact, wearable computing allows user to use her own body to get in touch with the environment.There is a growing interest in designing new natural interaction models, using gestural interaction and a *body in action* according to the embodied cognition theory and its applications [3]. The exploitation of the richness of the body, of the senses and of the movements considers actions as the most relevant part of cognition [4]. As a result, the growth of studies about full-body interaction restores the importance of the body as a controller able to move with several degrees of freedom and, at the same time, as an interface to exchange information with the surrounding environment.

As a consequence, there is a huge space of new perspectives for the design of ubiquitous natural interaction exploiting the body and senses, taking inspiration from the matching between the capabilities offered by innovative technologies such as wearable computing, tangible interfaces and the renewed interest in body and senses (Fig. 2).

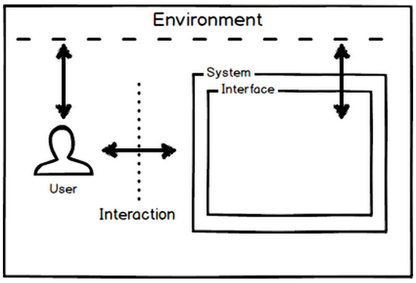


Fig. 1: Traditional interaction

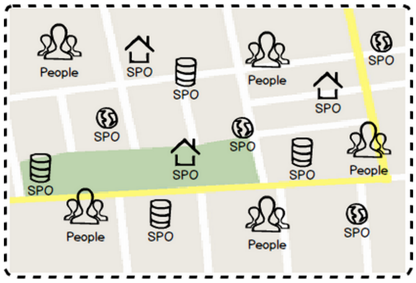


Fig. 2: Interaction in PSMA

4 Status of current research

The current status of my research project is as follows: I analyzed what an SPO is and in particular how intelligence can be characterized, decomposing it into several dimensions, discussing and analysing a notion of "granularity of its intelligence" [9]. I then introduced new affordances to communicate the augmented functionalities of SPOs [12]. I introduced natural interaction paradigms for spaces, focusing on wearable computing [11], full body experience and multisensory experience [10]. In this way I explored the body as a natural interface to keep in contact with the surroundings. Next steps will be toward understanding new frontiers in natural interaction in spaces.

5 Research directions and future steps

The main aim of my work is twofold: first, choosing a set of new interactive tools that stimulate a natural interaction in the spaces allowing a continuous exchange of information between people and a SMA in order to create a PSMA; second, designing a new interaction paradigm to support a direct interaction with spaces exploiting body as a natural interface and adopting gesture and senses as the only tools to accomplish these tasks.

The interaction in a PSMA could be based on a new code as a corpus to exchange information in user-friendly way with the environment (Fig. 2). The constant exchange between descriptions of people derived from user models and capabilities and knowledge embedded in a SMA will allow us to build a new concept of PSMA completely based on the user features, needs and preferences, without any mediation in interaction. This will allow people to interact using their body in a more spontaneous way and allow each one of them to build her own PSMA.

Taking inspiration from the steps already defined in Section 2, the ultimate goal will be the definition of a spatial framework based on these components and able to provide a set of instruments and guidelines to build a PSMA starting from the definition of the space itself.

References

1. Beigl, M., Gellersen, H., Schmidt, A.: Mediacups: experience with design and use of computer-augmented everyday artefacts. Computer Networks 35(4) (2001)
2. Brusilovsky, P., Millán, E.: User models for adaptive hypermedia and adaptive educational systems. In: The adaptive web. Springer-Verlag (2007)
3. van Dijk, J., van der Lugt, R., Hummels, C.: Beyond distributed representation: embodied cognition design supporting socio-sensorimotor couplings. In: Proc. of the TEI. ACM (2014)
4. Dourish, P.: Where the action is: the foundations of embodied interaction. MIT press (2004)
5. Featherstone, M.: Body image/body without image. Theory, Culture & Society 23(2-3), 233–236 (2006)
6. Gibson, J.: The theory of affordances. Hilldale, USA (1977)
7. Harrison, C., Tan, D., Morris, D.: Skinput: appropriating the body as an input surface. In: Proc. of the CHI. ACM (2010)
8. Kawsar, F.: A document-based framework for user centric smart object systems (2009)
9. Matassa, A., Cena, F., Console, L., Torre, I.: Smart physical objects manifesto. under revision
10. Matassa, A., Console, L., Angelini, L., Caon, M., Khaled, O.: 1st Workshop on Full-Body and Multisensory Experience in Ubiquitous Interaction. `https://sites.google.com/site/bodysenseux/` (2015), [Online; accessed 11-June-2015]
11. Matassa, A., Rapp, A., Simeoni, R.: Wearable accessories for cycling: tracking memories in urban spaces. In: Proc. of the UbiComp. ACM (2013)
12. Matassa, A., Simeoni, R.: Eliciting affordances for smart objects in iot era. In: Proc. of the COIOTE (2014)
13. McEwen, A., Cassimally, H.: Designing the internet of things. John Wiley & Sons (2013)
14. Merleau-Ponty, M., Smith, C.: Phenomenology of perception. Motilal Banarsidass Publishe (1996)
15. MoraGuiard, J., Pares, N.: Child as the measure of all things: the body as a referent in designing a museum exhibit to understand the nanoscale. In: Proc. of the IDC. ACM (2014)
16. Norman, D.A.: The design of everyday things: Revised and expanded edition. Basic books (2013)
17. Perez Hernandez, M., Reiff-Marganiec, S.: Classifying smart objects using capabilities. In: Proc. of the SMARTCOMP (2014)
18. Shilling, R., Shinn-Cunningham, B.: Virtual auditory displays. Handbook of virtual environment technology (2002)
19. Streitz, N., Röcker, C., Prante, T., Van Alphen, D., Stenzel, R., Magerkurth, C., et al.: Designing smart artifacts for smart environments. Computer 38(3) (2005)
20. Tuan, Y.: Space and place: The perspective of experience. U of Minnesota Press (1977)
21. Weiser, M., Brown, J.: The coming age of calm technology. In: Beyond calculation. Springer (1997)