5th International Symposium on Rules (RuleML 2011@IJCAI)

Barcelona, Spain
July 19th, 2011

Proceedings of the Doctoral Consortium and Poster Session

Editors:

Carlos Viegas Damásio
Centro Inteligência Artificial (CENTRIA)
Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
Caparica, Portugal

Alun Preece
School of Computer Science & Informatics
Cardiff University, UK

Umberto Straccia
ISTI-CNR
Pisa, Tuscany, Italy

Preface

This volume contains the papers presented at the first edition of the Doctoral Consortium of the 5th International Symposium on Rules (RuleML 2011@IJCAI) held on July 19th, 2011 in Barcelona, as well as the poster session papers of the RuleML 2011@IJCAI main conference.

The RuleML 2011 doctoral consortium is a new initiative of the International Symposium on Rules, RuleML, to attract and promote Ph.D. research in the area of Rules and Markup Languages. The doctoral symposium offers to students a close contact with leading experts on the field, as well as the opportunity to present and discuss their ideas in a dynamic and friendly setting. The program committee of RuleML 2011 doctoral consortium coincides with the one of RuleML 2011@IJCAI to obtain reviewing and feedback from the leading experts on rules and semantic technology.

There were 9 submissions to the doctoral consortium and one poster submission. Each submission was thoroughly peer-reviewed, receiving on average between 4 and 5 reviews from program committee members. We hope that all submitting students benefitted from this feedback. Following the review process, the committee decided to accept 4 doctoral consortium papers and one poster paper.

The reviewing and proceedings generating processes have been conducted using EasyChair.

The organisers would like to thank the entire RuleML 2011@IJCAI organising team and program committee members for their competence, enthusiasm and support in making this first RuleML Doctoral Consortium, and 2011 poster session, a success.

July 5, 2011

Carlos Viegas Damásio,
Alun Preece,
and Umberto Straccia

Table of Contents

RuleML-2011@IJCAI Doctoral Consortium Papers

RuleML-2011@IJCAI Poster Papers

Program Committee

Alexander Artikis	NCSR ”Demokritos”
Colin Atkinson	University of Mannheim
Costin Badica	University of Craiova, Software Engineering Department, Romania
Ebrahim Bagheri	National Research Council Canada
Matteo Baldoni	Dipartimento di Informatica, Univ. di Torino
Mikael Berndtsson	University of Skövde
Pedro Bizarro	University of Coimbra
Guido Boella	University of Torino
Lars Braubach	University of Hamburg
Christoph Bussler	Xtime, Inc.
Horatiu Cirstea	Loria
Claudia D’Amato	Computer Science Department - University of Bari
Célia Da Costa Pereira	University of Milan
Weichang Du	University of New Brunswick
Jenny Eriksson Lundström	Uppsala University
Vadim Ermolayev	Zaporozhye National Univ.
François Fages	INRIA Rocquencourt
Luis Ferreira Pires	University of Twente
Michael Fink	Vienna University of Technology
Paul Fodor	State University of New York at Stony Brook
Enrico Francesconi	ITTIG-CNR
Fred Freitas	CIn-UFPE
Aldo Gangemi	CNR-ISTC
Giancarlo Guizzardi	Ontology and Conceptual Modeling Research Group (NEMO)/Federal University of Espirito Santo (UFES)
Ioannis Hatzilygeroudis	University of Patras, Dept of Computer Engineering & Informatics
Stijn Heymans	SemanticBits, LLC
Rinke Hoekstra	University of Amsterdam/Vrije Universiteit Amsterdam
Yuh-Jong Hu	National Chengchi University
Joris Hulstijn	
Giovambattista Ianni	University of Calabria
Minsu Jang	Electronics & Telecommunications Research Institute
Krzysztof Janowicz	GeoVISTA Center, Department of Geography; Pennsylvania State University, USA
Mustafa Jarrar	Birzeit University
Paul Krause	University of Surrey
Jérôme Lang	LAMSADE
Domenico Lembo	DIS, Univ. di Roma ”La Sapienza”

Francesca Alessandra Lisi	Dipartimento di Informatica - Università degli Studi di Bari
Jorge Lobo	IBM T. J. Watson Research Center
Emiliano Lorini	IRIT
Thomas Lukasiewicz	Oxford University
Michael Maher	National ICT Australia and University of New South Wales
Jing Mei	IBM China Research Lab
Leora Morgenstern	New York University
Grzegorz J. Nalepa	AGH University of Science and Technology
Georgios Paliouras	Institute of Informatics & Telecommunications, NCSR "Demokritos"
Monica Palmirani	CIRSFID
Jeff Z. Pan	University of Aberdeen
Jeffrey Parsons	Memorial University of Newfoundland
Fabio Porto	LNCC
Alun Preece	Cardiff University
Michael Rosemann	Queensland University of Technology
Antonino Rotolo	CIRSFID, University of Bologna
Marco Seiriö	RuleCore
Guy Sharon	IBM
Davide Sottara	DEIS, Faculty of Engineering, University of Bologna (Italy)
Giorgos Stamou	National Technical University of Athens
Giorgos Stoilos	Oxford University Computing Laboratory, UK
Nenad Stojanovic	FZI at the University of Karlsruhe
Umberto Straccia	ISTI-CNR
Leon Van Der Torre	ILIAS
Wamberto Vasconcelos	Department of Computing Science, University of Aberdeen
Carlos Viegas Damásio	Universidade Nova de Lisboa
George Vouros	University of the Aegean
Kewen Wang	Griffith University
Renata Wassermann	University of Sao Paulo

Extend Commitment Protocols with Temporal Regulations: Why and How

Elisa Marengo, Matteo Baldoni, and Cristina Baroglio*

Dipartimento di Informatica, Università degli Studi di Torino
{emarengo, baldoni, baroglio}@di.unito.it

Abstract. The proposal of Elisa Marengo's thesis is to extend commitment protocols to explicitly account for *temporal regulations*. This extension will satisfy two needs: (1) it will allow representing, in a flexible and modular way, temporal regulations with a normative force, posed on the interaction, so as to represent conventions, laws and suchlike; (2) it will allow committing to complex conditions, which describe not only what will be achieved but to some extent also *how*. These two aspects will be deeply investigated in the proposal of a unified framework, which is part of the ongoing work and will be included in the thesis.

1 Commitment-based Interaction Protocols

The issues of communication and cooperation are crucial in the area of Multiagent systems (MAS). The common solution is to rely on agent interaction protocols. Among different proposals, *commitment protocols* [24] have been widely adopted. All agents, involved in an interaction ruled by a commitment protocol, share the semantics of a set of actions which affect the social state. This semantics is based on the notion of *social commitment*. The idea is that if an agent takes a commitment towards another agent to bring about a condition, then, it will behave in such a way to fulfill the engagement sooner or later. In this respect, commitment protocols have a *deontic nature*, because a commitment introduces a social expectation on the responsibility of some agent towards some other agent to perform something or to achieve some result. Commitment protocols suit well open MAS because they are *respectful* of the agents' autonomy, since no introspection to the agents' mental states is required [26]; they are *dynamic* because commitments can be created, released, deleted and suchlike, and *flexible* because agents are free to take advantage of opportunities or to follow shortcuts [29].

Commitment protocols have fundamentally changed the process of protocol specification from a procedural approach (i.e., prescribing *how* an interaction is to be executed) to a declarative one (i.e., describing *what* interaction is to take place) [28]. Agents decide which action to perform depending on the commitments they have taken and this is because they want to comply with the protocol and fulfill the engagements they have taken [28]. However, in many practical situations this is not sufficient. *Why*? Because, in many cases it is necessary to express some hints on how the interaction should evolve [7, 18]. For example, it is necessary to express that some ways to fulfill the commitments

[*] M. Baldoni and C. Baroglio are the advisor and co-advisor.

are preferred over others, or that only some of them are legal. This does not mean going back to procedural approaches, but it means reconsidering the *how*. In particular, it could be necessary to express *commitments to temporal regulations* and to represent *legal patterns of interaction*. The former are needed to express the engagement of someone to achieve something and in a specific order. For instance, an insurance company commits to paying an in-network surgeon for a procedure only after a covered patient has undergone the procedure. Patterns of interaction, instead, can capture conventions, laws, preferences, habits, or, in general, rules that hold a given reality. For example, in a democratic assembly, a participant cannot speak if she has not obtained the floor.

The thesis, therefore, focuses on the *specification of interactions*, which require a degree of expressiveness that commitments alone do not have. We propose an extension of commitment protocols in order to (i) supply a way for expressing patterns of interaction, capturing laws, conventions and whatever constrains the interaction and (ii) extend the regulative nature of commitments with the possibility of explicitly committing to temporal regulations. A further challenge is how to provide a specification of the interaction in which agents can recognize the normative force of the temporal regulations and explicitly accept them. Indeed, since an agent is free to violate or to behave in accordance with a norm, for a regulation to *influence* the agents' behaviour it must be ascribed of a normative force and, then, it must be accepted as a norm by the agent [13]. The final step will investigate this aspect and propose a unified framework in which both patterns of interactions and commitments to temporal regulations can find place.

2 Temporal Regulations and Commitment Protocols

We discuss how temporal regulations and commitment protocols can be combined: Section 2.1 describes our proposal for including patterns of interaction in protocol specification; Section 2.2 describes commitments to temporal regulations.

2.1 *Why (and How)* expressing legal patterns of interaction

Commitment-based protocols represent a valid solution for interaction protocols specification in open and heterogeneous MAS, mostly because they take into account and respect the agents' and MAS characteristics. For instance, they are respectful of the agents' autonomy, and they do not require a particular implementation or architecture to the agents that are part of the system (heterogeneity). However, to be considered a complete tool for interaction specification, commitment protocols cannot disregard the possibility of specifying *patterns of interaction* as temporal regulations. This requirement is supported by many proposals in the literature (see Section 3) and is due to the need of discriminating those possible executions that are legal from those that are not. These patterns can specify rules of different nature, like habits, conventions, laws, protocol compositions or simply preferences. Basically, they capture a partial ordering between certain actions (or states of affair to be achieved).

Our proposal, described deeply in [4, 7], relies on Searle's definition of a social reality [23]. In particular, he identifies a *constitutive* and a *regulative* specification. The former defines a set of actions as foundational of a certain context. In commitment protocols this corresponds to the *social meaning* of the actions, i.e. to actions' semantics, usually given in terms of effects on the social state. The latter, instead, captures how things should be carried on. In current proposals there is not a clear distinction between the constitutive and the regulative part of the specification. Some approaches only care of the regulative nature of commitments (once a commitment is taken, it must be be fulfilled), but completely disregard a specification of *how* things should be done. The thesis proposes an explicit and declarative definition of the regulative specification, given by means of a set of constraints expressed in Linear-time Temporal Logic (LTL). Such constraints [4] define a relative order among different conditions (facts and commitments) that become true in the social state. For example, it allows to express that something can become true only after something else holds, or that if something becomes true, something else must hold sooner or later. The choice of a declarative representation of constraints allows for the specification of what is mandatory and what is forbidden in a protocol, without the need to enumerate the allowed executions (on the contrary to procedural approaches). Indeed, such enumeration is often a huge task, when considering open and dynamics MAS, it limits reusability and the agents' autonomy.

One of the main differences w.r.t. other works from the literature, e.g. [25, 17, 20], is that temporal constraints are defined in terms commitments and facts, which are, broadly speaking, the effects of the social actions, and not directly on actions (events). This improves flexibility and easiness of reuse in different contexts. Suppose, for instance, to have a specification with the action "pay-by-cash" with semantics *paid*, and to have the constraint "$paid\ before\ sent$". Then, suppose that a change in the context requires that also payment by credit cards can be performed. In this case it is necessary to add a new action"pay-by-credit-card" with the same semantics of "pay-by-cash", i.e. *paid*. Since the constraint is not defined directly on actions, it is not necessary to change it or add a new constraint. Indeed, $paid\ before\ sent$ already constrains the execution of both actions. These aspects are studied in [6], where the adaptation of the Contract Net Protocol to different contexts is discussed. However, where needed 2CL constraints can be used to rule directly actions. The way this is done is by adding a specific effect for each action and then by using these effects in the definition of the constraints.

Orders among actions could be obtained also by adding ad-hoc preconditions to the executability of actions, as done in [28, 11, 16, 12]. However, this solution is not flexible, since regulations are hidden in the actions' definitions and thus difficult to be recognized, updated or modified. Moreover, agents should be always free to decide whether sticking to regulations. If regulations are realized by means of preconditions, agents cannot but choose which action to perform among those that are executable. Thus, they are forced to respect the rules and this is against the normative nature of regulations [13]. In our approach, which is orthogonal to preconditions definition, agents are free to evaluate different alternative paths and to take advantage from opportunities by choosing, among these, the most convenient for them. The role of the constraints is to restrict the set of legal executions, but an agent is free to decide to stick at the rules or to violate them. In the second case the agent knows it could be punished (sanctioned).

A real case study in which this approach has been tested is for the representation of *MiFID: Markets in Financial Instruments Directive* [8]. This directive by the European

Union regulates the interaction of banks, clients, and financial intermediaries in order to guarantee the investor from the intermediaries. The complete example can be found at `http://www.di.unito.it/~alice/2CL/`.

2.2 *Why (and How)* Committing to regulations

In many practical situations, commitments involve rich temporal structures rather than simple conditions to achieve. Let us consider a few examples: (a) an insurance company commits to reimbursing a covered patient for a health procedure provided the patient obtains approval from the company prior to the health procedure; (b) a pharmacy commits to provide medicine only if the patient obtains a prescription for that medicine; (c) an insurance company commits to paying an innetwork surgeon for a procedure only after a covered patient has undergone the procedure. Presumably, the surgeon would bill the insurance company after performing the procedure.

Commitments alone do not have the degree of expressiveness required by these conditions. Indeed, conditions in conditional commitments do not impose a temporal ordering: the consequent condition can be achieved even if the antecedent condition does not hold. The contribution of the thesis for capturing these aspects is described in [18] and consists in a new formalization of commitments, where temporal regulations are incorporated as content of commitments themselves. In this way regulations assume a normative force which is due to the regulative nature of commitments. For example, $C(x, y, \top, a\ before\ b)$ expresses the engagement of x towards y not only to make a and b happen but also to make them happen in the given order. Participants to the interaction will be able to guide their actions locally, in order to not violate any commitment they have taken, and to judge the compliance of their counter-parties. Indeed, since regulations are placed inside commitments, the debtor will be considered responsible and thus liable for violations. Consider, for example, a regulation saying that a physician's referral should precede a surgeon's procedure; in this situation, in case of violation, it is not clear whether the physician is responsible for moving first or the surgeon is responsible for moving second. By placing the regulations in commitments, we make it explicit that it is the debtor of the commitment who needs to ensure its satisfaction.

For this reason it becomes fundamental for an agent to establish, before taking a commitment, if it has a sufficient support by the other agents. The elements the agent has to consider are both the set of actions it can perform and the cooperation it can get from the others, via the set of commitments of which it is the creditor. To this aim, we formalized the notions of *control* and *safety*. The former captures the capability, for an agent, to bring about a regulation. It depends on the actions a certain agent can perform and on commitments directed towards it. The latter is strictly related to the notion of control: a commitment is safe if its debtor has established sufficient control to guarantee being able to discharge it.

To the best of our knowledge, no approach for protocol specification based on commitments allows to express commitments to temporal regulations. Placing temporal regulation inside commitments, however, allows for the representation of a debtor and thus allows to precisely identify who is responsible for each regulation and potentially liable for a violation. This is an advantage w.r.t. approaches based on expectations [1] which are not scoped by a debtor and a creditor. Moreover, it helps make the regulations explicit within the system of interacting agents and thereby facilitates their coordination. Accordingly to [13] it allows also to explicitly represent the recognition and the acceptance of a regulation by the agents.

3 Related Works

The need of expressing temporal regulations is supported by many attempts in the literature to rule actions' execution along the interaction. However, in our opinion, all these attempts can be improved in order to better take into account regulative aspects without compromising the flexibility and all the good properties of commitment protocols.

Fornara and Colombetti [14] propose a model based on *interaction diagrams*, a kind of specification which is similar to UML sequence diagrams. The choice of relying on interaction diagrams is very strong because it forces the ordering of action execution defining a strict set of allowed sequences. It basically can be classified as a procedural approach, thus presenting the same shortcomings [20]: it weakens agents' autonomy to decide which action to perform and their capability to take advantage of opportunities; it is too rigid, where instead the openness and dynamicity of MASs require higher flexibility of the specification.

The use of a declarative approach is proposed by Singh [25] and by Mallya and Singh [17]. In these works they define a *before relation* applied to events. The idea is that when a before relation among two activities is specified, the only thing that matters is the order among the two, no matter what happens in-between. Even if the choice of adopting a declarative specification overcomes many limits of the proposals described before, the main limitation is that temporal regulations are defined over actions (events). As described in the previous section, a greater degree of decoupling between actions and temporal specifications can, in our opinion, support better the openness of MAS. Moreover, this kind of regulation is conceived as a solution for service composition external from protocol specification. In our proposal, instead, temporal regulations actively contribute to the definition of the protocol.

The same shortcoming can be found in other proposals. It is hard to be exhaustive but let us consider a proposal inspired from the neighboring area of business processes. Pesic and van der Aalst [22] propose ConDec, a declarative language for business process representation. ConDec is a graphical language grounded in Linear-time Temporal Logic, which is used to rule the activities that compose a process. Montali and colleagues [9, 20] integrate ConDec with SCIFF thus giving a semantics to actions that is based on expectations. The authors use this approach to specify interaction protocols and service choreographies. As the previous one, also this proposal is based on actions, thus suffering the same shortcomings.

Dialogue games are another solution for communication specification. Different kinds of dialogues basically define different kind of schema according which the agents can interact. The differences among them are given by the aim of the communication (e.g. persuade, inform, negotiate). Our approach is more general, since it provide the basic components for interaction specification and since it is not limited to communication (message exchange), as in [15, 19], but to interaction in general. By means of this tool the desired interaction can be declaratively drawn according to the needs and to the aim of the system, without having to choose one among predefined schema. This is along the line of the claim by Singh in [10]. The idea is that a standard is difficult to be used “as-is” for modelling a desired system. To model a desired interaction he proposes standards for standard definition. Similar considerations holds for works that propose to use commitments for ACL semantics: they define predefined schema (type) for actions specification, bringing to an undesired rigidity. Our approach, and in general approaches based on the meaning of messages [10], are more flexible.

4 Ongoing and Future Work

Our proposal, for patterns specification and commitments to temporal regulations, allows facing two different lacks of commitment protocols related to temporal regulations. The thesis will finally investigate a unified framework in which both aspects can be reconciled under a common normative force. In other words, agents will be provided of the necessary means to explicitly recognize and accept temporal regulations, thus accepting their behaviour to be influenced by them [13]. Of course, agents will be free to decide to violate them in every moment. Thus, this framework will allow agents to commit to complex conditions, it will allow for the specification of patterns of interaction representing norms, conventions and rules, and it will provide the tools necessary to the agents to verify their ability to fulfill the engagements (along the line of control and of safety).

Reconciling these two aspects, which are *strictly connected to one another* [5], opens the way to interesting considerations. In particular, a set of constraints restricts the set of commitments that can be taken by the agents to those that can be considered legal. For example, before getting on a train a person has to punch the ticket. Only after, he/she is allowed to travel on the train. However, think to a person that commits to travel to his/her destination first and, once he/she reached it, to punch the ticket. In this situation, it is impossible for the person to fulfill his/her commitment without violating the norm. More generally, in order to propose a unified framework some important questions are to be answered. For example, given a set of norms expressed in terms of patterns of interaction, how can one establish which commitments are compliant and which are incompatible? If norms change, how do these changes affect the set of commitments? Moreover, how can one monitor the interaction of the agents and discover violations? In this respect, a solutions could be to lean on e-institutions. In [3] an initial proposal is described, where the idea is to define specific artifacts [21] able to detect violations. The kinds of reasoning that can be performed in this way are many. For example, it is possible not only to detect a violation, but also to classify different violations according to how relevant they are or how costly would be to repair from the damage caused by the violation.

The modularity of our proposal suits well the needs of the dynamic specification of protocols, along the line of [2]. Artikis’ proposal is to define a set of *meta-actions* that can be performed by the agents at run-time, and that can change the set of rules that define the protocol. During this phase, the interaction is suspended. It will be resumed once the definition of the new rules is finished. In our proposal, it is possible to define a set of meta-actions whose effects are to change the set of constraints representing the norms that must be respected in the MAS. By performing those actions, agents would be able to change at run-time and dynamically, i.e. without suspending the interaction, the set of rules. Also this extension opens the way to some important question. For instance, who and how is allowed to change the rules? As part of the future work we will investigate also these aspects.

Finally, our proposal can be applied also to business process representations, and in particular to those situations in which a sequential representation is not adequate, due to the many alternative executions. In these contexts the high number of possible sequences suggests that a declarative representation, based on rules or constraints, is preferable with respect to procedural approaches. We plan to investigate more deeply these aspects along the line of [27], where a business process is described in terms of the commitments of the actors, that are involved in the process. One advantage of adopting declarative specifications and a modular representation of the constitutive (actions) and the regulative (constraints) part, is a gain of time and money in the operation of update of the business process due, for example, to norms changes.

5 Acknowledgments

We would like to thank Viviana Patti, Munindar P. Singh, and Amit K. Chopra for the helpful discussions and the work done together. We would like to thank also the reviewers for the helpful suggestions.

References

1. Alberti, M., Chesani, F., Daolio, D., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Specification and Verification of Agent Interaction Protocols in a Logic-based System. Scalable Computing: Practice and Experience. Spec. Iss. on Foundational Underpinnings for Pragmatic Agent-based Systems 8(1), 1–13 (2007)
2. Artikis, A.: Formalising Dynamic Protocols for Open Agent Systems. In: Proc. of the 12th International Conference on AIL. pp. 68–77 (2009)
3. Baldoni, M., Baroglio, C., Bergenti, F., Boccalatte, A., Marengo, E., Martelli, M., Mascardi, V., Padovani, L., Patti, V., Ricci, A., Rossi, G., Santi, A.: MERCURIO: An Iteraction-oriented Framework for Designing, Verifying and Programming Multi-Agent Systems. In: Proc. of COIN@MALLOW. pp. 66–83 (2010)
4. Baldoni, M., Baroglio, C., Marengo, E.: Behavior-Oriented Commitment-Based Protocols. In: Proc. of the European Conf. on Art. Int. (ECAI). pp. 137–142 (2010)
5. Baldoni, M., Baroglio, C., Marengo, E.: Commitment-based Protocols with Behavioral Rules and Correctness Properties of MAS. In: Proc. of Workshop DALT. pp. 66–83 (2010)
6. Baldoni, M., Baroglio, C., Marengo, E.: Constraints among Commitments: Regulative Specification of Interaction Protocols. In: Proc. of Int. Workshop on Agent Communication. pp. 2–18 (2010)
7. Baldoni, M., Baroglio, C., Marengo, E., Patti, V.: Constitutive and Regulative Specifications of Commitment Protocols: a Decoupled Approach. ACM Transactions on Intelligent Systems and Technology, Special Issue on Agent Communication (2011). To appear.
8. Baldoni, M., Baroglio, C., Marengo, E., Patti, V., Capuzzimati, F.: Learn the rules so you know how to break them properly. In: Proc. of WOA 2011 (2011)

9. Chesani, F., Mello, P., Montali, M., Torroni, P.: Verifying A-Priori the Composition of Declarative Specified Services. In: Proc. of MALLOW. CEUR Workshop Proceedings, vol. 494. CEUR-WS.org, Turin, Italy (September 2009)
10. Chopra, A.K., Artikis, A., Bentahar, J., Colombetti, M., Dignum, F., Fornara, N., Jones, A.J.I., Singh, M.P., Yolum, P.: Research Directions in Agent Communication. ACM Trans. on Int. Sys. and Tech., Spec. Iss. on Agent Communication (2011). To appear.
11. Chopra, A.K., Singh, M.P.: Contextualizing Commitment Protocol. In: Proc. of AAMAS 2006. pp. 1345–1352. ACM (2006)
12. Chopra, A., Singh, M.P.: Constitutive interoperability. In: Proc. of AAMAS'08. pp. 797–804 (2008)
13. Conte, R., Castelfranchi, C., Dignum, F.: Autonomous Norm Acceptance. In: ATAL. Lecture Notes in Computer Science, vol. 1555, pp. 99–112. Springer (1998)
14. Fornara, N., Colombetti, M.: A Commitment-Based Approach To Agent Communication. Applied Artificial Intelligence 18(9-10), 853–866 (2004)
15. Gaudou, B., Herzig, A., Longin, D.: A Logical Framework for Grounding-based Dialogue Analysis. Electr. Notes Theor. Comput. Sci. 157(4), 117–137 (2006)
16. Giordano, L., Martelli, A., Schwind, C.: Specifying and verifying interaction protocols in a temporal action logic. J. Applied Logic 5(2), 214–234 (2007)
17. Mallya, A.U., Singh, M.P.: Introducing Preferences into Commitment Protocols. In: Agent Communication II, International Workshops on Agent Communication (AC 2005 and AC 2006). LNCS, vol. 3859, pp. 136–149. Springer, Utrecht, Netherlands (2006)
18. Marengo, E., Baldoni, M., Baroglio, C., Chopra, A.K., Patti, V., Singh, M.P.: Commitments with Regulations: Reasoning about Safety and Control in REGULA. In: Proc. of AAMAS. pp. 843–850. IFAAMAS (2011)
19. McBurney, P., Parsons, S.: Games That Agents Play: A Formal Framework for Dialogues between Autonomous Agents. Journal of Logic, Language and Information 11(3), 315–334 (2002)
20. Montali, M.: Specification and Verification of Declarative Open Interaction Models: a Logic-Based Approach, LNBIP, vol. 56. Springer, Heidelberg (2010)
21. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems 17(3), 432–456 (2008)
22. Pesic, M., van der Aalst, W.M.P.: A Declarative Approach for Flexible Business Processes Management. In: Proc. of BPM 2006 International Workshops. LNCS, vol. 4103, pp. 169–180. Springer (2006)
23. Searle, J.: The construction of social reality. Free Press, New York (1995)
24. Singh, M.P.: An Ontology for Commitments in Multiagent Systems. Artif. Intell. Law 7(1), 97–113 (1999)
25. Singh, M.P.: Distributed Enactment of Multiagent Workflows: Temporal Logic for Web Service Composition. In: Proc. of AAMAS. pp. 907–914. ACM (2003)
26. Singh, M.P.: Agent Communication Languages: Rethinking the Principles. In: Communication in Multiagent Systems. LNCS, vol. 2650. Springer (2003)
27. Telang, P.R., Singh, M.P.: Business Modeling via Commitments. In: SOCASE. LNCS, vol. 5907, pp. 111–125. Springer (2009)
28. Winikoff, M., Liu, W., Harland, J.: Enhancing commitment machines. In: DALT 2004. LNCS, vol. 3476, pp. 198–220. Springer (2005)
29. Yolum, P., Singh, M.P.: Commitment Machines. In: Intelligent Agents VIII, ATAL 2001. LNCS, vol. 2333, pp. 235–247. Springer (2002)

Rule-Based Semantic Sensing

Przemyslaw Woznowski and Alun Preece

Cardiff University, School of Computer Science, 5 The Parade, Cardiff, UK
{p.r.woznowski,a.d.preece}@cs.cf.ac.uk

Abstract. Rule-Based Systems have been in use for decades to solve a variety of problems but not in the sensor informatics domain. Rules aid the aggregation of low-level sensor readings to form a more complete picture of the real world and help to address 10 identified challenges for sensor network middleware. This paper presents the reader with an overview of a system architecture and a pilot application to demonstrate the usefulness of a system integrating rules with sensor middleware.

Keywords: RBS, SNM, RFID, rules, tracking, sensors, JESS, GSN.

1 Introduction & Motivation

A single sensor provides only partial information on the actual physical condition measured, e.g. an acoustic sensor only records audio signals. For an application to reason over sensor data, raw sensor readings have to be captured and often aggregated to form a more complete picture of the real-world condition measured. Sensor Network Middleware (SNM) aids this process. As defined in [1], "The main purpose of middleware for sensor networks is to support the development, maintenance, deployment, and execution of sensing-based applications". However, existing SNMs don't give the user – who can be an expert in some area that is not computer science – an opportunity to easily specify data aggregation logic themselves.

It is hypothesised that rules help to address this problem and can greatly improve the SNM. Moreover, such an approach to sensor networks addresses many of the 10 challenges for SNM, listed in [2], in the following way:

Data Fusion - Rules fuse simple facts to infer higher-level facts about the real world.
Application Knowledge - Expert's knowledge encoded into an automated system.
Adaptability - Applicable to any domain, non-programmers can write rules.
Abstraction Support - Each fact is an interpretation of data. How the data is interpreted is determined by an expert via rules.
QoS Support - Multiple combinations of rules and facts can often answer the same query. Solution can be explained by retracing the reasoning.

The remaining challenges: Network Heterogeneity, Dynamic Topology, Resource Constraints, Security and Scalability – need to be met by SNM. Additional benefits come from well-known advantages of using RBS systems: reproducibility, permanence, consistency, timeliness, efficiency, breadth, completeness, documentation, etc. – as identified in [3]. Finally, representing sensor data in the form of facts adds semantics. We propose the Rule-Based Semantic Sensor System (RBS3) which employs a Rule-Based System (RBS) on top of existing off-the-shelf SNM. The pilot application described in Section 3 was implemented to test the hypothesis that rules help to address the 10 Challenges and ease the development, maintenance, execution and extensibility of sensing-based applications.

2 Proposed System Architecture

The proposed system architecture in Figure 1 consists of four layers. The SNM layer serves as a bridge between physical sensors and the layer above it. It abstracts away the network heterogeneity by modelling hardware entities, and the output they produce, in software. The Interface layer is responsible for injecting sensor data, coming from the layer below it, into the Reasoning Engine layer in the form of facts. Its main function is to translate the SNMs output into facts, defined in terms of a semantic data model (for which we currently use RDF Schema for simplicity, although details of this are not included in this paper due to lack of space). The Reasoning Engine layer is the heart of the system. It continuously reasons over incoming facts and those already in the Working Memory (WM) to produce new, more complex facts. The more complex the facts, the higher the semantic enrichment and therefore more detailed picture of the real world. The Application layer bridges the user's interface with the Reasoning Engine. It exposes facts and queries, which persist in the KB, to the application. Moreover, it takes user's queries, pushes them to the layer below and returns the results in the format easily consumable by the application.

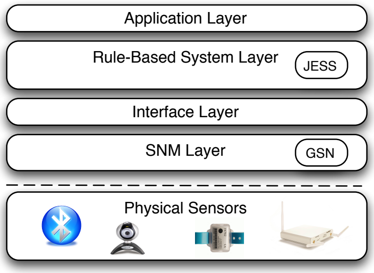

Fig. 1. System Architecture

The system architecture in Figure 1 is implemented in our RBS3 system as follows. The SNM layer currently consists of Global Sensor Networks (GSN) middleware, which serves XML data in response to queries. GSN (GNU GPL license) is a SNM, which deals with sensor network heterogeneity via use of Virtual Sensor (VS) abstraction. Any type of sensors, whether hardware or software, is represented by a single VS XML file, which specifies its inputs and output structure [4]. However, other SNMs such as: Pachube, ITA Sensor Fabric or SWE compliant middleware could replace GSN. The Interface Layer parses the XML data to JavaBeans, which are then injected into the Reasoning Engine (Jess) in the form of facts. Alternatively SweetRules, which is much more compact and offers extra features, could replace Jess, as both rule engines accept rules in CLIPS format. Queries, their arguments, and return parameters are available through the Application Layer, which serves data in JSON(JavaScript Object Notation) format, because it is a lightweight data-interchange format, which is easy for humans to read/write and easy for machines to parse.

3 Pilot Application

The aim of this application is to provide information on people's indoor locations, their history of visited locations, and information on walking speed between the locations - later referred in this paper as "corridor tests". The basic assumption for the system to work is that the tracked person wears either the RFID tag or any Bluetooth(BT) enabled device pre-registered with the mobileDevStore VS. Also corridor entities need to be defined in the corridorStore VS in order for the system to log corridor tests. This is part of a larger project looking at people's recovery from physical injury.

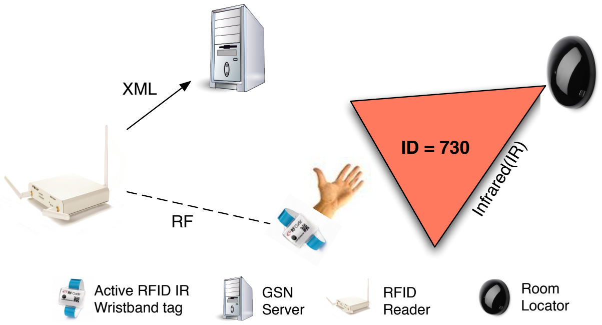

Fig. 2. Localisation with the Room Locator

The room-level localisation of active RFID tags is possible via use of the Room Locators, which broadcast a pre-set location code via IR (Figure 2). The active RFID wristband tags are IR enabled, therefore report the IR location code to the RFID reader. For this a direct line-of-sight between tag and the Room Locator is required. In the experiment, the network consisted of 1 laptop, 1 RFID reader, 2 Room Locators and 2 standard desktop PCs with Bluetooth, placed in two rooms, both running an instance of the GSN server. To clarify, the software/hardware components used in the experiment have the following functions:

RFID Active Tag: Every 2 seconds broadcasts it's unique ID, IR location code, motion status, etc. to the RFID Reader.

Bluetooth: Alternative source of information on user's location.
GSN: Connects to sensors, logs their readings and exposes them via a web server. Also serves as a source of information for static data.
RFID Reader: Receives active RFID tags' signals.
Room Locator: Transmits an IR pulse pattern containing a unique 3-digit location code to enable room-level accuracy localisation.
mobileDevStore VS: Lookup service. Stores name to RFID/BT address mappings.
corridorStore VS: Lookup service. Stores corridor entities (endA, endB, length).
btReader VS: Logs device's discovery time, BT address and reader's location.
rfidReader VS: Logs tag's discovery time, ID and reader's location.

3.1 Facts

Shadow fact, as described in [5], is "an unordered fact whose slots correspond to the properties of a JavaBean". Three shadow fact templates are defined in the Knowledge Base (KB): `MobileTrace, Person` and `Corridor`. They allow for quick insertion of JavaBean objects into the Working Memory (WM) and they directly represent GSN Virtual Sensor's outputs.

```
(deftemplate MobileTrace (declare (from-class javaBeans.MobileTrace)))
;Java class members/slots: location, address, time.
(deftemplate Person (declare (from-class javaBeans.Person)))
;Java class members/slots: name, deviceAddress.
(deftemplate Corridor (declare (from-class javaBeans.Corridor)))
;Java class members/slots: enda, endb, length.
```

Apart from shadow facts described above, the following set of unordered facts exists in the KB. All these facts originate from rules defined in the KB. To summarise, in this implementation, shadow facts (capitalised) represent sensor readings and unordered facts are used internally in Jess to represent fused sensor data. These fact templates are the semantic interface and we do have the RDF Schema for them, however, this is not included due to lack of space.

```
(deftemplate is-seen-at (slot name)(slot location)(slot time))
(deftemplate is-currently-at (slot name)(slot location)(slot tStart)
 (slot tFinish))
(deftemplate was-at(slot name)(slot location)(slot tStart)(slot tFinish))
(deftemplate was-tracked (slot name) (slot endA) (slot endB)
 (slot tStart)(slot tFinish)(slot distance)(slot tTaken)(slot velocity))
```

3.2 Rules

The set of rules defined in the KB, allows the system to infer four types of observations from sensor and static data: `is-seen-at, is-currently-at, was-at` and `was-tracked`. First rule, `seen_at`, simply aggregates `Person` and `MobileTrace` facts to assert `is-seen-at` to the WM. It also retracts all the `MobileTraces` that are successfully fused with `Person` facts.

```
(defrule seen_at
 (Person (deviceAddress ?address)(name ?name))
 ?mob <- (MobileTrace (location ?loc)(time ?time)(address ?address))
 =>  (retract ?mob)
     (assert (is-seen-at(name ?name)(location ?loc)(time ?time))))
```

The next rule, was_at, asserts two facts to the WM: was-at and is-currently-at. The latter contains information about a person's current location; therefore whenever the same person is seen at different location, the is-currently-at fact becomes was-at and a new is-currently-at fact is added. This time both facts, which are used to infer new information (is-seen-at and is-currently-at) are retracted from the WM, as at any point in time there should only exist one of each of these facts, simply because some person can only be seen at one location at any time. However, was_at will never fire unless the initial is-currently-at fact is inserted as is-currently-at facts are only produced by this rule. Therefore, a dummy fact is defined for each person tracked by the system, e.g. for Pete we have (assert(is-currently-at(name ''Pete'')(location ''dummyLoc'')(tStart 0)(tFinish 0))).

```
(defrule was_at
 ?c <- (is-currently-at(name ?n)(location ?l1)(tStart ?tS)(tFinish ?tF))
 ?seen <- (is-seen-at (name ?n)(location ?l2)(time ?t))
 =>(retract ?c ?seen)
   (assert(was-at(name ?n)(location ?l1)(tStart ?tS)(tFinish ?tF)))
   (assert(is-currently-at(name ?n)(location ?l2)(tStart ?t)(tFinish ?t))))
```

As opposed to was_at, the update_current_loc rule deals with the situation when the location reported by is-seen-at is the same: it simply updates the tFinish of the is-currently-at fact.

```
(defrule update_current_loc
 ?c <- (is-currently-at (name ?n)(location ?loc)(tStart ?tS)(tFinish ?tF))
 ?seen <- (is-seen-at (name ?n)(location ?loc)(time ?time))
 (test(< ?tF ?time))
 => (retract ?seen)(modify ?c (tFinish ?time)))
```

The three rules discussed above can already provide information on a subject's current location and history of visited locations. If a human expert was to analyse this data, they could easily answer questions on where the person currently is/was at any point in time. Additionally, it wouldn't be a problem to tell how much time it took somebody to transfer from one location to another, as this can be worked out from was-at facts. Find_corridor_events does exactly this, but in a slightly different way. Instead of analysing consecutive was-at facts it works with is-currently-at and was-at facts, whose locations are defined as ends of some Corridor in the KB. However, was-tracked fact is asserted if the subject travels from A to B and not B to A. This logic is there in purpose, as one may be interested in journeys in only one direction.

```
(defrule find_corridor_events
 (was-at (name ?name)(location ?loc1)(tStart ?t1S)(tFinish ?t1F))
 (is-currently-at (name ?name)(location ?loc2)(tStart ?t2S)(tFinish ?2tF))
 (Corridor (enda ?loc1)(endb ?loc2)(length ?length))
 => (bind ?tTaken (- ?t2S ?t1F))
    (assert (was-tracked (name ?name)(endA ?loc1)(endB ?loc2)(tStart ?t1F)
    (tFinish ?t2S)(distance ?length)(tTaken ?tTaken)
    (velocity (/ ?length (/ ?tTaken 1000)))))
```

It seemed to be enough to define only four rules in the KB. However, test results have revealed the missing logic. Assuming a scenario where somebody visits locations in the following order: 730, 000, 740, 000, 730, 000, 740 and the corridor is defined as (Corridor (enda 730)(endb 740)(length 20)) any person would know that there are two journeys of interest: 2 x “730 through 000 to 740”. However, the system inferred one additional fact: “730 through 000, 740, 000, 730, 000 to 740”. Since it does not make sense to consider cyclic journeys, we also have rules to retract these from the WM. Obviously the `find_corridor_events` rule could be replaced with a query, which looks for `was-at` and `Corridor` facts, however the general idea is to infer new, more complex facts from existing lower-level facts, rather that to come up with a clever query which can provide information on one’s journeys. By inserting new and often more complex facts, the KB is populated with more data what allows for defining new rules that can simply look at existing facts and infer even more complex ones. `Find_corridor_events` is an example of a rule that does not modify facts that are already in the WM but instead populates new facts, which can then be used by other rules.

3.3 Results

Three queries, that take name as the parameter, are defined in the KB: `find_journeys`, `where_is` and `location_history`. They simply look for `was-tracked`, `is-currently-at` and `was-at` facts respectively for some person. Querying the WM becomes very simple, as neither new data needs to be inferred, nor any calculations done - simply query parameter needs defining. Hence query of the form `"find_journeys Pete"` lists all the `was-tracked` facts (corridor test results) for Pete.

To validate this application some tests were carried out. The table below contains results of the corridor tests recorded by the system, contrasted with times recorded by the subject of these tests via use of an ordinary watch synchronised with system’s time. For simplicity, times represented in the table are of form HH:MM:SS and do not include milliseconds. From Table 1 it is easy to see that the system never underestimates the tTaken but is rather an overestimate of it. This behaviour was predictable due to the following two factors. Firstly, RFID tags broadcast their signal every 2 seconds (when in motion) and therefore introduce a maximum delay of 2 seconds on both ends of the corridor. Therefore, if somebody arrives at some location, this information may not be injected into the system until the next round of broadcasting, which in worst case is 2 seconds later. Secondly, in order for the tag to report it’s location it has to receive the IR signature of some location. If there is no direct line-of-sight between the tag and Room Locator, the tag reports location code 000 instead of the broadcasted location code. To account for both these factors the system could subtract the average delay time from the results returned.

Table 1. Experiment Results

Recorded by the system			Recorded by hand		
tStart	tFinish	tTaken	tStart	tFinish	tTaken
13:30:44	13:31:26	42	13:30:43	13:31:21	38
13:36:21	13:37:00	39	13:36:18	13:36:56	38
13:59:25	14:00:08	43	13:59:22	14:00:03	41
14:13:41	14:14:16	35	14:13:38	14:14:08	30

4 Related Work

Many of the popular Sensor Network Middlewares, such as GSN, ITA Sensor Fabric, Pachube or SWE-compatible products are rather low-level [4, 6–8]. They simply provide sensor data (using different models and abstractions) and do not make it very easy for the programmer to program with them. We consider these as candidates for the SNM layer rather than complete solutions that meet all the 10 challenges to a satisfactory level. Application knowledge, adaptability and abstraction support are not very well addressed by these SNMs.

To the best of our knowledge there are no systems that implement a Rule-Based System on the top of SNM. The most similar is Semantic Streams (SS) – "...a framework ...that allows users to pose declarative queries over semantic interpretations of sensor data" [9]. SS and RBS3 are both very high-level in terms of ability to query for real world facts. RBS3 adapts ideas from SS in a sense that rules have analogous function to the semantic services – both take some inputs and produce outputs as a result of data aggregation. Moreover, streams of data (in case of RBS3 - facts) are reused in both systems. However, SS uses a modified version of Prolog and connects to sensors using MSR Sense toolkit, hence lacks openness at the lower layer, and can only use sensors compatible with this toolkit – according to Microsoft [10] "MSR Sense has only been tested on TinyOS-based sensor motes, although in theory, it should work with any 802.15.4 compatible wireless sensors". Therefore SS is hard to extend with new sensors or other sensor middleware. In addition rules are coded implicitly using SS markup language another specification to learn in order to use the system. RBS3, on the other hand, defines rules explicitly in a well-known "standard" form of rule (CLIPS) and allows adding new rules at the runtime. Semantic Streams use a variant of backward chaining to find semantic services that can satisfy the query. In contrast, RBS3 implements forward-chaining mechanism and only allows the user to query the system using queries defined in the KB.

5 Conclusion & Future Work

In this paper, we have proposed a system architecture which combines rules and sensor middleware to better address 10 identified challenges for sensing systems. The proposed system architecture provides several benefits amongst which are: flexibility and extensibility. This approach also aids application development, maintenance, deployment, and execution. Other benefits come from using a Rule-Based System and they help to address half of the 10 challenges for SNM: Data Fusion, Application Knowledge, Adaptability, QoS and Abstraction Support.

The current implementation of the system only has GSN in the SNM layer. In the next version of RBS3, wrappers to interface with other popular sensor middleware, such as Pachube, ITA Sensor Fabric or SWE, will be present. The proposed system architecture makes the system extensible – if the user is constrained to use a specific type of SNM they can implement their own wrapper for it; and flexible – if the user does not want to be limited to use one SNM but wants to use sensor data from various sources. Another improvement to the system would be to modify the Interface Layer, so that RDF data serialised in JSON is parsed and injected into the Reasoning Engine, instead of XML parsed to JavaBeans – "since XML just describes grammars there is no way of recognising a semantic unit from a particular domain of interest" [11]. The Reasoning Engine would then be processing semantically rich data.

Because the system works in a forward-chaining way, only when a rule that produces certain type of facts is specified, these facts become available for queries. The next version of the system may use both: backward- and forward-chaining mechanisms to allow the user to query for data for which production rules are specified just before the query – therefore, historical data stored in DBs can participate to the query. Another, desirable enhancement to the system is at the Application layer – the system is easier to interface with if the user has a choice whether to receive data in JSON or RDF format.

Scalability is something that the entire system, once fully implemented, has to be extensively tested for in order to provide good response times and good level of reliability – the more sensors used, the more data to parse and store. However, the system as it is, is proven to work correctly and starts to reveal it's potential. As facts are injected into the system, they are not only aggregated together but also they are re-used across multiple rules. The more complex the facts are the better they re-create the real world conditions measured by sensors and can answer more complex queries.

Acknowledgements

We would like to thank Prof. Robert van Deursen, Arshi Iqbal (School of Healthcare Studies, Cardiff University) and Dr Allison Cooper (Swansea University) for providing RFID equipment and continuous help with system development and testing. Many thanks to Chris Gwilliams for proofreading this paper.

References

1. K.Römer, O.Kasten, and F.Mattern, "Middleware challenges for wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol.6, pp.59–61, Oct. 2002.
2. M.Molla and S.Ahamed, "A survey of middleware for sensor network and challenges," in *Parallel Processing Workshops, 2006. ICPP 2006 Workshops. 2006 International Conference on Parallel Processing Workshops*, p.6, 2006.
3. C.E. Brown and D.E. O'Leary, "Introduction to Artificial Intelligence and Expert Systems." available at `http://www.carfield.com.hk/document/ai/Expert+Systems.html`.
4. K.Aberer, M.Hauswirth, and A.Salehi, "A Middleware For Fast And Flexible Sensor Network Deployment," in *Proceedings of the 32nd international conference on Very large data bases ACM*, 2006.
5. E.Friedman-Hill, *Jess in Action: Java Rule-based Systems.* Greenwich, CT: Manning, 2003.
6. U.Hague, "Pachube" available at `http://www.pachube.com`.
7. J.Wright, C.Gibson, F.Bergamaschi, K.Marcus, R.Pressley, G.Verma, and G.Whipps, "A dynamic infrastructure for interconnecting disparate isr/istar assets (the ita sensor fabric)," in *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pp. 1393 –1400, 2009.
8. M.Botts, G.Percivall, C.Reed, and J.Davidson, "OGC Sensor Web Enablement: Overview and high level architecture," in *GeoSensor Networks*, vol.4540 of *Lecture Notes in Computer Science*, pp.175–190, Springer Berlin / Heidelberg, 2008.
9. K.Whitehouse, F.Zhao and J.Liu, "Semantic Streams: a Framework for Declarative Queries and Automatic Data Interpretation," *Technical Report*, 2005.
10. Microsoft Corporation, "MSR Sense" available at `http://research.microsoft.com/en-us/projects/msrsense/default.aspx`.
11. S.Decker, F.V. Harmelen, and J.Broekstra, "The Semantic Web - on the respective Roles of XML and RDF," *IEEE Internet Computing*, 2000.

Advancing Multi-Context Systems by Inconsistency Management*

Antonius Weinzierl

Institute of Information Systems
Vienna University of Technology
Favoritenstraße 9-11, A-1040 Vienna, Austria
weinzierl@kr.tuwien.ac.at

Abstract. Multi-Context Systems are an expressive formalism to model (possibly) non-monotonic information exchange between heterogeneous knowledge bases. Such information exchange, however, often comes with unforseen side-effects leading to violation of constraints, making the system inconsistent, and thus unusable. Although there are many approaches to assess and repair a single inconsistent knowledge base, the heterogeneous nature of Multi-Context Systems poses problems which have not yet been addressed in a satisfying way: How to identify and explain a inconsistency that spreads over multiple knowledge bases with different logical formalisms (e.g., logic programs and ontologies)? What are the causes of inconsistency if inference/information exchange is non-monotonic (e.g., absent information as cause)? How to deal with inconsistency if access to knowledge bases is restricted (e.g., companies exchange information, but do not allow arbitrary modifications to their knowledge bases)? Many traditional approaches solely aim for a consistent system, but automatic removal of inconsistency is not always desireable. Therefore a human operator has to be supported in finding the erroneous parts contributing to the inconsistency. In my thesis those issues will be adressed mainly from a foundational perspective, while our research project also provides algorithms and prototype implementations.

1 Introduction

Multi-Context Systems (MCSs) are an expressive formalism for (possibly) non-monotonic knowledge exchange between heterogeneous knowledge sources. These sources are called contexts and formalized as abstract 'logics'. Information flow between contexts is specified using bridge rules which look and behave similar to rules in non-monotonic logic programming (cf. [15]):

$$(k : s) \leftarrow (c_1 : p_1), \ldots, (c_j : p_j), not(c_{j+1} : p_{j+1}), \ldots, not(c_m : p_m). \tag{1}$$

Such a rule states that information s is added to context k if for $1 \leq i \leq j$ knowledge p_i is present in context c_i and for $j+1 \leq i \leq m$ knowledge p_i is absent

* Supported by the Vienna Science and Technology Fund (WWTF), grant ICT08-020.

in c_i. Following common terminology $p_1, \dots, p_m$ are called beliefs (each of their respective context) and s is the head formula of the bridge rule.

Consider a hospital where a database with patient records, a medical ontology, and an expert system shall be working together giving decision support on patient medications. The MCS framework is a good choice to realize this. Assume for patient Sue, the database knows that a) her X-Ray result indicates pneumonia, b) a certain blood marker is present, and c) she has no known allergies. The ontology imports information on X-Ray and blood tests using bridge rules

$$(C_{onto} : \mathit{xray}(Sue)) \leftarrow (C_{patients} : \mathit{labresult}(Sue, \mathit{xray})).$$
$$(C_{onto} : \mathit{marker}(Sue)) \leftarrow (C_{patients} : \mathit{labresult}(Sue, \mathit{marker})).$$

As the ontology contains the axiom $\mathit{xray} \sqcap \mathit{marker} \sqsubseteq \mathit{atyp\_pneu}$ it concludes that Sue has a atypical pneumonaia, severe kind of pneumonia. Finally, the expert system, a logic program containing rules $\mathit{give\_weak} \vee \mathit{give\_strong} : -\mathit{pneumonia}.$ and $\mathit{give\_strong} : -\mathit{atyp\_pneumonia}.$ suggests one out of two kinds of antibiotics if a patient has pneumonia. But it also respects potential allergies by the constraint $: -\mathit{give\_strong}, \mathit{not}\ \mathit{allowed\_strong}$. As Sue has atypical pneumonia, only the strong antibiotic will help, so the logic program suggests this.

Now assume that Sue is allergic to strong antibiotics, a case that actually happens in the real world. Then the expert system can give no valid suggestion as strong antibiotics have to be given, but at the same time they are forbidden to be applied. This results in the whole system having no 'model' satisfying deductions of all knowledge bases and bridge rules. We call such an MCS inconsistent. [1]

By this example, we identify the following *open problems*:

- the inconsistency above is present due to tuples in the database, terminological assertions in the ontology, logic programming rules in the expert system and, a set of bridge rules establishing the information exchange. In what terms should the inconsistency be described and is there a uniform description irrespective of the specific formalisms used in contexts? Non-monotonicity of bridge rules and contexts is an additional challenge to such a description.
- Given such a description it is very likely that multiple ways exist to restore consistency. Removing some bridge rules would make the above example consistent, but also removal of tuples describing lab results. Similarly, addition of new bridge rules could resolve the inconsistency. If multiple options exist, which is the most preferred to restore consistency? Is it possible to do this in a heterogeneous way, i.e., can the designer of an MCS use a formalism of his own choice to specify his preference? Can such preference be given only for specific parts of an MCS and preference for other parts differently expressed?
- In the above example, the inconsistency can be dealt with locally, e.g., the expert system could switch to use paracoherent semantics and the MCS becomes consistent. For MCSs with cyclic information flow, however, this might be impossible as cyclic information flow can be such that each context returns valid belief sets ("models"), but still for the overall system it does

[1] A complete formalisation of this example is available in [12].

not fit together. How far does local inconsistency management help to resolve inconsistency, e.g., for MCSs with acyclic information flow?
- Besides inconsistency, is the MCS framework so versatile as to use other kinds of rules to connect contexts, e.g., SPARQL queries for information exchange?

As research on these topics has been started two years ago, the rest of this paper will briefly present results adressing above questions. Regarding research methodologies, we built analogies from existing techniques, e.g., Reiter's diagnosis. For algorithms we resorted to reductions to computational logic and meta-reasoning transformations, e.g., preference is handled in this way. Whenever possible, our invented methods are open so that legacy systems may be integrated to achieve certain tasks, e.g., local inconsistency management.

Contributions summary:

- we developed a uniform representation of inconsistency in terms of bridge rules. This representation leads a) to the notion of inconsistency explanation which separates different sources of inconsistency and points out those bridge rules creating inconsistency and b) to the notion of diagnosis which induce all possible repairs of an inconsistent MCS. Notably, both notions coincide on the overall set of bridge rules which are marked 'faulty.
- on top of those notions, we developed a transformation-based technique to allow meta-reasoning on diagnoses of an inconsistent MCS. This allows system designers to express preferences over diagnoses in a formalism of their own choice. The same techinque also allows to filter out undesired diagnoses.
- for local inconsistency management, a generalization of the MCS formalism was developed allowing to use existing methods of inconsistency management locally for a context. The introduced notion of a context manager allows to employ arbitrary knowledge management techniques locally at a context. It is important that the employed manager can change a knowledge base in a broad range and therefore it can also do other operations like view updates, belief revision, logic program updates, etc.
- for above notions the computational complexity also was analysed.
- to show the versatility of the ideas behind MCS, we also introduced a modified notion of MCS where knowledge exchange is specified using SPARQL queries.

Finally, we also implemented prototypes for evaluating MCSs and computing diagnoses and explanations of inconsistent MCSs.

The remainder of this paper is organized as follows: In Section 2 related work is discussed while Section 3 recapitulates the formal semantics of MCS and our basic notion for inconsistency diagnosis/explanation, it is followed by a short presentation of major achievements in the last two years in Section 4. Finally, Section 5 is an outlook on future work.

2 Related Work

With the seminal work of [19] and [16] the notion of context has been introduced to artificial intelligence and logic. In these works, a context is a regarded as a certain point of view in which formal reasoning takes place. The Trento school (cf. [17, 22]) formalized and improved this understanding of context. It is notable, however, that those first frameworks consider homogeneous, monotonic logics for representing a context. With [9, 21] non-monotonicity was introduced to Multi-Context Systems. Although default negation is added to bridge rules, contexts still are homogeneous or monotonic. Only with [7] the framework has been generalized for non-monotonic bridge rules and heterogeneous contexts. This finally allows to use arbitrary knowledge sources that are connected by (possibly) non-monotonic bridge rules. Our research is based on this notion of MCSs.

To deal with inconsistency, in [5] defeasible rules are introduced as a way of establishing information exchange in MCS. Defeasible rules are similar to bridge rules, but their semantics differs as a defeasible rule does not fire if it would cause an inconsistency by doing so. Several algorithms based on preference orders (or argumentation frameworks [4]) have been proposed. Inconsistency is resolved inherently, but no deeper inconsistency analysis is possible. For our hospital example this would mean that some information simply would not be passed along, e.g., forgetting the illness of *Sue*. Most of the proposed algorithms are based on provenance, which means that context internals have to be exhibited to other contexts. A company making profit by allowing third parties to use its knowledge base, however, will not risk its business by providing such information.

Aside from MCS, other areas deal with knowledge integration and its issues. Peer-to-Peer (p2p) systems [24, 10] are similar as knowledge sources interchange pieces of information. Although the notion of a peer is very similar to a context in MCS, the essential feature of p2p systems is that peers may leave and join the system arbitrarily. Therefore research seeks to cope with inconsistency by isolating faulty contexts and simply ignore their information instead of analysing the inconsistency and aiming for a consistent system.

Information integration on the other hand deals extensively with issues like constraint violations that stem from the integration of several databases into a single one (cf. [6] for a survey on data fusion). Its main differences to MCS are that the result of data fusion is one single database which usually uses relational algebra for knowledge representation. MCSs, however, require inconsistency management for multiple, heterogeneous knowledge bases which are not restricted to a relational setting.

For many formalisms, methods of inconsistency handling have been invented, e.g., belief revision or possibilistic reasoning (e.g. [3]) for classical logic, paracoherent semantics for logic programs, etc. These methods can resolve inconsistency locally at a context (cf. Section 4), but they can not guarantee a consistent system. Also, most of those methods are only applicable to a specific formalism instead of a heterogeneous non-monotonic system.

3 MCS Preliminaries

Each context of an MCS is seen as a knowledge base built on an underlying logic. To capture different kinds of logics, this notion is general and not defined in the bottom-up style of inductive definitions for syntax and semantics. Instead, its approach is top-down, directly working with sets of well-formed formulas (wffs) and models (called belief sets). The semantics of a logic then only maps each set of wffs to a set of belief sets, i.e., the models of the wffs.

Formally, a logic $L = (\mathbf{KB}_L, \mathbf{BS}_L, \mathbf{ACC}_L)$ consists, of the following components: 1) $\mathbf{KB}_L$ is the set of well-formed knowledge bases of L where each element of $\mathbf{KB}_L$ is a set (of formulas). 2) $\mathbf{BS}_L$ is the set of possible belief sets where we assume that each element of $\mathbf{BS}_L$ is a set (i.e.,a model containing all formulas that are considered true). 3) $\mathbf{ACC}_L : \mathbf{KB}_L \to 2^{\mathbf{BS}_L}$ is a function describing the semantics of L by assigning each knowledge base a set of acceptable belief sets. This concept of a logic captures many monotonic and non-monotonic logics, e.g., classical logic, description logics, modal, default, and autoepistemic logics, circumscription, and logic programs under the answer set semantics.

A *Multi-Context System* $M = (C_1, \ldots, C_n)$ is a collection of contexts $C_i = (L_i, kb_i, br_i)$, $1 \leq i \leq n$, where $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$ is a logic, $kb_i \in \mathbf{KB}_i$ a knowledge base, and br_i is a set of bridge rules of form (1) over logics $(L_1, \ldots, L_n)$. Furthermore, for each bridge rule $r \in br_i$ its head formula s is compatible with C_i, i.e., for each $H \subseteq \{s \mid r \in br \text{ and } (i : s) \text{ is the head of } r\}$ holds $kb \cup H \in \mathbf{KB}_{L_i}$.

A belief state $S = (S_1, \ldots, S_n)$ of an MCS $M = (C_1, \ldots, C_n)$ is a belief set for every context, i.e., $S_i \in \mathbf{BS}_i$ for all $1 \leq i \leq n$. The semantics of MCS is defined in terms of equilibria, i.e., belief states that reproduce themselves under the application of bridge rules. Formally, let M be an MCS, C_i a context of M and $S = (S_1, \ldots, S_n)$ a belief state of M, then an bridge rule r of form (1) is applicable wrt. S, denoted by $S \models body(r)$, iff $p_\ell \in S_{c_\ell}$ for $1 \leq \ell \leq j$ and $p_\ell \notin S_{c_\ell}$ for $j < \ell \leq m$. Let $app_i(S) = \{hd(r) \mid r \in br_i \wedge S \models body(r)\}$ denote the heads of all applicable bridge rules of context C_i under S, then $S = (S_1, \ldots, S_n)$ is an *equilibrium* of M if and only if $S_i \in \mathbf{ACC}_i(app_i(S))$ for $1 \leq i \leq n$.

Basic Notions for Inconsistency Analysis (cf. [12]): We call an MCS M *inconsistent* iff no belief state of M is an equilibrium. To analyse and explain the inconsistency in an MCS, two notions have been developed: consistency-based diagnosis and entailment-based inconsistency explanation. Both notions use bridge rules to characterize 'faulty' information exchange. Intuitively, a diagnosis states how an inconsistent MCS can be changed to get a consistent system and an explanation shows what parts of the system create the inconsistency.

For an MCS M, br_M denotes the set of all bridge rules occuring in M, $M[R]$ denotes a modified MCS where all bridge rules of M are replaced by those of R, and $M \models \bot$ denotes that M is inconsistent. Given an MCS M, a *diagnosis* of M is a pair (D_1, D_2),$D_1, D_2 \subseteq br_M$, s.t. $M[br_M \setminus D_1 \cup heads(D_2)] \not\models \bot$. An *explanation* of M is a pair (E_1, E_2) of sets $E_1, E_2 \subseteq br_M$ of bridge rules s.t. for all (R_1, R_2) where $E_1 \subseteq R_1 \subseteq br_M$ and $R_2 \subseteq br_M \setminus E_2$, it holds that $M[R_1 \cup heads(R_2)] \models \bot$.

For a concise characterization, one usually focuses on subset-minimal diagnoses and explanations. The basic ideas behind both notions appear also in Reiter's seminal work on diagnosis [20]. Our diagnosis is similar to his notion and our explanation is similar to (minimal) inconsistent sets. For differences, we assume the source of inconsistency to be some faulty information exchange, so we only consider bridge rules, and because of the non-monotonic nature of MCSs, a bridge rule can be faulty by firing when it should not and also by not firing when it should. In classical diagnosis, only the former is relevant as monotonic logics only become inconsistent by that. The set of minimal diagnoses can also be seen as describing all minimal repairs, while the set of minimal explanations show hows inconsistency is caused in the system. The set E_2 in an explanation also shares some ideas with consistency restoring rules (cf. [2]) of logic programs.

4 Contributions: Methods of Inconsistency Management

This section presents contributions and answers the motivational questions raised in the introduction. These are the major published results of my graduate research. Note that authors are listed alphabetically for the respective publications.

Inconsistency Assessment: Having jointly developed and investigated, the basic notions for inconsistency analysis, the next step was developing methods to assess inconsistency qualitatively, i.e., filter diagnoses with undesired properties and select most preferred ones. In the spirit of MCS, we do not apply a specific formalism for preference or filters on diagnoses, but rather show how a transformation of the MCS and slight adaption of the notion of diagnosis is sufficient to achieve the desired effects in [13].

As one of the strengths of MCS is the ability to allow arbitrary formalisms for knowledge representation inside contexts, we do not want to restrict the users to a specific kind of representation of filters (or preferences). We therefore devised a meta-reasoning transformation which allows certain contexts to observe which diagnosis is applied to the MCS. The desired filter then is realized inside such an observer context (in a formalisms which is best suited for this task). So an MCS M is transformed into an MCS M_f where an additional observer context ob is added together with some additional bridge rules (details cf. [13]). As M_f contains all contexts and bridge rules of M, every diagnosis of M can also be applied to M_f. If ob detects an undesired diagnosis D', then ob simply becomes inconsistent, i.e., having no acceptable belief set. Therefore D' is no diagnosis of M_f, but all other diagnoses of M are diagnoses of M_f. This allows to compute all filtered diagnoses with the same algorithm as for computing subset-minimal diagnoses and it also allows to specify the filter in any desired formalism.

The meta-reasoning transformation also can be applied for multiple observation contexts where each observer only sees some bridge rules instead of all, thus preserving information hiding. As a similar meta-reasoning transformation can be used for comparison of diagnoses, it is possible to realize any given preference order on diagnoses and select the most preferred one. In general, however, this requires exponentially many more bridge rules in the transformed system, but for restricted classes of preference orders it is feasible.

Inconsistency management at the level of contexts: For many specific logics and knowledge formalisms, solutions to deal with inconsistency have been developed in the past, e.g., belief revision and paraconsistency for logics, paracoherent logic programming for logic programs, etc. For contexts using the underlying formalism it is desireable that MCSs also offer the same methods of inconsistency handling. Those methods, however, require to modify a knowledge base in more ways, than just the addition of formulas as bridge rules can do.

We therefore propose *managed Multi-Context Systems* (mMCS) in [8] where each context is equiped with a manager that can apply arbitrary changes to the context's knowledge base. Bridge rules in an mMCS are like those of MCS, but their head contains a unary command *op*, e.g., $\mathit{revise}(s)$, $\mathit{delete}(s)$, $\mathit{add}(s)$, to apply the resp. operation on the formula s and the knowledge base of the context.

Managed MCS are a significant generalization of MCS as management functions can be used to realize a multitude of tasks: belief revision, view updates, updates of logic programs. To us, the most interesting is to ensure that contexts have a 'model' for any input. Such contexts are called *totally coherent*. Most notably even mMCS with totally coherent contexts cannot guarantee that the overall system has an equilibrium, but they ensure that inconsistency is only caused by odd-cyclic information flow. It directly follows that any acyclic mMCS with totally coherent contexts is consistent, thus proving local inconsistency management sufficient for acyclic MCS.

Beyond bridge rules: In [23] we introduce MCS where knowledge exchange is realised using SPARQL construct-queries. This is surprisingly simple and again shows the versatility of MCS. The resulting SPARQL-MCS framework is related to the MWeb approach [1], but our treatment of variables is different.

5 Future Work

As shown above, we were able to answer several foundational questions, give a uniform representation of inconsistency in heterogeneous MCSs, an open integration of preference-based inconsistency assessment, investigating the impact of local inconsistency handling, and making the MCS formalism capable of dealing with arbitrary changes to the knowledge bases of an MCS.

To evaluate the feasibility of the developed methods, we also aim for a reference application which is currently in the making: querying of a DNA database posing questions in (almost) natural language using an ontology and answer-set programs. Intital steps towards exchanging large amounts of information (cf. [14]) also showed that more specialised algorithms are needed.

Investigations whether approximation operators of [11] for logic programs can be translated to MCSs and transferring optimisations for abductive diagnosis (e.g.,[18]) to MCSs are also open tasks.

6 Acknowledgements

I am very grateful to my advisor Thomas Eiter, the principal investigator of our research project Michael Fink, and my colleague Peter Schüller who provided guidance, and helped with many fruitful discussions. Thank you.

References

1. Analyti, A., Antoniou, G., Damasio, C.V.: MWeb: A principled framework for modular web rule bases and its semantics. ACM Trans. Comput. Logic 12(2) (2011)
2. Balduccini, M., Gelfond, M.: Logic programs with consistency-restoring rules. In: International Symposium on Logical Formalization of Commonsense Reasoning, AAAI 2003 Spring Symposium Series. pp. 9–18 (2003)
3. Benferhat, S., Lagrue, S., Yahi, S.: Bridging possibilistic conditional knowledge bases and partially ordered bases. In: JELIA. pp. 38–50 (2010)
4. Bikakis, A., Antoniou, G.: Contextual argumentation in ambient intelligence. In: LPNMR. pp. 30–43 (2009)
5. Bikakis, A., Antoniou, G., Hassapis, P.: Alternative strategies for conflict resolution in multi-context systems. In: AIAI. pp. 31–40 (2009)
6. Bleiholder, J., Naumann, F.: Data fusion. ACM Comput. Surv. 41(1), 1–41 (2008)
7. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: AAAI. pp. 385–390 (2007)
8. Brewka, G., Eiter, T., Fink, M., Weinzierl, A.: Managed multi-context systems. In: IJCAI (2011), to appear.
9. Brewka, G., Roelofsen, F., Serafini, L.: Contextual default reasoning. In: IJCAI. pp. 268–273 (2007)
10. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Inconsistency tolerance in p2p data integration: An epistemic logic approach. Inf. Syst. 33(4-5), 360–384 (2008)
11. Denecker, M., Marek, V.W., Truszczynski, M.: Ultimate approximation and its application in nonmonotonic knowledge representation systems. Inf. Comput. 192(1), 84–121 (2004)
12. Eiter, T., Fink, M., Schüller, P., Weinzierl, A.: Finding explanations of inconsistency in multi-context systems. In: KR (2010)
13. Eiter, T., Fink, M., Weinzierl, A.: Preference-based inconsistency assessment in multi-context systems. In: JELIA. pp. 143–155 (2010)
14. Fink, M., Ghionna, L., Weinzierl, A.: Relational information exchange and aggregation in multi-context systems. In: LPNMR. pp. 120–133 (2011)
15. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Comput. 9(3/4), 365–386 (1991)
16. Giunchiglia, F.: Abstract contextual reasoning (1993)
17. Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics or: How we can do without modal logics. Artif. Intell. 65(1), 29–70 (1994)
18. de Kleer, J.: Focusing on probable diagnoses. In: AAAI. pp. 842–848 (1991)
19. McCarthy, J.: Notes on formalizing context. In: IJCAI. pp. 555–562 (1993)
20. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. 32(1), 57–95 (1987)
21. Roelofsen, F., Serafini, L.: Minimal and absent information in contexts. In: IJCAI. pp. 558–563 (2005)
22. Roelofsen, F., Serafini, L.: Minimality and non-determinism in multi-context systems. In: CONTEXT. pp. 424–435 (2005)
23. Schüller, P., Weinzierl, A.: Semantic reasoning with sparql in heterogeneous multi-context systems. In: Semantic Search over the Web (2011), to appear
24. Serafini, L., Giunchiglia, F., Mylopoulos, J., Bernstein, P.: Local relational model: A logical formalization of database coordination. In: CONTEXT. pp. 286–299 (2003)

Rule-based query answering method for a knowledge base of economic crimes

Jaroslaw Bak
Institute of Control and Information Engineering,
Poznan University of Technology,
M. Sklodowskiej-Curie Sqr. 5, 60-965 Poznan, Poland
Jaroslaw.Bak@put.poznan.pl

Abstract. We present a description of the PhD thesis which aims to propose a rule-based query answering method for relational data. In this approach we use an additional knowledge which is represented as a set of rules and describes the source data at concept (ontological) level. Queries are posed in the terms of abstract level. We present two methods. The first one uses hybrid reasoning and the second one exploits only forward chaining. These two methods are demonstrated by the prototypical implementation of the system coupled with the Jess engine. Tests are performed on the knowledge base of the selected economic crimes: fraudulent disbursement and money laundering.

Keywords. Rule-based query answering, relational database access, Jess engine, economic crimes, SDL library

1 Introduction

Data stored in relational databases are described only by their schema (syntactic structure of data). Therefore, it is often difficult to pose a query at a higher level of abstraction than in a language of database relations and attributes. There is also a mismatching problem with table and column names without strictly defined semantics. A lack of a conceptual knowledge can be overcome by introducing ontologies which for evaluation purposes can be transformed into a set of rules. This kind of additional rule-based knowledge allows reasoning and query answering at an appropriate abstract level and relieves a user of using structural constructions from SQL. This kind of query evaluation is called a rule-based query answering method.

In our rule-based system we apply rules that are Horn clauses [8]. If there is conjunction of several predicates in the head, the rule can be easily transformed into Horn clauses with the Lloyd-Topor transformation [8].

We assume that only unary or binary predicates exist in our system, according to the terms that appear in OWL language [2] (since we decided to use this standard as a way to express conceptual knowledge).

Every rule consists of two parts: the left-hand-side, which is called the body, and the right-hand-side, which is called the head. Generally both parts are the sets of atoms that are interpreted conjunctively. In the body of the rule we use premises (patterns, conditions), which have to be satisfied by the appropriate atoms (facts) to allow the rule to be fired and produce conclusions from the rule's head. Next section describes the problem statement of the proposed PhD thesis. Section 3 presents current knowledge of the problem domain and existing solutions. Section 4 contains results achieved so far, the current state of the work and author's contributions. In Section 5 the concluding remarks are given.

2 Problem statement

The presented PhD thesis is trying to cope with the following research question:

How to efficiently query a relational database
at the conceptual level defined in a rule-based system?

This question is strongly connected with the following three main problems:

i. Rule-based query answering,
ii. The combination of a rule-based system and a relational database,
iii. The construction of the knowledge base (i.e. knowledge of economic crimes).

In a rule-based query answering method we assume that there exists a knowledge base which contains two parts: intensional and extensional. The intensional knowledge is represented as a set of rules and describes the source data at a conceptual (ontological) level. The extensional knowledge consists of facts that are stored in the relational database as well as facts that were derived in the reasoning process. Queries can be posed in the terms of the conceptual level. Thus, one gets an easier way to create a query than using structural constructions from SQL (Structured Query Language). The rule-based query answering method uses the reasoning process to obtain an answer for a given query. During this process facts from database are gathered and used to derive new facts according to a given set of rules. Next, the answer is constructed and presented.

In the first two problems (i, ii), we need to deal with the following questions:

1. What kind of rule-based system do we want to use?
2. How to express and represent the conceptual knowledge in the form of rules?
3. What is the language of the queries that can be evaluated by the system?
4. What kind of reasoning is involved in the rule-based query answering?
5. How to ensure the decidability of the query answering method?
6. How to combine a relational database with a rule-based system?
7. Which reasoning engine should be used for the prototypical implementation?
8. What are other potential applications of the proposed method and system?

We also assume that the rule-based query answering method will be used with the knowledge base of the selected economic crimes: fraudulent disbursement and money laundering. Particularly, we assume our system to be aimed at determining legal sanctions for crime perpetrators and to discover crime activities and roles (of particular types of owners, managers, directors and chairmen) using concepts, appropriate relations and rules.

The answers to the majority of the given questions and current achievements are presented in Section 4.

3 Overview of existing solutions

The presented problem, the rule-based query answering task [14], has many times been subject to research. Generally, there are two kinds of reasoning method applied in the rule-based query answering task. The first one is a backward chaining method, where reasoning is goal-driven. In this case our goal is the query posed to the system. This scheme of reasoning is implemented, for instance, in Prolog engine, and takes the form of the Selective Linear Definite clause resolution (SLD). In the backward reasoning technique facts are obtained only when they are needed in derivations.

On the contrary a forward chaining approach, which is data-driven, needs reasoning about all facts. In the working memory some of the inferred facts are useless and many rules are fired unnecessarily. It has a negative impact on the efficiency of the answering process. Moreover, because all facts should exist in the working memory, the scalability of reasoning task is poor due to the limited RAM memory. This drawback occurs also in the backward chaining.

The rule-based query answering task in rule-based systems, which exploits forward chaining is generally an inefficient method. The results of the OpenRuleBench initiative [1] show that efficiency of tabling Prolog and deductive database technologies surpasses the ones obtained from the corresponding pure rule-based forward chaining engines.

The most comprehensive approaches concerning optimizations of bottom-up query evaluation (in forward chaining) were given in [14, 15]. The general method relies on the transformation of a program *P* (set of rules) and a query *Q* into a new program, *magic(P ∪ Q)*, as shown in [15]. This *magic transformation* modifies each original rule by additional predicates to ensure that the rule will fire only when the values for these predicates are available. There were also other improvements and modifications of magic approach [14]. According to the work presented in [12] we also believe that the bottom-up approach has still room for improvements in order to increase the performance of the rule-based query answering task.

There exist also some works about the combination of rules (or logic programming) with relational databases. Notable are approaches presented in [18], [10] and [19] where ontology-based data access is performed with Prolog rules or Disjunctive Datalog.

The problem of applying rules in economic crimes is quite new. Most of the research work in the legal area relies on using ontologies in the field of information management and exchange [23, 24], not reasoning [16]. Other solutions, developed for instance in FFPoirot project [25, 26], concern descriptions of financial frauds, mainly the Nigerian letter fraud and fraudulent Internet investment pages. The ontologies developed in this project are not publicly available.

In our approach rules and queries are used to reflect data concerning documents and their attributes, formal hierarchy in a company, parameters of transactions, engaged people actions and their legal qualifications. To the best of our knowledge it is the first such approach in the field of economic crimes, besides the work presented in [17], which concerns cybercrimes.

4 Achievements and the current work

4.1 General assumptions

As mentioned in Section 2, the first two tasks include eight questions. For most of them, in the current state of our work, the answers are already known:

1. We wanted to use a production rule system because we need to apply our solutions in the real world applications.
2. We decided to express the conceptual knowledge with the Horn-SHIQ ontology combined with SWRL (Horn-like) [3] rules. We are aware of the restrictions that Horn-SHIQ imposes on ontology creation [21], but this fragment of OWL is sufficient for our needs.
3. Currently we assume conjunctive queries only, which are built of the terms from ontology (concepts and relations).
4. We developed two ways of applying reasoning process in the rule-based query answering task. In the first one [6] *hybrid* reasoning (forward and backward chaining) is used and in the second one only *forward* chaining and *extended* rules are executed. The second approach is still in progress.
5. We use the *Datalog Safety* [11] restriction in the rule-based query answering and *DL-safe* rules in ontology creation [22].
6. We developed the special mapping method which is presented later in this section and also in [6].
7. We decided to use the *Jess* (Java Expert System Shell) engine [4, 5], since it is one of the fastest commercial engines (with the free academic use) and it can be easily integrated with the Java language (which is the implementation language of our tool). The Jess engine also supports both forward and backward chaining.
8. We are convinced that our knowledge base of economic crimes [27, 30] would not be the only application of the defined system. Our methods are general and can be used in every application, which requires additional knowledge for query evaluation or need to offer an easier way of query creation than with the traditional SQL.

Our current results were presented in Polish and English papers [6, 27, 28, 29, 30].

4.2 Query answering with the hybrid reasoning

The approach described in [6] concerns the hybrid reasoning in the rule-based query answering task. In this work we described also the method of mapping between an ontology and a relational database. We presented our prototypical implementation of a library tool, the Semantic Data Library (SDL), which integrates the Jess engine, rules and ontology to effectively query a relational database.

In our hybrid reasoning process the backward chaining engine is responsible only for gathering data from a relational database. Data is added (asserted in Jess terminology) as triples into the working memory. The forward chaining engine can answer a query with all constraints put on variables in a given query (=, !=, <, > etc.).

The queries are constructed in Jess language in terms of ontology concepts. The mapping between the ontology classes and properties and the relational database schema is defined to fit syntactic structures and to preserve the semantics of the data.

Extensional data itself is stored in a relational database. The ontology and the mapping rules transformed into Jess language format provide the additional semantic layer to the relational database. Such an approach allows for answering queries to a relational database with a reasoning process performed in the Jess system over rules and ontology. The hybrid reasoning and query execution is supported by the SDL library. More details are given in [6].

4.3 Mapping between ontology terms and relational database

A mapping between ontology terms and relational data [6] is defined as a set of rules where each rule is of the following form:

$$SQL\ query => essential\ predicate$$

where "essential" means that the instance of the term cannot be derived from the rules. We assume that every "essential" ontology term has its appropriate SQL query and can be obtained only in a direct way, as a result of the SQL query. For example, in the following OWL hierarchy of classes *Mother is-a Woman is-a Person*, where the class *Mother* is a subclass of the class *Woman* etc., every instance of the class *Mother* is an "essential" term.

We assume that every SQL query has the following form:

$$SELECT\ [R]\ FROM\ [T]\ <WHERE>\ <C, AND, OR>$$

where:

- R denotes the result attributes (columns) – one or two according to the unary or binary terms (OWL Class, OWL DataProperty or OWL ObjectProperty),
- T stands for the tables, which are queried,
- $WHERE$ is an optional clause to specify the constraints,
- C abbreviates the constraints in the following form: <attribute, comparator, value>, for example: $Age > 21$,
- AND, OR are the optional SQL commands.

As an example, let us assume that we have a table *Person* with the following attributes: *Id, Name, Age* and *Gender*. To obtain all adult men, we would define the following SQL query: *SELECT Id FROM Person WHERE Age>21 AND Gender='Male'*.

The mapping process requires defining SQL queries for all "essential" classes and properties. Other terms can be mapped too, but this is not necessary, since instances of them can be derived in the reasoning process.

4.4 Knowledge base of economic crimes

The approaches presented in [27, 28, 29] concern construction of the knowledge base of the selected economic crimes: fraudulent disbursement and money laundering. We analysed current related works and proposed the formal model of these economic crimes. We developed the ontology, which is the result of an analysis of about 10 crime cases. This means that the ontology is crafted to a task rather than attempting to describe the whole conceivable space of concepts and relations (top ontologies). The intensional part of the knowledge base contains also SWRL rules, which are very important when we want to determine legal sanctions for crime perpetrators and to discover crime activities and roles (not only to describe a crime scheme).

The methodology consists of several steps:

1. Design of a hierarchical data representation with 'minimal' ontology, which is used to uncover a crime scheme. This means using only necessary concepts that follow in the logical order of uncovering a crime. In the first stage goods/services transfer data is analyzed with relation to 3 basic flows: money, invoices, and documents (i.e., confirming that the service or goods have been delivered). In addition, responsible or relevant people within companies are associated with particular illegal activities.
2. Provision of a framework in which the graph building process and queries are executed.
3. Relating answers to queries with crime qualifications.

This approach is limited, but provides an essential model for evidence-building of a very important class of financial crimes: among them acting to do damage to a company and money laundering. Both crimes occurred in the example *Hydra Case* which was tested with the hybrid approach and artificially generated data. The work and results are presented in [30].

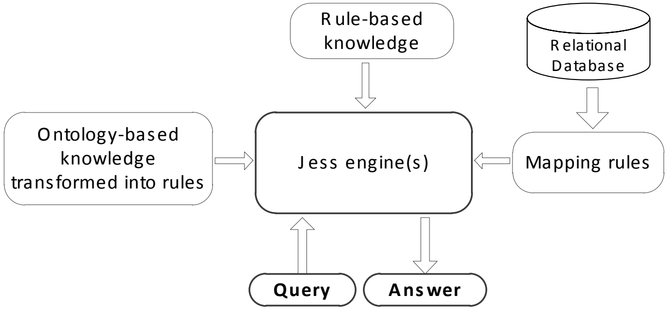

Figure 1. The architecture of the rule-based query answering system

4.5 Current work

In the current state of our work we are focused on the new rule-based query answering method which uses *extended* rules. *Extended* means that these rules are generated automatically from the basic ones for the evaluation purposes, and the modification is strongly connected with the magic transformation method. The extended rules are generated in the goal- and dependency-directed transformation. In this method we are interested in dependencies between variables appearing in predicates inside each rule.

The rule-based query answering method in this approach needs the different assumptions from the hybrid one because we use only one Jess engine to obtain relational data and answer a query. Obviously, we have to modify our query answering algorithm prepared for the hybrid system. This work is still in progress and results will be presented as soon as possible.

Figure 1 presents the architecture of our system which covers both solutions (hybrid reasoning and reasoning with extended rules).

5 Conclusions

In this paper we have outlined the content of the PhD thesis titled *Rule-based query answering method for a knowledge base of economic crimes*. Up to date we have obtained some achievements in the research, particularly related to the special crime-oriented ontological knowledge, its representation in rules of the Jess system, the connection with extensional data in a database and query answering by reasoning over the different data representations. We continue our research aiming to elaborate a new method of rules transformation, which will allow for more efficient application of rules in query answering task. We have to manage with problems presented in Section 2 and to provide a precise, clear and formal description of our solutions. We have already obtained positive results of tests performed on the prototype system but we also plan to execute queries prepared by the OpenRuleBench initiative. The comparison of our results and those obtained in a pure Jess system seems to be an adequate and objective assessment of usefulness of our work.

References

1. Liang S., Fodor P., Wan H., Kifer M., OpenRuleBench: An Analysis of the Performance of Rule Engines, Proceedings of the 18th international conference on World wide web, ACM, 2009, o. 601-610.
2. McGuinness D., van Harmelen, F.: Owl web ontology language overview. W3C Recommendation, 10 February 2004, http://www.w3.org/TR/owl-features/
3. Horrocks I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: Swrl: A semantic web rule language combining owl and ruleml. W3C Member Submission (May 21 2004), http://www.w3.org/Submission/SWRL/
4. Jess (Java Expert System Shell), http://jessrules.com/
5. Friedman-Hill, E.: Jess in Action, Manning Publications Co. (2003)
6. Bak, J., Jedrzejek, C., Falkowski, M.: Usage of the Jess engine, rules and ontology to query a relational database. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 216–230. Springer, Heidelberg (2009)
7. Forgy C., Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artificial Intelligence, 19, pp. 17-37, 1982.
8. Lloyd J.W., Foundations of logic programming (second, extended edition). Springer series in symbolic computation. Springer-Verlag, New York, 1987.
9. Horridge M., Bechhofer S. The OWL API: A Java API for Working with OWL 2 Ontologies. OWLED 2009, 6th OWL Experienced and Directions Workshop, Chantilly, Virginia, October 2009.
10. Poggi A., Lembo D., Calvanese D., De Giacomo G., Lenzerini M., Rosati R., Linking Data to Ontologies, Journal on Data Semantics, Volume 10, 2008, p. 133-173.

11. Gallaire H., Minker J. (Eds.): Logic and Data Bases, Symposium on Logic and Data Bases, Centre d'études et de recherches de Toulouse, 1977. Advances in Data Base Theory, Plenum Press, New York, 1978
12. Brass, S., Implementation Alternatives for Bottom-Up Evaluation, 26th International Conference on Logic Programming, ICLP (Technical Communications), Edinburgh 2010, 44-54.
13. Liang S., Fodor P., Wan H., Kifer M., OpenRuleBench: Detailed Report, May 29, 2009, http://projects.semwebcentral.org/docman/view.php/158/69/report.pdf.
14. F. Bry, N. Eisinger, T. Eiter, T. Furche, G. Gottlob, C. Ley, B. Linse, R. Pichler, and F. Wei. Foundations of Rule-Based Query Answering. In Proceedings of Summer School Reasoning Web 2007, Dresden, Germany (3rd–7th September 2007), volume 4634 of LNCS, pages 1–153. REWERSE, 2007.
15. Beeri C., Ramakrishnan R., On the power of magic, Journal of Logic Programming, v.10 n.3-4, p.255-299, April/May 1991.
16. Breuker, J. (2009). Dreams, awakenings and paradoxes of ontologies, invited talk presentation, 3rd Workshop on Legal Ontologies and Artificial Intelligence Techniques, http://ontobra.comp.ime.eb.br/apresentacoes/keynoteontobra-2009.ppt.
17. Bezzazi, H. (2007). Building an ontology that helps identify articles that apply to a cybercrime case, Proceedings of the Second International Conference on Software and Data Technologies, ICSOFT 2007, Barcelona, Spain, pp. 179–185.
18. Lukácsy G., Szeredi P., Scalable Web Reasoning Using Logic Programming Techniques. In A. Polleres and T. Swift, editors, Proceedings of the Third International Conference on Web Reasoning and Rule Systems, RR 2009, Chantilly, VA, USA, October 25-26, 2009, volume 5837 of Lecture Notes in Computer Science, pages 102{117. Springer, 2009.
19. U. Hustadt, B. Motik, U. Sattler. Reducing SHIQ-Description Logic to Disjunctive Datalog Programs. Proc. of the 9th International Conference on Knowledge Representation and Reasoning (KR2004), June 2004, Whistler, Canada, pp. 152-162.
20. Hustand U., Motik B., Sattler U., Data Complexity in Very Expressive Description Logics, In Proc. of the 19th Joint Int. Conf. on Artificial Intelligence (IJCAI 2005), 2005.
21. Horn-SHIQ restrictions: http://www.w3.org/2007/OWL/wiki/Tractable_Fragments#5
22. Motik B., Sattler U., and Studer R., Query answering for owl-dl with rules. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 3(1):41–60, 2005.
23. Biasiotti, M., Francesconi, E., Palmirani, M., Sartor, G. and Vitali, F. (2008). Legal Informatics and Management of Legislative Documents. Global Centre for ICT in Parliament, Working Paper No. 2, United Nations, Department of Economics and Social Affairs.
24. Casellas, N. (2008). Modelling Legal Knowledge through Ontologies. OPJK: the Ontology of Professional Judicial Knowledge, PhD thesis, Universitat Autónoma de Barcelona, Barcelona.
25. Kerremans, K. and Zhao, G. (2005). Topical ontology of VAT, Technical Report of the FFPOIROT IP project (IST-2001- 38248). Deliverable D2.3 (WP 2), STARLab VUB.
26. Zhao, G. and Leary, R. (2005). AKEM: an ontology engineering methodology in FFPoirot, Technical Report of the FFPOIROT IP project (IST-2001-38248). Deliverable D6.8 (WP 6), STARLab VUB.
27. Bak, J., Jedrzejek, C., Falkowski, M.: Application of an Ontology-Based and Rule-Based Model to Selected Economic Crimes: Fraudulent Disbursement and Money Laundering, Lecture Notes in Computer Science, 2010, Volume 6403/2010, 210-224
28. Bak J., Jedrzejek C., Application of an Ontology-based Model to a Selected Fraudulent Disbursement Economic Crime. In P. Casanovas, U. Pagallo, G. Ajani, and G. Sartor, editors, AI approaches to the complexity of legal systems, LNAI, Vol 6237, 2010
29. Jedrzejek C., Cybulka J., Bak J., Towards Ontology of Fraudulent Disbursement, Agent and Multi-Agent Systems: Technology and Applications, Proceedings of 5th International Conference, LNCS 2011, to be published.
30. Bak J., Jedrzejek C., Falkowski M., Application of the SDL Library to Reveal Legal Sanctions for Crime Perpetrators in Selected Economic Crimes: Fraudulent Disbursement and Money Laundering, In Proceedings of the 4th International RuleML-2010 Challenge, Washington, DC, USA, October, 21-23, 2010. Edited by: Palmirani M., Omair Shafiq M., Francesconi E., Vitali F., Volume 649.

Semantic-ontological combination of Business Rules and Business Processes in IT Service Management

Alexander Sellner[1], Christopher Schwarz[1], Erwin Zinser[1]

[1]FH JOANNEUM University of Applied Sciences, Department of Information Management
Alte Poststrasse 147, 8020 Graz, Austria
{Alexander.Sellner|Christopher.Schwarz|Erwin.Zinser}
@fh-joanneum.at

Abstract. IT Service Management deals with managing a broad range of items related to complex system environments. As there is both, a close connection to business interests and IT infrastructure, the application of semantic expressions which are seamlessly integrated within applications for managing ITSM environments, can help to improve transparency and profitability. This paper focuses on the challenges regarding the integration of semantics and ontologies within ITSM environments. It will describe the paradigm of relationships and inheritance within complex service trees and will present an approach of ontologically expressing them. Furthermore, the application of SBVR-based rules as executable SQL triggers will be discussed. Finally, the broad range of topics for further research, derived from the findings, will be presented.

Keywords: Semantic IT Service Management, SBVR-based SQL statements, Ontological ITSM service trees

1 Introduction

IT Service Management (ITSM) can be seen as a large and complex environment for business processes and rules with a certain potential for automation. On the one hand, ITSM puts a strong focus on providing tools for managing business topics such as outsourcing costs, licensing fees and negotiated prices within Service Level Agreements (SLAs), on the other hand the goal is to provide clearly defined processes for managing IT resources, which are, for instance defined in the ITIL framework [1].

Having a look at ITSM from large IT service providers' perspectives, there is an tremendous amount of so-called configuration items which are combined together to services, which are ultimately sold to clients. Due to changes in system environment or technological development, these services are in a constant state of change, turning the task of outsourced service provision to a rather stable price to a quite difficult challenge.

Most ITSM activities affecting existing relationships between IT service providers and their clients are triggered by so called "requests for changes" (RFCs). RFCs usually add, remove, or change existing services e.g. upsize the RAM of a server or add an SLA to a database application. Almost every ITSM item will have a certain dependency to another ITSM item and, depending on the complexity of the services, these relationships will lead to the formation of a so-called service tree. The service tree can be seen as one or more graphs, as there is no single parent node and items, such as SLAs can exist multiple times.

A well-suited approach for making such graphs human-readable is to make use of ontologies using semantic expressions [2]. By applying ontologies, it becomes possible to create graphical representations of the complex service trees, making it possible to discover all dependencies and to keep them well-managed. A further advantage becomes obvious, having a look at natural-language based semantic expressions used by ontologies. Besides adding further advances to the comprehensibility and integrity of service trees, this also creates the possibility of performing commonly understandable modifications of service relationships and therefore can help to create a mutual understanding between ITSM service providers and their customers [3].

Because of the strong business connection and great involvement of rules within SLAs, the idea of establishing a rule repository for service level definitions which are based on natural language, seems obvious. Over the last years, several standards for such definitions have been created such as RuleSpeak[1], R2ML[2] and Semantics of Business Vocabulary and Business Rules (SBVR)[3]. Most recently, it seems that there is a strong support for the SBVR standard both within the academic community as well as the industry.

This paper focuses on the challenge of displaying complex graphs of service relations to human-readable ontologies, based on semantic models. Furthermore the paper will discuss on the special topic of enriching these complex service trees with SLAs assembled by SBVR-based business rules. Another topic being discussed will be the establishment of SBVR-based business rules using DBMS triggers for execution [4].

The paper is structured as follows: Section 2 will have a closer look at service trees, discuss the paradigm of service inheritance and present ways of ontologically displaying existing service structures. Section 3 will explain the main categories of SLA rules within service trees and present samples for converting SBVR definitions to database-executable statements. Section 4 will discuss topics for further research within the presented areas.

[1] www.rulespeak.com/

[2] http://rewerse.net/

[3] http://www.omg.org/spec/SBVR/1.0/

2 Establishing ontological views of service trees based on existing semantic definitions

As already briefly discussed in the previous section, the structure of service trees in real-world business environments can be quite complex and is therefore hard to handle and maintain by people carrying out various tasks within ITSM processes. The following example will provide closer insights into the process of establishing RFCs as well as the related legacy data and the necessary procedures to involve ontological views which allow establishing a well-structured perspective on the configuration items' relationships and corresponding inherited attributes.

Within the given case, which is based on a project carried out for a large ITSM provider, the following types of configuration items (CIs) can be identified:

- Service: A service is a collection of ITSM assets with a business-relevant impact. This means that they are the elementary constructs within contracts between service providers and customers. Having a look at the function of services within the service tree, they can also be used as logical containers for putting together various "low-level" services to more sophisticated "high-level" services. Therefore, a low level service can for instance be a domain name resolution service and a high level service could be a billing application. As a matter of fact, services can exist multiple times within the service tree.
- Host: Hosts are the fundamental basis for establishing services. As a result, a service can be put together by one or more hosts. Because of a strong advance in virtualization technology over the recent years, hosts don't necessarily need to be actual hardware but multiple hosts could also depend on a host-service relationship providing the actual hardware for virtualization as well as on an additional host-service relationship for data storage. The same host can be used by multiple services and therefore can appear multiple times within the service tree.
- Service Level Agreement: SLAs include legal implications regarding the quality of services. For instance, the loss of an email service for a certain time could have a significant business impact for organizations. As a result, SLAs also handle penalty fees which need to be paid under certain circumstances. This can, for instance, be as soon as a service becomes unavailable (first failure fines and concurrent failure fines) or if a certain value for the availability of a service over a certain period is not met (availability percentage per day, month or year). SLAs follow the principle of inheritance and are therefore valid for all services or hosts which are hierarchically subordinate within the service tree. In addition, SLAs can appear multiple times within the service tree and they need to be prioritized in order to prevent discrepancies by allowing only one valid SLA per host or service.
- Maintenance Contract (MTC): MTCs contain contractually defined regulations regarding repair and renewal tasks on hardware. This can for instance be the replacement of a defect part or the regular upgrade to newer hardware parts. MTCs can be closed for services or hosts have the same characteristics as SLAs regarding inheritance. In contrast to SLAs, there can be multiple MTCs valid at the same time for one service or host which leads to accumulated maintenance liabilities.

Besides these CIs, there can be a wide range of additional CIs involved in real-world ITSM processes [5]. The above-mentioned items will serve to establish a simplified demonstration of a sample ITSM service tree which already contains a great variety of characteristics worth to undergo closer investigation. The sample service construct is demonstrated in Figure 1.

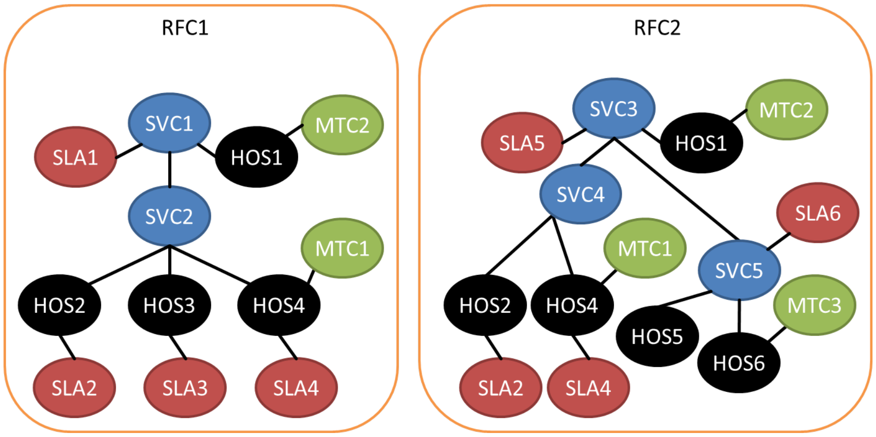

Figure 1. Sample ITSM service construct

The sample demonstrates two RFCs with their respective hierarchically subordinate services, hosts, SLAs, and MTCs. RFC1 might, for instance, be a database service hosted on a virtual machine (HOS1) with HOS2, HOS3 and HOS4 being redundant database servers, each having different SLAs. RFC2 could be a Web-based application consisting of SVC4 as database backend and SVC5 as Web servers, whereas SVC4 consists of two machines (HOS2 and HOS4) already being used by RFC1.

Considering the characteristic of inheritance for SLAs and MTCs within service trees, the example clearly illustrates why more complex service trees are almost impossible to manage and handle from a business perspective.

Having a look at the SLAs in RFC1, it clearly needs to be determined whether SLA1 or one of the respective SLAs for HOS2, HOS3 and HOS4 is valid. This is a prime example where priorities need to be assigned to SLAs in order to ensure legal correctness and optimized business benefit. Such a prioritization even needs to be put in place across boundaries of RFCs. For instance within RFC2 it also needs to be determined whether SLA4 or SLA2 and SLA5 respectively are valid for HOS2 and HOS4. Changing priority of SLA2 or SLA4 would again have an impact on the overall situation of RFC1

When focusing closer on the question of how prioritization between two SLAs needs to be done, it becomes clear that sometimes it is quite difficult to say which SLA is preferable to be applied . Assuming that within RFC1, the fines for first failure are higher in SLA1 than in SLA2 and the availability fines are higher in SLA2 than in

SLA1, it is impossible to instantly determine the most cost effective solution. It is rather imaginable that the availability of service or hosts is monitored over a certain period leading to statistical forecast regarding its behavior. Only then it would be possible to pick the optimal SLA, taking all types of fines into account.

In contrast, the challenge regarding MTCs is rather on deciding whether individual MTCs should be discharged because of redundancies or existing MTCs should be extended. As a result, the accumulated MTC benefits should be considered for every individual host or service.

3 Invoking semantics within ITSM service trees

To overcome the challenge of managing complex relations within ITSM service trees, it seems quite obvious to introduce ontologies and semantic expressions. This makes it possible to display dependencies between configuration items in a human-readable way and supports extending or rearranging the service tree [6]. An example of such an ontological view taken from the prototype of an ITSM application of a large IT service provider is given in Figure 2. The figure displays an RFC which is linked to an SLA, an MTC, a hardware asset and a high level infrastructure service which is then linked to a low level infrastructure service. The figure also reveals the concept of service instances used to express configuration items which appear multiple times within a service tree.

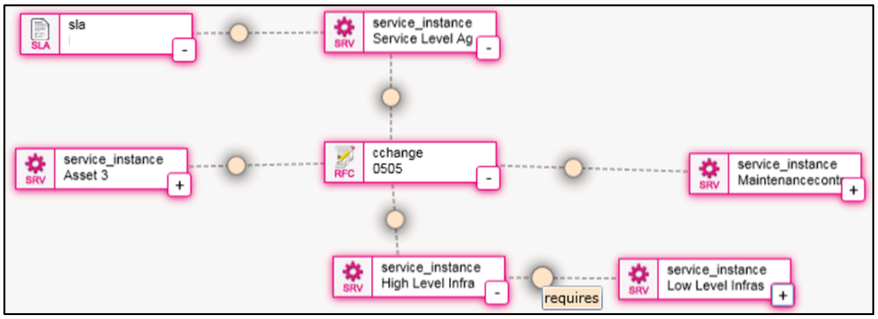

Figure 2. Ontological view of ITSM service tree

Integrating semantic expressions, adds a further level of complexity regarding the underlying data structure [7]. This can lead to quite a heavy integration effort regarding database design and operations performed to read, write or update data.

On the other hand, the advantages for integrating semantic expressions are obvious. It becomes possible to create human-understandable illustrations of service trees based on natural language. Additionally, rules can be applied on the service structure based on the existing semantic expressions. Consequently, these rules can contribute to handle the situation regarding SLA and MTC inheritance within the service tree.

4 Integrating SBVR-based rules using DBMS triggers for execution

In general, three categories of rules can be identified within ITSM service trees:

1. Rules to prevent adding new SLAs or MTCs due to inconsistencies or disadvantageous business impacts
2. Rules to analyze and improve existing SLA and MTC structure
3. Rules within SLAs themselves

There are various ways following the goal of achieving a loosely coupled execution of these SLA rules within a repository on the specified service tree. A quite feasible approach is to aim for rule execution within Database Management Systems (DBMS) using SQL triggers. This approach leverages characteristics of DBMS to establish as rule engines allowing the execution of Event-Condition-Action (ECA) rules [8].

Recent findings dealing with the conversion of SBVR definitions to SQL statements were taken as basis for developing a prototype making it possible to express the first category of rules for SLAs in structured English and carry out the respective database operations to put them in action [9,10].

Listing 1 gives an example for the conversion of the first category of rules from structured English to an ECA rule which is placed on the DBMS.

Structured English statement:

```
T:SLA
T:SVC
T:total fines
F: SLA has total fines
F:SLA is linked to SVC
NR: For an SLA that is linked to an SVC it is obligatory that
the total fines of the new SLA are less than the total fines
of the old SLA.
```

SQL expression:

```
CREATE TRIGGER "NR1" BEFORE UPDATE OF “SLA_id”
  ON "SLA-is_linked_to-SVC"
WHEN NOT
(SELECT "total fines" from "SLA" where id=new.SLA_id)<
(SELECT "total fines" from "SLA" where id=old.SLA_id)
  BEGIN
    SELECT RAISE(ABORT, ‘Requirement of NR1 not met’);
  END;
```

Listing 1. Conversion from structured English to DBMS trigger statement

The terms “SLA” and “SVC” and “total fines” are expressed as lines starting with “T”. The fact types “SLA has total fines” and “SLA is linked to SVC” are marked by a preceding “T”. Finally, the normative rule is denoted by “NR”.

Regarding the definition of fact types, the identification of table attributes takes place using conjugations of the predicate “have” and a table relationship is established by the predicate “being linked to”.

By choosing “NR” as markup, the creation of the SQL trigger is either based on a fact type involving a relationship or on a table related to a single term. The identification is based on the phrase being placed in front of the rule. If a fact type is identified as input, the first term used to describe the identity column for the SQL trigger.

Based on the fact type definition for the attribute “total fines”, the reverse order “total fines of the new SLA” forms that basis for the SELECT statement (SELECT "total fines" from "SLA" where id=new.SLA_id), whereas the definitions “new” and “old”, are explicitly required because of the targeted syntax of the SQL trigger.

Because of the fact that the trigger should not abort when the defined rule is met, the WHEN clause needs to be followed by NOT in order to display an error message in any other cases.

5 Conclusions and Future Work

The approach presented, shows the advantages and challenges regarding the application of ontologies and semantic expressions. Findings derived, led to conclusions regarding prerequisites which need to be fulfilled, especially when aiming for a close connection to underlying data structure.

Adding semantic expressions to ITSM service trees can increase presentability and manageability. On the other hand, such an integration also implies a challenge towards the underlying data, because of the paradigm of inherited SLAs and MTCs within the service tree.

The application of SBVR based business rules adds further needs which must be met by database models. In fact, the database design needs to be carried out in accordance with naming conventions established within SQL statements converted from SBVR.

Having a look at the originating statements defined in structured English, a strongly controlled natural language needs to be applied in order to enable a conversion to SQL definitions. The approach presented, focused on SBVR-based rules for preventing negative business impacts in connection with the addition of new SLAs. It needs to be investigated how these statements need to be verbalized for MTCs. It will also be necessary to look at the conversion to stored procedures, which would allow apply rules not only on a data manipulation level.

Further research will also be necessary to mathematically investigate paradigm of inheritance within ITSM service trees. Most likely, this will be closely related to graph theory and provide further insights how service trees can be optimized.

Another area worth performing research on will be investigating the application of upper level ontologies within the specific scenario. This might lead to an enterprise ontology for combining business processes and rules or to an ontology being closely focused on the area of ITSM.

References

1. Addy, R.: Effective IT Service Management: To ITIL and Beyond! Springer, Berlin (2007)
2. Guarino, N., Giaretta, P.: Ontologies and Knowledge Bases - Towards a Terminological Clarification. Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (1995)
3. Baioco, G., Costa, A., Calvi, C., Garcia, A.: IT service management and governance modeling an ITSM Configuration process: A foundational ontology approach. Integrated Network Management-Workshops (2009)
4. Vasilecas, O., Bugaite, D.: Ontology-Based Information Systems Development: The Problem of Automation of Information Processing Rules. ADVIS 2006, Proceedings (2006)
5. Ayachitula, N., Buco, M., Diao, Y., Surendra, M., Pavuluri, R., Shwartz, L., Ward, C.: IT service management automation - A hybrid methodology to integrate and orchestrate collaborative human centric and automation centric workflows. Proceedings of IEEE SCC'2007 (2007)
6. Betz, C.: Architecture and Patterns for IT Service Management, Resource Planning, and Governance: Making Shoes for the Cobbler’s Children. Morgan Kaufmann (2006)
7. Storey, V.: Understanding and Representing Relationship Semantics in Database Design. Lecture Notes in Computer Science (2001)
8. Paschke, A.: ECA-RuleML: An Approach combining ECA Rules with temporal interval-based KR Event/Action Logics and Transactional Update Logics. Internet-based Information Systems (2005)
9. Moschoyiannis, , Marinos, A., Krause, P.: Generating SQL queries from SBVR rules. RuleML'10 Proceedings of the 2010 international conference on Semantic web rules (2010)
10. Marinos, A., Moschoyiannis, S., Krause, P.: SBVR to SQL Compiler. Proceedings of the RuleML-2010 Challenge (2010)

Author Index