Logiche multimodali per ragionare sull'interazione
Multimodal Logics for Reasoning about Interaction

Matteo Baldoni, Cristina Baroglio and Viviana Patti

SOMMARIO/*ABSTRACT*

Questo articolo presenta alcune delle attività condotte nel corso degli ultimi anni dal gruppo di ricerca guidato da Alberto Martelli. In particolare verrà presentato un percorso che comprende la specifica, lo sviluppo e la verifica di protocolli di interazione. Il filo conduttore è costituito dall'uso di logiche multimodali e di formalismi dichiarativi e tecniche di ragionamento basati sulla logica computazionale.

In this paper, we report some of the activities carried on in the last years by the research group leaded by Alberto Martelli. In particular, it presents a research line that encompasses the specification, the development and the verification of interaction protocols. The leading thread is given by the use of multimodal logics and of declarative formalisms and reasoning techniques, based on computational logic.

Keywords: Interaction protocols, multimodal logics, web services, semantic web.

1 Introduction

Modal logics are widely used in Artificial Intelligence for representing *knowledge* and *beliefs* together with other attitudes like, for instance, *goals*, *intentions* and *obligations*. Moreover, modal logics are well suited for representing *dynamic* aspects in *agent systems* and, in particular, to formalize reasoning about *actions* and *time* [16, 22]. In this context, one of the main research lines of the last years concerns the specification of interaction and the forms of reasoning that can be applied to it, and gives particular attention to the verification of properties of the interaction and of the interacting agents [22]. The work that we summarize in these pages begins with the construction of a logical framework, based on a class of normal multimodal logic (called *grammar logics*). This framework has a computational counterpart, which is particularly suitable for representing the behavior of interacting and communicative agents and which lead to the implementation of the declarative programming language Dynamics in LOGic. The framework has been successfully applied to as diverse application domains as web-based adaptive tutoring, web service selection and composition, reasoning about choreographies, semantic web. In particular, it has been adopted in various national and international research projects, e.g. MASSiVE, SVP, and REWERSE.

Future directions

Declarative languages are becoming very important in the context of Semantic Web, especially in the most recent years, when the focus moved from the ontology layer to the logic layer and the need of expressing rules and apply various forms of reasoning have emerged. This interest is witnessed also by the creation of a W3C working group to define a Rule Interchange Format. The effort done for representing and reasoning about interactions, in the framework presented in this paper, finds a natural grounding in the development of negotiation or personalization policies, expressed by rules. Challenging applications can be identified also in the context of web services. Here, we are interested in applications aimed at fostering the re-use of software, task that requires abilities, e.g. flexibility, which are supplied by declarative languages and by the reasoning techniques that they allow to apply. In particular, a very promising direction of research is the study of methods and approaches to verify the interoperability of services and the conformance of a service to a choreography role.

2 The origins: A class of Normal Multimodal Logics

In [12, 3] a class of *normal multimodal logics*, called *grammar logics*, is studied. The class is characterized by a set of logical axioms of the form:

$$[p_1] \ldots [p_n]\varphi \supset [s_1] \ldots [s_m]\varphi \quad (n > 0 \;\; m \geq 0) \qquad (1)$$

called *inclusion axioms*, where the p_i's and s_j's are modalities. This class includes some well-known modal systems such as K, $K4$, $S4$ and their multimodal versions. Differently from other logics, such as those in [16], these systems can be *non-homogeneous* (i.e., every modal operator is not restricted to belong to the same system) and can contain some *interaction axioms* (i.e., modal operators are not restricted to be mutually independent).

This class of logics has been introduced by Fariñas del Cerro and Penttonen in [14] to simulate the behavior of *formal grammars*. Given a formal grammar, a modality is associated to each terminal and nonterminal symbol, while, for each production rule of the form $p_1 \cdots p_n \rightarrow s_1 \cdots s_m$, an inclusion axiom $[p_1] \ldots [p_n]\varphi \supset [s_1] \ldots [s_m]\varphi$ is defined. In [14] it is shown that testing whether a word is generated by the formal grammar is equivalent to proving a theorem in the logic, thus showing the *undecidability* of the whole class of logics.

In [12, 3] an *analytic tableau calculus* for the class of *grammar logics* is presented. The calculus is parametric w.r.t. the logics of this class. In particular, they deal with *non-homogeneous* multimodal systems with arbitrary *interaction axioms* of form (1). The calculus is a prefixed tableaux extension of those in [18, 15]. Prefixes are given an atomic name and the accessibility relationships among them are explicitly represented in a graph. The key idea is using the characterizing axioms of the logic as "*rewrite rules*" which create new paths among worlds in the counter-model construction. The works prove the *undecidability* of modal systems based on *context-sensitive* and *context-free* grammars, while *right regular* grammars are *decidable* (by using an extension of the *filtration methods* by the Fischer-Ladner closure for modal logics). In the particular case when m is 1, the axiom schema reduces to

$$\langle s_0 \rangle \varphi \subset \langle t_1 \rangle \langle t_2 \rangle \ldots \langle t_n \rangle \varphi \qquad (2)$$

and the rewriting rules for describing accessibility relations become similar to a Prolog goal-directed proof procedure. This observation has allowed the definition of the language Dynamics in LOGic. This class of logics has also been used in the study of description logics [20, 17] and extended to more general forms of interaction in [4].

3 Dynamics in LOGic: An agent Programming Language

Dynamics in LOGic [13] has been developed as a language for programming agents and is based on a logical theory for reasoning about actions and change in a modal logic programming setting. An agent's behavior is described in a non-deterministic way by giving the set of actions that it can perform. Specifically, it can be specified by a *domain description*, which includes: a) *action and precondition laws*, describing the atomic world actions the agent may perform; b) a set of *sensing axioms* describing the agent's atomic sensing actions; c) a set of *procedure axioms* describing the agent's complex behavior. Each atomic action can have preconditions to its application (that decide if the action is executable) and effects due to its application. Moreover, effects can be subject to further conditions in order to become true. For instance, the executability precondition to the action "paying by credit card" is that I hold a valid credit card. A conditional effect of this action could be "to be notified by SMS about payments". This effect will become true only if I subscribed the service (precondition to the effect).

Given this view of actions, we can think to the problem of reasoning as the act of building or of traversing a sequence of transitions between *states*. Technically speaking, a state is a set of *fluents*, i.e., properties whose truth value can change over time. Such properties encode the information that flows during the execution of a policy: for instance, if a requester communicates to pay by miles, this information will be included in the state of the provider as a fluent. In general, we cannot assume that the value of each fluent in a state is known, so we want to have both the possibility of representing that some fluents are unknown and the ability of reasoning about the execution of actions on incomplete states. To explicitly represent the unknown value of some fluents, we introduced an epistemic operator $\mathcal{B}$, to represent the beliefs an entity has about the world: $\mathcal{B}f$ means that the fluent f is known to be true, $\mathcal{B}\neg f$ means that the fluent f is known to be false. A fluent f is undefined when both $\neg\mathcal{B}f$ and $\neg\mathcal{B}\neg f$ hold. Thus each fluent in a state can have one of the three values: *true*, *false* or *unknown*.

Complex behaviors can be specified by means of *procedures*, Prolog-like clauses built upon other actions. Formally, a complex action is a collection of *inclusion axiom schemas* of the modal logic, of form (2). s_0 is a *procedure name* and the p_i's are *procedure names*, *atomic actions*, or *test actions* (f). Procedure definitions may be *recursive* and procedure clauses can be executed in a *goal-directed* way, similarly to standard logic programs.

The language allows the specification of communicative behaviors [10]. Indeed, we define the *communication kit* [19] for an agent as consisting of a predefined set of primitive speech acts the agent can perform/recognize, modeled in terms of action and preconditions laws, a set of special sensing actions for getting new information by external communications, defined by sensing axioms, and a set of interaction protocols specified by procedure axioms. Usually a communicative action modifies not only the beliefs of the executor about the world but also its beliefs about the interlocutor's mental state.

Given a domain description, we can reason about it and formalize the *temporal projection* and the *planning* problems by means of existential queries of form:

$$\langle p_1\rangle\langle p_2\rangle \dots \langle p_m\rangle Fs \quad (3)$$

where each p_k, $k = 1, \dots, m$ may be an (atomic or complex) action executed by the agent. Checking if a query of form (3) succeeds corresponds to answering the question "Is there an execution trace of the sequence $p_1, \dots, p_m$ that leads to a state where the conjunction of belief fluents *Fs* holds for our agent?". In case all the p_k's are atomic actions, it amounts to predict if the condition of interest will be true after their execution. In case complex actions are involved, the execution trace that is returned in the end is a *plan* to bring about *Fs*. The procedure definition constrains the search space. The plan can be conditional because whenever a sensing action is involved, none of the possible answers of the interlocutor can be excluded.

A goal-directed proof procedure, based on negation as failure, allows to (dis)prove queries of form (3). An interpreter for the language has been implemented in Sicstus Prolog [8]. This implementation allows the language to be used as an ordinary programming language for *executing* procedures which specify the behavior of an agent, but also for *reasoning* about them, by extracting linear or conditional plans. The plan extraction process of the interpreter is a straightforward implementation of the proof procedure contained in the theoretical specification of the language.

3.1 Web-based Adaptive Tutoring

DyLOG has been used to implement an adaptive tutoring system [11] with a multi-agent architecture, that can produce *personalized study* plans and that can validate study plans built by a user. A key feature that allows the tutoring system agents to *adapt to users* is their ability to tackle mental attitudes, such as beliefs and intentions. The agent can adopt the user's learning goal and find a way for achieving it, which fits the specific student's interests and takes into account his/her current knowledge. A natural evolution of this work opened the way to the activity carried on in the REWERSE network of excellence [2, 9, 7].

4 Reasoning about WS Composition and Choreographies

In the last years distributed applications over the World-Wide Web have obtained wide popularity and uniform mechanisms have been developed for handling computing problems which involve a large number of heterogeneous components, that are physically distributed and that interoperate. These developments have begun to coalesce around the web service paradigm, where a service can be seen as a component available over the web [1]. Each service has an interface that is accessible through standard protocols and that describes the interaction capabilities of the service.

The work presented in [10] faces the problem of automatic selection and composition of web services, discussing the advantages that derive from the inclusion, in a web service declarative description, of the high-level interaction protocol, that is used by the service for interacting with its partners, allowing a rational inspection of it. Communication can, in fact, be considered as the behavior resulting from the application of a special kind of actions: communication actions. The reasoning problem that this proposal faces can intuitively be described as looking for an answer to the question "Is it possible to make a deal with this service respecting the user's goals?". Given a logic-based representation of the service policies and a representation of the customer's needs as abstract goals, expressed by a logic formula, logic programming reasoning techniques are used for understanding if the constraints of the customer fit in with the policy of the service.

In this issue it is possible to distinguish three necessary components: (*i*) web services capabilities must be represented according to some declarative formalism with a well-defined semantics, as also observed by van der Aalst et al. [21]; (*ii*) automated tools for reasoning about such a description and performing tasks of interest must be developed; (*iii*) in order to gain flexibility in fulfilling the user's request, reasoning tools should represent such requests as *abstract goals*.

Our proposal is set in the Semantic Web field of research and inherits from research in the field of multi-agent systems. In particular, the declarative descriptions of services are based on the modal logic programming framework introduced in Section 3. Web services are viewed as software agents, communicating by predefined sharable interaction protocols, where the protocol-based interactions are formalized as DyLOG procedures; reasoning about actions and change techniques (planning) are used for performing the selection and composition of web services in a way that is personalized w.r.t. the user's request. Applying reasoning techniques on a declarative specification of the service interactions allows to gain flexibility in *fulfilling the user preference* in the context of a web service matchmaking process. As a quick example, consider a web service that allows buying products, alternatively paying cash or by credit card: a user might have preferences on the form of payment to enact. In order to decide whether or not buying at this shop, it is necessary to single out the specific course of interaction that allows buying cash. This form of personalization requires to reason about a description of the service interaction policy.

A declarative specification of the interaction is useful also in the process of selecting the services which will play the various roles of the given choreography, in the particular case in which a condition of interest is to be preserved (the goal for which the service is sought). In [6, 5] we show that current semantic matchmaking techniques do not guarantee goal preservation. The approach is based on an action-based representation of the operations of a service: each operation is described in terms of its preconditions and effects. Also in this work, the Dynamics in LOGic framework was used to represent service interaction policies as well as roles. This representation allow to reason for checking if it is possible to reach a goal by adopting a certain role, and if the goal is preserved after the substitution of the service capabilities to the abstract requirements specified in the role. We show that, by exploiting reasoning mechanisms and the choreography definition, it is possible to overcome the limits of the current semantic matchmaking techniques and we have proposed a variant of the *plugin match* which guarantees goal preservation.

5 Acknowledgements

The authors would like to thank all people who contributed to this line of research over the last years, in particular Alberto Martelli, Alessandro Chiarotto, Laura Giordano, Claudio Schifanella and Laura Torasso, co-authoring most of the papers.

REFERENCES

[1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services*. Springer, 2004.

[2] G. Antoniou, M. Baldoni, C. Baroglio, R. Baungartner, F. Bry, T. Eiter, N. Henze, M. Herzog, W. May, V. Patti, S. Schaffert, R. Schidlauer, and H. Tompits. Reasoning Methods for Personalization on the Semantic Web. *Ann. of Math., Comp. & Teleinf. (AMCT)*, 2(1):1–24, 2004.

[3] M. Baldoni. *Normal Multimodal Logics: Automatic Deduction and Logic Programming Extension*. PhD thesis, Università degli Studi di Torino, Italy, 1998.

[4] M. Baldoni. Normal Multimodal Logics with Interaction Axioms. In *Labelled Deduction*, *Applied Logic Series* 17, pp. 33–53. Kluwer Ac. Publisher, 2000.

[5] M. Baldoni, C. Baroglio, A. Martelli, V. Patti, and C. Schifanella. Reasoning on choreographies and capability requirements. *Int. J. of Business Process Integration and Management*, 2(4), 2007.

[6] M. Baldoni, C. Baroglio, A. Martelli, V. Patti, and C. Schifanella. Service selection by choreography-driven matching. In *Proc. of WEWST'07*, vol. 313 of *CEUR, Workshop Proc.*, pp. 1–17, 2008.

[7] M. Baldoni, C. Baroglio, I. Brunkhorst, E .Marengo, and V. Patti. Reasoning-based curriculum sequencing and validation: Integration in a service-oriented architecture. In *EC-TEL*, *LNCS* 4753, pp. 426–431. Springer, 2007.

[8] M .Baldoni, C. Baroglio, A. Chiarotto, and V. Patti. Programming goal-driven web sites using an agent logic language. In *PADL*, *LNCS* 1990, pp. 60–75. Springer, 2001.

[9] M. Baldoni, C. Baroglio, and N. Henze. Personalization for the semantic web. In *Reasoning Web*, *LNCS* 3564, pp. 173–212. Springer, 2005.

[10] M. Baldoni, C. Baroglio, A. Martelli, and V. Patti. Reasoning about interaction protocols for customizing web service selection and composition. *J. Log. Algebr. Program.*, 70(1):53–73, 2007.

[11] M. Baldoni, C. Baroglio, and V. Patti. Web-based adaptive tutoring: An approach based on logic agents and reasoning about actions. *Artif. Intell. Rev.*, 22(1):3–39, 2004.

[12] M. Baldoni, L. Giordano, and A. Martelli. A tableau for multimodal logics and some (un)decidability results. In *TABLEAUX*, *LNCS* 1397, pp. 44–59. Springer, 1998.

[13] M. Baldoni, A. Martelli, V. Patti, and L. Giordano. Programming rational agents in a modal action logic. *Ann. Math. Artif. Intell.*, 41(2-4):207–257, 2004.

[14] L. Fariñas del Cerro and M. Penttonen. Grammar Logics. *Logique et Analyse*, 121-122:123–134, 1988.

[15] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*, volume 169 of *Synthese library*. D. Reidel, Dordrecht, Holland, 1983.

[16] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artif. Intell.*, 54(2):319–379, 1992.

[17] Ian Horrocks and Ulrike Sattler. Decidability of shiq with complex role inclusion axioms. *Artif. Intell.*, 160(1-2):79–104, 2004.

[18] A. Nerode. Some Lectures on Modal Logic. In F. L. Bauer, editor, *Logic, Algebra, and Computation*, volume 79 of *NATO ASI Series*. Springer-Verlag, 1989.

[19] V. Patti. *Programming Rational Agents: a Modal Approach in a Logic Programming Setting*. PhD thesis, Università degli Studi di Torino, Italy, 2002.

[20] Ulrike Sattler. Description Logics for Ontologies. In *Proc. of ICCS 2003*, *LNAI* 2746, pap. 96–116, 2003.

[21] Wil M. P. van der Aalst, Marlon Dumas, Arthur H. M. ter Hofstede, Nick Russell, H. M. W. (Eric) Verbeek, and Petia Wohed. Life after bpel? In *EPEW/WS-FM*, *LNCS* 3670, pp. 35–50. Springer, 2005.

[22] Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2002.

6 Contacts

Matteo Baldoni (Corresponding Author)
Dipartimento di Informatica — Università degli Studi di Torino
c.so Svizzera, 185
I-10149 Torino (Italy)
Tel. +39 011 6706756
baldoni@di.unito.it

Cristina Baroglio
Dipartimento di Informatica — Università degli Studi di Torino
c.so Svizzera, 185
I-10149 Torino (Italy)
Tel. +39 011 6706703
baroglio@di.unito.it

Viviana Patti
Dipartimento di Informatica — Università degli Studi di Torino
c.so Svizzera, 185
I-10149 Torino (Italy)
Tel. +39 011 6706804
patti@di.unito.it

7 Biography

Matteo Baldoni (http://www.di.unito.it/ baldoni) is associate professor at the University of Torino since 2006. He received a Ph.D. in Computer Science from the same university. He has a background in computational logic, multi-modal and non-monotonic extensions of logic programming and reasoning about actions. Current research interests: communication protocol design and implementation, conformance and interoperability of services, personalization by reasoning in the semantic web. He is co-chair of the workshop AWESOME@MALLOW'007, and has been co-chair of the last four editions of the DALT@AAMAS international workshop. He chairs the working group "Sistemi ad Agenti e Multiagente" of the Italian Association for Artificial Intelligence.

Cristina Baroglio (http://www.di.unito.it/ baroglio) is associate professor of Computer Science at the University of Torino since 2005. She has a Ph.D. in Cognitive Sciences from the same university, and has a background in Machine Learning and Automated Reasoning. She is author of over 70 papers, co-chair of the workshop "Agents, Web Services, and Ontologies: Integrated Methodologies" (AWESOME@MALLOW'007), and member of the PC of the Reasoning web Summer School 2007 and 2008. Her research interests include: semantic web and semantic web services, adaptation based on reasoning, conformance and interoperability of services to choreographies/protocols, automatic teaching to artificial agents, formal approaches to e-learning.

Viviana Patti (http://www.di.unito.it/ patti) is a researcher associate in Computer Science at the University of Torino since 2005. She received her Ph.D. in Computer Science in 2002. She is author of more than 50 scientific papers. Her research interests include: computational logics for agent programming, reasoning enabling personalization in the semantic web, web service interoperability and conformance verification, web-based education courseware and curricula. She has been involved in the organization of several international events mainly in the fields "Personalization in the Semantic Web", "Web Information systems and Technologies" and "Agents". She has been member of the European Network of Excellence REWERSE.