Verifying web service conformance and interoperability w.r.t. a global choreography

Matteo Baldoni, Cristina Baroglio, Alberto Martelli,
Viviana Patti
Dipartimento di Informatica
Università degli Studi di Torino
C.so Svizzera, 185 — I-10149 Torino (Italy)
Email: {baldoni,baroglio,mrt,patti}@di.unito.it

***Abstract*— Global choreographies define the rules that peers should respect in their interaction, with the aim of guaranteeing interoperability. An abstract choreography can be seen as a protocol specification; it does not refer to specific peers and, especially in an open application domain, it might be necessary to retrieve a set of web services that fit in it. A crucial issue, that is raising attention, is verifying whether the business process of some peers, in particular the parts that encode the communicative behavior, will produce interactions which are conformant to the agreed protocol (legality issue). Such issue is tackled by the so called *conformance test*, which is a means for certifying the capability of interacting of the involved parts: two peers that are proved conformant to a same protocol will actually *interoperate* by producing a legal conversation. This work proposes an approach to the verification of a priori conformance of a business process to a protocol, which is based on the theory of formal languages and guarantees the interoperability of peers that are individually proved conformant.**

I. Introduction

The next Web generation promises to deliver Semantic Web Services, that can be retrieved and combined in a way that satisfies the user. It opens the way to many forms of *service oriented personalization*. Indeed web services provide a suitable infrastructure for constructing Plug&Play-like environments, where the user can select and combine the kinds of services s/he prefers. Personalization can be obtained by taking different approaches, e.g. by developing services that offer personalization functionalities, or by personalizing the way in which services are discovered, selected, invoked and composed so to meet specific user's requirements or by customizing the composition of different services offering personalization.

A prerequisite to this is the definition of an infrastructure for *semantic interoperability* of web services provided by the evolution of the Semantic Web initiatives. Indeed functionalities for performing personalization require a machine-processable knowledge layer that is not supplied by the current web. Web services should be augmented with public machine-interpretable semantic descriptions of their capabilities, such that a rational inspection of their behavior is enabled and new applications encapsulating personalization functionalities can be developed on this basis.

So far, most of the current standard technologies for web services (e.g WSDL [16], BPEL4WS [9], WS-CDL [33]) provide descriptions of the service capabilities, business process orchestration and choreography at the syntactic level. Such descriptions do not rely on *well-founded models* that make possible to define access and usage mechanisms without necessitating human intervention and to perform the analysis of the described process. But the capability of performing this analysis is fundamental to the real implementation of those sophisticate forms of flexibility and composition that one expects in the context of the personalization on web. To meet this requirement, one possibility is to focus on giving to the standard languages a formal semantics, by translation into formal models. Part of the formal methods community focussed the attention on capturing the behavior of BPEL and WS-CDL in a formal way, and many translations of such languages into models supporting analysis and verification (process algebras, petri nets, finite state machines) are currently under investigation [11], [10]. In parallel to the industrial standards, some proposals of standards for describing *Semantic Web Services* have been developed within the Semantic Web Initiative [15]. In this area we can distinguish two main approaches: IRS III [27], which is based on a knowledge oriented approach and relies on WSMO ontology [34] and OWL-S [30], which is based on an agent-oriented approach. The common goal of such proposals is augmenting web services with semantic descriptions that enable some kind of automatic service discovery and composition. Such semantic descriptions may concern services goal and capabilities as well as the possible compositions or choreographies.

Let us focus on the description of the choreography of a set of services. Recently W3C proposed the choreography language WS-CDL [33], well-characterized and distinguished from languages for business process representation, like BPEL. Choreographies aim at expressing *global interaction protocols*, i.e. rules that define the global behavior of a system of cooperating parties. The respect of these rules guarantees the interoperability of the parties (i.e. the capability of *actually* producing an interaction), and that the interactions will satisfy given requirements.

In the context of personalization, semantic descriptions of choreographies can provide a basis for enabling service discovery/composition personalized w.r.t. the user's goal or for developing semantic personalization service that can be combined or customized w.r.t to the user's requirements. This is the idea behind the approach taken in [4] (see Section II). Services are augmented with a high-level description of their *global interaction protocols*, about which agent applications can reason so as to personalize the selection and the composition of services and meet specific user's requirements.

However one key issue for enabling open and interoperable personalization functionalities is the development of formal methods for verifying if the behavior of a service respects a choreography. Such kind of verification is known in the literature as *conformance test*. The applications would be various. A choreography could be used *at design time* (a priori) for verifying if the internal processes of a new personalization service enable it to participate appropriately in the interaction with other personalization services, interaction that is specified by the choreography. At *run-time*, choreographies could be used to verify if everything is proceeding according to the current agreements. Section III reports our results concerning the proof of interoperability and conformance of services to a global description of their interaction.

II. Personalization of the interaction with web services

One of the needs that have inspired recent research [8] is the study of declarative descriptions of web services, aimed at allowing forms of automated interoperation that include, the automation of matchmaking, of execution, and of service *selection and composition*, in a way that is customized w.r.t. the *user's goals* and *needs*; indeed, a form of *personalization* [2]. In particular, selection and composition not always are to be performed on the sole basis of general properties of the services themselves, such as their category or their functional compositionality, but they should also take into account the user's intentions and purposes. As a quick example, consider a service that allows buying products, alternatively paying cash or by credit card: a user might have preferences on the form of payment to enact. This information should be taken into account when buying at an e-shop, singleing out a specific course of interaction that allows buying cash. This form of personalization can be obtained by applying *reasoning techniques* on a description of the service *process*, that has a well-defined meaning for all the parties involved. In this issue it is possible to distinguish three necessary components:

- web services capabilities must be represented according to some declarative formalism with a well-defined semantics, as also recently observed by van der Aalst [32];
- automated tools for reasoning about such a description must be developed;
- in order to gain flexibility, reasoning tools should represent such requests as *abstract goals*.

The approach that we propose in [4] inherits from the experience of the research community that studies MAS and, in particular, logic-based formalizations of interaction aspects. Indeed, communication has intensively been studied in the context of formal theories of agency [18], [17] and a great deal of attention has been devoted to the definition of standard agent communication languages (ACL), e.g. FIPA [23] and KQML [22]. Recently, most of the efforts have been devoted to the definition of formal models of interaction, based on *conversation protocols*. Protocols improve the interoperability of the various components (often separately developed) and allow the verification of compliance to the desired standards.

The basic idea is to consider a service as a software agent and the problem of composing a set of web services as the problem of making a set of software agents interact and cooperate within a MAS. This interpretation is quite natural, and shared in proposals that are closer to the agent research community or more properly set in the *Semantic Web* research field [12], [31]. Among the others, let us recall the OWL-S [30] experience. In [12] the goal of providing greater expressiveness to service description in a way that can be *reasoned about* has been pursued by exploiting agent technologies based on the *action metaphor*; in particular, at the level of the process model, a service is described in a way inspired by the agent language GOLOG and its extensions [28], [24], [29]. Reasoning techniques supported by the language are used to produce composite and customized services.

On this line, we have studied the benefits provided by a *declarative description* of the communicative behavior, in terms of personalization of the service selection and composition. A better personalization can be achieved by focussing on the abstraction of web services as entities, that communicate by following predefined, public and sharable interaction protocols and by allowing agents to reason about high level descriptions of the *interaction protocols* followed by web services. We model the interaction protocols provided by web services by a set of logic clauses,using an extension of the agent programming language DyLOG [7], [4].

Having a logic specification of the protocol, it is possible to reason about the effects of engaging specific conversations. We have proposed to use techniques for *reasoning about actions* for performing the selection and composition of web services, in a way that is customized w.r.t. the users's request. Communication can, in fact, be considered as the behavior resulting from the application of a special kind of actions: *speech acts*. The reasoning problem to face can intuitively be described as looking for a 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.

Our proposal can be considered as an approach based on the process ontology, a *white box* approach in which part of the behavior of the services is available for a rational inspection. A description of the communicative behavior by policies is definitely richer than the list of input and output, precondition and effect properties usually taken into account for the matchmaking. Actually, the approach can be considered as a *second step* in the matchmaking process, which narrows a set of already selected services and performs a *customization* of the interaction with them.

Present web service technology is quite primitive w.r.t. the framework we propose, and the realization of the picture sketched above requires an integration, in the current tools for web service representation and handling, of knowledge representation languages –in the line of those developed in the Semantic Web area– and of techniques for reasoning and dealing with communication, inspired by those studied in the area of MAS. Even if the integration is still far from being real let us describe our *vision* of the steps to be taken towards the realization. Public descriptions of interaction protocols should be mapped to public descriptions of choreographies (e.g. WS-CDL-like descriptions). A choreography defines a global view of the protocol followed by a certain service, e.g. a *cinema service*, for accomplishing the cooperative task of booking a cinema ticket. A *costumer* service, that in principle is interested to participate to the cooperation for booking a ticket for its user, *translates* such a description in the declarative language and uses reasoning techniques, supported by the language, plus its local knowledge on the user's preferences for checking whether the contract, defined by the choreography, satisfies its user. The outcome is meaningful under the following assumption: the implementation of the cinema service behavior (that could be written in an execution language like BPEL) must be *conformant* w.r.t. the choreography specification that is used. Verifying the conformance and the interoperability of web services to a global protocol definition (to be provided at the choreography level) is definitely one of the hot topics at the moment and is the issue addressed by the next section.

III. WEB SERVICE INTEROPERABILITY

According to *Agent-Oriented Software Engineering* [26], a distinction is made between the global and the individual points of view of interaction. The *global* viewpoint is captured by an *abstract protocol*, expressed by formalisms like AUML, automata or Petri Nets. The *local* viewpoint, instead, is captured by tthe agent's policy; being part of the agent's implementation, the policy is usually written in some executable language. Having these two levels of description it is possible to decide whether an agent can take a role in an interaction: this problem can be read as proving if the agent's policy *conforms* to the abstract protocol specification.

A similar need of distinguishing a global and a local view of the interaction is recently emerging also in the area of *Service Oriented Architectures*. Here a distinction is made between the *choreography* of a set of services, a global specification of the desired interaction, and the concept of *behavioral interface*, seen as the specification of the interaction from the point of view of the individual service. The recent W3C proposal of the choreography language WS-CDL [33], well-characterized and distinguished from languages for business process representation, like BPEL, is emblematic.

Taking this perspective, choreographies and agent communication protocols undoubtedly share a common purpose. In fact, they both aim at expressing the rules that define the global behavior of a system of cooperating parties. The respect of these rules guarantees the interoperability of the parties (i.e. the capability of *actually* producing an interaction), and that the interactions will satisfy given requirements.

In this context, a crucial problem is the development of formal methods for verifying if a service respects a choreography. The applications would be various. A choreography could be used *at design time* for verifying if the internal processes of a service enable it to participate appropriately in the interaction. At *run-time*, choreographies could be used to verify if everything is proceeding according to the agreements. A choreography could also be used unilaterally to detect exceptions (e.g. a message was expected but not received) or help a participant in sending messages in the right order and at the right time.

In the last years the agent community already started to face the two above mentioned kinds of conformance w.r.t. MAS [25] (e.g. see [19], [20], [5], [3] for *a priori* conformance, and [1] for *run-time* conformance). In the web service community the problem of conformance is arising only recently [13] because so far the focus has been posed on the specification of single services and on standards for their remote invocation. The new interest is emerging due to the growing need of making services, that are heterogeneous (in kind of platform or in language implementation), to interoperate. Therefore, there is a need of giving more abstract representations of the interactions that allow to perform reasoning in order to select and compose services disregarding the specific implementation details. Given our experience in the area of MASs, where the heterogeneity of the components is a fundamental characteristic, we agree with the observation by van der Aalst [32] that there is a need for a more declarative representation of the behaviour of services.

In this line, the work in [5], [3] about conformance of agent implementations w.r.t. protocol specifications has been adapted to the case of web services in [6]. In particular, in [6] we focus on testing *a priori conformance* and develop a framework based on the use of formal languages. In this framework a global interaction protocol (a choreography), is represented as a finite state automaton, whose alphabet is the set of messages exchanged among services. It specifies permitted conversations. Atomic services, that have to be composed according to the choreography, are described as finite state automata as well. Given such a representation we capture a concept of conformance that answers positively to all these questions: *is it possible to verify that a service, playing a role in a given global protocol, produces at least those conversations which guarantee interoperability with other conformant service? Will such a service always follow one of these conversations when interacting with the other parties in the context of the protocol? Will it always be able to conclude the legal conversations it is involved in?* Technically, the conformance test is based on the acceptance of both the service behavior and the global protocol by a special finite state automaton. Briefly, at every point of a conversation, we expect that a conformant policy never utters speech acts that are not expected, according to the protocol, and we also expect it to be able to handle any message that can possibly be received, once again according to the protocol. However, the policy is not obliged to foresee (at every point of conversation) an outgoing message for every alternative included in the protocol (but it must foresee at least one of them).

The interesting characteristic of this test is that it guarantees the interoperability of services that are proved conformant *individually* and *independently* from one another. By interoperability we mean the capability of an agent of actually producing a conversation when interacting with another. The conformance test has been proved *decidable* when the languages used to represent all the possible conversations w.r.t. the policy and w.r.t. the protocol are *regular*.

The application of our approach is particularly easy in case a logic-based declarative language is used to implement the policies. In logic languages indeed policies are usually expressed by Prolog-like rules, which can be easily converted in a formal language representation. In [3] we show this by means of a concrete example where the language DyLOG [7], based on computational logic, is used for implementing the agents' policies. On the side of the protocol specification languages, currently there is a great interest in using informal, graphical languages (e.g. UML-based) for specifying protocols and in the translation of such languages in formal languages [14], [21]. By this translation it is, in fact, possible to prove properties that the original representation does not allow. In this context, in [5] we have shown an easy algorithm for translating AUML sequence diagrams to finite state automata thus enabling the verification of conformance. Of course, having a declarative representation of the choreographies as well, would help the proof of these properties in the context of the web services.

References

[1] M. Alberti, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni, "Specification and verification of agent interactions using social integrity constraints," in *Proc. of the Workshop on Logic and Communication in Multi-Agent Systems, LCMAS 2003*, ser. ENTCS, W. van der Hoek, A. Lomuscio, E. de Vink, and M. Wooldridge, Eds., vol. 85(2). Eindhoven, the Netherlands: Elsevier, 2003.

[2] M. Baldoni, C. Baroglio, and N. Henze, "Personalization for the Semantic Web," in *Reasoning Web*, ser. LNCS Tutorial, vol. 3564. Springer, 2005, pp. 173–212.

[3] M. Baldoni, C. Baroglio, A. Martelli, and V. Patti, "Verification of protocol conformance and agent interoperability," in *Pre-proc. of Sixth International Workshop on Computational Logic in Multi-Agent Systems, CLIMA VI*, 2005, pp. 12–27.

[4] ——, "Reasoning about interaction protocols for customizing web service selection and composition," *Journal of Logic and Algebraic Programming, special issue on Web Services and Formal Methods*, 2006, to appear.

[5] M. Baldoni, C. Baroglio, A. Martelli, V. Patti, and C. Schifanella, "Verifying protocol conformance for logic-based communicating agents," in *Proc. of 5th Int. Workshop on Computational Logic in Multi-Agent Systems, CLIMA V*, ser. LNCS, no. 3487, 2005, pp. 192–212.

[6] ——, "Verifying the conformance of web services to global interaction protocols: a first step," in *Proc. of 2nd Int. Workshop on Web Services and Formal Methods, WS-FM 2005*, ser. LNCS, no. 3670, 2005, pp. 257–271.

[7] M. Baldoni, L. Giordano, A. Martelli, and V. Patti, "Programming Rational Agents in a Modal Action Logic," *Annals of Mathematics and Artificial Intelligence, Special issue on Logic-Based Agent Implementation*, vol. 41, no. 2–4, pp. 207–257, 2004.

[8] A. Barros, M. Dumas, and P. Oaks, "A critical overview of the web services choreography description language(ws-cdl)," *Business Process Trends*, 2005, http://www.bptrends.com.

[9] BPEL4WS, "http://www-106.ibm.com/developerworks/library/ws-bpel," 2003.

[10] M. Bravetti, L. Kloul, and G. Zavattaro, Eds., *Proc. of the 2nd International Workshop on Web Services and Formal Methods (WS-FM 2005)*, ser. LNCS, no. 3670. Springer, 2005.

[11] M. Bravetti and G. Zavattaro, Eds., *Proc. of the 1st Int. Workshop on Web Services and Formal Methods (WS-FM 2004)*. Elsevier Science Direct, 2004, vol. 105 of Electronic Notes in Theoretical Computer Science.

[12] J. Bryson, D. Martin, S. McIlraith, and L. A. Stein, "Agent-based composite services in DAML-S: The behavior-oriented design of an intelligent semantic web," 2002. [Online]. Available: citeseer.nj.nec.com/bryson02agentbased.html

[13] N. Busi, R. Gorrieri, C. Guidi, R. Lucchi, and G. Zavattaro, "Choreography and Orchestration: a synergic approach for system design," in *Proc. the 3rd Int. Conf. on Service Oriented Computing*, 2005.

[14] L. Cabac and D. Moldt, "Formal semantics for auml agent interaction protocol diagrams," in *Proc. of AOSE 2004*, 2004, pp. 47–61.

[15] T. Cabral, J. Domingue, E. Motta, T. Payne, and F. Hakimpour, "Approaches to semantic web services: An overview andcomparisons," in *proceedings of the First European Semantic Web Symposium (ESWS2004)*, Heraklion, Crete, Greece, 2004.

[16] R. Chinnici, M. Gudgin, J. J. Moreau, and S. Weerawarana, "Web Services Pescription Language (WSDL) version 1.2," 2003, working Draft.

[17] F. Dignum, Ed., *Advances in agent communication languages*, ser. LNAI, vol. 2922. Springer-Verlag, 2004.

[18] F. Dignum and M. Greaves, "Issues in agent communication," in *Issues in Agent Communication*, ser. LNCS, vol. 1916. Springer, 2000, pp. 1–16.

[19] U. Endriss, N. Maudet, F. Sadri, and F. Toni, "Protocol conformance for logic-based agents," in *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*, G. Gottlob and T. Walsh, Eds. Morgan Kaufmann Publishers, August 2003, pp. 679–684.

[20] ——, "Logic-based agent communication protocols," in *Advances in agent communication languages*, ser. LNAI, vol. 2922. Springer-Verlag, 2004, pp. 91–107, invited contribution.

[21] R. Eshuis and R. Wieringa, "Tool support for verifying UML activity diagrams," *IEEE Trans. on Software Eng.*, vol. 7, no. 30, 2004.

[22] T. Finin, Y. Labrou, and J. Mayfield, "KQML as an Agent Communication Language," in *Software Agents*, J. Bradshaw, Ed. MIT Press, 1995.

[23] FIPA, "Communicative act library specification," FIPA (Foundation for Intelligent Physical Agents), Tech. Rep., 2002.

[24] G. D. Giacomo, Y. Lesperance, and H. Levesque, "Congolog, a concurrent programming language based on the situation calculus," *Artificial Intelligence*, vol. 121, pp. 109–169, 2000.

[25] F. Guerin and J. Pitt, "Verification and Compliance Testing," in *Communication in Multiagent Systems*, ser. LNAI, M. Huget, Ed., vol. 2650. Springer, 2003, pp. 98–112.

[26] M. P. Huget and J. Koning, "Interaction Protocol Engineering," in *Communication in Multiagent Systems*, ser. LNAI, H. Huget, Ed., vol. 2650. Springer, 2003, pp. 179–193.

[27] I. III, "http://kmi.open.ac.uk/projects/irs/."

[28] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl, "GOLOG: A Logic Programming Language for Dynamic Domains," *J. of Logic Programming*, vol. 31, pp. 59–83, 1997.

[29] S. McIlraith and T. Son, "Adapting Golog for Programmin the Semantic Web," in *5th Int. Symp. on Logical Formalization of Commonsense Reasoning*, 2001, pp. 195–202.

[30] OWL-S, "http://www.daml.org/services/owl-s/1.1/," 2004.

[31] K. Sycara, "Brokering and matchmaking for coordination of agent societies: A survey," in *Coordination of Internet Agents*, A. O. et al., Ed. Springer, 2001.

[32] W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede, N. Russell, H. M. W. Verbeek, and P. Wohed, "Life after BPEL?" in *Proc. of WS-FM'05*, ser. LNCS, vol. 3670. Springer, 2005, pp. 35–50, invited speaker.

[33] WS-CDL, "http://www.w3.org/tr/2004/wd-ws-cdl-10-20041217/," 2004.

[34] WSMO, "http://www.wsmo.org/," 2005.