



Requirements-driven design of service-oriented interactions

Ayman Mahfouz and Leonor Barroca and Robin Laney and Bashar
Nuseibeh

13 May, 2010

Department of Computing
Faculty of Mathematics, Computing and Technology
The Open University

Walton Hall, Milton Keynes, MK7 6AA
United Kingdom

<http://computing.open.ac.uk>

Requirements-Driven Design of Service-Oriented Interactions

Ayman Mahfouz, Webalo Inc.

Leonor Barroca and Robin Laney, The Open University, UK.

Bashar Nuseibeh, The Open University, UK, and Lero, Ireland.

Designing service-oriented interactions requires addressing concerns of many stakeholders across enterprise boundaries. Throughout the design process, software artifacts of various levels of abstraction are analyzed and produced. To aid architects with reconciling concerns of the stakeholders as well as managing consistency between software artifacts, we describe four viewpoints for designing service-oriented interactions. The viewpoints enable a requirements-driven collaborative interaction design process.

Specifying Service-Oriented Interactions

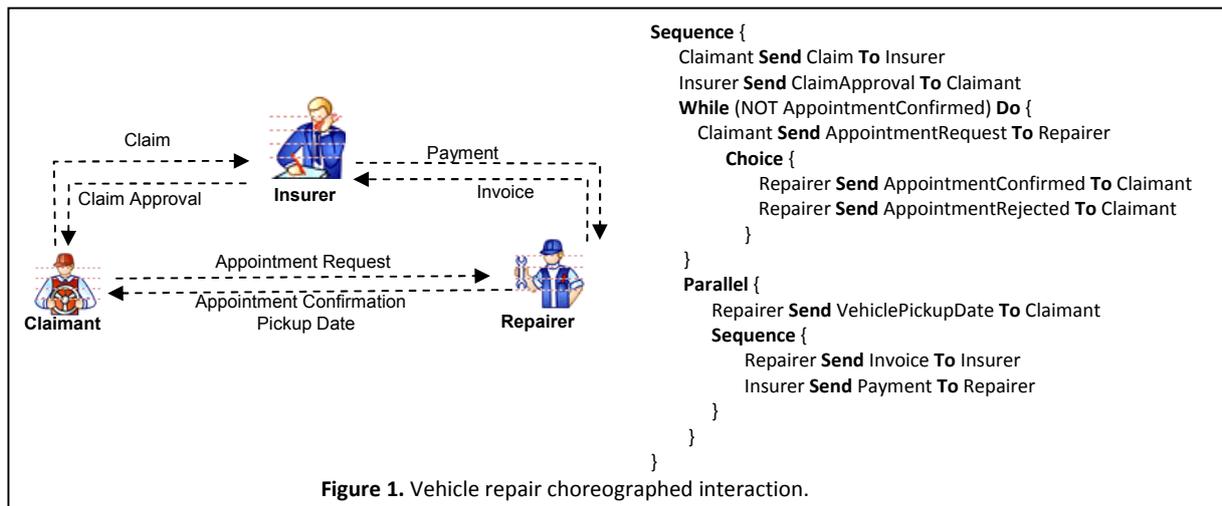
Service-Oriented Computing (SOC) is emerging as an enabler for inter-enterprise interactions. Services provide platform-independent abstractions around software systems thereby enabling interoperability between heterogeneous systems. Service interfaces describe data structures of messages exchanged between participants in an interaction using platform-independent standards such as SOAP and WSDL [1].

Real-world interactions are complex, long-running, and involve exchanging sequences of messages. Ensuring interoperability thus also requires specifying sequences of messages exchanged between interaction participants, i.e. an interaction protocol. Choreography [2] is emerging as a means for specifying interoperable inter-enterprise service interaction protocols between a set of abstract roles. At runtime, actual participants assume the choreographed roles; messaging between them must abide by the established protocol between the roles they are playing in the choreographed interaction.

Choreography specifies the observable messaging between the interacting roles from a global point of view, i.e. it abstracts away from internal business process specifics of any participant playing that role, thereby providing interoperable protocol specification. Choreography specifies messaging protocols from an observer's point of view, where the observer is a neutral party overseeing the interaction. The observer is typically a regulatory agency that handles disputes and ensures compliance of participants to the protocol.

Figure 1 shows an example messaging protocol for an interaction between three roles: Insurer, Claimant, and Repairer. The protocol specifies the claim processing and vehicle repair interaction between the three roles. We use a pseudo-language for expressing the protocol with typical constructs for specifying messaging and control flow:

- **Send...To:** specifies sending of a message of a certain type from a sender role to a recipient role.
- **Sequence:** encloses activities that must execute in order
- **Parallel:** encloses activities that can execute concurrently.
- **Choice:** represents conditional choice between mutually-exclusive options.
- **While (condition) Do:** represents repetition.



The protocol specifies that the Claimant submits a claim to the Insurer then obtains an appointment to get their vehicle repaired at the Repairer’s shop. Eventually, the Repairer notifies the Claimant that the repairs are done (by specifying a date-time when the vehicle can be picked up) and they bill the Insurer for the cost.

The global observer in this example could be the State’s Department of Insurance, as is the case in the US, which regulates the insurance business and handles disputes about non-compliance.

So far so good? Not really. Even though emerging languages such as WS-CDL [3] provide standards for specifying interoperable interaction protocols, these languages have serious limitations. Most prominently:

- They provide only a partial representation of the interaction. Physical activities that are a crucial part of the interaction are not captured. Furthermore, their relative ordering to the electronic messaging is not specified, which severely limits the utility of the protocol. For instance, we cannot specify that the Repairer is obliged to finish all repairs before billing the Insurer. Similarly, we cannot constraint vehicle hauling and inspection activities as they are simply not represented in the protocol.
- The specification of messaging focuses on operational aspects of the interaction and hence is detached from the participants’ business goals. Why do we care? Because goals and business

policies change over time and we need to adapt the messaging protocol accordingly. It is hard to prove that the messaging protocol satisfies the goals it should achieve without explicitly representing these goals and relating them to messaging activities.

These deficiencies call for a richer specification of the interaction. In particular, the need for capturing business goals and physical activities lead us to resort to specifying the interaction at the level of Models of Organizational Requirements (MOR) motivating the messaging [4]. MOR capture goals motivating participants to interact, organizational dependencies that make the interaction possible, and all activities that constitute the interaction, including physical activities.

Whereas the messaging protocol is adequate as a machine-readable specification, the high-level nature of MOR makes them useful for business-level reasoning during interaction design. Not only do the two representations address different concerns, but they also serve different purposes for different stakeholders. To ensure the interaction design process properly serves all stakeholders we need to dissect and understand these concerns.

Separation of Design Concerns

To disentangle the various concerns of interaction design, we separate them along two fundamental axes: The Stakeholder axis and the Abstraction axis.

The Stakeholder Axis

In addition to participants in the interaction being stakeholders, the global observer overseeing the interaction is another stakeholder with a distinct set of concerns.

Participants

Each participant in the interaction is a stakeholder that wishes to fulfill business needs relevant from their local point of view. The main concern of each participant is ensuring that their goals from joining the interaction are achieved. To ensure that each goal is addressed adequately, a participant needs to determine how each goal is to be achieved, i.e. which activities performed in the course of the interaction contribute towards the fulfillment of the goal.

Equally important is the need to enforce business constraints, such as data flow between business activities or pre-conditions on their execution, imposed by their internal business policies. A participant needs to ensure that their adherence to the interaction protocol does not lead to violation of any of their internal business policies, and vice versa.

A participant also needs to identify and mitigate risks, especially those arising from delegating control of business activity execution to other participants.

Global Observer

The global observer is often a regulatory agency overseeing the interaction from a neutral point of view. The observer is a stakeholder whose concerns are to facilitate the interaction and encourage participants to interact.

- To encourage participants to interact, the global stakeholder helps potential participants assess and mitigate risks involved in the interaction. The global stakeholder also needs to ensure fairness by rationalizing the balance between obligations and rights of each participant; unfair rules will deter participants from joining the interaction.
- To facilitate the interaction, the global stakeholder aims to ensure interoperability, for which specifying upfront the interacting roles and their obligations is essential. The specification of obligations becomes a standard contract for participants wishing to join the interaction and play one of the roles.

Concerns of the global stakeholder are global in that they are not specific to any participant, but rather broadly benefit all potential participants in the interaction. For instance, the objectives of the global stakeholder could be promoting trade, enabling advancement across an industry sector as a whole, or ensuring public safety.

The Abstraction Axis

We have identified two levels that are essential for designing service-oriented interactions, namely the messaging specification and MOR motivating the interaction. Whereas the messaging specification describes “how” the interaction is carried out, MOR address “why” aspects of interaction design.

Messaging Specification

Messaging specification addresses concerns about correctness of message content and messaging sequences exchanged during the interaction. Messaging protocols are the basis for ensuring that runtime inter-enterprise messaging between participants adheres to their obligations. Ultimately, the protocol is intended for use by machines. Participants deploy services and software clients that exchange electronic messages and carry out the interaction. For these services and clients to adhere to the protocol, it has to be made available to them in some machine-readable language.

Messaging specification also addresses concerns about intra-enterprise messaging coordination. An enterprise may be participating in many different interactions with several participants at the same time. In addition to fulfilling their obligation towards each interaction, an enterprise is also concerned with coordinating their overall messaging activities to ensure that their internal business process complies with their business policies.

Organizational Requirements

MOR exhibit a high level of abstraction, which makes them a closer match to business concepts than the machine-oriented messaging specification. MOR are thus more suited for processing by humans, e.g. business analysts and architects acting on behalf of the stakeholders in the interaction.

Concerns of analysts and architects are centered on identifying goals of their enterprise and reasoning about means for their fulfillment. Analysts need to identify, represent, and decompose business problems in ways that allow them to deepen their understanding of the problems and share business domain knowledge[5]. Architects need means to explore and evaluate alternative solutions for business problems and rationalize decisions made in choosing solutions. To specify a business solution, architects need to identify business activities, electronic or physical, involved in implementing the solution and ensure that the execution of these activities satisfies the business goals.

Even though stakeholders in the interaction share the concern of inter-enterprise interaction viability, their local business needs may conflict. MOR captures business processes inter-dependencies of interacting roles, thereby providing means for reconciling their conflicting needs.

Four Viewpoints for Service Interaction Design

Segregating the concerns along two axes produces the four viewpoints for interaction design represented by the four quadrants in figure 2. Each viewpoint embodies a sub-set of concerns of a certain stakeholder.

(Q1) Global Requirements

This view embodies the global stakeholder's concerns of specifying the context of the interaction. The interaction context is specified in terms of the interacting roles, their high-level motivations for interacting, dependencies that make the interaction possible, and risk that comes with these dependencies. Role-Dependency (RD) diagrams are suitable for this view; they are used to specify the roles in the interaction and analyze their intentions and inter-dependencies from a global (neutral) point.

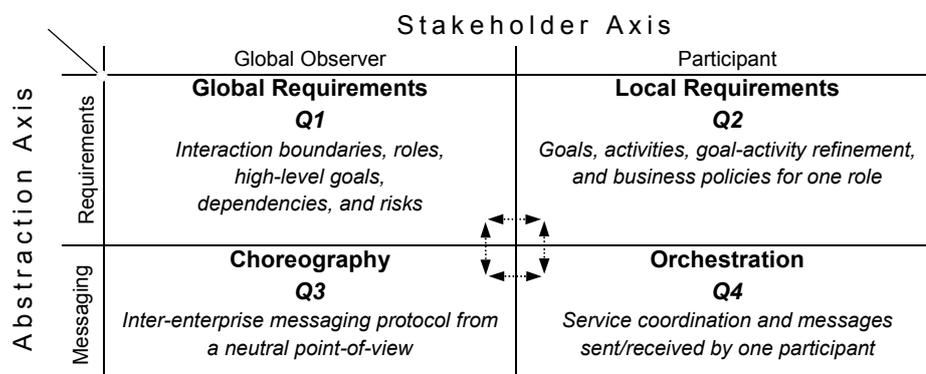


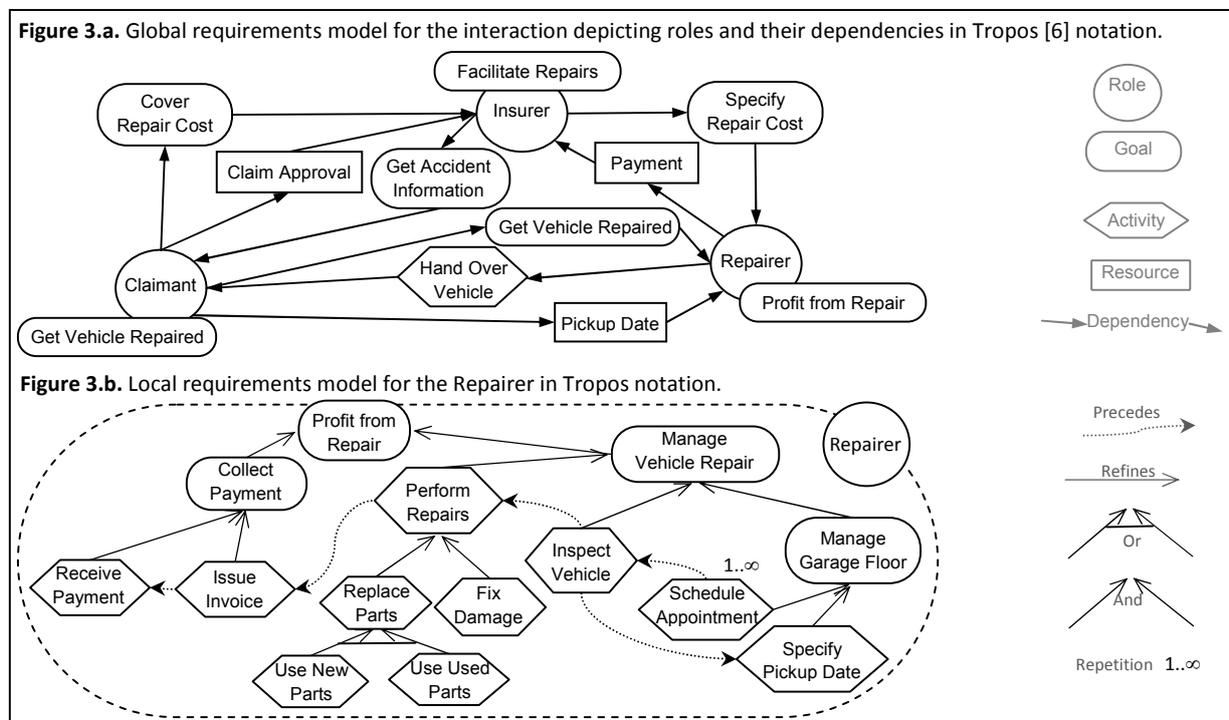
Figure 2. Four viewpoints for interaction design.

Figure 3.a is an RD diagram[6] depicting a high level representation of the vehicle repair interaction. Each role is represented as an oval with goals corresponding to that role attached to it. Dependencies between roles motivate the interaction between them. Roles depend on each other for fulfilling goals, performing activities, or furnishing resources. For example, the Claimant depends on the Repairer to get their vehicle fixed, including all entailed activities.

Dependencies are fulfilled either electronically (i.e. via messaging) or physically. Some dependencies are physical by nature; for instance, the Claimant has to haul their vehicle to fulfill “Hand Over Vehicle”. Otherwise, MOR provide the flexibility of making a design decision as to how to fulfill each dependency. For instance, the interaction can be designed such that the Insurer either mails a check or provides an electronic payment to fulfill “Payment”.

RD diagrams enable the rationalization of responsibilities of goal fulfillment. For example, the Claimant’s expectation that the Insurer will “Cover Repair Cost” is consistent with the Repairer’s reliance on the Insurer for “Payment”.

Delegating responsibility is not risk-free, and RD diagrams enable reasoning about risks. For example, although it reasonable to assume that the Repairer has the necessary expertise to fulfill the “Specify Repair Cost” goal, it arguably entails risks of fraud. Identifying such risks drives further analysis to mitigate them or explore alternative responsibility assignment.



Note also that outlining the interaction context includes specifying what roles and goals are NOT part of the interaction. For instance, the role of “Parts Supplier” and goals related to ordering vehicle parts are not part of the interaction.

(Q2) Local Requirements

This view embodies business-level concerns of one participant which are to specify their business goals, determine what activities are required to fulfill the goals, and ensure that these activities comply with business policies. Goal-Activity (GA) diagrams are suitable for representing this view.

GA diagrams provide mechanisms for successively refining high level goals into finer-grained goals and eventually activities[6] for one role. A GA diagram is constructed from the point of view of one role, and hence may include goals and activities relevant only to that role and not necessarily to the global view. Figure 3.b shows a GA diagram for the Repairer role.

GA diagrams specify what activities, including both physical and messaging activities, are carried out by a participant to achieve their goals. Through refinement, relations between high-level goals and operational activities (e.g. messaging) are established, thereby allowing for reasoning about how the activities contribute towards goal achievement. For example, the Repairer needs to “Specify Pickup Date” as part of achieving “Manage Garage Floor”, whereas in the RD diagram the pickup date appeared to serve a purpose only for the Claimant.

GA diagrams also capture business policies. Data flow and ordering constraints between activities are represented as activity precedence links. For instance, it can now be seen that the Repairer is obliged to finish all repairs before issuing an invoice. Note how using MOR we explicitly represent the ordering of physical activities relative to messaging activities.

RD and GA diagrams support various quantitative and qualitative analyses for complex models [5, 7].

(Q3) Choreography

This view is concerned with specifying the messaging protocol from the global stakeholder’s point of view using languages such as WS-CDL[3]. The protocol describes obligations of interacting roles as constraints on messaging sequences they are allowed to exchange. The protocol provides a standard against which the global stakeholder assesses participants’ compliance to the roles they play.

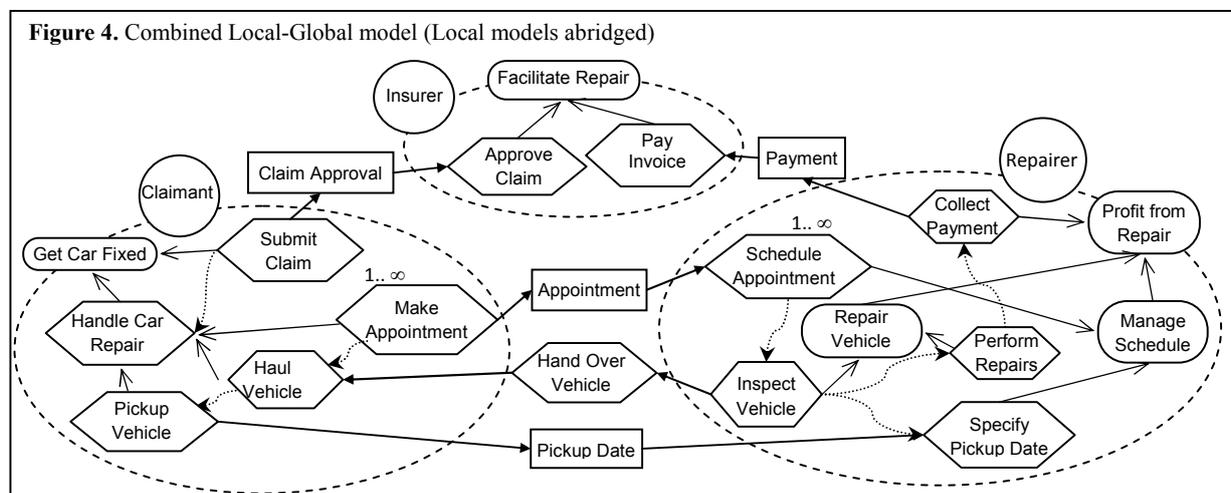
(Q4) Orchestration

This view is concerned with specifying messaging exchanged between services implemented by a single participant, either internally or with the outside world, using standard process description languages such as BPEL[8].

Relating the Four Viewpoints

To maintain consistency between the viewpoints, we establish the following relations.

Q1-Q2: Each dependency in the RD model ties together a depending activity in the GA model of a depender role to a dependee activity in the GA model of the role fulfilling the dependency. By combining together the GA models for all roles we establish inter-enterprise ordering of activity execution. The order is such that the depending activity can only execute to completion when the dependee activity has fulfilled the dependency. Figure 4 shows the result of using dependencies to relate the local GA models of the vehicle repair interaction roles into a combined local-global model. By tying together the GA models we enable participants to negotiate reconciliation of their needs.



Q1-Q3: We also use dependencies to infer what messages will be exchanged between participants[4]. A dependency fulfilled electronically typically implies two messages: a request message from the depender and a response message from the dependee providing information that fulfills the dependency. For example, the “Appointment” dependency implies that the Claimant sends a message requesting an appointment and the Repairer replies with the date and time of the appointment. While we determine messages in the interaction by examining dependencies, we determine message ordering by examining constraints on the execution of activities at both ends of each dependency.

Q2-Q4: Through refinement, the relations of business activities of a participant to their messaging activities are established. Compliance of messaging with business policies of the participant can then be automatically verified [9].

Q3-Q4: By relating the inter-enterprise messaging to messaging activities of a participant, the participant’s compliance with the messaging protocol can also be automatically verified [10].

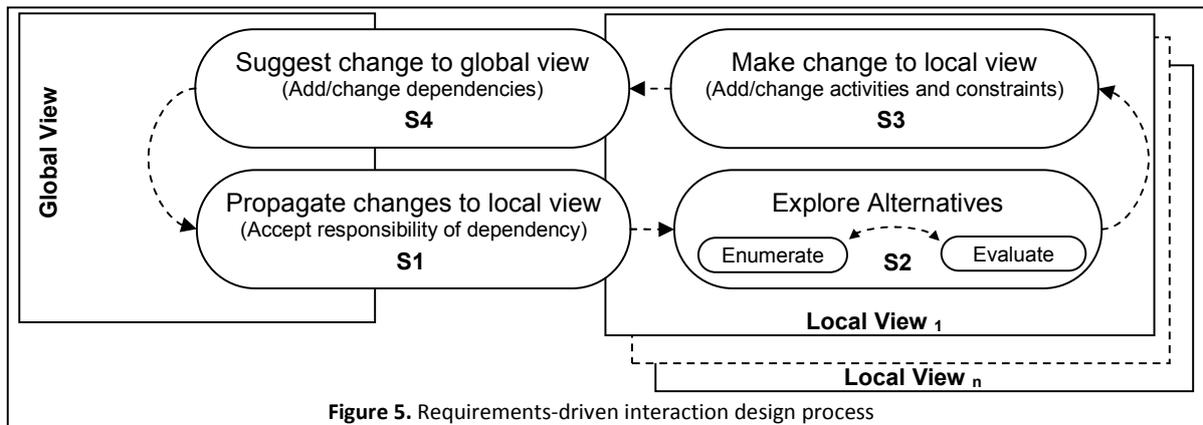
Requirements-Driven Interaction Design

By elevating the level of abstraction at which the interaction is specified, we enable a design process that focuses on the requirements of the stakeholders. Additionally, by relating the local viewpoints to the global viewpoint we enable the participants to collaborate with the global stakeholder on reconciling their needs. In the forward-engineering version of the process we start with specifying interaction requirements and then deriving the messaging protocol from the combined local-global requirements model.

Collaborative Specification of Requirements

Participants collaborate on specifying interaction requirements while the regulatory agency mediates negotiations between them. During an iteration of the process a participant can perform one of (see figure 5):

- S1.** Accept responsibility of a dependency that was assigned to them.
- S2.** Enumerate and evaluate alternatives for fulfilling a newly accepted dependency.
- S3.** Adapt their local view to comply with business policies, fulfill a goal, or fulfill a dependency.
- S4.** Suggest an addition of a dependency to the global model, subject to approval by the regulatory agency and other participants.



The iterative nature of the process allows us to apply it to an existing model[11]. Assuming that the model in figure 4 is the starting point, the design process may proceed as follows:

- (S3) To guarantee fulfillment of “Collect Payment” goal (in figure 3.b) the Repairer decides to add to their local model a “Verify Claim Approval” activity to be performed prior to inspecting the vehicle or performing repairs.
- (S4) Realizing that this activity requires the Claimant to provide information, the Repairer suggests adding a “Proof of Claim Approval” dependency to the global model and suggests it is

to be fulfilled right before car inspection, i.e. the Claimant brings a physical copy of claim approval when they drive the vehicle in. The State's Department of Insurance (the regulatory agency) deems this suggestion reasonable and agrees to it.

- (S1) The Claimant accepts the new responsibility of providing proof of claim approval.
- (S2) Evaluating alternatives, the Claimant finds that providing a physical copy of the approval is undesirable. One reason is that if they forget to carry it to the Repairer, they will be denied inspection and they will have wasted time hauling the car.
- (S2) The Claimant enumerates other alternatives and decides that providing proof of approval (electronically) prior to hauling the car to the Repairer is preferred.
- (S3) The Claimant adds an activity to their local model for providing the approval.
- (S4, S1, S3) The process continues: Claimant suggests the alternative and Department of Insurance agrees to it, Repairer accepts alternative, and Repairer adapts their local model by making "Verify Claim Approval" precede "Schedule Appointment".

Designing the interaction at the level of MOR allowed us to:

- a. Take into account physical activities as design constraints (e.g. hauling the car).
- b. Explore alternatives to electronic messaging (e.g. providing hard copy of claim).
- c. Rationalize design choices based on business policies.

Deriving Messaging Protocol from Requirements Models

Once an agreement on the requirements models is reached, the stakeholders need to specify a messaging protocol that satisfies these requirements. MOR embody sufficient semantics for automatically deriving the messaging protocol[12]. We implemented an automated tool that accepts MOR as input and generates the messaging protocol as output. The messaging protocol of figure 1 can be obtained by applying our tool to MOR of figure 4. The tool can be obtained by contacting the authors.

Closing Remarks

When designing inter-enterprise interactions we need to address the concerns of multiple stakeholders. On one hand, each interaction participant is concerned with fulfilling their business goals and enforcing business policies while observing their obligations towards others. On the other hand, the regulatory agency overseeing the interaction is concerned with fairness, interoperability, and protocol compliance. To manage these concerns and enable a collaborative design process, we proposed four design viewpoints.

Our requirements-driven design process allowed us to explore design alternatives and rationalize choices using business policies. Requirements models also allowed us to incorporate physical activities into the design process both as constraints and implementation choices. Furthermore, by exploiting the semantics of requirements models we derive the messaging protocol automatically, thereby ensuring consistency between the two artifacts.

References

- [1] M. P. Papazoglou and D. Georgakopoulos, "Service-Oriented Computing," *Communications of the ACM*, vol. 46, 2003, pp.25-28.
- [2] C. Peltz, "Web Services Orchestration and Choreography," *IEEE Computer*, vol. 36, 2003, pp.46-52.
- [3] "Web Services Choreography Description Language Version 1.0, <http://www.w3.org/TR/ws-cdl-10/>," W3C, 2005.
- [4] A. Mahfouz, L. Barroca, R. Laney, and B. Nuseibeh, "Customizing Choreography: Deriving Conversations from Organizational Dependencies," Proc. 12th Int'l IEEE Enterprise Distributed Object Computing Conference (EDOC'08), IEEE Computer Society, 2008, pp. 181-190.
- [5] E. Yu and J. Mylopoulos, "Understanding "Why" in Software Process Modelling, Analysis, and Design," Proc. 16th International Conference on Software Engineering (ICSE'94), IEEE Computer Society, 1994, pp. 159-168.
- [6] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An Agent-Oriented Software Development Methodology," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 8, 2004, pp.203-236.
- [7] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Formal Reasoning Techniques for Goal Models," *Journal on Data Semantics*, vol. 2800, 2003, pp.1-20.
- [8] T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana, "Business Process Execution Language for Web Services Version 1.1," 2003.
- [9] R. Kazhamiakin, M. Pistore, and M. Roveri, "A Framework for Integrating Business Processes and Business Requirements," Proc. 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC'04), IEEE Computer Society, 2004, pp. 9-20.
- [10] H. Foster, S. Uchitel, J. Magee, and J. Kramer, "Model-based Verification of Web Service Compositions," Proc. 18th International Conference on Automated Software Engineering (ASE'03), IEEE Computer Society, 2003, pp. 152.
- [11] A. Mahfouz, L. Barroca, R. Laney, and B. Nuseibeh, "Requirements-Driven Collaborative Choreography Customization," Proc. International Conference on Service-Oriented Computing (ICSOC'09), Springer, 2009, pp. 144-158.
- [12] A. Mahfouz, L. Barroca, R. Laney, and B. Nuseibeh, "From Organizational Requirements to Service Choreography," in *Proceedings of the 2009 Congress on Services - I - Volume 00*: IEEE Computer Society, 2009, pp. 546-553.

About The Authors

Ayman Mahfouz is Chief Architect at Webalo, Inc (www.webalo.com). He has been developing enterprise and mobile software for the past 15 years. He has a Masters in software engineering and is currently finishing his PhD in the topic of Requirements-Driven Adaptation of Service-Oriented Interactions. He is a member of the IEEE and ACM. Contact him at amahfouz@gmail.com

Leonor Barroca is a Senior Lecturer in Computing in the Open University, UK, with a PhD in Computer Science from the University of Southampton, UK. Contact her at l.barroca@open.ac.uk

Robin Laney is a Senior Lecturer in Computing at the Open University, UK, with a PhD in Computer Science from King's College, University of London, UK. Contact him at r.c.laney@open.ac.uk.

Bashar Nuseibeh is a Professor of Software Engineering and Chief Scientist of Lero – The Irish Software Engineering Research Centre, and a Professor of Computing at The Open University, UK. He is Editor-in-Chief of IEEE Transactions on Software Engineering, and holds PhD in Software Engineering from Imperial College London. Contact him at b.nuseibeh@open.ac.uk