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.

Abstract

Designing service-oriented interactions requires addressing concerns of many stakeholders across enterprise boundaries. To ensure that stakeholders' concerns are well-understood and properly addressed, we modularize them into four viewpoints that cover representations ranging from business goals to service messaging protocol. We propose a framework for interaction design that helps maintain consistency between representations across the viewpoints. The framework allows stakeholders to collaborate on reconciling their business needs and automatically obtain messaging protocols that satisfy these needs. The viewpoints and the framework enable a requirements-driven collaborative interaction design process.

Specifying Service-Oriented Interactions

Service-Oriented Architecture (SOA) is emerging as an enabler for inter-enterprise interactions. Services provide platform-independent abstractions around software systems thereby enabling interoperability between heterogeneous systems [1]. Several languages are emerging as standards for describing service interfaces, service architectures, and inter-enterprise interaction protocols (sidebar 1). Inter-enterprise service interaction protocols specify expected message exchanges between a set of abstract roles [2]. At runtime, messaging between actual participants must abide by the established protocol between the roles they play in the interaction.

Sidebar 1: Relevant SOA and Web Service (WS) Standards

Language	Used to Specify
WS- Description Language (WSDL) www.w3.org/TR/wsdl	Message types and service operation signatures.
WS-Choreography Description Language (WS-CDL) www.w3.org/TR/ws-cdl-10/	Multi-participant messaging protocol from a global/neutral point of view.
WS-Business Process Execution Language (WS-BPEL) www.oasis-open.org/committees/wsbpel/	Service messaging coordination from a single participant point of view.
WS-BPEL Extension for People (BPEL4People) www.oasis-open.org/committees/bpel4people/	Human tasks within a WS-BEPL specification
SOA-Modeling Language (SoaML) www.omg.org/spec/SoaML/	High-level service architecture, business information model, and service component architecture.

In WS-CDL, messaging and control flow of an interaction protocol are specified using these constructs (pseudo-language used for brevity):

- **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 may execute concurrently.
- **Choice:** represents conditional choice between mutually-exclusive options.
- **While** (condition) **Do:** represents repetition.

Consider the messaging protocol for a vehicle repair interaction between three roles; Insurer, Claimant, and Repairer:

```
Sequence {
    Claimant Send Claim To Insurer
    Insurer Send ClaimApproval To Claimant
    While (NOT AppointmentConfirmed) Do {
      Claimant Send AppointmentRequest To Repairer
      Choice {
           Repairer Send AppointmentConfirmed To Claimant
           Repairer Send AppointmentRejected To Claimant
      }
    }
    Parallel {
        Repairer Send VehiclePickupDate To Claimant
        Sequence {
            Repairer Send Invoice To Insurer
            Insurer Send Payment To Repairer
        }
    }
}
```

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 vehicle pick-up date) and they bill the Insurer for the cost.

Often times the design and execution of inter-enterprise interactions are overseen by a global observer, e.g. regulatory agency, that ensures compliance of participants to the protocol. The global observer in this example is the State's Department of Insurance which regulates the insurance business and handles disputes about non-compliance. WS-CDL specifies messaging between the interacting roles from the point of view of a global observer, thereby catering for these needs. By doing so, WS-CDL also abstracts away from internal business process specifics of participants, thereby providing interoperable protocol specification.

Even though these emerging languages (sidebar 1) provide interoperable specifications, they have serious limitations:

- ***They focus on operational aspects of the interaction and hence are detached from the participants' business goals***. It is hard ensure that a messaging protocol, say in WS-CDL, satisfies the participants' goals without explicitly representing these goals and relating them to messaging activities. SoaML attempts to represent high-level service architectures, nevertheless it does not

provide mechanisms for refining these architectures into messaging protocols. An architect using SoaML is left to manually construct a multi-participant messaging protocol with no systematic means for ensuring that it is consistent with the high-level architecture or that it satisfies the goals of participants. The matter is complicated when considering that goals of participants often conflict and SoaML does not help reconcile these goals.

- ***They only specify electronic messaging and leave out physical activities, i.e. activities carried out by humans in a non-electronic medium.*** Physical activities are often crucial to achieving the goals of the interaction. For instance, the vehicle repair interaction is pointless if the Repairer does not physically perform vehicle repair, even if all required messaging takes place as specified by the protocol. Furthermore, the ordering of physical activities relative 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. Even though BPEL4People tackles human activities, it is limited to specifying a human's interaction with the electronic system, and only from one participant's point of view.

These deficiencies call for a richer specification of the interaction. In particular, the need for capturing business goals and their refinement into activities, messaging and otherwise, call for specifying the interaction at the level of Models of Organizational Requirements (MOR) motivating the messaging [3]. MOR capture goals motivating participants to interact and all activities that constitute the interaction, including physical activities.

Whereas the high-level nature of MOR makes them useful for business-level reasoning, messaging protocols are adequate as a machine-readable specification. Not only do the two representations address different concerns, but they also serve different purposes for the two different types of stakeholders, i.e. interaction participants and the global observer. To ensure the interaction design process properly serves all stakeholders we need to disentangle these concerns.

Separation of Design Concerns

Having identified two types of stakeholders and two distinct levels of abstraction, concerns of interaction design can be separated along two fundamental axes: the stakeholder axis and the abstraction axis.

The Stakeholder Axis

The stakeholder axis separates concerns of interaction participants from those of the global observer.

Participants

Each interaction participant 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 and 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.

Global Observer

The global observer, i.e. regulatory agency, is a stakeholder whose concerns are to facilitate the interaction and encourage participants to interact.

- To encourage participants to interact, the global observer helps potential participants assess and mitigate risks involved in the interaction. The global observer 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 observer aims to ensure interoperability, for which specifying upfront the obligations of the interacting roles is essential. The specification of obligations becomes a standard contract for participants wishing to play one of the roles in the interaction.

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

The Abstraction Axis

The abstraction axis separates business-level concerns, captured in MOR, from concerns related to messaging specification.

Organizational Requirements

Organizational requirements 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.

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 [4].

- 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, required for 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 capture inter-connections between business processes of interacting roles, thereby providing means for reconciling their conflicting needs.

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, i.e. 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 needs to coordinate their overall messaging activities to ensure that their internal business process complies with their business policies.

Four Viewpoints for Service Interaction Design

Segregating concerns along two axes produces the four viewpoints for interaction protocol design represented by the four quadrants in figure 1. Each viewpoint embodies a sub-set of concerns of a certain stakeholder.

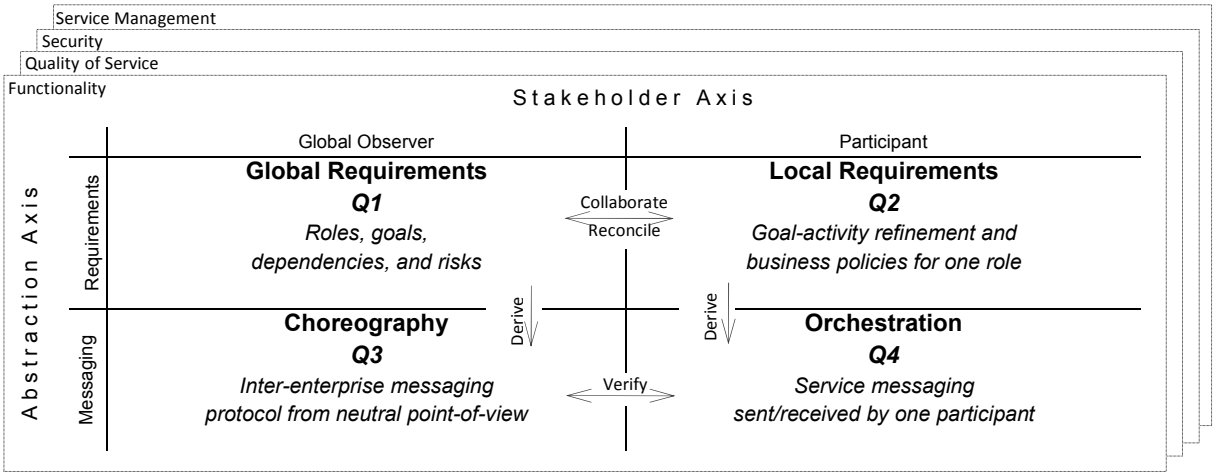


Figure 1. Four viewpoints for interaction protocol design and how they fit in the larger SOA picture [1]. (Note how non-functional concerns orthogonal to protocol specification are represented as parallel planes.)

(Q1) Global Requirements

This view embodies the global stakeholder's concerns of specifying the context of the interaction 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 [5] are suitable for this view; they are used to specify the interacting roles and analyze their inter-dependencies from a global point.

Figure 2.a depicts our proposed usage of RD diagrams to represent the global requirements 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.

Figure 2.a. Global requirements model for the interaction depicting roles and their dependencies.

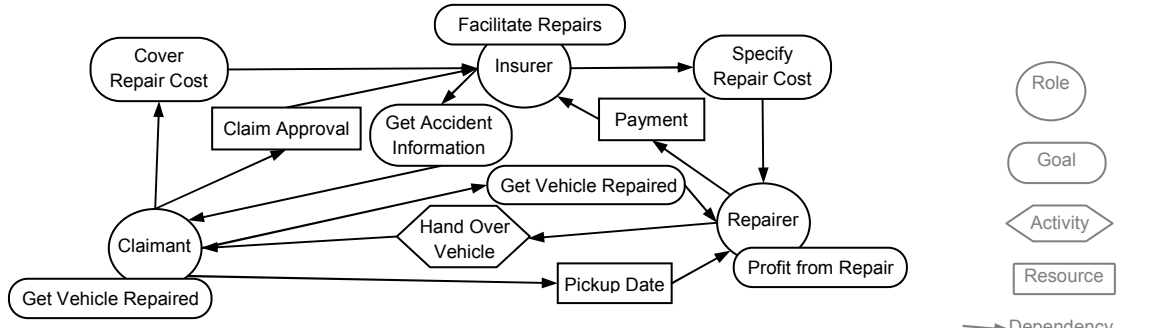
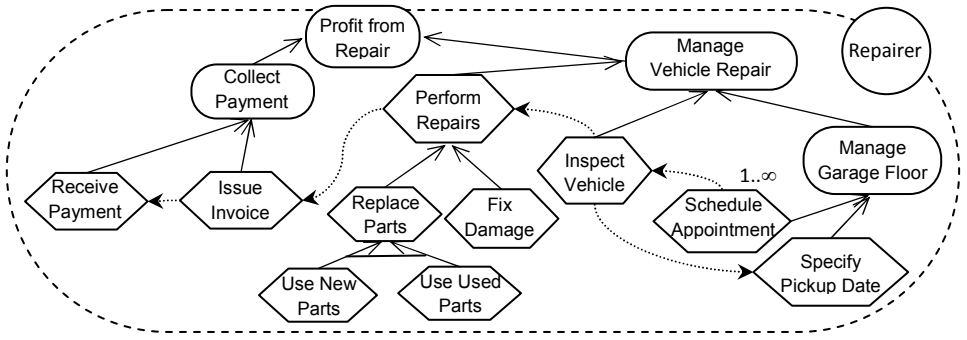


Figure 2.b. Local requirements model for the Repairer.



Dependencies are fulfilled either via electronic 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 design decisions 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 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".

RD diagrams also enable reasoning about risks that involved in delegating responsibility. 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.

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 [5] are suitable for representing this view.

GA diagrams provide mechanisms for successively refining high level goals into finer-grained goals and eventually activities[5] 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 2.b shows how we capture the local view of the Repairer role in a GA diagram.

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 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 the ordering of physical activities relative to messaging activities is explicitly represented.

RD and GA diagrams support various quantitative and qualitative analyses for complex models [4, 6].

(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. The protocol describes valid messaging sequences that the

interacting roles 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.

Framework for Service Interaction Design

By relating the representations in the four viewpoints we construct a framework that maintains consistency between them. In this article, we focus on relating Q1-Q2 and Q1-Q3.

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, thereby providing linkage between the local models. Figure 3.a. 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 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 3.b). By tying together the GA models we enable participants to negotiate reconciliation of their needs.

Figure 3.a Combined Local-Global model (local models abridged).

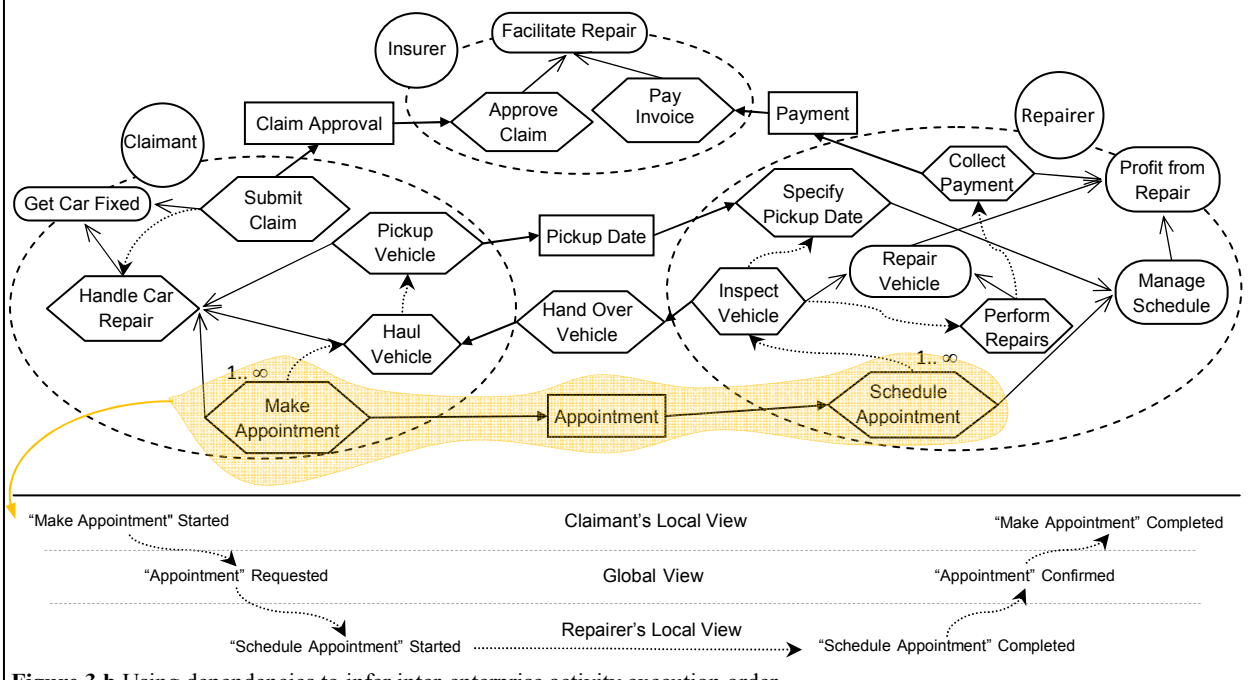


Figure 3.b Using dependencies to infer inter-enterprise activity execution order

Q1-Q3: Dependencies also imply what messages will be exchanged between participants[3]. 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 protocol messages are determined by examining dependencies, message ordering is determined by examining constraints on the execution of activities at both ends of each dependency. Figure 4 summarizes the basic rules for automatically deriving messaging protocol from MOR.

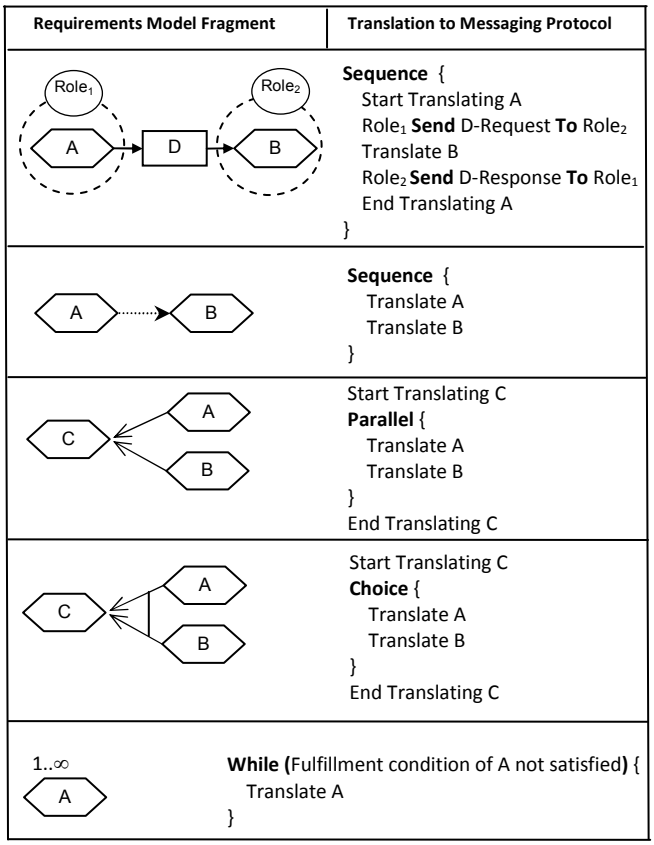


Figure 4. Rules for deriving messaging protocol from requirements models

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 participants to collaborate with the global stakeholder on reconciling their needs. The forward-engineering version of the design process starts with collaborative specification of interaction requirements and then deriving the messaging protocol from the combined

local-global requirements model. The process provides a path to proceed systematically from possibly conflicting business requirements of multiple enterprises all the way to the specification of inter-enterprise messaging.

Collaborative Specification of Requirements

Participants collaborate on specifying interaction requirements while the regulatory agency mediates negotiations between them. The design process proceeds in iterations as follows:

(Q2) A participant P_1 makes a change to their local view to comply with business policies or fulfill an emergent goal.

(Q2 to Q1) If the change to the local model of P_1 involves adding or changing activities that participate in dependencies, the change is propagated to the global model.

(Q1) The regulatory agency reviews the requested change to the global model made by P_1 and approves it if it finds it reasonable.

(Q1 to Q2) The regulatory agency notifies the participant at the other end of the dependency, P_2, of the added/changed responsibility. P_2 can then accept the new dependency and propagate its impact to their local model.

(Q2) P_2 adapts their local model to fulfill their responsibility towards the added dependency.

The iterative nature of the process makes it suitable for application to an existing model[7]. Assuming that the model in figure 3 is the starting point, the design process may proceed as follows:

- (Q2) To guarantee fulfillment of "Collect Payment" goal the Repairer decides to add to their local model a "Verify Claim Approval" activity to be performed prior to inspecting the vehicle.
- (Q2 to Q1) 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 by the Claimant before car inspection.
- (Q1) The State's Department of Insurance deems this suggestion reasonable and agrees to it.
- (Q1 to Q2) The Claimant is notified of the new dependency. They accept the new responsibility of providing proof of claim approval.
- (Q2) The Claimant adds an activity to their local model for providing the approval prior to handing the vehicle to the Repairer.

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. We implemented an automated tool (sidebar 2) that accepts MOR as input and, utilizing the precise semantics of MOR[8] and rules in figure 4, generates the messaging protocol. The messaging protocol of the vehicle repair example is obtained by applying

our tool to MOR of figure 3. Our tool also generates comments interleaved with the messaging protocol to indicate the points at which physical activities are expected to execute. We have also developed transformations from the pseudo language used in this article to WS-CDL constructs.

Sidebar 2: Download Tool and Case Studies

The vehicle repair and healthcare case study results are available at http://tinyurl.com/chreq-eval-rep. Our CHoreography REQuirements tool (CHREQ) is downloadable from https://sourceforge.net/projects/chreq. The download includes source files for the example in this article and those of the case studies.

Evaluation

The vehicle repair example in this article is an abridged version of a real-world case study built for a European insurance company. We applied our approach to the full version of the case study as follows:

1. Modeled the original requirements for the European market.
2. Analyzed requirements from real public documents published by Departments of Insurance in several States in the US and Canada.
3. Applied our process to the original model to re-design it to the North American context.
4. Generated the messaging protocol for the re-designed models.

We also applied our approach to a case study from the healthcare domain (sidebar 2). In both cases results were encouraging:

- The majority of requirements in the public documents were easily captured using our design process in an iterative manner.
- Our design process allowed systematic exploration of design alternatives and rationalizing choices using business policies.
- Physical activities were naturally incorporated into the design both as design constraints and alternative implementation choices to electronic messaging.
- Messaging protocols were derived automatically using our tool, thereby ensuring consistency between the requirements and the protocol. In fact, the tool helped identify errors in hand-constructed messaging protocol published earlier.

The evaluation helped identify areas where our approach can be improved:

- Even though MOR are built from a few primitive constructs, there is a curve to learning how to create robust models. To smooth this learning we built a set of patterns that architects can apply to incrementally create requirements models.

- MOR diagrams can get complicated quickly. We need to develop techniques for modularizing MOR into reusable parts, especially for optional and exceptional execution paths. To help manage the complexity we plan to integrate our tool with an automatic graph layout tool.
- It remains to be seen how our approach supports reverse engineering, i.e. re-constructing MOR from existing messaging protocols in a semi-automated manner.
- Some problematic aspects of WS-CDL remain challenging at the level of MOR. Business needs requiring synchronization of multiple instances of an interaction stands out as one.

These results encouraged us to plan further evaluations including getting other practitioners to apply our design process to their business cases.

Acknowledgments

Nuseibeh is supported, in part, by SFI grant 03/CE2/I303_1

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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