An Overview of Business Process Adaptations via Protocols *

[Extended Abstract]

Nirmit Desai Amit K. Chopra Munindar P. Singh
Department of Computer Science
North Carolina State University, Raleigh, NC 27695-8206
{nvdesai, akchopra, singh}@ncsu.edu

ABSTRACT

Business process management in open environments poses special challenges. In particular, such environments are dynamic, thereby requiring frequent changes in business processes. Current business process modeling approaches handle such changes in an ad hoc manner, and lack a principled means of determining what needs to be changed and where.

This paper provides an overview of process adaptability through a novel application of business protocols, especially of protocol composition, introduced in our previous work. Through a real business scenario of auto-insurance claim processing, this paper briefly describes how a wide range of adaptations can be handled naturally and systematically via protocol composition. The illustrated adaptations have been evaluated via a prototype.

1. INTRODUCTION

Successful business process management requires supporting three key properties of open environments, namely, *autonomy*, *heterogeneity*, and *dynamism* [4, pp. 7–10]. Supporting autonomy means modeling and enacting business processes in a manner that offers maximal flexibility to the participants by only minimally constraining their behavior. Supporting heterogeneity means making as few assumptions as possible about the construction of the components for the parties, concentrating instead on characterizing the interactions among them. For open environments, interconnections among components, not the components themselves, are critical.

Dynamism has two flavors. *Membership dynamism* means that the set of components may change. Components may be added and removed at either design-time (*design-time membership dynamism*) or run-time (*runtime membership dynamism*). *Structural dynamism* means that the set of interconnections is not fixed. Dynamism reflects changes in the business requirements; modern businesses face intense pressure and must repeatedly reconfigure themselves in order to thrive, if not to survive. As changes in requirements are routine, an elegant way of handling such changes is vital. Thus, dynamism poses a difficult challenge, one that has not been adequately addressed in the literature, and is the theme of this paper. Supporting dynamism implies supporting process adaptability.

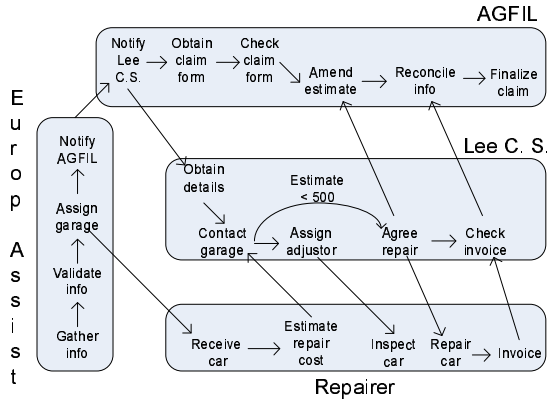

Figure 1: CrossFlow Insurance Claim Processing

Previously, Desai et al. [2] reported a methodology (named OWL-P) for developing business processes that supports autonomy and heterogeneity. The essence of OWL-P is an abstraction of business protocols. A *business protocol* is an abstract, modular, and publishable specification of rules that govern a business interaction among two or more *roles*. A protocol is abstract because it does not model the proprietary business logic of the agents enacting the roles. It is modular because it achieves a specific goal which may be a part of another bigger goal. Due to their abstract and modular nature, business protocols, each achieving a goal, can be combined to form a *composite protocol* that achieves a bigger goal. Each role's perspective of a protocol is captured in its *role skeleton*. A role skeleton, when integrated with the private business logic of an agent enacting the role, yields the agent's *local process*. A *business process* is an aggregation of the local processes of all the agents involved.

In this way, protocols partition a process in a cross-cutting manner whereas conventionally, flows are taken as partitions. For instance, peeking ahead on an example studied in great detail below, each of the boxes in Figure 1 represents a flow. Such boxes would be the modules of abstraction in conventional approaches. By contrast, OWL-P focuses on the interconnections among the boxes; such interconnections are a natural fit for the interorganizational nature of SOC environments [2]. As a striking illustration of the

*This research was partially supported by the NSF under grant DST-0139037 and by DARPA under contract F30603-00-C-0178.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

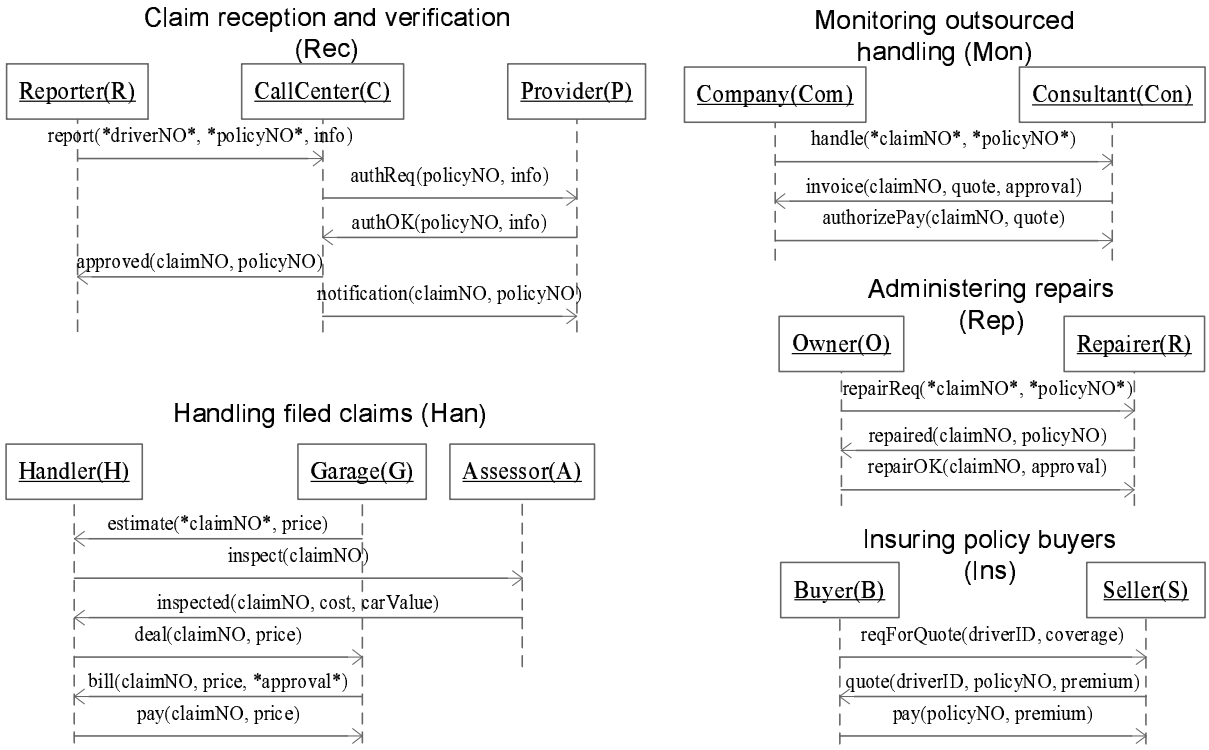

Figure 2: Scenarios from the protocols corresponding to the insurance claim process of Figure 1

importance of focusing on the interconnections, notice that the process flow (taken from [1]) in Figure 1 misses the insurance holder and its interactions altogether. Although the internal flow of the insurance holder's process (its box) may not be important to AGFIL, its external interactions with other parties (i.e., the insurance holder's interconnections) are crucial for modeling the business of AGFIL. Figure 2 shows how the above example is modeled via protocols.

Previous work on protocols provides the basic mechanisms to specify, compose, and enact them to produce software engineering benefits such as reusability and flexibility. This paper is a sketch of protocol-based treatment of dynamism in a real business scenario. Adaptability in general involves a broad variety of research challenges. Hence, this paper concentrates on three kinds of adaptations: (1) exceptions, (2) changes in business policies, and (3) changes in business models.

To illustrate the protocol-based approach, this paper considers the above insurance claim processing case studied under the CrossFlow project [1]. Being a real-life business scenario previously studied by others, this example provides an independent and significant test-case for this approach. Figure 1 shows the parties involved in the process and the process flow. AGFIL is an insurance company in Ireland. The present scenario deals with automobile insurance. AGFIL underwrites the insurance policy and covers any losses incurred. Europ Assist provides a 24-hour help-line service for receiving claims. Lee CS is a consulting firm that coordinates with AGFIL and deals with repairers to handle the claims. A network of approved repairers provide repair services. AGFIL holds ultimate control in deciding if a given claim is valid and if payment will be made to the repairer.

2. PROTOCOLS

Figure 2 shows scenarios from the protocols involved in the above insurance example. Such protocols are independently developed and stored in a repository for reuse.

Any realistic business process would involve multiple interaction protocols. For example, AGFIL would need protocols for selling insurance policies and for receiving and handling claims. Consider how protocols can be combined to form a composite protocol. Let's assume AGFIL outsources the help-line service for receiving claim reports to a call center, but handles the claims itself. The requisite process can be obtained by combining the *Ins* and *Rec* protocols in a certain way yielding a new *Bas* (Basic insurance claim processing) composite protocol.

As the ontologies of the protocols may not match, and they may have interdependencies, simply unioning the rules of the component protocols is not enough. The constraints on combining protocols are specified as a set of *composition axioms* in a composition profile. These axioms identify roles across protocols, specify data flow dependencies, event ordering constraints and ontological dependencies. Composite protocols can be treated and enacted like other protocols. The details of protocol specification and composition are discussed in our previous work [2].

Typically, the protocols are enacted with some of the roles having business relationships established via participating in some other protocol, e.g., *Ins* establishes the contract between the *buyer* and the *seller*. This business relationship is a contract represented by a set of commitments. For brevity, this presentation skips such relationship formation protocols. Figure 3 shows all such contractual commitments among the roles after all the relationships are formed and commitments delegated.

3. ADAPTING PROCESSES

Here, we describe what kinds of adaptations can be handled

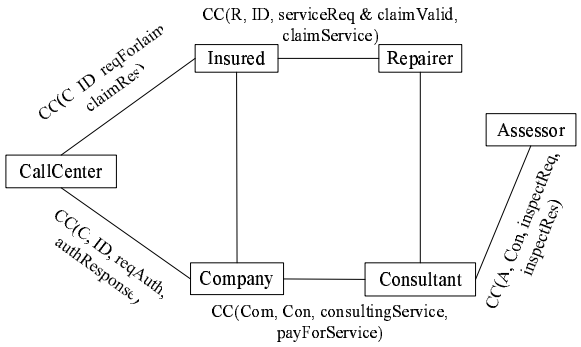

Figure 3: Commitments among parties

by applying the mechanism of protocol composition. Each of the adaptations feature varying degrees of autonomy, heterogeneity, structural dynamism, and design-time membership dynamism. The following assumes that a composite protocol *Picp* (Partial insurance claim processing) is constructed from the composition of *Bas*, *Mon*, *Han*, and *Rep* of Figure 2.

3.1 Exception Handling

Exceptions are (typically) uncommon conditions arising during a business interaction, e.g., fraudulent auto-insurance claims. Protocols facilitate incorporating the handling of an exception into an existing process.

A composite protocol *Ficp* (Fraudulent insurance claim processing) can be constructed by combining *Picp* with a protocol *Fra* (fraud detection and reporting) When AGFIL and Lee CS form their relationship, a conditional commitment CC(Com, Con, consultingService, payForService) is created meaning that AGFIL will authorize payments for handling individual claims if Lee CS provides the consulting service. In case of fraud, the *company* cancels the policy coverage of the *insured* and the *consultant* releases the *repairer* from the commitment to perform repairs. The *assessor*'s advising of a fraud should discharge CC(A, Con, inspectReq, inspectRes) and the *consultant*'s forwarding it to the *company* should mean the consulting service was provided.

The resulting interaction allows AGFIL to end the process by canceling the commitments in case of a fraud. Simply adjusting the commitments can allow greater flexibility without affecting other parties adversely. Here, the support for structural dynamism comes from the added interconnections among the existing partners to handle the exceptions. Support for autonomy comes from the ability to control the effects of foul behavior of a participant.

3.2 Changes in Business Policy

In handling claims where the value of the car is less than the estimated cost of repairs, the *company* may want to scrap the car, i.e., declare it a total loss. To settle this, it pays the *insured* a sum equal to the value of the car and takes possession of the car instead of administering repairs. Also, if the damage is minor, the *insured* may accept a cash settlement instead of having the car repaired. A protocol *Pcsc* (Pay cash and scrap car) can specify the interactions needed for such a policy.

AGFIL would have delegated its conditional commitment CC(S, B, serviceReq $\wedge$ claimValid, claimService) to Lee CS, thereby making Lee CS responsible for servicing claims, who would in turn delegate it to the *repairer*. When the *consultant* advises scrapping the car, it fulfills its commitment to provide the consulting service and the commitment to service the claim falls back to the *company* as the car is not to be repaired but to be paid for. Similarly, when the *consultant* advises a cash payment for a minor damage, and the *insured* agrees to it, it fulfills the commitment for the consulting service and the *company* becomes committed to paying the *insured*. This can be modeled by delegation. Assuming that *Picp* is the current interaction protocol among the partners, *Pcsc* can be composed with *Picp* yielding the composite protocol *Icp* (Insurance claim processing) that handles this policy change.

In *Icp*, a business policy change internal to AGFIL is accommodated across the business process. Here, support for structural dynamism comes from the modified interconnections among the existing partners to handle the policy change. Support for heterogeneity comes from the fact that a change of business logic internal to AGFIL could be accommodated.

3.3 Changes in Business Model

Quite often, enterprises change their business models to respond to changes in their business environment. Business process outsourcing (BPO) [3] has recently become a popular approach for conducting business.

The composition of *Bas* discussed in Section 2 already illustrated this type of adaptation. Let's assume AGFIL could handle claims itself. AGFIL outsourced its call center to Europ Assist to focus on its core business of selling insurance policies. Thus, the business model of AGFIL changed to have a partnership with Europ Assist.

Obviously, a change in the business model is a big shift for an enterprise; new partnerships are formed and new interconnections among the parties emerge. Thus, such a change features not only structural dynamism, but also design-time membership dynamism. Both are supported via composition with protocols having new roles.

4. REFERENCES

[1] CrossFlow consortium / AGFIL. Insurance (motor damage claims) scenario. Technical report, CrossFlow consortium, 1999. Document identifier: D1.a, http://www.crossflow.org.

[2] N. Desai, A. U. Mallya, A. K. Chopra, and M. P. Singh. Interaction protocols as design abstractions for business processes. *IEEE Transactions on Software Engineering*, 31(12):1015–1027, 2005.

[3] P. Harmon. *Business Process Change: A Manager's Guide to Improving, Redesigning, and Automating Processes*. Morgan Kaufmann, 2002.

[4] M. P. Singh and M. N. Huhns. *Service-Oriented Computing: Semantics, Processes, Agents*. Wiley, 2005.