

# Supporting Process Flexibility through Service Relationship Modeling

Malinda Kapuruge, Jun Han, Alan Colman and Muhammad Adeel Talib  
Faculty of Information and Communication Technologies  
Swinburne University of Technology,  
Melbourne, Australia  
{mkapuruge, jhan, acolman, mtalib}@swin.edu.au

**Abstract**— Traditional business process modeling and enactment approaches for service compositions do not explicitly represent service relationships. Such an explicit representation is required to safeguard the business invariants of the composition as well as of the composed services. In this paper we introduce a new approach to business process modeling and enactment based on explicitly defined service relationships. It allows business processes to be defined in a flexible manner while safeguarding the business invariants amidst runtime modifications.

**Keywords**- SOA, BPM, Flexibility, Business Invariants, Service Relationships

## I. MOTIVATION

Business process modeling (BPM) and enactment approaches in information systems are used to support and coordinate the business activities in an automated manner. With the advancement in service based technologies, the execution of activities become more distributed and autonomous. As a result, several process model standards have been developed to coordinate these distributed and autonomous service activities.

However, while improving the process support, the current standards for business process modeling and enactment (e.g. WS-BPEL[1]) lack the *flexibility* to adapt to the runtime process variability requirements in a service composition. One of the causes of this inflexibility is the inadequate representation of business invariants as well as mutual obligations of collaborating services. Such a lack of representation may lead to inaccurate modifications of business processes that violate the goals of collaborating service providers as well as of the composition itself. Therefore it is necessary to modularize the business logic of the composition in way that it explicitly represents the relationships among the collaborating services.

As an example think of a business that provide road side assistance to motorists. The business may compose many other services such as garages, tow cars, paramedics, hotels, case officers etc. During runtime the business may add more services, replace the existing service providers, or add more business processes to achieve some other goals using the existing set of services. For example, an additional process might provide safety checks of vehicles, car-towing as another business offering. Further, the business owners might need to optimize the existing business operations by changing the coordination of the composed services. Therefore it is necessary to model business processes to tackle above issues at runtime to minimize the system downtime and the maintenance cost.

## II. THE APPROACH

### A. Overview

We adopt an organizational approach where an organizational structure is placed between the business process and the concrete services as shown in Figure 1. The organizational structure consists of the defined service relationships between the participants/collaborators and provides a flexible but managed platform for defining and adapting business processes at runtime. Such a structure must have the following characteristics. It should be able to:

1. Absorb the runtime variability of existing services, e.g. variability in service behavior.
2. Accommodate the runtime changes in business processes.
3. Ensure that the change requirements in both services and business processes do not violate the business objectives and constraints of
  - a. the organization itself, and
  - b. the collaborators bound by the organization.

The Role Oriented Adaptive Design (ROAD) approach to adaptive software systems has such an organizational structure as the basis for service composition. ROAD[2], which is made of binary relationships between Roles (positions) that represent the requirements of the services and are bound to concrete services. However, it does not provide support for process definitions.

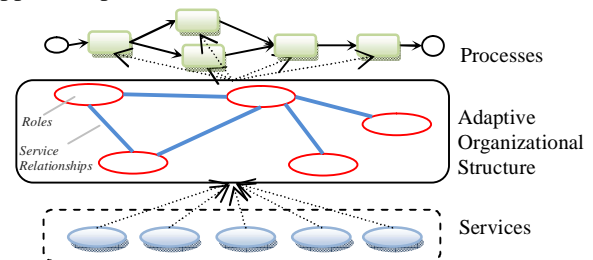


Figure 1: Adaptive Organizational Structure

In this research, we adopt ROAD and further extend it to realize the process support as described above. We take the declarations that define expected behaviors between pairs of services in the ROAD composite and use them as building blocks to construct one or more meaningful business process definitions and change these definitions and their runtime instantiations.

### B. Concepts

**Service relationships:** We define an organization as a collection of *inter-service relationships*. A service relationship describes the mutual obligations and the

expected behavior between two collaborating parties. As shown in Figure 2, the Case Officer has certain obligations towards the Garage and vice versa. An aggregation of all the service relationships that a Case Officer is involved in, defines the position or Role of the Case Officer. A human such as *Mr. Smith* may play the role of Case Officer.

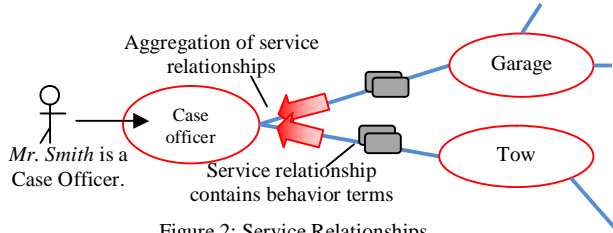


Figure 2: Service Relationships

**Unit of behavior:** A service relationship is described by a number of terms that correspond to a *unit of behavior* between the two parties. Such a unit encapsulates one or more *tasks* that should be performed as part of the behavior. Further it defines the business invariants that should be protected to maintain the relationship.

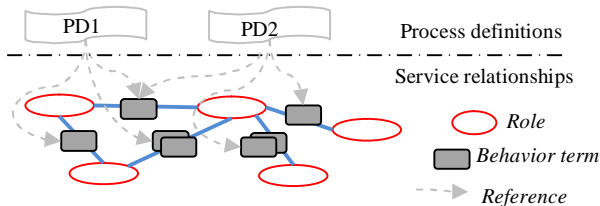


Figure 3: A process definition logically groups behavior terms

**Business Processes:** In SeRenDiP, a process definition is a logical grouping of such behaviors in service relationships as illustrated in Figure 3. Multiple process definitions may share a behavior term, e.g. client login to the system. Changes applied to behavior terms of service relationships automatically get reflected in the business processes. Therefore it is not necessary to change multiple business processes as the service relationships get changed with existing service providers.

**Business invariants:** Business invariants can be defined in service relationships level as well as in process level. Service relationship level invariants ensure that no modification of business processes violates the business objectives of the collaborators. Similarly, the business invariants defined in the process level ensures that no modification of service relationships violates business goals of the process.

**Adaptation:** The adaptation of business processes is achieved via regulation and reconfiguration of behavior terms defined in the service relationships. Changes to behavior terms at runtime are accommodated as long as they do not violate the invariants defined at both levels. This safeguards the goals of collaborators as well as of the composition.

**Process progression:** An instantiated process progresses by executing tasks defined in behavior terms. In order to define the progression, we adopt an event-action based approach, where a *true* evaluation of an event pattern leads to an execution of a task defined in behavior terms.

**Role behavior determination:** The possible behavior of a role is determined by its relationships with others as shown in Figure 2. Each relationship defines a number of tasks that should be performed by an obligated role. If these task descriptions (pre-conditions, messages, events) get changed, the behavior defined of the obligated roles get changed too.

### III. METAMODEL

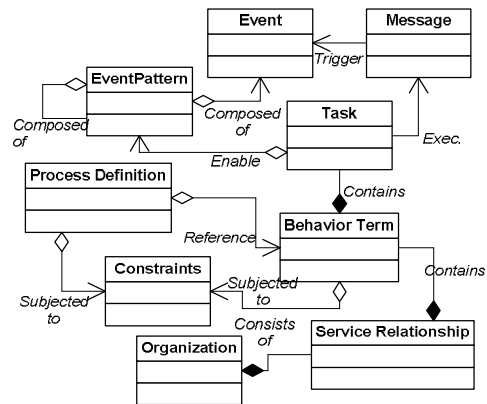


Figure 4: The process meta-model

Figure 4 shows how these concepts are expressed in a meta-model. The central concept of meta-model is the Service Relationship defined in an *organization*. Each *service relationship* contains a number of *behavior terms* that defines the expected behavior of the two participating Roles. A *process definition* is a logically grouped set of behavior terms. A behavior term consists of a number of *tasks* that should be executed to maintain the relationship. Each task becomes doable as a result of a truth evaluation of an *event pattern*. An event pattern is a logical expression that combines *events*. As tasks get executed a number of *messages* get exchanged among the collaborators. These messages are interpreted by the organization in terms of relationships to generate more events. In order to protect the business invariants, runtime modifications in behavior terms and process definitions are subject defined *constraints*.

### IV. CONCLUSION

In this paper, we have introduced a business process modeling and enactment approach based on explicitly defined service relationships. We presented the concepts and the meta-model of our approach. The service relationships modularize the possible interactions and the business invariants that should be protected. That provides the platform to define many business processes without violating the business goals of the composition as well as its collaborators.

### REFERENCES

[1] BEA, IBM, Microsoft, SAP, AG, Siebel Systems, Business Process Execution Language for Web Services V1.1 specification 2003, e.g., <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>  
 [2] A. Colman and J. Han, "Roles, players and adaptable organizations," *Applied Ontology*, vol. 2, no. 2, 2007, pp. 105-126.