A Provision-centric Model for Electronic Services[♣]

Clare Gryce and Giacomo Piccinelli
Software Systems Engineering Group
University College London, London (UK)
{c.gryce,g.piccinelli}@cs.ucl.ac.uk

Christian Zirpins
Distributed Systems Group- VSIS
University of Hamburg, Hamburg (Germany)
zirpins@informatik.uni-hamburg.de

Abstract

In addition to producing a pervasive standard and a technology framework, the Web Service initiative has prompted businesses to re-examine their service delivery channels. Back-end systems are already in place to capture business knowledge and manage operational strategies and procedures. Web Services enable access to such systems, as well as basic orchestration. The infrastructure is in place to recreate the business-level notion of an electronic service. However, the gap between technology-oriented and business-oriented models for services is still considerable.

In this paper, we outline a model for electronic services as defined by the FRESCO project. The model constitutes an architectural blueprint of the technical and business infrastructure for an electronic service. The focus is on the provisional aspects of electronic services.

1. Introduction

Electronic services can be considered from different viewpoints. From a technical perspective, electronic services tend to be related to specific functions of server-type applications (e.g. login services). The application-oriented viewpoint of electronic services has recently been reinforced by the Web Service initiative [2] from the W3C. From a business perspective, a service is a self-contained capability for which the realisation details are hidden (e.g. a postal service, a freight service). Such capabilities can be used indirectly as part of a business process [3], or directly by end users.

In the FRESCO (Foundational Research on Service Composition) [4], the business and technical perspectives on electronic services are combined in a single model. The main feature of the FRESCO service model is the distinction between content and provision. The content of a service refers to the core capabilities that are enabled by the service. For example, the content of a freight service refers to the capability of moving goods from one place to the other. Provision refers to the business channel between the provider and the consumer of a service. For example, provision covers the formation and management of a contractual agreement between the freight company and a user.

Content and provision are complementary aspects of a service. On the one hand, the provision logic of a service depends upon the capabilities that the provider is able to offer to consumers. On the other, the capabilities that are actually made available to consumers depend upon the provision logic defined by the provider. As an example relating to quality of service (QoS) capabilities, the fact that a freight provider has different aircraft makes it possible for the company to provide different speeds of delivery for goods (content). Still, the actual speed of delivery offered to a given consumer derives from the specific service-level agreement negotiated with the consumer (provision). The FRESCO model focuses on the provision aspects of services.

2. Outline of the Service Model

The FRESCO model assumes that the content of an electronic service is embodied in a set of core Assets. For example, in the context of freight services, transport, packaging, and insurance capabilities are among the Assets that a service may be built upon. Around this core, the model defines a provision layer referred to as the Service Shell. The Service Shell includes capabilities related to Publication, Negotiation, Contract Management, Interaction, Accounting, and Billing for the electronic service. Each cluster of capabilities contained in the Service Shell is referred to as a Shell Stripe.

A common Information Base is shared among all the capabilities in the Service Shell. Information about the Assets can be generated and stored in the Information Base. Information can also be fed back from the Information Base to the Assets. In both cases, the transfer must happen within the scope of the execution of a Service Shell capability. Similar considerations apply to the flow of information between the Service Shell and the consumers of the service.

In accordance with indications provided by models such as RM-ODP (Reference Model for Open Distributed Processing) [1], the FRESCO service model includes the concept of roles. Roles mainly represent centres of rights

[♣] This research is partially funded by HP and the EU project EGSO

and responsibilities. They provide a level of indirection between provision logic and the actual resources involved in realising Service Shell capabilities. As an example, the role Invoice Manager can utilise specific resources that are part of an ERP (Enterprise Resource Planning) in order to realise the provision logic of the system.

3. Service Content

The content of an electronic service derives mainly from the Assets of the service provider. Assets are abstracted using a basic object-oriented model (e.g. the model assumed by Web Services [2]). The only requirement that is imposed on Assets by the FRESCO service model is for them to expose specific interfaces to the Service Shell.

Using the example of the freight service, the progress tracking system of the transport company is an Asset. Such an Asset could be required to expose an interface that will be needed for contract management (e.g. monitoring delivery speeds as defined by service-level agreements). The interface can be as simple as a Web Service [2] that returns the percentage of completion of a physical delivery. The majority of existing IT systems expose interfaces based on some form of object-oriented model. Further assumptions would be useful (e.g. explicit modelling of interaction flows), but in most cases, not available in practice.

4. Service Shell

The purpose of the Service Shell is to define a common framework for the engineering and management of the various resources and activities that are involved in the provision of an electronic service.

The atomic unit adopted by the model is referred to as a Capability. The level of granularity for Capabilities may vary according to context. An example of a Capability is credit checking on a potential customer. Different Capabilities can be loosely related based upon the area of service provision to which they contribute. The concept of the Shell Stripe is used to model this relation. For example, capabilities that relate to metering and invoicing are clustered into the Accounting Shell Stripe.

The FRESCO model defines three main aspects of a Capability: content, operational logic, and results. The concept of content for a Capability is very similar to that previously suggested for the overall service. For example, the invoicing Capability is concerned with sending an invoice to a customer. The description framework that FRESCO provides for the content of a capability is ontology based.

The operational logic derives from the business process that the Capability is based on. Emphasis is given to the steps within the process that involve interaction with other Capabilities, as well as with entities that are external to the service provider. In the invoicing example, the interaction process with the customer also needs to be specified. In FRESCO, business processes are formalised using the WfMC (Workflow Management Coalition) model for workflows [3].

The explicit modelling of results focuses on the impact of a Capability, independently of the operational logic used. The results associated with a Capability are modelled in terms of the information and artefacts that are produced during the execution of the Capability. In the invoicing example, an invoice and a receipt from the customer should be produced.

Capabilities can be interdependent. Interdependence can take a hierarchical form, whereby a set of capabilities {C1, C2, …} is instrumental in the creation of a new capability C. The new Capability cannot exist without the sub-Capabilities, but the sub-Capabilities can exist independently of C. For example, the Capability of sending information with recorded delivery can be instrumental to the invoicing Capability, but recorded delivery can exist independently of invoicing. Interdependence can also take the form of a peer-to-peer network of relations between capabilities. In this case, if any of the capabilities in a set {C1, C2, …} are removed, either the remaining capabilities cannot exist or their individual value is diminished. For example, the relation between invoicing and payment is one of interdependence. The amount of the invoice might change depending upon the payment method, while payment can only be justified if an invoice has been issued.

5. Conclusions

The conceptual framework proposed by FRESCO moves in the direction of bridging the gap between the business and technical views on electronic services. In particular, the emphasis is on the provision channel of electronic services and the cooperation requirements for the related entities and resources. From a technical perspective, the FRESCO approach is based on a multi-level architecture that is aligned with OMG's MDA (Model-Driven Architecture) methodology.

6. References

[1] AA.VV. "Reference Model of Open Distributed Processing, 1-4", ISO 10746, 1995.

[2] E. Cerami, "Web Services Essentials", O'Rielly and Associates, 2002.

[3] L. Fisher (Ed.), "Workflow Handbook", Workflow Management Coalition and Future Strategy Inc., 2002.

[4] G. Piccinelli, C. Zirpins, and W. Lamersdorf, "The FRESCO Framework: An Overview", IEEE-SAINT, 2003.