Concept of Service Provisioning and Realization

Raymond Wu

IBM Software Group

Abstract

Service transformation is one of the big challenges in this century from a scale and complexity perspective. The main objective is to increase service levels, enhancing both customer satisfaction and productivity. To reach this goal requires multi-dimensional approaches from business innovation, service strategy, solution architecture, implementation and integration. Therefore, alignment, coherence and a roadmap are very important. In this research paper we introduce conceptual service provisioning with extension to interconnectivity as the service foundation. We further explore service provisioning based on the concept we introduced and realization based on service design pattern foundation.

Keywords: Service transformation, Conceptual service provisioning

1. Introduction

The concept of service provisioning requires service identification, decision, validation, analysis and design. Our previous research within enterprise integration has been divided into four areas: decision, analysis, design and virtualization. Within the life cycle, political-model decision-making and rational-model analysis were applied to the key areas in which the components of business, service and physical elements are selected and validated in provisioning a particular business context goal or KPI within the project scope of the enterprise integration dimension.

In the research on metadata interoperability we revealed, in relation to both political and rational models, how the components are derived through the validation process and how the strategy of component implementation achieves layer interoperability. In this paper we extend our research in service concept and service provisioning, and define the service ontology (Ψ) as follows:

In a full service interoperability within micro and macro processes:

Demand = Supply = Provisioning, means: $\Psi = \Sigma (\Psi(d_i) \times \Psi(s_i) \times \Psi(p_i))$

This implies that service provisioning is implemented through the micro process and component alignment between service, demand and supply, so each required element, in either new or enhanced form, can be narrowed down to one or more mapped components which are consolidated by the object, symbol and service concept. The realization of this top-down vertical integration approach requires robust middleware to support the inter-communication between component- and-component and component-and- repository. In other words, the concept needs physical support from an integration service bus such as the middleware in runtime services.

Figure 1 shows the micro interoperability within service ontology. Common service and information is a top-down approach which follows the nature of the business implementation direction. From the technical perspective the idea was to make integration easier by using the “virtualization” concept. From the organization perspective this enhances the flexibility of service provisioning and the agility between organization and service, thus increasing the interoperability across organizations.

t

2. Literature review

Oberle, Staab and Eberhart (2006), in their Kaon server design, show several potential sources of semantic metadata. These sources provide input for the framework regarding the semantic management of software components; this includes deployment descriptors, web and application server configuration files, source code information or annotations and database metadata. The component metadata collector parses this information and automatically integrates it into the ontology as semantic metadata (instances). The inference engine is embedded in the application server and applies the ontology and semantic metadata for reasoning and querying. The application server’s core functionality and the administration console both use the reasoning capability. The latter allows the developer to browse and query the ontology at runtime so that configurations may be queried or checked to avoid inconsistent system configurations during development and at runtime Semantic mediation between object and symbol has been with us for quite some time; in Greek philosophy, semantic ontology was developed in semiotics to include all of linguistics as a special case. Aristotle distinguished objects, the words that refer to them, and the corresponding experiences in the conceptual experience (Sowa, 2000). The triangle of integration ontology consists of object, symbol and concept. In our conventional design we emphasized the use of a symbol to describe the object, and attempted to synchronize for the moving object.

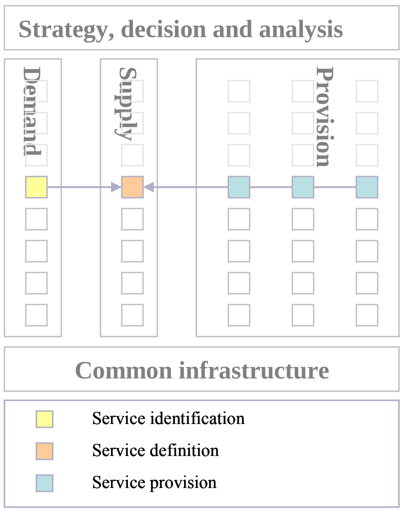

Figure 1: Demand, supply and provision.

In many large scale enterprises, architecture such as SOA (service oriented architecture), MDA (model driven architecture) and EDA (event driven architecture) use many conceptual metadata to mediate between the "dynamic" and "static."

3. Service concept

In an advanced CRM (customer relationship management) system, we assume that all the customer master data needs to be static and consistent; this is due to information and process not having sufficient "speed" to synchronize with changes in market, customer preference and industry trends if the concept does not support dimensional strategy.

What we need is a service concept mediation between object and symbol. Figure 2 demonstrates this mediation. A racing car consists of an ontology of three parts: object (physical car), symbol (description) and concept (a conceptual car and racing). Our focus is on concept instead of object or symbol, as the object in this case is the car itself and does not contribute to the objective, and the symbol notation contributes no semantic information.

Let 'τ' denote the object, the physical car

'χ' denote characteristics of the car (weight, engine, etc.) - the symbol

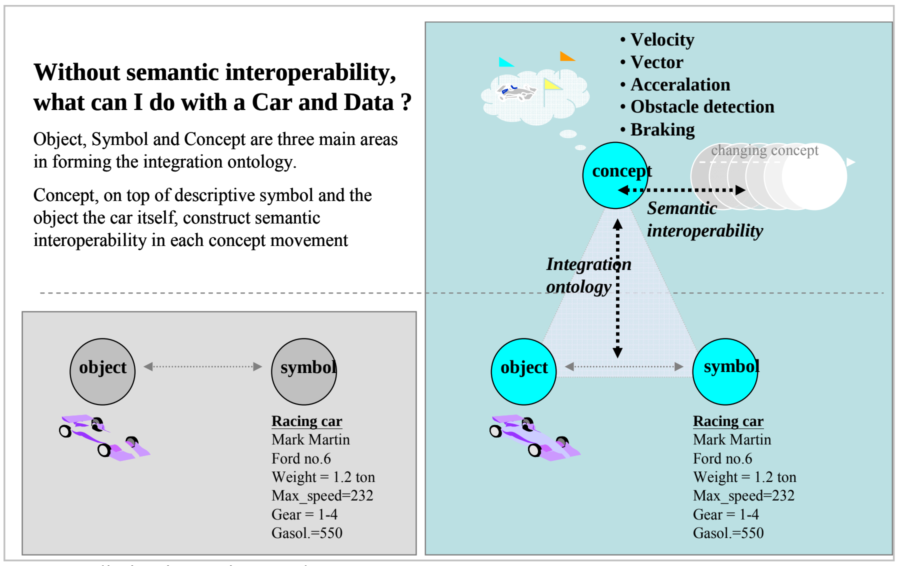
Without semantic interoperability, what can I do with a Car and Data ?
Object, Symbol and Concept are three main areas in forming the integration ontology.
Concept, on top of descriptive symbol and the object the car itself, construct semantic interoperability in each concept movement
• Velocity
• Vector
• Accelaration
• Obstacle detection
• Braking
changing concept
concept
Semantic interoperability
Integration ontology
object
symbol
Racing car
Mark Martin
Ford no.6
Weight = 1.2 ton
Max_speed=232
Gear = 1-4
Gasol.=550

Figure 2: Service concept mediation in service ontology.

'ν' denote velocity

'α' denote acceleration

'ω' denote direction

'β' denote braking

'δ' denote distance between car and obstacle

'Ψ' denote the ontology interoperability between ν, α and ω

Let the concept of "win in racing" be constructed by semantic integration

$\Gamma = \Psi\ (\nu, \alpha, \omega, \beta, \delta)$ and $\Psi^1, \Psi^2 \ldots \Psi^n$ denote the change of strategy in coping with environment, trend, competition, KPI (key performance identifiers) or market need (win). In the case of the racing car this can be the competition strategy and environment.

Concept Γ in a moving environment produces the semantic interoperability in each change; we use $\tilde{O} = \Gamma\ (\Psi, \tau, \chi)$ to denote the ontology of enterprise integration. This clearly shows that the ontology $\tilde{O}$ built on top of the interoperability between object, symbol and concept needs recursive interference to adjust its interoperability $\Gamma^{*} = \Gamma + \Delta\Gamma$

The condition in producing $\Delta\Gamma$ is based on $\tilde{O} = \Gamma\ (\Psi, \tau, \chi)$, instead of $\Gamma = \Psi\ (\nu, \alpha, \omega, \beta, \delta)$

In other words, our assumptions are that design mode $\tilde{O} = \Gamma\ (\Psi, \tau, \chi)$ is the origin in moving execution mode from Ψ^{i-1} to Ψ^{i}, concept Γ mediates "racing" and "car" in competition, semantic interoperability creates the dimensional strategy in mediating object and symbol, and the componentization is the foundation in this ontology.

The design of service components follows our previous strategy by using business semantic injection to take the context into the semantic and micro processes; this means each component has its contract-level responsibility in provisioning a subset of service. This interface helps the coordination between components and metadata to build the foundation of interoperability within the service layer and between layers. As we are still at a pioneer stage in service provisioning and aggregation, we understand this cannot be accomplished within a short period. Further research efforts will be undertaken following the current work (Barcia and Brent, 2005).

4. Service mediation

As service provision requires service mediation to serve as foundation. Now this study need to shed light on the realization for how mediaters support the concept this paper proposes. D. Roman et al. in their "Web Service Modeling Ontology (WSMO)" state that semantic services need to meet the following requirements: web compliance, ontological base, strict decoupling, centrality of mediation, ontological role separation, and execution semantics (Roman, 2005). These requirements were addressed in our research and the strategy is identical to that which we proposed. Basically, mediation patterns manipulate messages in-flight on the bus by requests or events.

According to IBM service mediation, there are several basic patterns for mediations; the following more complex patterns can be built by combining the simple patterns:

Protocol switch: Enables service requesters to dispatch their messages using a variety of interaction protocols or APIs such as SOAP/HTTP, JMS and the MQ Integrator (MQI).

Transform: Translates the message payload (content) from the requester's schema to the provider's schema. May include enveloping, de-enveloping or encryption.

Enrich: Augments the message payload by adding information from external data sources such as customization parameters defined by the mediation or from database queries.

Route: Changes the route of a message, selecting among service providers that support the requester's intent. Selection criteria can include message content and context as well as a target's capabilities.

Distribute: Distributes the message to a set of interested parties and is usually driven by subscriber interest profiles.

Monitor: Observes messages as they pass through the mediation unchanged. Can be used to monitor service levels, assist in problem determination or meter usage for subsequent billing to users.

Correlate: Derives complex events from message or event streams. Includes rules for pattern identification and rules that react to pattern discovery; for example, by generating a complex event derived from the content of the triggering event stream.

Mediations can be explicitly configured within a solution. As well as being attributes of the services, policies can be set by the solution administrator for a single interaction or set of interactions (Hutchison, Schmidt, and Wolfson, 2005). Once the vertical skeleton and service extension are built, the interoperability reflected from the business layer has the capability of integrating cross-organization and cross-geographical constraints, as fully transparent services become possible if each system or organization follows an industry standard by using metadata strategy and an industry framework. An example of this would be a case in which several key industry vendors have built up an industry framework for service providers in that particular industry to follow, starting from business domain analysis and working through to component deployment. Each service provider would thus benefit from the industry-common services and the common interface, particularly those involving merger and acquire cases (B2B, B2C). Any business activity becomes easier when the various parties involved have a common understanding and a consensus on business terms, service definitions, interfaces and component specification, particularly if these become transparent and common practices.

The foundation of service interoperability is achieved by interconnectivity between service components by using strategy of common component and an aggregated component. We further develop the service virtualization on top of this interoperability skeleton, which we call enterprise vertical integration, in extending the integration across industry and geographical areas. This is the virtualization service in a horizontal direction; once the industries of different regions build up their semantic interoperability in three metadata repositories (business, service and information), they can synchronize their business, operation and strategy even if they have their local implementation extended from the interoperability skeleton.

In enterprise service provisioning and virtualization, we constructed the skeleton of enterprise service with leverage to each layer and the interoperability of the layers. Service virtualization is a process of leveraging enterprise integration strategy within a specific geographical region or across regions. This can be an industry standard scenario such as a credit card transaction in which the service, geography and systems are fully transparent to both cardholders and merchants. A full virtualization environment needs to be built on top of the foundation of component design and service transformation process in order to apply the vertical enterprise integration approach to this final stage, which is the goal of our research work. The standardization of metadata technology and common services and components are all requirements in this stage.

5. Conclusions

In the new service provisioning industry, the object and the symbol – the description of the object – have been the focus of interest; however, the service concept behind the object and the symbol, is the real key in provisioning the demanding service and maintaining the consistency between object, symbol and service. The service concept mediates the object and the symbol within the service ontology as concepts change from time to time. We use the statistics information from the symbol which pointed to the object, we then aggregate the symbol using such data as analytical information about the behavior of the object and finally input the metadata into the concept layer; in this way more strategic governance will be produced from concept to cope with a changing environment. In service realization, web compliance, ontology, decoupling, centrality and semantics are keys in building the patterns. Basically, mediation patterns manipulate messages in-flight on the bus by requests or events.

References

[1] D. Oberle, S. Staab, A. Eberhart, Semantic Management of Distributed Web Application. *IEEE Distributed Systems Online*, 7(4), art. no. 0605-o, 5001, 2006.

[2] J. Sowa, *Ontology, Metadata, and Semiotics, ICCS 2000 AI #1867,* Springer-Verlag, pp. 55-81, 2000.

[3] D. Oberle, S. Staab, A. Eberhart, Building SOA solutions with the Service Component Architecture. *IBM Research*, 2006.

[4] D. Roman, Web Service Modeling Ontology (WSMO), *European Commission under the projects DIP 2005.*

[5] B. Hutchison, M. Schmidt, D. Wolfson, SOA programming model for implementing Web services,. *IBM Research*, 2005.