1

SWS Challenge Scenarios

Holger Lausen[1], Ulrich Küster[2], Charles Petrie[3], Michal Zaramba[1], and Srdjan Komazec[1]

[1] Semantic Technology Institute Innsbruck, University of Innsbruck, Technikerstr. 21, 6020 Innsbruck, Austria, firstname.lastname@sti2.at
[2] Institute of Computer Science, Friedrich-Schiller-University Jena, 07743 Jena, Germany, ukuester@informatik.uni-jena.de
[3] Computer Science Dept. University of Stanford, Gates Building, Stanford, CA 94305-9020, USA petrie@stanford.edu

Summary. The Semantic Web Service Challenge defines problem scenarios that serve as the basis for the certification and comparison of approaches participating in the challenge. These scenarios are classified in two broad types: Mediation and Discovery. The first primarily address aspects of data and process mediation whereas the latter focus on problems around automated service discovery. Currently, there are two main scenarios in each type which are described in detail in this chapter.

1.1 Introduction

From the very beginning it was the intention of the challenge organizers to provide problems that are as close to reality as possible. However all of the challenge organizers (at that time) have been engaged full-time in research jobs. We have been thinking about various possibilities, and after several iterations we decided to start with a problem based up on the analysis of a current industry standard.

Instead of having a single giant scenario, we choose to have several problems on different levels, which ideally build one upon the other. The first problem is centering around process and data mediation during a purchase order process. The second basic scenario is about discovering shipment providers given specific constraints such as destination country, weight, etc. Subsequent organizers and participants have together developed an enhanced discovery scenario as well as an orchestration scenario. In this section we familiarize the reader with the basic details of the scenarios. This details are intended to help understanding the solutions presented in subsequent chapters.

1.2 The Mediation Scenario

After deciding that the mediation scenario has to be in the area of mediating a purchase order between various systems we started looking closer at currently used solutions. We identified three standards that could potentially be used to derive a problem scenario:

- EDIFACT[4] is the United Nations/Electronic Data Interchange For Administration, Commerce, and Transport (UN/EDIFACT). It is a plain text format for data exchange developed by the United Nations.
- ebXML[5] is sponsored by sponsored by OASIS and UN/CEFACT and commonly known as e-business XML. ebXML is a family of XML based standards whose mission is to provide an open, XML-based infrastructure that enables the global use of electronic business information in an interoperable, secure, and consistent manner by all trading partners.
- RosettaNet[6] is a non-profit consortium aimed at establishing standard processes for the sharing of business information (B2B). The standard is based on XML and defines message guidelines, business processes interface and implementation frameworks for interactions between companies.

As EDIFACT is based on plain text messages, we decided not to take it as basis since this would push too many low level data transformation issues to the participants. In opposite to the two XML based standards no parsers and other tools could be used. The choice between ebXML and RosettaNet was taken on the base of available documentation. Since only the RosettaNet messages and protocols are freely and easily accessible we decided to use RosettaNet.

In the mediation scenario we focus on interoperability problems of existing systems. The aim is to show how semantic Web technologies can help to overcome the need for manual development of mediation systems.

In our initial scenario description we provide relevant information about the systems involved in two forms: using current Web Service description (WSDL) and natural language text annotations. Using current state-of-the-art technologies a programmer has to interpret the information given and to code components that overcome the heterogeneity between the different systems. In the SWS Challenge participants are asked to extend the syntactic descriptions in a way that their algorithms/systems can perform the necessary translation tasks in a semi or fully automatic manner.

We focus on the scenario of purchasing goods using a simplified version of the RosettaNet specification. While the external interfaces must follow the RosettaNet specification, internally Moon uses a propriety legacy system in which data model and message exchange patterns differ from those of RosettaNet. Participants shall basically enable Moon to "talk RosettaNet" and

[4] `http://www.unece.org/trade/untdid/welcome.htm`
[5] `http://www.ebxml.org/`
[6] `http://www.rosettanet.org/`

implement the Purchase Order receiving role part of the interaction described in the RosettaNet PIP 3A4.

There are three main components taking part in the process are depicted in Figure 1.1:

- Company Blue, which is a customer (service requester) ordering products,
- Mediator which is a piece of technology providing automatic or semi-automatic mediation for the Moon company
- Legacy System of the Moon Company While the external interfaces must follow the RosettaNet specification, internally Moon uses a propriety legacy system in which data model and message exchange patterns differ from those of RosettaNet.

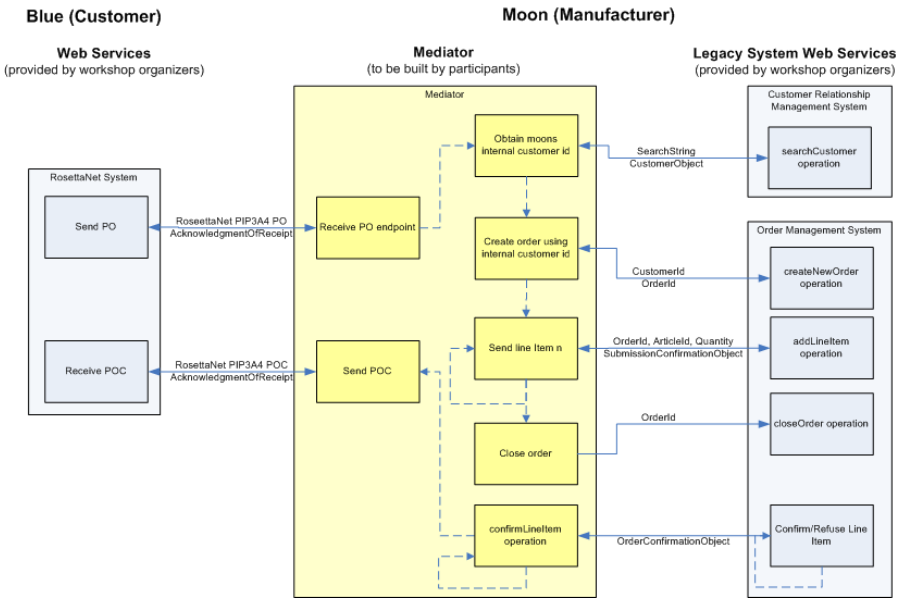

Fig. 1.1. The Mediation Scenario: Process and Data Mediation

The Moon legacy systems and the customer Web Services (Blue) are provided by the challenge organizers and can not in principle be altered (although their description may be semantically enriched). The sketch of the mediator shall be implemented by the participants.

In the mediation scenario, Moon uses two backend systems to manage its order processing, namely a Customer Relationship Management system (CRM) and an Order Management System (OMS). The challenge provides access to both systems through public Web Services described using WSDL.

The scenario describes how Moon has signed agreements to exchange purchase order messages with its client company called Blue using the RosettaNet PIP 3A4 specification.

In order to address integration of Blue and Moon companies, the participating groups are encouraged to use Semantic Web Service technology to facilitate conversation between all systems, to mediate between the PIP 3A4 and the XML schema used by Moon, as well as to ensure that the message exchange between all parties is correctly choreographed. In particular,

- Data mediation is involved in mapping the Blue RosettaNet PIP 3A4 message to the messages of the Moon back-end systems.
- Process mediation is involved in mapping of message exchanges defined by the RosettaNet PIP 3A4 process to those defined in the WSDL of the Moon back-end systems.

The messages used in the challenge are simplified versions of the original specification. To describe context of messages we provide simplified PIP3A4 as RosettaNet XML Schemas. Within the RosettaNet PIP3A4 specification the information is given using a DTD. We have converted this DTD to XML Schema and removed some fields to make the message less complex. Tag names, their meaning and structure have not been changed. The PIP 3A4 enables a buyer to issue a purchase order and to obtain a response from the provider that acknowledges which of the purchase order product line items are accepted, rejected, or pending.

A purchase process is initiated by the buyer when it sends the Purchase Order message to the endpoint exposed by a mediator (this one has to be provided by challenge participants). The Purchase Order message must be synchronously confirmed by an Acknowledgement of Receipt message. The original RosettaNet specification allows 24 hours for confirmation of the Purchase Order Action. We changed it and for the sake of practicability, the Purchase Order Confirmation should be issued no later than 5 minutes since the Mediator has received Purchase Order.

RosettaNet messages contain no specific information about products, but refer only to a global unique product identifier. For the purpose of the challenge we provide a list of products, which can be ordered from Moon. We recognize that a pure identifier remains quite meaningless from the perspective of Semantic Web. Nevertheless we decided not to change existing specification.

In the RosettaNet standard a purchase order is sent using just a single message, however, in order for Moon to be able to process an order, several steps have to be made. The overall ordering process of Legacy System is more complex that the one defined by RosettaNet protocol and the Mediator must take care of this. This process is illustrated in Figure 1.2

First, the Mediator must communicate with the Legacy Customer Relationship Management (CRM) System to obtain relevant customer details. With the data from the CRM system the mediator can assess if the order is eligible, i.e. if the customer is known and authorized to do business with. As

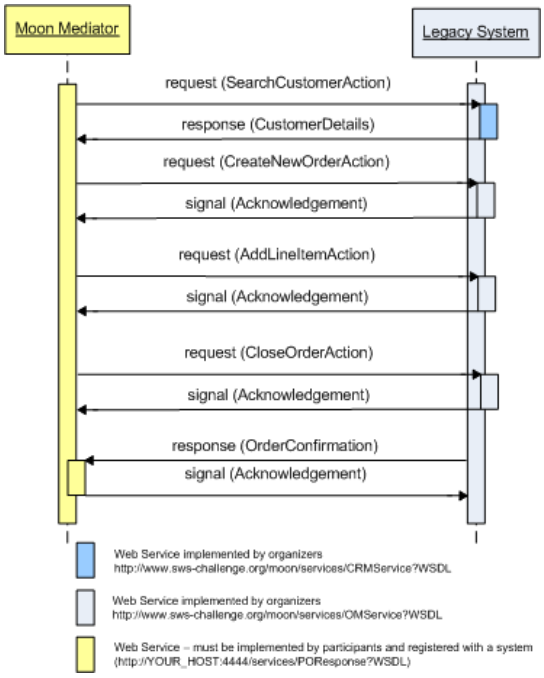

Fig. 1.2. The Mediation Scenario: Legacy System Interaction

a next step a new order must be created with the Legacy Order Management System. Now individual line items can be added to the order created. Once all the line items have been submitted, the order has to be closed. Finally the Order Management System sends a response back containing the products that are on stock and can be delivered. Challenge participants must provide an endpoint for their mediators to which this response can be sent.

Finally the mediator must aggregate all the information received and send it back to the originating party using the Purchase Order Confirmation message of the RosettaNet standard.

In the second phase of the challenge a Production Management (PM) system is added. With the Production Management system products that are not on stock can be scheduled for production. This means the mediator needs to perform an additional step, i.e. to enquiry the production costs and completion dates for every item that is not on stock and confirm production for those where the constraints fit those specified in the original Purchase Order.

1.3 The Discovery Scenarios

The discovery scenarios are orthogonal to the integration/mediation problem. The integration problem can be solved with current syntactic technologies, however it shall be shown how semantic annotation can be used to make this task easier and more flexible. The discovery scenarios - service providers have to be located, selected and invoked dynamically - are more visionary scenarios, since in present business scenarios this task always involves a human in the loop. Two complementary scenarios have been defined, one concerned with shipping of packages, the second dealing with requests to purchase computer hardware.

1.3.1 Shipping Discovery Scenario

The first discovery scenario defines five shipping services (described via their WSDL and natural language documentations). It presents a set of increasingly complex shipping requests. Given a request, a suitable shipper needs to be discovered and invoked. Thus, participants have to create (semantic) descriptions for the available shippers and the given shipping requests such that the discovery and invocation task can be performed by an automated autonomous agent.

Shipping services are characterized by the following properties:

- Operation range: Shippers operate worldwide or in a set of listed countries or continents.
- Package limitations: Shippers define maximum bounds on the dimensions and the weight of packages. Additionally the notion of a dimensional weight is used: Packages with a low weight, but big dimensions need to use the dimensional weight (computed from the dimensions of the package) instead of the actual weight.
- Price: Four shippers statically specify the price as rules how to compute the price of a package depending on shipping location and package dimensions or weight. One shipper requires to dynamically call a Web Service endpoint to gather the current price providing the same information. Thus for goals specifying an upper price limit for the shipping operation, this service could not be discovered by exploiting static descriptions alone, but required dynamic negotiation during the discovery process.
- Package collection: Shippers offer collection of packages and define various constraints on the minimum or maximum advance notice for collection or the total length of the collection interval.
- Shipping time: Shippers specify rules about the maximum shipping time depending on the location of the shipment and the time of the pickup.

Predefined shipping requests specify a required shipping operation, characterized by concrete pickup and delivery addresses as well as concrete package dimensions and weight. The more complex goals additionally specify a

maximum price for the shipping operation or constraints regarding the maximum shipping time. During discovery, participants have to filter unsuitable shippers, automatically choose a suitable one and invoke it by calling the corresponding Web Service endpoint. Since the shipper do not use a common XML-Schema for their messages, participants also have to deal with issues of data mediation to create the properly formatted messages.

One advanced goal requests sending of two packages instead of one. Since none of the shippers support multiple packages, this goal has to be mapped to multiple invocations of the same or different shippers.

Similar to the mediation scenario we provided implementations of all shipping services. The implementation behavior is to be used in case of ambiguity in the textual descriptions. With an invocation of one of the corresponding Web Services an order is triggered which allows the organizer to automatically verify if a particular solution has chosen the right shipment provider.

To illustrate the scenario we present the details of one shipper and a sample goal.

Racer

The rates are composed of a flat fee and a fee per pound different for every continent: Europe($41/$6.75), Asia($47.5/$7.15), North America($26.25/$4.15), Rates for South America like North America, Rates for Oceania like Asia. Furthermore for each collection order $12.50 are added! Racer ships to 46 countries which are listed in its interface specification (WSDL file) The maximum package weight is 70lbs. Racer requires at least a pick-up interval of 120 minutes for collection and the latest possible collection time is 8pm. If a package is collected by 6pm, it is shipped in 2 business days within a country and 3 business days internationally.

Example Goal

One package with dimensions 40/24/10 (l/w/h) (in inch) weighing 40 pounds shall be shipped from an address in California to an address in Bristol, UK. As we can see Racer is one of the shipment providers that match. Others must be excluded since the dimensional weight is either exceeding the specified limits or they are not shipping to the UK.

1.3.2 Hardware Purchasing Scenario

In the second discovery scenario, a customer wants to buy computer hardware with fairly clear requirements on the products to buy. We will provide some examples below. Additionally, three services (called Bargainer, Hawker, and Rummage) are defined, which sell products. Each of the services offers an endpoint that allows to inquire about the products (and their detailed properties) currently on stock. Like in the first scenario, the task is to select the right

service and invoke it with the right input parameters to purchase the products that best match the customer's expectations. The hardware purchasing scenario was designed to extend the shipping scenario along three dimensions of difficulty.

- Currently, the available services offer 19 products alltogether which are identified by a global product id (GTIN). Clearly more realistic services offer way more different products. It may or may not be feasible to specify all different options and all the product details in the offer description(s). Solutions to the scenario should indicate how they attempt to address this issue in more realistic scenarios with hundreds of products available. For the future, it is planned to extend the scenario in this direction.
- Some requests contain competing preferences as is usual for realistic matchmaking: price should be as low as possible, processor power, hard disk drive size and memory size should be as big as possible. The scenario request definitions clearly define rankings among such competing preferences. The semantic task is to represent these ranking rules clearly and execute them.
- The scenario requests involves increasingly difficult requirements of basic service composition:
  1. Uncorrelated composition: Some requests ask for several products that may or may not need to be purchased from different providers. Thus, a single request needs to be mapped to multiple invocations of the same or different services.
  2. Correlated composition: Some requests ask for several products but not all possible pairings of requested products are compatible to each others. Making a choice for one product may limit the choices for the remaining products to purchase or even make it impossible to fullfil the goal.
  3. Composition with global optimization goal: Some requests ask for several products with global optimization goals and constraints. A power minimum or a price maximum are examples of constraints that should not be violated for the total order. Therefore, like in the previous case, products can not be chosen independently of each other.

Example Goals

We illustrate the scenario by two exemplary goals.

Goal B2

Purchase a 13 inch Apple MacBook with a 2.0 GHz Intel Core Duo processor. It should have at least 1 GB RAM and at least 100 GB HDD. The price should be around $ 1500, at the very most $ 1800. If the white version is significantly cheaper than the black one (at least $ 100) buy the white one, otherwise buy the black version.

The resulting preferred solution is a white MacBook for $ 1449 by Bargainer. Another, albeit less preferred, solution is a black MacBook for $ 1699 by Rummage.

Goal C4

Purchase a 13 inch Apple MacBook with at least 2.0 GHz Intel Duo Core Processor, 512 MB RAM and 80 GB HDD. Additionally buy a web cam for notebooks with a resolution of at least VGA (640*480) and a 13 inch notebook sleeve. The total price must not exceed $ 1750. As long as the price limit is satisfied, choose the better product: The processor power of the notebook is most important to me. Besides that I rather need more RAM than a bigger HDD. If possible prefer webcams with a higher resolution.

The resulting solutions are as follows: The MacBook can be purchased by Hawker or Bargainer (preferred since better product). The products offered by Rummage either have not enough processor power or are too expensive after the web cam is added. The web cam needs to be purchased from Rummage since other web cam offers either do not specify a resolution or the specified resolution is too low. Hawker is the only service that offers sleeves.

1.3.3 Status and Future of the Discovery Scenarios

As of now there are two comprehensive scenarios related to service discovery and matchmaking. The first, original scenario, involves the discovery of an appropriate shipment service out of five offers, each with different peculiarities regarding price, supported locations, maximum package weight, constraints on the pickup time and the speed of delivery. A second scenario deals with purchasing computer hardware from a set of available vendors. The task is to determine which combination of products suits the needs of the client best and then to invoke the various vendor services properly to purchase the desired products.

Based on a hierarchy of increasingly difficult given goals for both scenarios (i.e. shipping and purchasing requests), submitted solutions are evaluated by determining the the problem levels that they are able to solve. For the first discovery scenario the problem levels are as follows:

1. discovery based on location,
2. discovery with arithmetic price and weight computations,
3. discovery including request for quote,
4. discovery including sending multiple packages (which had to be resolved to multiple service invocations), and
5. discovery with temporal semantics, i.e. pickup times and required speed of delivery.

The problem levels for the second discovery scenario have been defined as follows:

1. discovery based on clear product specifications,
2. discovery including (competing) preferences (like as cheap as possible),
3. discovery for multiple products that must be resolved to multiple service invocations,
4. discovery for multiple correlated products (like a notebook and a compatible docking station),
5. discovery for multiple products with a global optimization goal (e.g. overall minimal price), and
6. discovery for multiple products with a global optimization goal and preferences.

Submissions for new goals and also completely new scenarios that extend the coverage of the complete problem space are encouraged any time. Such submissions will be evaluated by the challenge organizers to become part of the official SWS-Challenge test bed.

For the near future, two extensions to the scenarios are already planned. On the one hand we will add goals that require automated unit conversion to either of the discovery scenarios. This might e.g. be done by mixing products with a price stated in Dollars with products with a price stated in Euro in the purchasing scenario. Participants will have to detect that prices are given in different currencies and develop means to deal with this, e.g. by automatically invoking a currency conversion service during service matchmaking. This will be one further step towards really adaptable systems.

On the other hand we are currently working to include a realistic number of products into the supplier scenario. We are investigating whether it is possible to exploit the Amazon E-Commerce service to gather the necessary amount of realistic product data. Including a large number of products into the scenario will have major implications on the solutions. First, creating meaningful descriptions will become much more difficult. A broad generic description in the sense of "this service sells electronic products" will be of little use during discovery. On the other hand it might not be feasible to explicitly list all available products within a description for various reasons (privacy, dynamicity, . . .). Thus participants will have to balance their solution somewhere between these extremes, decide on the amount of statically encoded information versus the amount of information being dynamically gathered, and provide means how to integrate dynamic information into service descriptions and service matchmaking algorithms.

The first of the two planned extensions is targeted at increasing the complexity of the discovery problems at the process and reasoning level. Solutions being able to still tackle the problems will have proven an even higher level of adaptability to homogenous environments.

The second extension is complementary and increases the complexity with respect to the amount of information that needs to be processed and finally taken advantage of during discovery. Both extensions combined are aiming at making the discovery scenarios even more realistic than they already are,

thereby underlining the goal of the SWSChallenge to provide industrial level application scenarios.

1.4 Emerging Orchestration Scenario

The Orchestration Scenario aims at covering yet another aspect of the comprehensive SWS problem landscape. The aim of the scenario is to show how semantic Web technologies can contribute in overcoming problems of Web Service orchestrations. Although envisioned as a successor of the mediation scenario it will further challenge the problem solving capability domain of technologies used by participants.

After decision was made that the scenario will be in area of orchestrating various services in order to initiate purchase order payment existing solutions coming from financial market were evaluated. Unfortunately, RossettaNet used in preceding scenario doesn't provide support for communication with financial institutions (e.g. banks) in order to conclude purchase order with a payment. After some time spent in research it was identified that the gap between RossettaNet enabled systems and financial institutions could be bridged with a solution relying on ISO 20022 UNIversal Financial Industry (UNIFI) message scheme standard[7]. It is supported by major players in financial market (e.g. SWIFT[8] and TWIST[9]) and it provides common development platform for exchanging and processing financial messages encoded in a standardized XML. The standard covers wide range of possible cases found in respective domain (like Cash management, Payments Clearing and Settlement, Securities management, Trade Services, etc). Among them, especially interesting for the scenario, was Payments Initiation[10] case describing set of messages used to initiate and manage funds transfer between debtor (or customer) and creditor (or seller). This scenario uses simplified versions of messages than the messages provided in original specification. All message definitions are given as appropriate XML Schemas.

Like in the mediation scenario there are three main components participating in the conduction of payment initiation as depicted in Figure 1.3:

- Company Blue, which intends to pay for ordered goods,
- Integrator, which is a piece of technology providing capability to orchestrate a number of services on behalf of Blue company, and
- Moon Company, offering interface to retrieve creditor data (e.g. bank account number) needed to successfully complete payment initiation.

[7] http://www.iso20022.org
[8] http://www.swift.com
[9] http://www.twiststandards.org
[10] http://www.iso20022.org/index.cfm?item_id=59950

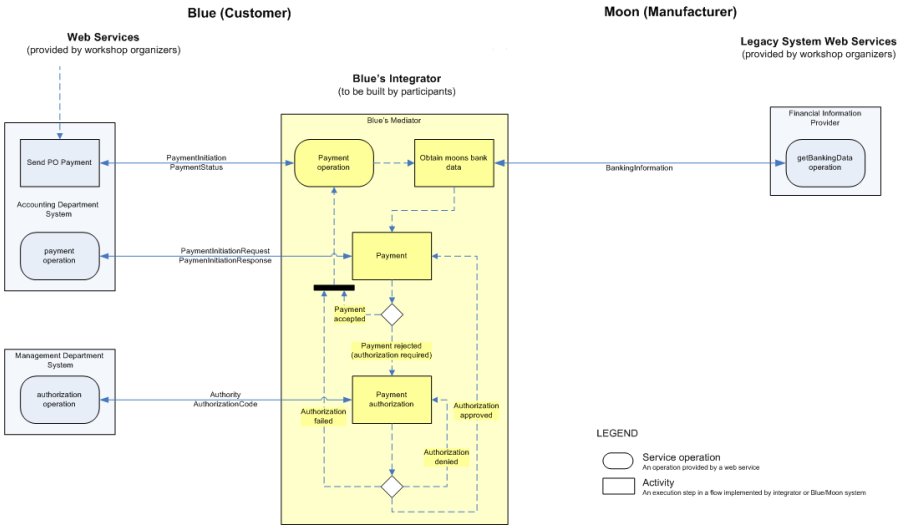

Fig. 1.3. The Orchestration scenario overview

The scenario description started where the mediation problem description ended, i.e. after reception of acknowledgment for a purchase order initiated by Blue. Intention of Blue to pay for the approved purchase order results in a payment instruction message sent to its Accounting Department system (i.e. Accounting Department service). The payment instruction must be completed with necessary data (Blue's and Moon's addresses, bank accounts and identifiers, purchase order amount) coming from various sources (e.g. Moon's Financial Information Provider service). Under certain circumstances payment instruction must be authorized by Blue's Management Department system (i.e. Management Department service). After payment instruction has been dispatched customer expects to receive a payment status report from its Accounting Department system. A solution should play the role of Blue's integrator concerned with the proper orchestration of before mentioned services.

Condition governing Integrator's need to consult Management Department system for payment authorization are based on the *threshold* amount. If the requested payment amount is below *threshold* there is no need for communication with the Management department, thus payment initiation orchestration can skip this step. Otherwise, Accounting Department will reject payment instruction, unless accompanied with an optional authorization code obtained by making a request to the Management Department service. The request must designate an Authority (i.e. a Blue employee) capable of approving or denying payment requests up to the specified amount. This system returns either an authorization code (after which payment initiation messages can be completed and provided to the Accounting Department) or a denial code. If a

denial code is returned, the service may be questioned again, but not with the same Authority as in previous call. Furthermore, Blue has a policy that the least senior Authority, as determined by the amount of money up to which an authorization could be made, should be requested first.

Like in the case of the mediation scenario challenge organizers provide a set of services which can not be altered by participants but which descriptions can be semantically enriched by participants. All services representing Blue and Moon systems are publicly accessible and described by accompanying WSDLs. It is expected that the Integrator component will be implemented by the participants.

To illustrate the scenario we will assume that the *threshold* amount is €2000, and that we have two authorities, Jackie Brown (authorizes amounts up to €5000), and Cathy Johnson (authorizes amounts up to €10000). Furthermore, in order to achieve compact description of service invocations we will annotate Blue's Accounting Department service as AD, Blue's Management Department service as MD and Moon's Financial Information Provider service as FIP.

Payment amount below *threshold*

In the case of payment amount below *threshold* there is no need to request payment authorization from MD, thus the Integrator can directly invoke AD service after successful compilation of the payment initiation message. Correct invocation sequence orchestrated by the Integrator should be FIP → AD.

Payment amount above *threshold*

Payment amount greater than *threshold* adds additional step in expected orchestration, i.e. the Integrator must consult MD to authorize payment initiation. If we assume that payment amount is €3000 than expected invocation sequence could be FIP → AD → MD → AD (FIP is contacted to gather Moon's financial data, AD refused to initiate payment because payment amount is above *threshold* and authorization code is missing, MD is contacted with Jackie Brown as least senior designated authority who approves payment and gives authorization code, and AD is contacted again with all necessary data needed to complete payment initiation). The expected service invocation sequence could also be FIP → AD → MD → MD → AD (if Jackie Brown as least senior authority refuses to give payment initiation approval the Integrator is contacting MD again but with Cathy Johnson as the next authority in hierarchy which decides to authorize payment).

1.5 Summary

The above described scenarios provide the current set of challenge problems. They are intended as common ground to discuss semantic (and other) Web

Service solutions. By providing implementations to every scenario we want to ensure that solutions are close to the real world and participants cannnot cheat with respect to the set of features supported by their solution.

Finally we want to emphasize that the SWS Challenge is open, not only to participation but also to the submission of new scenarios that extend the current coverage of problems.

Such scenarios can be "stand alone" or refine and extend the "Blue Moon" customer/company/supplier/shipper scenario. Eventually, this scenario should include the company fulfilling the customer order using a supply chain composed of the best suppliers and shippers for the specific customer order.