

Personal eBanking solutions based on Semantic Web Services

Oscar Corcho , Silvestre Losada , Richard Benjamins , José Luis Bas , Sergio Bellido

Intelligent Software Components, S.A.
C/ Pedro de Valdivia, 10. 28006 Madrid (Spain). Tel: +34913349797, Fax: +34913349799
{ocorcho, slosada, rbenjamins}@isoco.com
Fundación de la Innovación. Bankinter
Paseo Castellana, 29. 28046 Madrid (Spain). Tel: +34916234238
{jlbas, sbellidog}@bankinter.es

Abstract. We describe how Semantic Web Service technology can be used for the provision of personal e-banking online services. We describe two deployed applications: an overdraft notification service and a mortgage comparison service. The former accesses the bank accounts of a user as well as utility goods Web sites where invoicing information is stored and estimates whether the user will be in an overdraft situation in the near future, alerting him/her by e-mail or SMS. The latter accesses the mortgage information provided by the heterogeneous Web sites of different banks and allows users to compare them according to different types of criteria. The chapter not only focuses on the technological decisions followed to implement and deploy these services, but also on the added value of applying Semantic Web Services for them.

1 Introduction and motivation

The Internet has revolutionized our lives in areas such as communication (very low cost, large scope), online business (transactions), and access to content (millions of resources, irrespective of location and language). Also in our “financial lives”, the Internet has provoked significant changes: instead of going to a physical bank branch, we can go to the bank’s web site and make many different types of transactions.

There are important differences in the use of Internet banking between different countries [2], with a complex set of factors that influence adoption, such as access technology and infrastructure related factors, sector-specific Internet banking factors, and other socioeconomic factors. Most financial institutions allow their clients to access their accounts and consult their account information via Internet. Moreover, most of these institutions allow their clients to make transactions via Internet. In both cases, several options are made available for clients, ranging from regular bank services to associated services, such as information about grants, e-commerce services, mobility, shopping or on-line payments services, and so on.

In general, the main processes covered in banking operations can be classified into three categories:

- o **Inter-banking processes.** They refer to the exchange of documents or account entries (cheques, receipts, international and national transfers). For example, when a Bank pays a cheque from another bank, a movement of funds is produced from the second Bank (the payer) to the first (the payee). This movement of funds is always accompanied by information on the operation (e.g., the cheque number) and, in some cases, the document itself.
- o **Bank-provider processes.** This refers to basic supplies common to any industrial sector (paper, IT equipment, software, office furniture, etc.), with the sole exception of those information providers that are specific to banking (defaulter registry, real estate appraisal entities, etc.).
- o **Bank-client processes.** This refers to product sale processes and service usage processes through the different channels made available by the bank. Here, we must include as well the internal operations of the Bank, since they make the relationship with the client possible.

From another perspective, the banking business can be also divided according to the following categories: products, services and channels.

- o **Products.** They are contractual operations that involve the deposit of money (accounts, mortgages, deposits, investment and pension funds, etc.) or money loans (credit, loans, mortgages, guarantees, etc.).
- o **Services.** They are operations that involve the entry or exit of funds from a banking product (i.e.: credit cards, cheques, promissory notes, receipts, transfers). In a broad sense this category would include every kind of service that is offered by the bank within its corporate purpose, even if it may not be purely related to banking, such as virtual stores, Internet services, etc.
- o **Channels.** They are the means through which the bank reaches its clients: Branches, Telephone Banking Services, Internet, Agents, Commission Agents, etc. The most common situation to date is that the non-traditional channels offer only services. However, it is increasingly becoming more frequent that they also offer the possibility of contracting products.

Let us focus on the provision of products and services using the Internet channel, that is, on e-Banking. Internet banking services are considered as a cost-effective delivery channel, driven by cost reduction, market share increase and customer retention targets. However, profitability still remains a challenge. In line with this, there are still many fields subject to technological improvements. Within these, those most capable of being improved are the ones that fall in line with the requirements of the financial institutions and objectives, and usually are:

- o Those relating to task-performing by people. They consume valuable time and resources that may be used for marketing.
- o Those relating to cost reduction. One of major commitments of banks is efficiency in terms of competitiveness. The prices of the banking products and services are marked by the cost plus a differential (profit margin).
- o Those relating to new products/services/quality. This is highly related to the strategic position based on the differentiation in the market.

In this situation, the eBanking sector provides good opportunities for the deployment of Semantic Web Service technology. These opportunities are mainly related to those products and services that currently have a large complexity (involving the consumption of a large amount of time and human resources), and consequently are costly, and to those that have a large market potential (related to the provisioning of new added-value products and services that can differentiate the bank position in the market).

In this chapter we present two software prototypes that fall into the two previous categories: a mortgage comparison service (as an improvement of a service with a large complexity) and an overdraft notification service (as the provision of a new added-value service for bank clients. In both cases the services are deployed using Semantic Web Service technology. The application of such technology results in significant improvements and attractive return of investment (ROI).

The rest of this chapter is structured as follows: section 2 describes the mortgage comparison service, providing the service background as a set of general requirements, showing its architecture and describing in detail some of the aspects of its implementation and deployment. Section 3 describes the overdraft notification service, following the same structure as for the other one: general requirements, architecture and details of implementation and deployment. Section 4 provides some conclusions to this chapter, with a focus on the technological requirements that can be derived from the implementation of both applications: data mediation, service discovery and composition, invocation, etc., which will have to be considered in the future by Semantic Web Service research. Finally, section 5 will describe our expected future trends in the application of this and other similar technology in the e-banking domain.

2 Mortgage Comparison Service

The mortgage application process requires a large amount of effort in time and resources, mainly due to the need for the compilation of information from external sources. Figure 1 graphically depicts this process, which can be described as follows:

- 1 The client looks for a property in accordance with his/her requirements and which a priori considers he/she can deal with. The search for information is outside the scope of the financial institution service offering and, consequently, it can help little or nothing.
- 2 Once a market search has been done, the client makes his/her calculations.
 - a. Which type of mortgage can I deal with?
 - b. What are my current resources?
 - c. How much of my monthly income can I set aside for the payment of the loan to acquire a property?

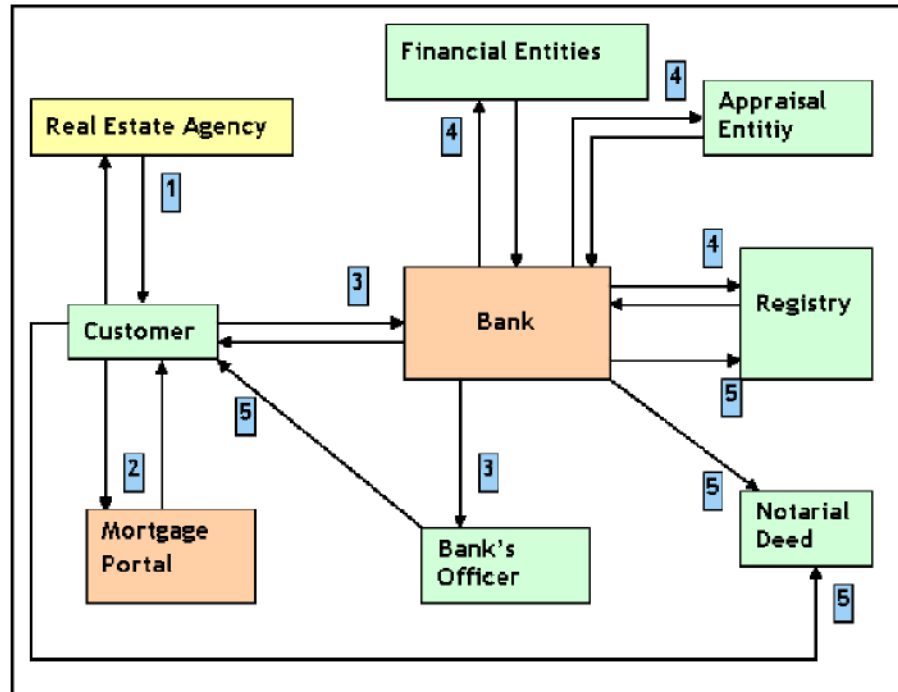


Figure 1. A simplified view of the mortgage application process.

This type of information can be provided by a mortgage simulator. Mortgage simulators follow a mechanism outside any financial institution (let's say that it follows the principle of an abstract mortgage). The simulator should be able to respond to the previous questions. Using a simple form that takes into account the client's income, it should be able to respond with the maximum amount that may be paid (taking into account a maximum depreciation time), or the number of instalments that should be paid if he/she wants to pay a fixed amount of money. A standard, fixed, reference can be used to make these calculations.

Once the reference information is obtained, the client knows the type of mortgage that can be contracted. Now, the client needs to know the mortgage offer which adapts best to his/her possibilities (or tastes). For this, information is obtained from different banks on the mortgage offers they provide. This data gathering can be done online and automatically, accessing a mortgage comparator. Given reference data, mortgage comparators calculate the amounts, the terms and conditions typified in the bank offers, and provide clients with an ordered list (in accordance with order parameters that the client can modify) of mortgage offers.

- 3 Once the mortgage offer to be contracted has been selected, the client contacts the bank providing the offer. The bank starts a client data gathering process, in which the client itself is involved.

- 4 During the client data gathering process, the bank obtains as much information as possible on the property to be acquired. This information gathering process is made by contacting different types of entities, such as appraisal entity agencies, public registries, notaries, etc. The bank also contacts other financial entities and its own records to retrieve information about the client.
- 5 Once all the necessary data has been gathered, it is necessary to make a decision on whether or not to provide the mortgage to the client. In case that the decision is positive, all the relevant information will be sent to the notary and the contract between the client and the financial entity will be signed.

The service described in this section is related to the second step in the mortgage application process: mortgage comparison and simulation.

The current situation is as follows: the client obtains information about the mortgages offered by different banks or building societies following a complex and tedious process: he/she is forced to access the Web sites of different organisations, find the relevant links inside them and perform a manual comparison taking into account many different factors, revealed by the heterogeneity of the existing mortgage offers (different vocabularies, different types of fees, etc.).

To alleviate the burden of performing such task, some financial institutions already offer comparison services¹ to their customers. The comparisons offered by these services are based on information that is input manually by persons (hence error prone and not always up to date) or using costly, and sometimes brittle, screen-scraping technologies. Furthermore, these services do not usually solve the heterogeneity problem highlighted above and hence are not so useful in the decision making process made by clients. Similarly, simulation services are currently accessible in many bank Web sites for the mortgages that they offer². These services are more common than those aimed at comparing mortgages. Third parties, such as intermediary webs sites that recommend mortgages, normally perform the simulations and access the results using screen-scraping techniques.

The short-term benefit of applying Semantic Web Service technology in mortgage comparison and simulation is that human intervention is reduced in the process (this means a cost reduction in the maintenance of these services) and that data reliability increases.

Furthermore, as a middle-term result, new added value services and market opportunities may be generated with the optimum development of applications based on the use of semantics. For instance, new kinds of mortgages may appear, which are rarely taken into account nowadays due to high costs/human-task mediation required (e.g., small amount operations in which the costs are far higher than the incomes). Besides, other new services may appear, which make use of the new banking service and may mediate in the sale of mortgages. In this case, there would be a market opportunity for

a bank, not only due to anticipating these intermediaries, but also its own competition. The bank could additionally become a wholesaler for these intermediaries.

2.1 Architecture of the Mortgage Comparison Service

As aforementioned, this service aims at easing the task of mortgage comparison by providing a simple interface where users specify constraints about the type of mortgage that they want, and receive back the information in a homogeneous way.

The main difference between this service and others currently available (as the ones whose URLs have been provided above) is that each time a client makes a request to the service, the data used for the comparison will be obtained on-line from each bank or building society, so that the information is never outdated. We rely on the existence of Web services in each of the accessed bank Web sites that can provide the mortgage simulations with the data provided by the client (be it modified or not according to the Web service requirements).

Furthermore, the different types of results will be homogenized, and filtered, in the user interface, so that the comparison can be performed according to the same set of parameters instead of those provided by each of the mortgage providers (since there is few standardization on the content of what a mortgage is).

Figure 2 summarises the architectural components needed to perform the operations of the mortgage comparison service. The architectural components in the middle box are based on the architecture defined in the context of the DIP project³, which is the architecture on top of which the mortgage comparison service is deployed⁴. Though it is not the purpose of this description, we will briefly explain the role that those components play in the context of the DIP architecture and in the context of this service:

- o The *discovery component* is used to find the services suitable to perform an action. In DIP the service discovery process is focused on finding services that comply with an abstract service description (that is, a set of concrete service instances that share a well-defined set of characteristics). In the context of the mortgage comparison service, service discovery is used to determine which of the services published by different banks provide simulations of mortgages that comply with the constraints specified by the users.
 - o The *mediation component* is used to overcome the communication problems that may arise between the service requestor and the service provider during the service invocation, due to the use of syntactically different data formats and different vocabularies to describe the data. In the context of the mortgage comparison service, service mediation will be needed to solve the problem of heterogeneity in the parameters used as inputs and outputs by each of the mortgage simulation services, which includes data formats (e.g., different data types for expressing
-

dates, currency amounts, etc.) and vocabularies (e.g., different representations of what quota means, differences in types of fees applicable, etc.).

- o The *invocation engine* is needed to provide a platform for the execution of the services. It makes it possible to execute the Web services to which the comparison service accesses and obtain the execution results from them.
- o The *repositories* are used for different purposes: the goal repository is used by the comparison service in the construction of the goal expressed by the client when a mortgage request is done, that is, it contains goal templates that are filled-in by the service; the Web service repository is used as a registry of the Web services that are available, and is used during the discovery process; finally, the ontology repository maintains the ontologies (vocabularies) used by the different Web services and is used during the task of ontology mediation.

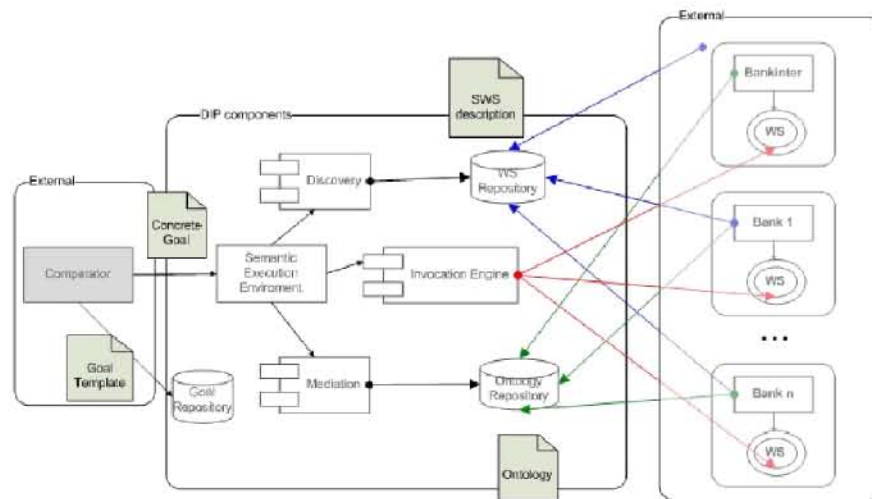


Figure 2. Architecture of the mortgage comparison service in relation to the Web services contacted and the components needed from a Semantic Web Service execution environment.

2.2 Implementation and Deployment of the Mortgage Comparison Service

The service has been implemented⁵ according to the architecture described in section 2.1. It is deployed on the WSMX execution environment⁶, which is one of the two platforms that realise the WSMO architecture (the other one being IRS-III [4]).

We will show now some examples of the goals and service descriptions that are used by this service. These descriptions are available in the WSML language⁷, which is the one proposed by the WSMO framework and used by the WSMX platform.

Goals denote what a user wants as the result of the Web Service invocation. Figure 3 shows a goal expressed in WSML, more specifically the goal used to obtain the number of payments of a mortgage given the total amount to be lent and the foreseen quota to be paid. First of all, we can see the namespace declaration, which provides the namespaces used in the goal description. Then some non-functional properties of the goal are provided, such as title, type, date, language, etc. Then the ontologies used in the goal description are provided (in this case a financial ontology developed for Bankinter, which contains not only terms related to the mortgage application process, but also other aspects related to the financial domain).

Finally, the capability is expressed, describing the restrictions on the information elements that the user wants to get from the service (preconditions, posconditions) and the state of the world (assumptions, effects).

Preconditions describe restrictions that have to be true before invoking the service. In the case of this goal, the loan capital has to be known and less than 200000, the quota has to be less than 900, and the maximum interest rate has to be 0.5 over the reference rate.

Postconditions describe restrictions that have to be true after invoking the service. In the case of this goal, the mortgage requires having a life insurance contract, but not a home insurance contract, and the opening commission will be less than 0.7% of the total loan capital.

Finally, since there are no changes in the real world as a result of the execution of a mortgage simulation, the goal expressed in figure 3 does not contain assumptions nor effects.

```

namespace {
  _ "http://users.isoco.net/~slosada/ontologies/bankinter/GoalGetNumberOfPayments.wsml#",
  dc _ "http://purl.org/dc/elements/1.1#",
  foaf _ "http://xmlns.com/foaf/0.1/",
  xsd _ "http://www.w3c.org/2001/XMLSchema#",
  wsmi _ "http://www.wsmo.org/2004/wsmi#",
  fin _ "http://users.isoco.net/~slosada/ontologies/bankinter/FinancialOntology.wsmi#"}

goal _ "http://users.isoco.net/~slosada/ontologies/bankinter/GoalGetNumberOfPayments.wsml"

  nfp
    dc#title hasValue "Goal to find mortgage simulator with value restrictions"
    dc#type hasValue _ "http://www.wsmo.org/2004/d2#goals"
    dc#description hasValue ""
    dc#subject hasValue {"Simulator", "Mortgage", "Financial", "Product"}
    dc#date hasValue _date("2005-03-07")
    dc#format hasValue "text/html"
    dc#language hasValue "en-US"
    dc#rights hasValue _ "http://www.isoco.com/privacy.html"
    wsmi#version hasValue "$Revision: 10 $"
  endnfp

importsOntology { _ "http://users.isoco.net/~slosada/ontologies/bankinter/FinancialOntology.wsmi#" }

capability
  sharedVariables ?mortgageLoan

```



```

precondition
nfp
  dc#description hasValue "The input has number of payments, type of interest and
  mortgage amount to simulate mortgage result is a monthly payment."
endnfp
definedBy
?mortgageLoan memberOf fin#MortgageLoan
[ loanCapital hasValue ?capital,
  initialQuota hasValue ?quota,
  interestRateType hasValue ?interest]
and ?capital < 200000
and ?quota < 900
and ?interest memberOf productRateApplicationVariable[interestRateValue hasValue 0.5,
  referenceType hasValue _# ].

postcondition
nfp
  dc#description hasValue "Result of webService is a list of mortgage. The person who wants
  contract this mortgage must have life insurance."
endnfp
definedBy
?mortgageLoan memberOf fin#MortgageLoan
and ?mortgageLoan [
  term hasValue _#,
  openingCommission hasValue ?opCommission,
  lifeInsurance hasValue true,
  homeInsurance hasValue false ]
and ?opCommission < 0.7.

```

Figure 3. Goal to find a mortgage simulator, given a known loan capital, the interest rate required and the maximum monthly payment.

Similarly, **Web Service** descriptions also consist of restrictions about their preconditions, postconditions, assumptions and effects. The description presented in figure 4 provides information about a mortgage simulator Web service that is available from an external financial institution. It contains, as in the case of the goal description from figure 3, a set of preconditions and postconditions, but not assumptions or effects, since the mortgage simulation itself does not have any impact on the external world.

From this description we can conclude that the service is compliant with the goal presented in figure 3, and consequently would be discovered by a discovery service.

```

namespace { _ "http://users.isoco.net/~slosada/ontologies/bankinter/WSGetMortgageCapital.wsmI#",
dc _ "http://purl.org/dc/elements/1.1#",
foaf _ "http://xmlns.com/foaf/01/",
xsd _ "http://www.w3c.org/2001/XMLSchema#",
wsmI _ "http://www.wsmo.org/2004/wsmI#",
fin _ "http://users.isoco.net/~slosada/ontologies/bankinter/FinancialOntology.wsmI#"

webService _ "http://users.isoco.net/~slosada/ontologies/bankinter/WSGetMortgageCapital.wsmI"

nfp
dc#title hasValue "Web Service that is a mortgage simulator"
dc#type hasValue { "http://www.wsmo.org/2004/d2#webservice" }
dc#description hasValue ""
dc#subject hasValue { "Simulator", "Mortgage", "Financial", "Product" }
dc#date hasValue _date("2005-03-30")
dc#format hasValue "text/html"
dc#language hasValue "en-US"
dc#rights hasValue { "http://www.isoco.com/privacy.html" }
wsmI#version hasValue "$Revision: 1.0 $"
endnfp

importsOntology { _ "http://users.isoco.net/~slosada/ontologies/bankinter/FinancialOntology.wsmI#"

capability
sharedVariables ?mortgageLoan
precondition
nfp
dc#description hasValue "The input has number of payments, type of interest and mortgage amount to simulate mortgage result is a monthly payment."
endnfp
definedBy
?mortgageLoan memberOf fin#MortgageLoan
[ term hasValue ?term,
initialQuota hasValue ?quota,
interestRateType hasValue ?interest
and ?term [ totalTerm hasValue ?totalTerm,
typeTerm hasValue MONTH ]
and ?totalTerm < 300
and ?quota > 600
and ?interest memberOf productRateApplicationVariable
[interestRateValue hasValue 0.5,
referenceType hasValue _# ].

postcondition
nfp
dc#description hasValue "Result of webService is a list of mortgage. The person who wants contract this mortgage must have life insurance. "
endnfp
definedBy
?mortgageLoan memberOf fin#MortgageLoan
and ?mortgageLoan [
loanCapital hasValue _#,
openingCommission hasValue ?opCommission,
lifeInsurance hasValue true,
homeInsurance hasValue false ]
and ?opCommission < 0.7.

```

Figure 4. Web service description for one of the mortgage simulators.

3 Overdraft Notification Service

People normally have several bank accounts, with the same or different banks, where they have different amounts of money. Besides, they normally have contracts with consumer goods companies, such as telephone companies, gas and electricity provid-

ers, broadband providers, etc., whose bills are charged to any of the bank accounts that they have.

Account aggregation has been identified as one of the key services that will have to be provided by financial institutions in the future [2]. Currently, account aggregation tools exist, such as GetSee®⁸, which perform this task, normally by means of screen-scraping techniques, and are available for customers in many bank Web sites.

The overdraft notification service described in this section goes a little bit further, and can be seen as one of the many added value services that can be created on top of account aggregation tools. This service detects whether any of the customer accounts is going to be in an overdraft situation, taking into account estimations of the next invoices that will be sent by consumer good companies to the customer's accounts. The service notifies the customer if the balance of the account is less than the expected invoice amount or is under a specific offset, using any of the notification channels available for the customer. Hence the customer can perform the corresponding transactions in order to avoid that situation⁹.

Figure 5 describes the typical operational scenario for this service. In this scenario, the following actors are involved: the customer, the banks, and the consumer goods companies. And the following services are involved: customer notification agent (CNA), Sentinel and some estimation services. Finally, the iSOCO GETsee® application is at the core of this scenario, in charge of the aggregation of data from bank accounts and consumer goods companies.

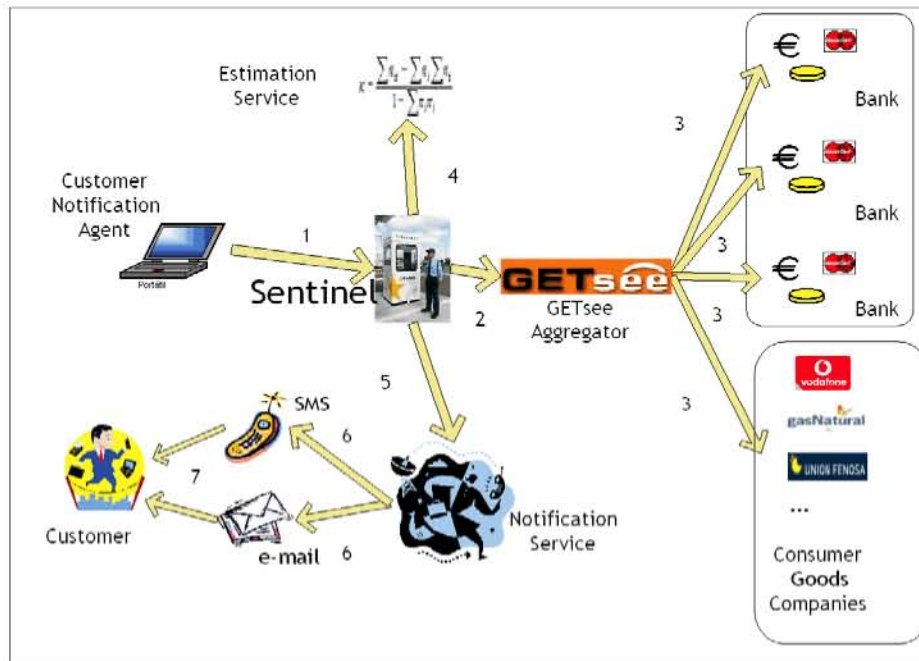


Figure 5. Sample scenario diagram for the Overdraft Notification Service.

The following steps will be normally done:

Step 1: Everyday, the Customer Notification Agent dynamically configures and invokes the Sentinel Service. This agent has the entire customer's information needed for invoking the composed service (online username, password and other data). The update frequency of this agent can be customized.

Step 2: The Sentinel Service uses iSOCO GETsee® for collecting information from the customer's accounts.

Step 3: iSOCO GETsee® collects the amount balance of all the customer's accounts (of banks B1, B2, ..., Bn). In one (or more) of this accounts some consumer goods companies (E1, E2, ..., En) can charge invoices. The invoices have their notification and value dates. The frequency of those invoices is always the same (weekly, monthly, bimonthly, annually).

Step 4: For each invoice of consumer goods companies (E1, E2, ..., En) associated with the account, the Estimation Service estimates the probable amount at the end of the period, Ae (estimated amount) in terms of heuristics or mathematical models. Ae has a relationship with a consumer good company (Ee) and an account of a bank (ABe). If the Ae is less than the established threshold for the account, then an alert has to be raised.

Step 5: The Notification Service looks in a (de)centralized registry different ways to communicate with the user. It can find different services involving many different devices (phone calls using VoIP, SMS, electronic mail, telegram) and personal data (phone number, cell phone number, e-mail, postal address). The services discovered must have the ability to perform the action defined in the Notification Service.

Step 6: The invocation engine sorts in terms of cost, time to deliver, etc., the different possibilities and chooses the first service in this particular ranking. Some data mediation could be needed if terms of the ontology used differ from the one used by the Notification Service. If the service chosen has an irrecoverable mismatching of process or data, or some communication error occurs in the invocation, the service has to be able to choose another service and invoke it.

Step 7: The service chosen is invoked and the user is notified.

3.1 Architecture of the Overdraft Notification Service

The general architecture of this service is shown in figure 6, which resembles the top-level diagram already presented in figure 5. In this case we have opted for presenting the Semantic Web Services involved in the scenario instead of providing details about the components of the architecture that are used at each time, since this is quite similar to what was presented for the mortgage comparison service.

The figure shows that there are three main services that are executed at some point in time during the execution of the service. These are the GETsee service, in charge of the aggregation of accounts from different bank and consumer good companies, the estimation service, in charge of providing estimations of the amounts of the invoices that will be sent to each bank account by the consumer good companies, and the notification service, in charge of notifying customers about their possible overdraft situation.

Furthermore, the GETsee Service is decomposed into five atomic services (openSession, getAccounts, getInvoices, getBalance, closeSession). These five services are annotated using the same ontology as the GETsee service (although this is not mandatory in our approach). Those atomic services invoke other services, which are annotated according to other ontologies. In these cases, data mediation is needed for the exchange of messages. At last, the Notification Service looks for a service able to notify something to a person and finds at least two services (notification by SMS and notification by e-mail), which might be annotated according to other two more ontologies.

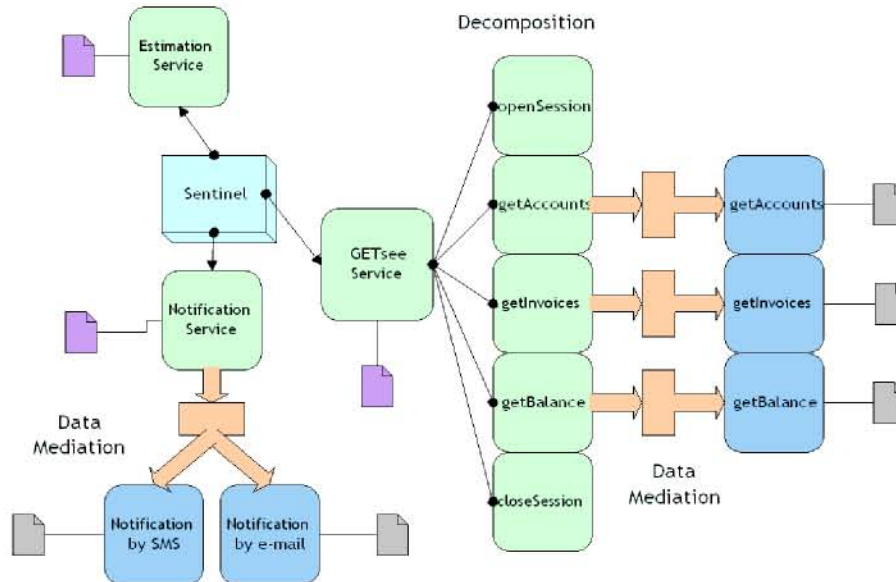


Figure 6. A diagram of the Semantic Web Services used for the notification scenario.

3.2 Implementation and Deployment of the Overdraft Notification Service

Unlike the mortgage comparison service presented in the previous section, this service has not been implemented on top of any existing Semantic Web Service execution platform, such as WSMX, IRS-III or the OWL-S [8] virtual machine, but using an ad-hoc approach. The reason for this decision is that this service was implemented and deployed when those platforms were still in an unstable stage of their development. However, it would be easy to adapt the service implementation to any of them since it follows a similar approach to that of those execution environments (using service discovery functions, data and ontology mediation functions, etc.), and the services are semantically described in a similar way to that required in those frameworks.

From all the processes involved in this service, the discovery process deserves special attention, since it is different to the one presented for the previous service on mortgage comparison. The discovery process works as a 2-step process, where the first stage is used to make a first filtering of the services that could provide the service requested, and the second stage is used to make a more fine-grained selection. This process is described in detail in [12].

For the first step the capabilities of services and the goals of service requestors are expressed using the description logic formalism [1] (more specifically using the OWL language [3]). As an example, below we have the description of the capabilities of two notification services (one for sending e-mail notifications and the another for

sending SMS notifications) and a request to send a notification. With this approach we do not consider as well the difference between preconditions, postconditions, assumptions and effects, as we did in the previous service, but we just consider that all the descriptions are of preconditions.

Capabilities and a Request:

```

CapA =
EmailNotification ⊆ ∃ from.User ⊆ ∃ to.User ⊆ ∀ to.User ⊆
∃ usedProvider.(ProviderA) ⊆ ∃ sendingTime.Timestamp ⊆ ∃ content.String ⊆
∀ acknowledgement.=E

CapB =
SMSNotification ⊆ ∃ from.User ⊆ ∃ to.CellphoneUser ⊆ ∀ to.CellphoneUser ⊆
∃ usedProvider.(ProviderB) ⊆ ∃ sendingTime.(Timestamp ⊆ ≤currentTime+week) ⊆
content.String

Req =
Notification ⊆ ∃ from.{UserX} ⊆ ∃ to.{UserY} ⊆ ∃ to.{UserZ} ⊆ =2to ⊆ ∀
usedProvider.Provider ⊆ ∃ sendTime ≤200406250900 ⊆ ∃ content.String ⊆ ∀
acknowledgment =T ⊆ ∀ cost ≤S

```

Besides, we have the following domain-level facts:

```

Notification ⊆ Action
EmailNotification ⊆ Notification
T ⊆ =1 from

```

The basic idea under DL-based discovery matching is to check whether the conjunction of a request and a capability is satisfiable, i.e. they can have at least one instance in common. If $\text{Request} \sqcap \text{Capability}_X \sqsubseteq \perp$ holds true there is no such common instance and the request cannot be fulfilled by this capability

However, with this type of discovery we are just checking which are the classes of services that can fulfil a request, but we cannot determine exactly which of the instances of those classes of services can actually perform the operations. Furthermore, some specific constraints related to actual values of some properties cannot be used in the reasoning process, and hence a second step is needed, using that information in the process.

For this second step we use individual service descriptions in the F-Logic language [6] and a F-Logic enabled reasoner like Flora-2¹⁰. In this approach for discovery we check whether the capability entails the goal (capability \leq goal). Current limitations with respect to available reasoners led to the current modeling, where the goal-postcondition is expressed as a fact (which may not be fully specified) and the capability-postcondition is expressed as a rule.

```

myGoal:goal[
  postCondition->myNotification].
myNotification:notification[

```

```

    ntf_userToBeNotified -> johndoe,
    ntf_date -> d040606:date[dayOfMonth->5, monthOfYear->5, year->2004],
    paymentMethod -> creditCard,
    cost -> 0.2,
    ntf_body -> "Your Account Z will be in minus in 2 weeks",
    ntf_from -> sentinel].
johndoe:user[
  nif -> 123,
  name -> "John Doe",
  password -> "p",
  login -> "l",
  firstPreference -> jdMobile,
  contacts ->>
  {jdEmail:eml_account[eml_account->"johndoe.com"],
   jdMobile:phone[phn_number->"0123456", phn_type->mobile],
   jdHome:phone[phn_number->"6543210", phn_type->home]}.
sentinel:user[
  name -> "Sentinel System",
  contacts ->> {jdEmail:eml_account[
    eml_account->"sentinel@isccc.com"]}.

```

The capability postcondition describes the state of the information space the service has after its execution. Here we use some prolog build in predicates, e.g. `//` which is an integer division, but that might also be replaced by more declarative predicate names like `integerDivision(X,Y,Z)`.

```

smsProvider[postcondition] :-
  _AnyNotification:notificationSMS[
    phn_number -> _X:phone[phn_type->mobile],
    ntf_receiptAcknowledgement -> false,
    ntf_time -> Time:dateAndTime,
    content -> AnyMessage:message,
    payment -> Payment],
  is_charlist(
    AnyMessage.msg_body, AnyMessageLength)@prolog(),
  AnyMessageLength < 800,
  Tokens is '//'(AnyMessageLength,160)@prolog()+1,
  Cost is Tokens * 0.05,
  Payment.cost >= Cost,
  (Payment.paymentMode = creditCard; Payment.paymentMode = account),
  secondsBetween(currentDate,Time,X), X < 5*60.

```

4 Conclusions

Internet technology is widely extended in the banking processes, especially in the context of bank-customer relationships, where Internet is used as another of the available channels that can be used by customers to access their accounts and perform transactions with them. Different banks and financial institutions adopt different strategies with respect to the types of services that they offer to their customers through this channel. Some of them only provide basic services to them and others are increasingly offering more ranges of products through their eBanking Websites.

However, new technologies are not only being applied to the bank-customer relations, but much more to the bank-to-bank and the bank-providers relations. These relations are highly standardised but not always fully integrated in bank proprietary systems, depending on the type of service that is provided.

In this chapter we have described two applications that fall under the category of bank-customer services, since the end user in both cases is the bank customer. However, they also imply the exchange of information between different organisations (in the case of mortgages, the service accesses external sites to be able to perform comparisons, while in the case of overdraft notification, the service aggregates accounts from different institutions and accesses personal accounts in consumer good companies).

SWS technology makes processes more efficient (in terms of costs and time) and simple to maintain. They optimise the manual processes currently carried out to build new added value services. For instance, they allow searching in available registries, so that the new Web services that have been deployed in the market can be discovered. They also allow composing new added value services that could not be foreseen without the use of this technology. Consequently, more services (product price comparators, broker information, deposits, etc.) can be offered by banks due to their low cost, since less human interaction is required to discover and invoke new available Semantic Web Services once the application is launched.

Though the main objective of this chapter is showing how Semantic Web Service technology can be used to create added value services for ebanking customers, these services also pose interesting technical requirements for the research being done in this area. These requirements are the following:

- o **Discovery.** Discovery capabilities are needed in order to find the Semantic Web Services able to solve the goals composed by service requestors. We have shown how different types of service discovery mechanisms can be used, depending on the degree of accuracy that we want to achieve out from the service discovery process. In general, better guidelines are needed for service developers with respect to the characteristics and limitations of the service discovery process in each Semantic Web Service execution framework.
- o **Mediation.** In both applications we have seen that mediation is needed, since the different information providers (e.g., the different financial entities), may use heterogeneous message syntaxes and vocabularies (ontologies). Mediation capabilities have to be provided in the different execution environments and they must be easy to configure, so that the use of external services is rather straightforward even if the vocabularies and syntaxes are too different from each other.
- o **Invocation.** Once the Semantic Web Services to be used have been selected, they will be invoked by the service requestor. Invocation engines are in charge of contacting the corresponding services and executing them, receiving their responses and sending them back to the service requestor.
- o **Security.** In some cases highly confidential data has to be transferred between services or used for reasoning purposes. This is the case of the overdraft notification service, which uses information about the amount of money in each account, the estimation of future invoices, etc. Security is also needed in the case of making transactions (e.g., bank transfers), to make sure that the right service is exe-

cuted using the correct user account and avoiding the exposure of some pieces of data.

5 Future trends in intelligent e-services applied to personal e-banking

The evolution of the relations between banks and their customers shows that banks will no longer be proprietors of the clients, as happened in the past, but will become instruments of the clients. Current generations tend to go less to the bank, understood as the branch placed in a physical location. For their daily operations they prefer to use electronic media (such as cash dispensers, Internet, and telephone banking services). As a consequence, bank employees have been transformed from 'storekeepers' to 'consultants', due to the fact that clients only go to the physical branch to ask for some specific pieces of information.

In [2] we can find some proposals about the factors that will have to be considered in the future for their Internet channels:

- o Intermediaries between the clients and the financial institutions will appear, as it has happened in other sectors with the appearance of Internet. Banks establish direct relations with their end clients through different channels (branches, post, telephone, proprietary software, Internet, etc.). New intermediaries will gather banking data to provide new services to their customers. From the traditional banking point of view, intermediation per se is not worrying, although the loss of direct relations with the client is. Each contact of the client with the bank, whatever channel used, is a potential sales opportunity.
- o CRM systems will be integrated with banking web sites. The customization of the offer, the simplicity of use and other factors, either in a direct client to/from bank relation or through intermediaries (e.g. real estate agents who would like to complete their range of services) probably has a different meaning than the current one, which is difficult to foresee.
- o There will be a more intensive use of recognition systems for written or spoken language. Banking will tend towards simplicity and the customization of more and more complex products.

The adoption of Semantic Web Service technology will imply an easier deployment of these types of services, since they will facilitate the creation of new services, the integration with other systems and even the interactions with end users.

However, there are still potential barriers for the uptake of this technology in the core of banking service provision. These are the following:

- o Banks do not normally exchange too much information with other banks, to avoid competition. This is the reason why screen-scraping techniques are still used as the main information source for account aggregators.
- o As a result of the previous aspect, many banks do not provide yet Web services to perform operations with them, since they do not want intermediaries to be able

to access easily to the data from their customers, even if their customers provide these intermediaries the access information to the bank services.

- o Finally, banks are very demanding regarding security, and current Semantic Web service technology is not too focused on this aspect, what could be seen as an important weakness for the real deployment of the technology in the future.

Acknowledgements

This work has been supported by the IST projects SWWS (IST-2001-37314) and DIP (FP6-507483).

References

1. Baader F, McGuinness D, Nardi D (2003) *The Description Logic Handbook*. Cambridge University Press.
2. Centeno C. Adoption of Internet Services in the Enlarge European Union. Lessons from the Internet Banking case. June 2003. <http://fiste.jrc.es/download/eur20822en.pdf>
3. Dean M, Schreiber G (2004) OWL Web Ontology Language Reference. W3C Recommendation. Latest version: <http://www.w3.org/TR/owl-ref/>
4. Domingue J, Cabral L, Hakimpour F, Sell D, Motta E (2004). *IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services*. Proceedings of the Workshop on WSMO Implementations (WIW 2004) Frankfurt, Germany, September 29-30, 2004, CEUR Workshop Proceedings, ISSN 1613-0073
5. Fensel D, Bussler C, Ding Y, Omelayenko B. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, 1(2), 2002.
6. Kifer M, Lausen G, Wu J. Logical Foundations of Object Oriented and Frame Based Languages. *Journal of ACM* 1995, vol. 42, p. 741-843
7. López-Cobo, J.M.; Losada, S.; Corcho, O.; Niño, M. A customer notification agent for financial overdrown using Semantic Web Services. *14th International Conference on Knowledge Engineering and Knowledge Management (EKAW'04)*. Springer-Verlag. Lecture Notes in Computer Science (LNCS) 3257:371-385. October 2004.
8. Martin D, Paolucci M, McIlraith S, Burstein M, McDermott D, McGuinness D, Parsia B, Payne T, Sabou M, Solanki M, Srinivasan N, Sycara K (2004) *Bringing Semantics to Web Services: The OWL-S Approach*. Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, California, USA
9. Martínez Montes M, Bas JL, Bellido S, Corcho O, Losada S (2004) Financial Ontology. DIP deliverable D10.3. <http://dip.semanticweb.org/>
10. Martínez Montes M, Bas JL, Bellido S, Corcho O, Losada S, Benjamins VR (2004) Design and Specification of ebanking application: Mortgage Comparison Service. DIP Deliverable D10.2. <http://dip.semanticweb.org/>
11. Martínez Montes M, Bas JL, Bellido S, López-Cobo JM, Losada S (2004) Analysis report on ebanking business needs. DIP Deliverable D10.1. <http://dip.semanticweb.org/>
12. Zaremba M, Moran M, Haselwanter T, Zaremba M, Oren E (2005) *DIP Revised Architecture*. DIP deliverable D6.5. <http://dip.semanticweb.org/>