Data, Information and Process Integration
with Semantic Web Sevices

DIP

Data, Information and Process Integration with Semantic Web Services

FP6 - 507483

Deliverable

WP10: e-Banking case study
D10.2
Design and Specification of Application

Mónica Martínez Montes

José Luís Bas

Sergio Bellido

Oscar Corcho

Silvestre Losada

Richard Benjamins

Jesús Contreras

15 April 2005

EXECUTIVE SUMMARY

This document describes how to deploy a mortgage comparator/simulator that takes advantage of the Semantic Web Services (SWS) technology provided by the DIP Consortium.

The application automates the process of collecting mortgage data from several banks, taking into account that the data can be accessed by executing Semantic Web Services from different banks. Then it provides this aggregated information to users, according to the data that they have filled-in in appropriate query forms.

Following the DIP standards, we present an extensive description of the application, the requirements (functional, non-functional and project constraints) and a sketch of the desired user interface.

Document Information

IST Project Number	FP6 – 507483	**Acronym**	DIP
Full title	Data, Information, and Process Integration with Semantic Web Services		
Project URL	http://dip.semanticweb.org		
Document URL			
EU Project officer	Kai Tullius		

Deliverable	**Number**	10.2	**Title**	Design and Specification of Application
Work package	**Number**	10	**Title**	Case Study eBanking

Date of delivery	**Contractual**	M 12	**Actual**	15-04-05
Status	Version. 1.3		final ☑	
Nature	Prototype ☐ Report ☑ Dissemination ☐			
Dissemination Level	Public ☐ Consortium ☑			

Authors (Partner)	Mónica Martínez Montes, José Luís Bas, Sergio Bellido (Bankinter), Oscar Corcho, Silvestre Losada, Richard Benjamins, Jesús Contreras (iSOCO)			
Responsible Author	Mónica Martínez Montes		**Email**	mmtnez@bankinter.es
	Partner	Fundación de la Innovación. Bankinter	**Phone**	

Abstract (for dissemination)	This document describes the mortgage comparator/simulator, which takes advantage of the Semantic Web Services (SWS) technology provided by the DIP Consortium.
Keywords	Mortgage simulation, mortgage comparison

Version Log			
Issue Date	**Rev No.**	**Author**	**Change**
19-10-04	**001**	Bankinter	First table of contents drafted
22-11-04	**002**	All	Table of contents aligned with other DIP partners
3-12-04	**003**	All	Version for QA
22-12-04	**004**	All	QA comments included
15-04-05	**005**	All	Comments from the first review included in introduction and in section 2.3

Project Consortium Information

Partner	Acronym	Contact
National University of Ireland Galway	NUIG	Prof. Dr. Christoph Bussler Digital Enterprise Research Institute (DERI) National University of Ireland, Galway Galway Ireland Email: chris.bussler@deri.ie Tel: +353 91 512460
Fundacion De La Innovacion.Bankinter	Bankinter	Monica Martinez Montes Fundacion de la Innovation. BankInter, Paseo Castellana, 29 28046 Madrid, Spain Email: mmtnez@bankinter.es Tel: 916234238
Berlecon Research GmbH	Berlecon	Dr. Thorsten Wichmann Berlecon Research GmbH, Oranienburger Str. 32 10117 Berlin, Germany Email: tw@berlecon.de Tel: +49 30 2852960
British Telecommunications Plc.	BT	Dr John Davies BT Exact (Orion Floor 5 pp12) Adastral Park Martlesham, Ipswich IP5 3RE, United Kingdom Email: john.nj.davies@bt.com Tel: +44 1473 609583
Swiss Federal Institute of Technology, Lausanne	EPFL	Prof. Karl Aberer Distributed Information Systems Laboratory École Polytechnique Féderale de Lausanne Bât. PSE-A 1015 Lausanne, Switzerland Email : Karl.Aberer@epfl.ch Tel: +41 21 693 4679
Essex County Council	Essex	Mary Rowlatt, Essex County Council, PO Box 11, County Hall, Duke Street, Chelmsford, Essex, CM1 1LX, United Kingdom. Email: maryr@essexcc.gov.uk Tel: +44 (0)1245 436524
Forschungszentrum Informatik	FZI	Andreas Abecker Forschungszentrum Informatik Haid-und-Neu Strasse 10-14 76131 Karlsruhe, Germany Email: abecker@fzi.de Tel: +49 721 9654 0

Institut für Informatik, Leopold-Franzens Universität Innsbruck	IFI	Prof. Dieter Fensel Institute of computer science University of Innsbruck Technikerstr. 25 A-6020 Innsbruck, Austria Email: dieter.fensel@uibk.ac.at Tel: +43 512 5076485
ILOG SA	ILOG	Christian de Sainte Marie 9 Rue de Verdun, 94253 Gentilly, France Email: csma@ilog.fr Tel: +33 1 49082981
inubit AG	Inubit	Torsten Schmale, inubit AG Lützowstraße 105-106 D-10785 Berlin, Germany Email: ts@inubit.com Tel: +49 30726112 0
Intelligent Software Components, S.A.	iSOCO	Dr. V. Richard Benjamins, Director R&D Intelligent Software Components, S.A. Pedro de Valdivia 10 28006 Madrid, Spain Email: rbenjamins@isoco.com Tel. +34 913 349 797
The Open University	OU	Dr. John Domingue Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, United Kingdom Email: j.b.domingue@open.ac.uk Tel.: +44 1908 655014
SAP AG	SAP	Dr. Elmar Dorner SAP Research, CEC Karlsruhe SAP AG Vincenz-Priessnitz-Str. 1 76131 Karlsruhe, Germany Email: elmar.dorner@sap.com Tel: +49 721 6902 31
Sirma AI Ltd.	Sirma	Atanas Kiryakov, Ontotext Lab, - Sirma AI EAD, Office Express IT Centre, 3rd Floor 135 Tzarigradsko Chausse, Sofia 1784, Bulgaria Email: atanas.kiryakov@sirma.bg Tel.: +359 2 9768 303
Tiscali Österreich Gmbh	Tiscali	Dieter Haacker Tiscali Österreich GmbH. Diefenbachgasse 35, A-1150 Vienna, Austria Email: Dieter.Haacker@at.tiscali.com

		Tel: +43 1 899 33 160
Unicorn Solution Ltd.	Unicorn	Jeff Eisenberg Unicorn Solutions Ltd, Malcha Technology Park 1 Jerusalem 96951, Israel Email: Jeff.Eisenberg@unicorn.com Tel.: +972 2 6491111
Vrije Universiteit Brussel	VUB	Carlo Wouters, Starlab- VUB Vrije Universiteit Brussel Pleinlaan 2, G-10 1050 Brussel ,Belgium Email: carlo.wouters@vub.ac.be Tel.: +32 (0) 2 629 3719

GLOSSARY

Bank. *A financial institution that accepts deposits and channels the money into lending activities*. In this deliverable, the terms **bank** and **financial entity** will be used indistinctly.

Comparator. *An instrument or machine for comparing anything that can be measured with a standard measure.*

Simulator. *A machine that simulates an environment for the purpose of training or research.*

Service. *Financial product offered by a bank or services that does not require a contract, like a bank transfer order*

Product. *Bank product that requires the signature of a contract between the customer and the bank*

Asset. *The land or property of a company or individual, payments due from bills, investments, and anything else owned that can be turned into cash*

SavingAccount. *Account without a chequebook and normally with a low interest rate*

Loan. *Money let out at interest*

MortgageLoan. *A long-term loan backed by real estate or valuable property, usually the item purchased with the loan. The creditor can claim that property if all payments are not made by the borrower when they are due*

Channel. *Communication means used in the relationship between the bank and its customers, including branches, phone, Internet, virtual banking, etc.*

Branch. *Physical bank office*

vBanking. *Virtual Banking. Banking without human intervention*

Customer. *Bank client, who usually has a contractual relationship with the bank*

Company. *A number of people grouped together as a business enterprise. Types of companies include public limited companies, partnerships, joint ventures and proprietorships, and branches of foreign companies*

Person. *Bank client that represents a single person (physical or juridical)*

SOHO. *Small Office, Home Office. It usually refers to professionals who work in their own offices*

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

The objective of this document is to describe the process of a financial application based on SWS (Semantic Web Services). As described in the project deliverable 10.1 [1], banks can act both as clients and as providers of Semantic Web Services. Consequently we will develop an application that includes SWS from both points of view.

In deliverable D10.1 we selected a first case study - mortgage processes – an interesting application to be developed in the SWS framework, due to the number of participants, the time-consumption and the amount of human tasks to be performed. While many SWS sub-processes could be deployed for this application, most of them are dependant on third party developments (i.e. appraisal entities), so we selected the one that allowed DIP to test the provider and the client sides. More precisely, we explained that a Mortgage Offers Simulator/Comparator would be suitable to test the technology provided by our core technology partners.

1.1 Added value of applying Semantic Web Services

As described in D10.1 [1], SWS technology can optimise several processes in the financial domain. These processes are mainly related with human interactions and, consequently, with the costs associated to them. Hence the main benefit of applying SWS technology is that it could permit to develop and maintain financial services with lower costs.

Bankinter is currently offering a free service at http://www.comparador.com/, described in the next section, which presents data about mortgages from a set of banks in Spain. This data is obtained manually by persons, by browsing the Internet services offered by banks (when available) or by calling each bank to gather the information.

The use of (semantic) web service technology can optimise this manual process by allowing search in available registries, so that the new Web services that have been deployed in the market can be discovered. Besides, these registries provide information about how to invoke the selected Web services so as to include them into other, more complex, services. Hence, the data gathering process is improved since the relevant information can be obtained more easily by means of executing those services.

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.

Some of the advantages of SWS over state-of-the-art Web Service technology are the following:

- When facing UDDI with a large number of exported web services, the lookup (aka discovery) becomes a serious problem. There is no standard for service goal or capabilities in current WSDL that prevents automatic service discovery. For example, a bank offering a mortgage information Web service only for fixed interest rates and with a maximum period of 20 years will not be able (or will have many difficulties) to publish such constraints in UDDI registries, so that the external parties looking for services that are compliant with those characteristics will not be able to know in advance whether the service is providing this information according to those constraints or not.

- When the discovered services have been defined according to a set of heterogeneous models, discrepancies may occur in the execution of those services. This is summarised as follows by Gartner Research (February 28, 2002): “Lack of technologies and products to dynamically mediate discrepancies in business semantics will limit the adoption of advanced Web services for large public communities whose participants have disparate business processes.”

Thus the possibilities of better discovery and mediation are the main advantages of SWS technology over current web service technology in the context of the financial application to be developed in this work package.

1.2 Introductory description of the application

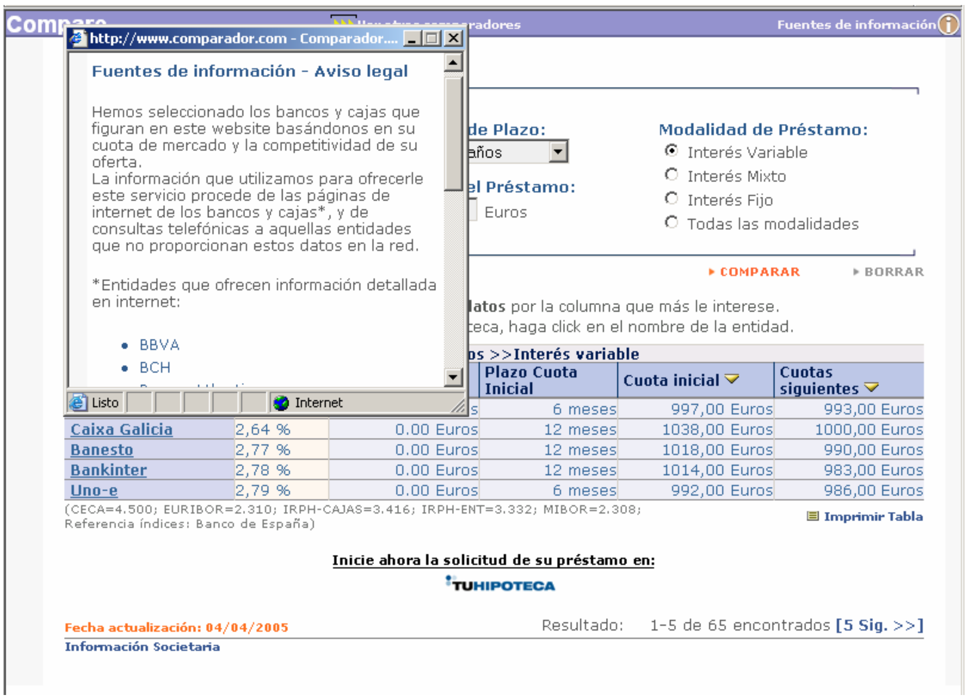
In this document we describe the application requirements (functional and non-functional, and other that depend on the project constraints) and the input/output data flows. Besides, we propose a first sketch of the user interface, although this is not so important from the SWS technology point of view.

The main interaction of the Mortgage Offers Simulator/Comparator is very simple: each time a client wants to know the mortgage market proposals, the application will give him/her actual simulations made on-line in each bank WS-based Simulator, and the results are presented in a human-capable interface in order to compare them. Some further filtering on the presented data could enhance the user experience.

To provide some degree of complexity, the application is designed to allow different requests (‘goals’), and, therefore, the use of different WS on those goals. The whole process implies the use of many of the components that DIP is committed to develop.

A complete market application would contain several hundreds of goals-requests. However, we have chosen a cut down sample in order to be workable within the project time constraints, which we hope to extend later when the prototype is deployed. Anyhow, we are keeping our eyes on a more complex future result. For instance, the mortgage market is changing as global low interest rates continue to be low and land prices are rising as a result while wages remain low. The market is devising differently presented mortgage solutions for its customer base (e.g., mortgages that have several months with only-interest payments).

In order to make the objective of the application more comprehensive, Webs similar to http://www.comparador.com have been considered. These kinds of services are fed with data obtained with screen-scraping techniques or filled in by humans. Each bank can use a different set of mathematic formulae to obtain the output data; consequently the information may not be completely accurate. It is easy to check the disclaimers that the aforementioned Webs show in their pages, as shown in figure 1.



Fuentes de información - Aviso legal

Hemos seleccionado los bancos y cajas que figuran en este website basándonos en su cuota de mercado y la competitividad de su oferta.
La información que utilizamos para ofrecerle este servicio procede de las páginas de internet de los bancos y cajas*, y de consultas telefónicas a aquellas entidades que no proporcionan estos datos en la red.

*Entidades que ofrecen información detallada en internet:

- BBVA
- BCH

de Plazo:
años
el Préstamo:
Euros

Modalidad de Préstamo:
[✓ Interés Variable]
Interés Mixto
Interés Fijo
Todas las modalidades

latos por la columna que más le interese.
teca, haga click en el nombre de la entidad.

os >>Interés variable

			Plazo Cuota Inicial	Cuota inicial	Cuotas siguientes
		s	6 meses	997,00 Euros	993,00 Euros
Caixa Galicia	2,64 %	0.00 Euros	12 meses	1038,00 Euros	1000,00 Euros
Banesto	2,77 %	0.00 Euros	12 meses	1018,00 Euros	990,00 Euros
Bankinter	2,78 %	0.00 Euros	12 meses	1014,00 Euros	983,00 Euros
Uno-e	2,79 %	0.00 Euros	6 meses	992,00 Euros	986,00 Euros

(CECA=4.500; EURIBOR=2.310; IRPH-CAJAS=3.416; IRPH-ENT=3.332; MIBOR=2.308;
Referencia índices: Banco de España)

Inicie ahora la solicitud de su préstamo en:

Fecha actualización: 04/04/2005

Resultado: 1-5 de 65 encontrados [5 Sig. >>]

Información Societaria

Figure 1. A sample mortgage comparison service offered by **http://www.comparador.com.** The disclaimer says that the comparator software retrieves the data from banks' web sites or from phone requests, for banks that do not publish this information on their webs.

In contrast with these kinds of services, in our Case Study data is to be calculated by each Bank provider, using its own formulae and, therefore, certifying the results in real time.

We do not want to forget another relevant matter either: the possible application of this case study to other industrial sectors. The present Case Study could be a useful experience for any other web-based product or service comparator/simulator, e-commerce related or not, due to the similar goal composition, especially when composing a personalized price offer. In the simplest case, only the data input/output and the formula must be changed to offer a similar service devoted to a non-financial product or service.

For example, similar Simulator/Comparator specific applications could be Travel, Real State or Insurance sector offers. As the complexity of the data increases, the easier is to see the basic benefits of SWS technology. A simple product price comparator is just a part of what we intend to do.

To conclude, the intention of this Case Study is to make use of the technologies deployed by DIP. Hence in this deliverable we will focus not only on the application requirements from the end user's point of view, but also on the relationships of the components to be deployed with the DIP architecture components.

2 Requirements

2.1 Functional requirements

In this section we provide the UML use case diagrams and sequence diagrams that describe the whole process. The system is made up of two modules:

- Portal simulator.
- Comparator.

The **Portal Simulator** is the user interface (it will be probably a web interface) that the user will use to input the parameters of the mortgage he/she is seeking for. This user interface connects to the most important system module, the comparator. The **Comparator** is devoted to get the user request and look for the available Semantic Web Services capable to solve the problem. Afterwards these SWS will be invoked and the results, after being ranked, will be passed to the portal simulator in order to be shown and presented to the user.

The UML use case diagram for the whole process is shown in figure 2. The customer sends a request to the simulation system, which uses DIP and other financial entities information providers to solve the customer's enquiry.

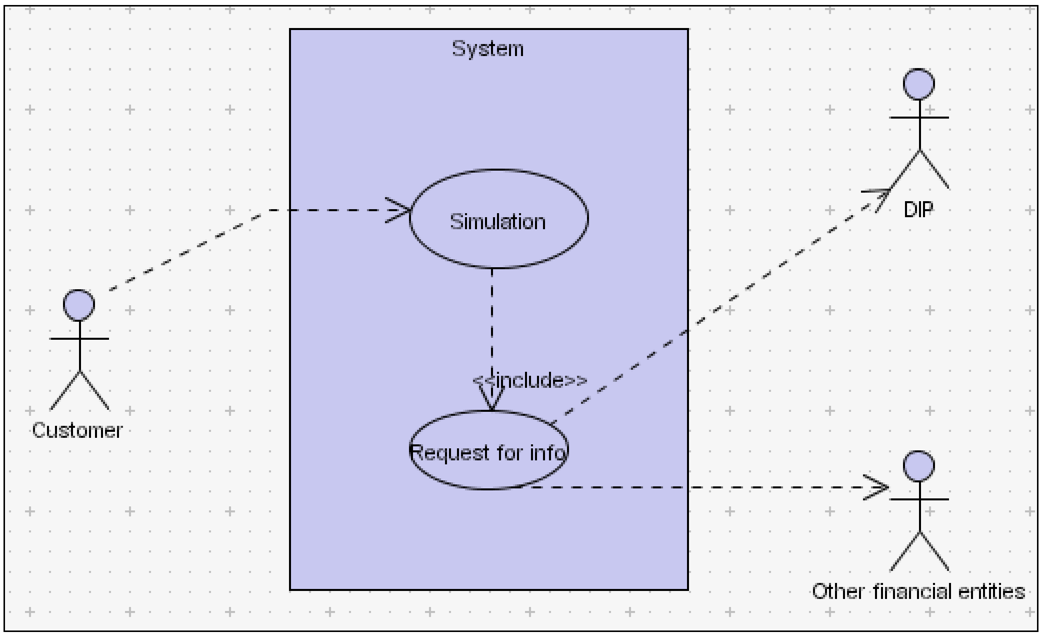


Figure 2. System use case diagram.

A key point is the integration with the DIP architecture and with their different modules. The UML sequence diagram in figure 3 shows at a high level basis how the system will use the DIP architecture, seen as a black box.

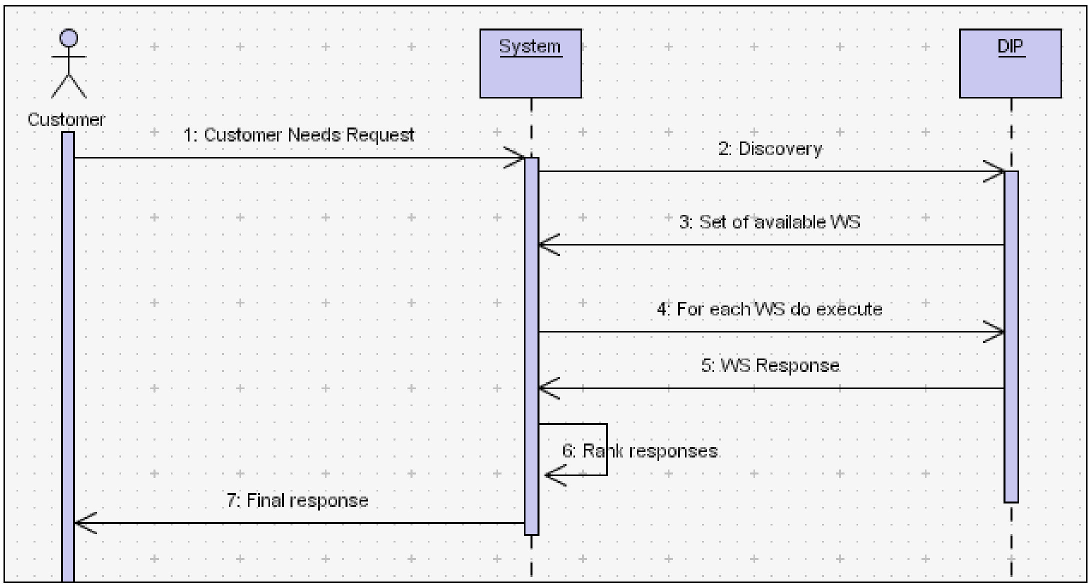


Figure 3. High level sequence diagram.

The customer sends the requests to the system, which discovers the Web Services available using components from the DIP architecture and uses other components of the DIP architecture to execute them. The responses are ranked before they are presented to the final user.

2.1.1 Portal simulator and comparator modules

2.1.1.1 Portal simulator

The portal simulator module is the simplest module. It has no interaction with any DIP architecture component and it is devoted to act as a user interface and to dispatch the user query to the comparator module.

User interface screens are described in section 3. The sequence diagram is shown in figure 4.

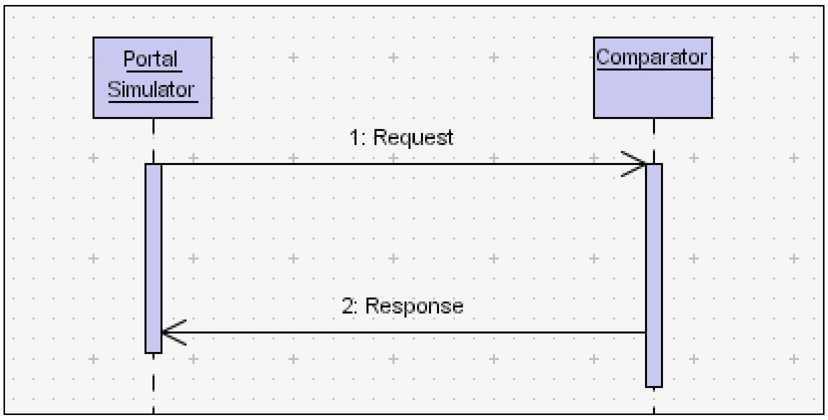


Figure 4. Portal simulator sequence diagram.

2.1.1.2 Comparator

The comparator is the module that interacts with the DIP architecture and with other financial entities.

In the next sequence diagram (figure 5) the System module from figure 3 is expanded into their two modules: the portal simulator and the comparator. The costumer requests will get to the portal and be redirected to the comparator.

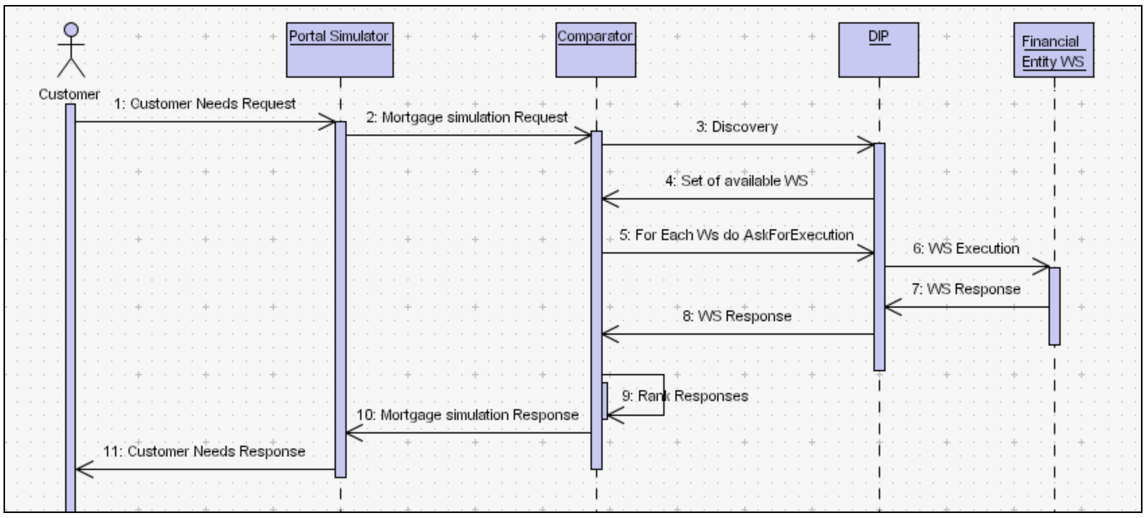


Figure 5. Detailed sequence diagram.

The first step form the comparator to be made is to “discover” the available Web Services by using the DIP architecture. It will provide a list of suitable ones and afterwards the Comparator module will ask the DIP Architecture to execute them. Those Web Services will belong to different financial entities that have previously registered themselves in order to be discovered.

Once the Web Services give any kind of response, it will be given back to DIP and to the Comparator, which will serve the ranked results to the portal simulator

2.1.2 Business data flow

In order to understand, from a high level point of view, the business data flow two figures (figures 6 and 7) are included. Figure 6 describes the data flow in the comparator module where a User Enquiry is entered into the comparator module and, using the WS provided by other financial entities, a goal is constructed and the output is returned.

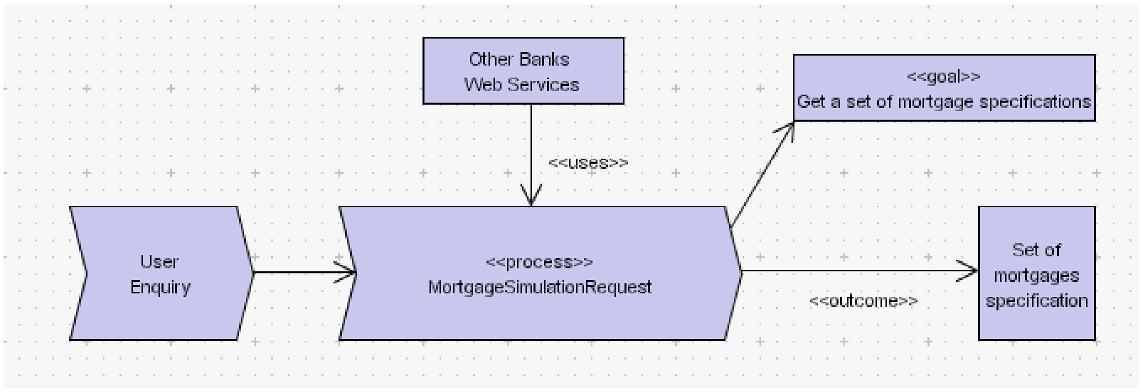

Other Banks Web Services
<<uses>>
<<goal>> Get a set of mortgage specifications
User Enquiry
<<process>> MortgageSimulationRequest
<<outcome>>
Set of mortgages specification

Figure 6. Business data flow.

Figure 7 describes the second business data flow, which ranks the information provided by the previous business flow according to the parameters given by the user. The Comparator module finally accomplishes the goal of the mortgage simulation.

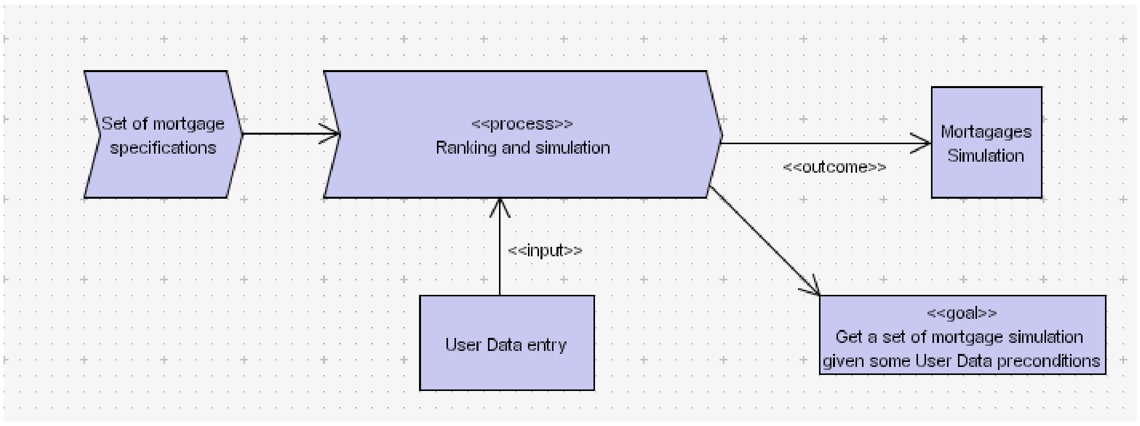

Set of mortgage specifications
<<process>>
Ranking and simulation
<<outcome>>
Mortagages Simulation
<<input>>
User Data entry
<<goal>>
Get a set of mortgage simulation given some User Data preconditions

Figure 7. Business data flow.

2.1.3 Required input-output data

The mandatory parameters for calculating the mortgage formula are:

- **Monthly payment**: The amount of money paid every month.
- **Number of payments** during the whole life of the mortgage.
- **Total mortgage amount**.
- **Interest rate** to be applied to the mortgage.
- **Type of interest**: Fixed type, variable type or mixed. Banks usually have different rates for each type of loan.

The specific interest rate for a given mortgage is supposed to be embedded in each bank Web Service, so it will not take part of the input parameters. The type of interest has to be provided by the user. From the other three remaining parameters only two have to be provided by the user: the other one will be calculated by the system, being the output data.

Therefore we have three combinations of possible input data:

Input	Output
Monthly payment + Number of Payments + Type of Interest	Mortgage Amount
Monthly payment + Mortgage Amount + Type of Interest	Number of Payments
Number of Payments + Mortgage Amount + Type of Interest	Monthly payment

For the sake of simplicity the following assumptions are considered by each bank:

- All the monthly payments are equal (the Web Service will provide the first payment amount).
- All the monthly payments include the amortization of the principal (not only-interest payments are involved).
- Operation commissions are not considered.

Also a set of optional output data can be considered; hence they may be provided or not by some SWS.

Optional Output data	**Description**
Starting interest rate	The mortgage starting interest rate
TAE	The interest rate including commissions (mandatory in Spain)
Commissions	The product commissions (usually a percentage of the loan amount)
Info	Any other kind of literal information

2.2 Non-functional requirements

2.2.1 External interface requirements

- **User related:** The user interface must run on the most common Web browsers, i.e., Microsoft Windows and Netscape in their latest versions so JavaScript, forms, java applets and other plug-ins are supported. The interface must be driven both by the keyboard and the mouse.
- **Hardware related:** No requirements.
- **Communication related:** The application must be accessible through the Internet.
- **Platform related:** The application must be platform independent. Consequently there is constraint to use Java.

2.2.2 Performance requirements

The maximum number of concurrent users for the prototype to be developed is established in 10 users. Moreover the time for the server response must be less than 10 seconds. This is common in all the prototypes developed in the organisation.

2.2.3 Development requirements

No requirements.

2.3 Project and solution constraints (relationship with the DIP architecture)

In the requirements section we have already shown the interaction between the comparator module and the DIP architecture (described in [2]), by means of a sequence diagram. In this section we provide more detail about the specific needs of the application with respect to the architecture proposed in DIP, and more specifically to the components inside the DIP architecture.

Though this kind of information would be more related to a design document, we consider it interesting for this document since it also provides requirements to be used by other DIP workpackages providing the implementation of these components.

Figure 8 summarises these needs.

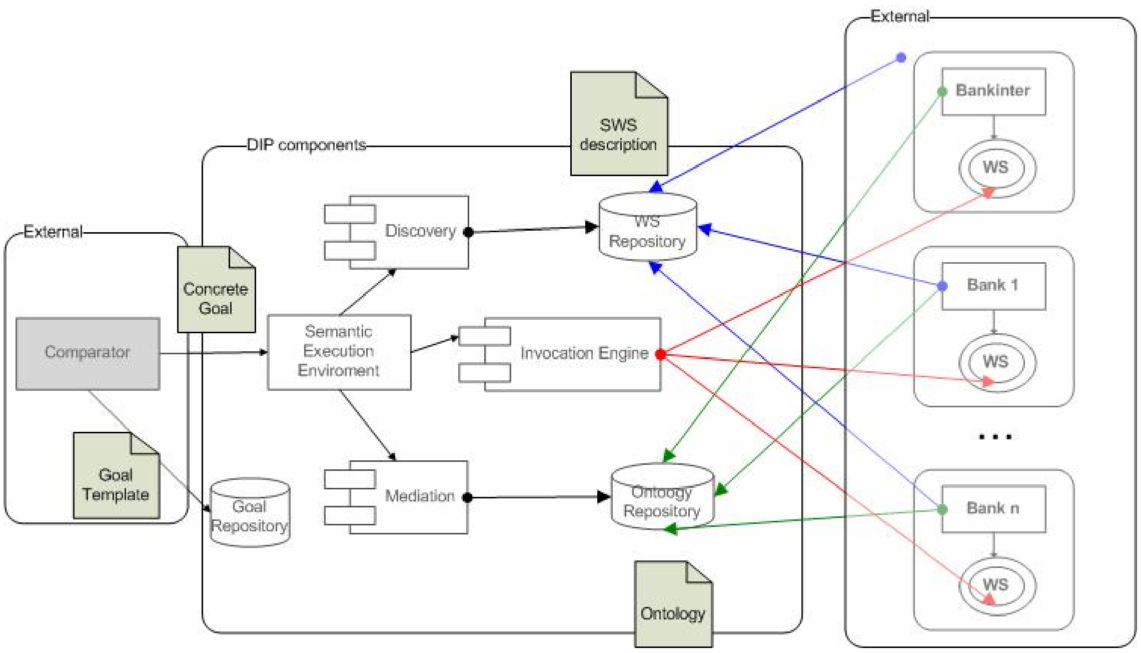


Figure 8. Relationship with the DIP architecture.

Discovery. Discovery capabilities are needed in order to find the Semantic Web Services able to solve the goals composed by the comparator module. WSMO interfaces are needed so that discovery can be executed in a proper way.

Mediation. Mediation is needed since the different information providers, e.g., the different financial entities, may use heterogeneous ontologies. The DIP architecture must mediate the results coming from the Semantic Web Services and return their responses homogeneously to the Comparator Module

WS Repository/Registry. Since discovery is needed, a registry infrastructure is mandatory for the case study purposes. This registry must contain the information of the Semantic Web Services, namely their WSML description including their capabilities, their physical location, their interfaces, etc.

Invocation. Once the Semantic Web Services to be used have been selected by the comparator, they will be invoked by the comparator. The invocation engine will contact the corresponding services and execute them, receiving their responses and sending them back to the comparator module.

Compensation. No compensation modules are foreseen to be used since no complex operations are performed during the sequence. If one of the SWS fails, the information it was to provide will be ignored and the correspondent bank will not partake in the final simulation/comparison process. A retry-philosophy can be applied if one of the Web Services fails. It will depend on the class of error and on the final WSMO implementation.

Composition. No composition modules are foreseen to be used in this case study, since no composed tasks will take place in the whole process.

Security. Although no private data will be transferred, we need ensure that the information comes from the source that is supposed to come from in order to effectively deploy this application in a real environment. This implies that data integrity mechanisms and e-signal procedures (e.g., when a SWS is executed in a registry) must be included in the WSMO implementation. However, since it is not yet clear how this will be addressed in WSMO and since the application will work, in its prototype state, in a closed environment, this requirement will not be considered in this development.

3 USER INTERFACE AND SCREEN FLOWING

The final user interface is limited to the following screens: the data input screen, the output screen and the error screen. Besides, we will include a processing screen that is used in case that the amount of time needed to process the query is longer than expected.

3.1 Input screen

The end user fills in two of three possible fields (monthly payment, number of payments and/or mortgage amount) and checks one or more types of interest (fixed, variable and/or mixed). By default, all of them are supplied marked.

Using check boxes the user may also ask for more information when available:

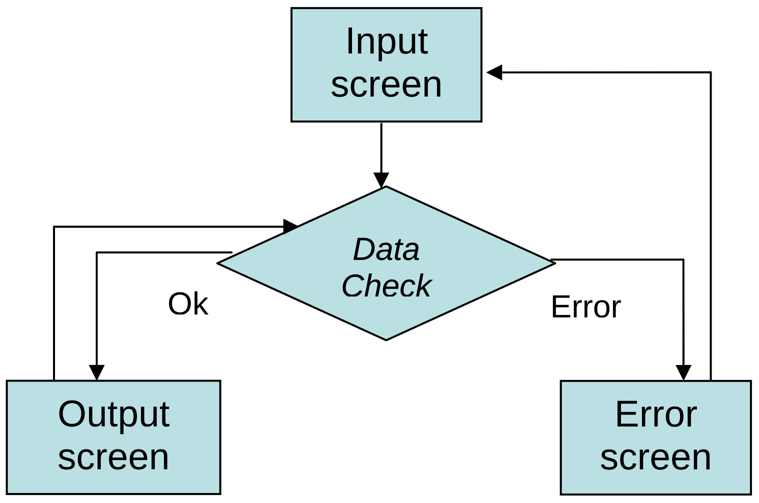


Figure 9. Simulator screen flow.

- The starting interest rate applied.
- The interest rate including commissions for the first period of time.
- The product commission.
- Other data.

All these components will be placed in the upper-half of the screen. In the bottom-half part of the screen an empty grid will be shown. The grid columns are described in the results screen section.

3.2 Processing screen

If the amount of time required to process the data and obtain a result is longer than usually expected for a Web Page, a screen with a text "'Processing your input" or similar will be showed. If possible, it will show a tracking of the status of the request using understandable texts for the average users.

If the waiting period is not too long, this screen should not be used/showed.

3.3 Results screen

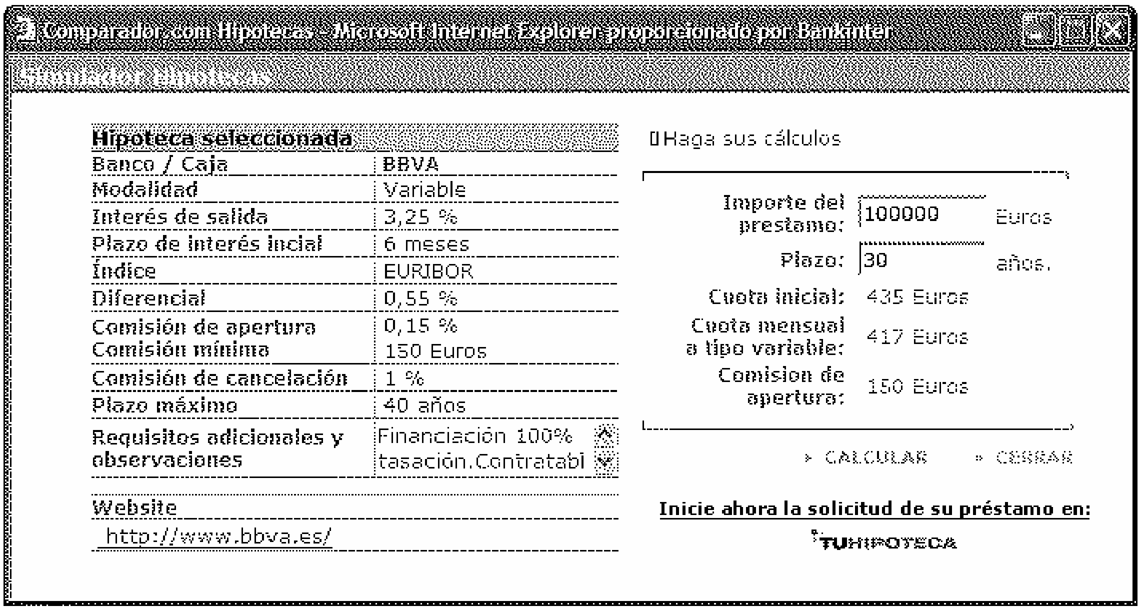
This will have three main areas:

- In the upper left corner a box will contain the list of banks providing information.
- The upper right corner shows the data provided by the final user (three data fields plus the selected type/s of mortgage). All this data should be editable in order to make a new request to all the banks quickly and easily. An action bottom must be showed to execute the new query.
- The bottom half of the screen will show a grid. Each row shows a different mortgage offer and columns will be filled with the following data:
    - The name of the Bank plus the name of the specific product offered (if supplied by the institution).
    - The type of interest applied.
    - The monthly payment.
    - The total amount.
    - The number of payments.
    - The interest rate applied.

The data will be showed sorted by the third column (monthly payment). The user could click on any column to order the data by the headline concept.

If the user clicks in a row, a window will appear with all the previous data plus the extra information fields required by user in the input screen. Again, the user should change any of the data of the upper right corner and deploy a specific request to a specific bank/product as showed below. The interface design should follow the example in Figure 9.

The results of a new query will be showed in the same format.


Comparador.com Hipotecas - Microsoft Internet Explorer proporcionado por Bankinter
Simulador Hipotecas

Hipoteca seleccionada	
Banco / Caja	BBVA
Modalidad	Variable
Interés de salida	3,25 %
Plazo de interés incial	6 meses
Índice	EURIBOR
Diferencial	0,55 %
Comisión de apertura Comisión mínima	0,15 % 150 Euros
Comisión de cancelación	1 %
Plazo máximo	40 años
Requisitos adicionales y observaciones	Financiación 100% tasación.Contratabi
Website	
http://www.bbva.es/	

Haga sus cálculos

Importe del prestamo:	100000	Euros
Plazo:	30	años.
Cuota inicial:	435 Euros	
Cuota mensual a tipo variable:	417 Euros	
Comision de apertura:	150 Euros	

Figure 10. Screen result pattern for a specific bank and product.

If a bank does not provide some of the required data and the user has asked for it, a blank text or a 'n.a' text will be showed in the output screen.

If the final user unmarks one or more of the extra data options, they will not be showed in the output screen even if they exist.

3.4 Error screen

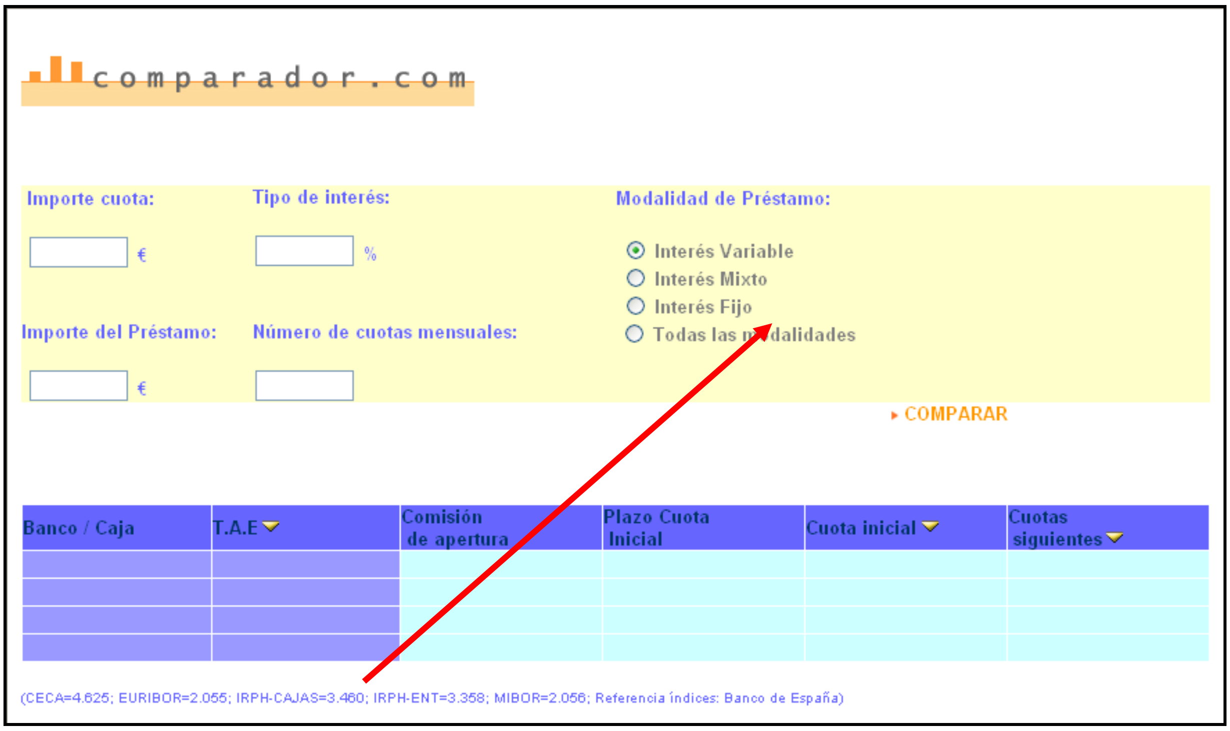
Before processing the query, data will be checked in order to ensure that enough data fields are filled and the provided data format is correct. If an error occurs, a descriptive text and an action button will appear in a pop-up box.

4 Mockup

A mockup of the application has been created in order to understand better the requirements of the application from the point of view of the user interface and of the Semantic Web Service descriptions needed to execute it. The following figures show screenshots of the mockup and some sample services that were developed in FLogic.

First Iteration

1) The comparator screenshot



Importe cuota: €

Tipo de interés: %

Modalidad de Préstamo:

[✓ Interés Variable]
Interés Mixto
Interés Fijo
Todas las modalidades

Importe del Préstamo: €

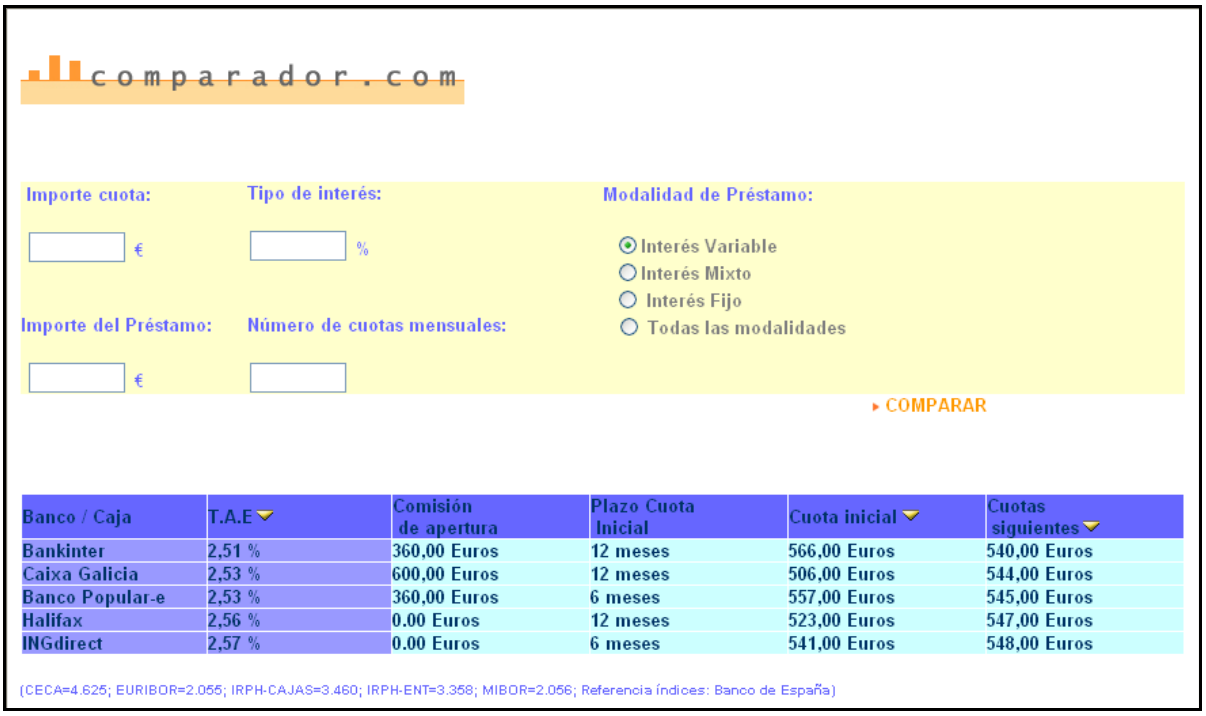
Número de cuotas mensuales:

Banco / Caja	T.A.E	Comisión de apertura	Plazo Cuota Inicial	Cuota inicial	Cuotas siguientes

(CECA=4.625; EURIBOR=2.055; IRPH-CAJAS=3.460; IRPH-ENT=3.358; MIBOR=2.056; Referencia índices: Banco de España)

Figure 11 Comparator home page

2) Clicking on the [COMPARE] button, we obtain the results



comparador.com

Importe cuota: €

Tipo de interés: %

Modalidad de Préstamo:
[✓ Interés Variable]
Interés Mixto
Interés Fijo
Todas las modalidades

Importe del Préstamo: €

Número de cuotas mensuales:

COMPARAR

Banco / Caja	T.A.E	Comisión de apertura	Plazo Cuota Inicial	Cuota inicial	Cuotas siguientes
Bankinter	2,51 %	360,00 Euros	12 meses	566,00 Euros	540,00 Euros
Caixa Galicia	2,53 %	600,00 Euros	12 meses	506,00 Euros	544,00 Euros
Banco Popular-e	2,53 %	360,00 Euros	6 meses	557,00 Euros	545,00 Euros
Halifax	2,56 %	0.00 Euros	12 meses	523,00 Euros	547,00 Euros
INGdirect	2,57 %	0.00 Euros	6 meses	541,00 Euros	548,00 Euros

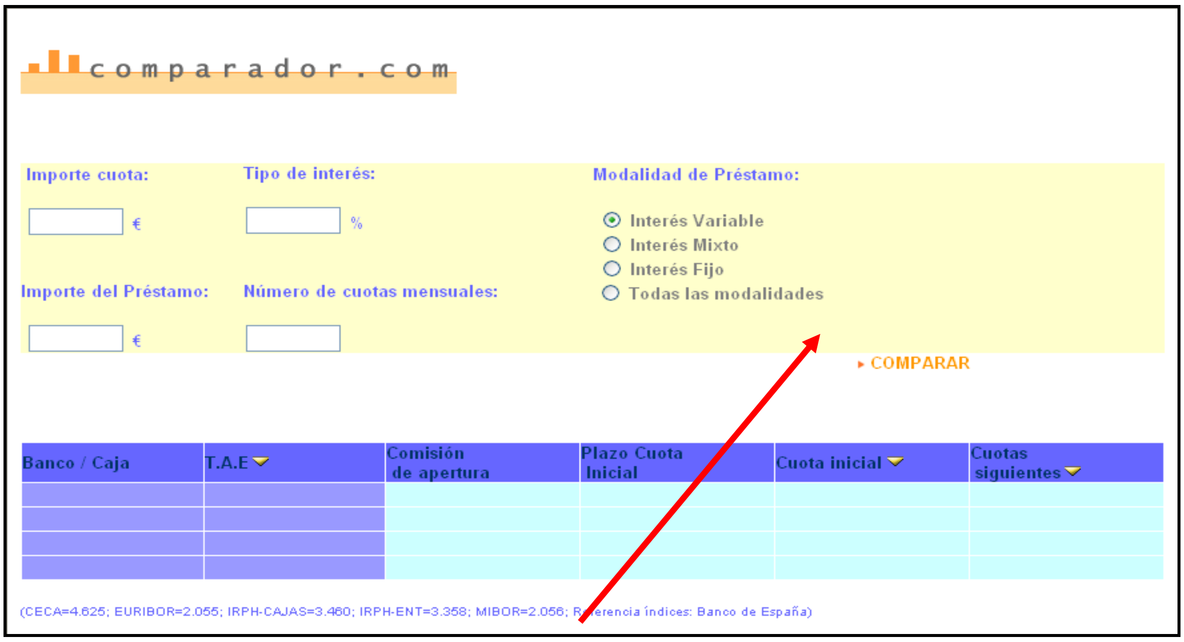
(CECA=4.625; EURIBOR=2.055; IRPH-CAJAS=3.460; IRPH-ENT=3.358; MIBOR=2.056; Referencia índices: Banco de España)

Figure 12 : Result offer list

Second iteration

In this interation we will show in detail how each mortgage is compared.

1) Comparator Home page



comparador.com

Importe cuota: €

Tipo de interés: %

Modalidad de Préstamo:
[✓ Interés Variable]
Interés Mixto
Interés Fijo
Todas las modalidades

Importe del Préstamo: €

Número de cuotas mensuales:

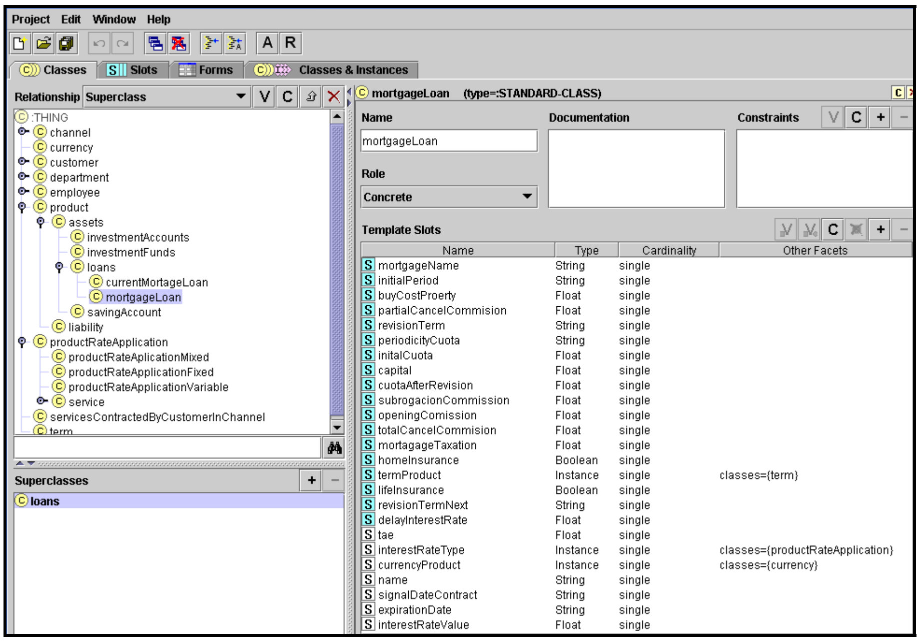
COMPARAR

Banco / Caja	T.A.E	Comisión de apertura	Plazo Cuota Inicial	Cuota inicial	Cuotas siguientes

(CECA=4.625; EURIBOR=2.055; IRPH-CAJAS=3.460; IRPH-ENT=3.358; MIBOR=2.056; Referencia índices: Banco de España)

Figure 13 : Comparator Home Page

2) When clicking on the [COMPARE] button, a screen with the used ontology for the offer description will come up.



Project Edit Window Help

Classes | Slots | Forms | Classes & Instances

Relationship Superclass

- :THING
  - channel
  - currency
  - customer
  - department
  - employee
  - product
    - assets
      - investmentAccounts
      - investmentFunds
      - loans
        - currentMortageLoan
        - mortgageLoan
      - savingAccount
    - liability
  - productRateApplication
    - productRateAplicationMixed
    - productRateApplicationFixed
    - productRateApplicationVariable
  - service
  - servicesContractedByCustomerInChannel
  - term

Superclasses
- loans

mortgageLoan (type=:STANDARD-CLASS)

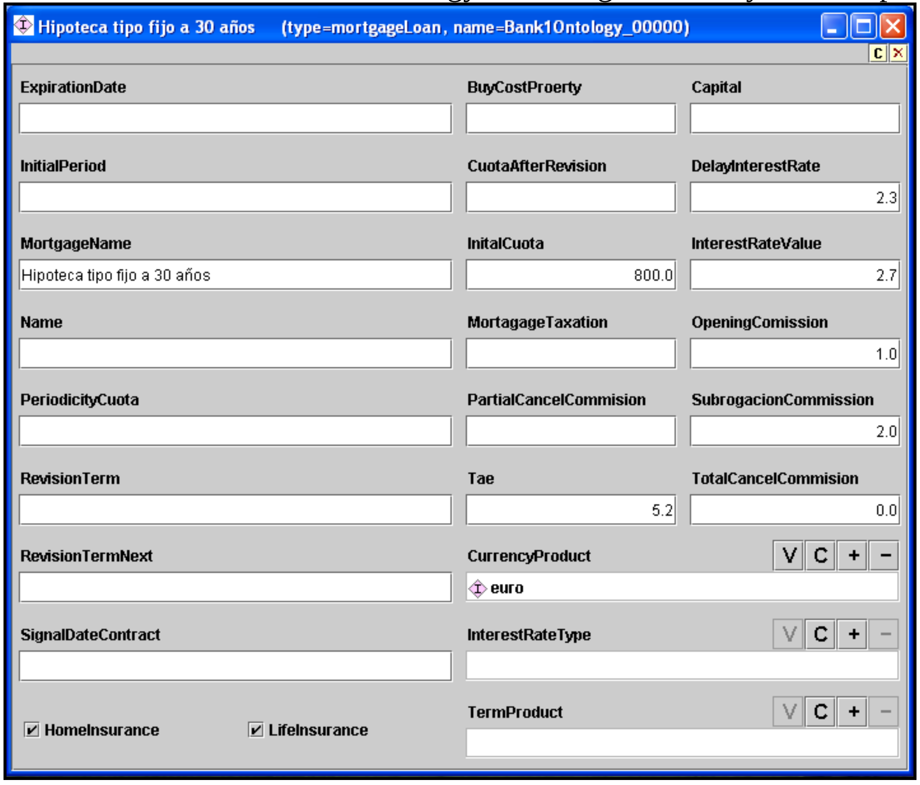
Name: mortgageLoan
Documentation
Constraints
Role: Concrete

Template Slots

Name	Type	Cardinality	Other Facets
mortgageName	String	single	
initialPeriod	String	single	
buyCostProerty	Float	single	
partialCancelCommision	Float	single	
revisionTerm	String	single	
periodicityCuota	String	single	
initalCuota	Float	single	
capital	Float	single	
cuotaAfterRevision	Float	single	
subrogacionCommission	Float	single	
openingComission	Float	single	
totalCancelCommision	Float	single	
mortagageTaxation	Float	single	
homeInsurance	Boolean	single	
termProduct	Instance	single	classes={term}
lifeInsurance	Boolean	single	
revisionTermNext	String	single	
delayInterestRate	Float	single	
tae	Float	single	
interestRateType	Instance	single	classes={productRateApplication}
currencyProduct	Instance	single	classes={currency}
name	String	single	
signalDateContract	String	single	
expirationDate	String	single	
interestRateValue	Float	single	

Figure 14 : Mortgage ontology edited in Protege

3) Here we show an instance of the ontology that wat generated by the Comparator.



Hipoteca tipo fijo a 30 años (type=mortgageLoan, name=Bank1Ontology_00000)

ExpirationDate	
BuyCostProerty	
Capital	
InitialPeriod	
CuotaAfterRevision	
DelayInterestRate	2.3
MortgageName	Hipoteca tipo fijo a 30 años
InitalCuota	800.0
InterestRateValue	2.7
Name	
MortagageTaxation	
OpeningComission	1.0
PeriodicityCuota	
PartialCancelCommision	
SubrogacionCommission	2.0
RevisionTerm	
Tae	5.2
TotalCancelCommision	0.0
RevisionTermNext	
CurrencyProduct	euro
SignalDateContract	
InterestRateType	
HomeInsurance	✓
LifeInsurance	✓
TermProduct	

Figure 15 : Ontology instance (Mortgage for 30 years with 2,7% of reate value)

4) The Comparator is an agent that invokes Web Services. This fugure bellow describes the GOAL (written in WSMO).

```
mygoal:goal.

mygoal[postcondition->>
  myMortgage:MortgageLoan[
      currency -> _#currency[
           currencyName ->EURO],
           term ->_#term:Term[
           type -> month,
           total -> 300,
           ],
           quota->_#quota[
           amount->800 ,
           currency->_currency:Currency[
                currencyName-> EURO
```

5) Comparator can be also seen as a Web Service offering an ordered list of mortgages.

```
MLCap ofclass capability

MLCap[precondition] :-
     mortgageloan ofclass MortgageLoan[
          interestRateNominal  ofvalue  2.7
          interestRateType  ofvalue
          _intrestType                                ofclass
     ProductRateApplicationVariable[
               referenceType ofvalue  EURIBOR
               interesRateVaulue  ofvalue  0.5
          ]
          term  ofvalue _term ofclass Term[
               type  ofvalue  typeTerm
               total  ofvalue  totalTerm
          ]
          quota[
          amount ofvalue quotaAmount
          currency ofvalue _currency ofclass Currency
          .currencyName ofvalue  EURO
          ]

mlCap[postcondition]  ofclass -
     _morgageLoan ofclass MortgageLoan[
          idMortgageLoan  ofvalue
```

6) Using the discovery againts the goals, there appears a list with all services found.
7) Each found service can be view in detail with information about the service, such as its Capability.
8) At the end we will obtain a list of mortgages that refers to found services.

5 REFERENCES

[1] DIP Deliverable D10.1. Analysis report on eBanking business needs. Martínez-Montes M, Bas JL, Bellido S, López-Cobo JM, Losada S, Benjamins VR. June 2004.

[2] DIP Deliverable D6.2. DIP Architecture. Hauswirth M et al. December 2004.