



Data, Information and Process Integration  
with Semantic Web Services

**DIP**

*Data, Information and Process Integration with Semantic Web Services*

**FP6 - 507483**

Deliverable

**WP10: Case study eBanking**

**D 10.7**

**Financial Ontology**

Silvestre Losada Alonso

Jose Luis Bas

Sergio Bellido

Jesús Contreras

Richard Benjamins

Jose Manuel Gomez

October 6<sup>th</sup>, 2005

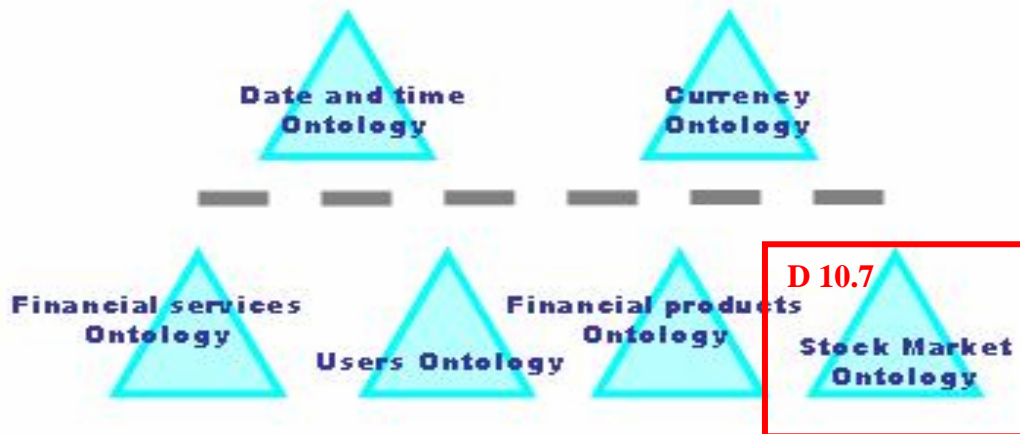




## EXECUTIVE SUMMARY

This document describes the Stock Market Ontology. This Ontology is part of Financial Ontology and has been created for Bankinter for the first eBanking case study in DIP, which is described in the deliverable D10.6. This application consists of a stock market application that allows bank customers to realize complex operations in the stock market, using a natural language interface.

The stock market ontology is one of the several modules that compose the Financial Ontology at different levels of abstraction: services and products; channels, users and currencies. Ontologies for the Financial Ontology are structured in a two level architecture. The higher level contains general-purpose ontologies, while the lower level contains more specific ontologies related to financial domain.



**Figure 1: Knowledge architecture of financial ontology.**

Stock Market ontology will be the backbone of all components developed in the context of second application of WP 10, focus on Stock Market operations, that will be delivered at M30 (June 2006). Financial Ontology was designed to be modular enough to allow refinements in the context of the current domain and extensions to other domains in the financial area.

In DIP D10.3 [1] we explained the reasons why it is difficult to build a ontology that covers the complete financial domain, including previous standardization experiences and business aspects.

The Financial Ontology [1] has been developed following principles specified in D3.11 [2] based in METHONTOLOGY [7]. For the Specification Phase, D3.11 [2] describes that, in conjunction with METHONTOLOGY [7], a technique proposed in the context of the kick-off phase of the On-to-knowledge methodology can be used, which has been proven to be useful to obtain the initial set of concepts and relations, and their classification, from non-experts in knowledge representation.

This deliverable is especially relevant to Workpackage 3.

Disclaimer: The DIP Consortium is proprietary. There is no warranty for the accuracy or completeness of the information, text, graphics, links or other items contained within

---

this material. This document represents the common view of the consortium and does not necessarily reflect the view of the individual partners.

## Document Information

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

<b>Deliverable</b>	<b>Number</b>	10.7	<b>Title</b>	Financial Ontology
<b>Work package</b>	<b>Number</b>	10	<b>Title</b>	Case Study eBanking

<b>Date of delivery</b>	<b>Contractual</b>	M 21	<b>Actual</b>	19-Sep-05
<b>Status</b>	version. 0.5		final	
<b>Nature</b>	Prototype <input type="checkbox"/> Report <input type="checkbox"/> Dissemination <input type="checkbox"/> Ontology <input checked="" type="checkbox"/>			
<b>Dissemination Level</b>	Public <input type="checkbox"/> Consortium <input checked="" type="checkbox"/>			

<b>Authors (Partner)</b>	Silvestre Losada (iSOCO), Jose Luis Bas (Bankinter), Sergio Bellido (Bankinter), Jesús Contreras, Monica Martinez (iSOCO), Richard Benjamins (iSOCO) Jose Manuel Gomez (iSOCO)			
<b>Responsible Author</b>	Silvestre Losada		<b>Email</b>	<a href="mailto:slosada@isoco.com">slosada@isoco.com</a>
	<b>Partner</b>	Bankinter	<b>Phone</b>	913349744


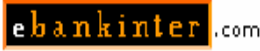



<b>Abstract (for dissemination)</b>	This document describes the Stock Market Ontology. This Ontology is part of Financial Ontology and has been created for Bankinter for the first eBanking case study in DIP, which is described in the deliverable D10.6. This application consists of a stock market application that allows bank customers realize complex operations in stock market, using a natural language interface
<b>Keywords</b>	Financial ontology, Stock Market

Version Log			
Issue Date	Rev No.	Author	Change
02-07-05	001	Silvestre Losada	Initial version Table of contents
07-07-05	002	Jose Luis Bas	Added terms of kick off
08-08-05	003	Sergio Bellido	Added knowledge sources




---

02-09-05	004	Silvestre Losada	Added conceptualization
15-09-05	0.5	Silvestre Losada	Added implementation

## 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: <a href="mailto:chris.bussler@deri.org">chris.bussler@deri.org</a> Tel: +353 91 512460
Fundacion De La Innovacion.Bankinter	Bankinter 	Monica Martinez Montes Fundacion de la Innovacion. BankInter Paseo Castellana, 29 28046 Madrid, Spain Email: <a href="mailto:mmtnez@bankinter.es">mmtnez@bankinter.es</a> Tel: 916234238
Berlecon Research GmbH	Berlecon 	Dr. Thorsten Wichmann Berlecon Research GmbH Oranienburger Str. 32 10117 Berlin, Germany Email: <a href="mailto:tw@berlecon.de">tw@berlecon.de</a> 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: <a href="mailto:john.nj.davies@bt.com">john.nj.davies@bt.com</a> Tel: +44 1473 609583
Swiss Federal Institute of Technology, Lausanne	EPFL 	Prof. Karl Aberer Distributed Information Systems Laboratory École Polytechnique Fédérale de Lausanne Bât. PSE-A 1015 Lausanne, Switzerland Email : <a href="mailto:Karl.Aberer@epfl.ch">Karl.Aberer@epfl.ch</a> Tel: +41 21 693 4679
Essex County Council	Essex 	Mary Rowlett, Essex County Council PO Box 11, County Hall, Duke Street Chelmsford, Essex, CM1 1LX United Kingdom. Email: <a href="mailto:maryr@essexcc.gov.uk">maryr@essexcc.gov.uk</a> Tel: +44 (0)1245 436524
Forschungszentrum Informatik	FZI 	Andreas Abecker Forschungszentrum Informatik Haid-und-Neu Strasse 10-14 76131 Karlsruhe Germany Email: <a href="mailto:abecker@fzi.de">abecker@fzi.de</a> Tel: +49 721 9654 0

Partner	Acronym	Contact
Institut für Informatik, Leopold-Franzens Universität Innsbruck	UIBK 	Prof. Dieter Fensel Institute of computer science University of Innsbruck Technikerstr. 25 A-6020 Innsbruck, Austria Email: <a href="mailto:dieter.fensel@deri.org">dieter.fensel@deri.org</a> Tel: +43 512 5076485
ILOG SA	ILOG 	Christian de Sainte Marie 9 Rue de Verdun, 94253 Gentilly, France Email: <a href="mailto:csma@ilog.fr">csma@ilog.fr</a> Tel: +33 1 49082981
inubit AG	Inubit 	Torsten Schmale inubit AG Lützowstraße 105-106 D-10785 Berlin Germany Email: <a href="mailto:ts@inubit.com">ts@inubit.com</a> 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: <a href="mailto:rbenjamins@isoco.com">rbenjamins@isoco.com</a> Tel. +34 913 349 797
NIWA WEB Solutions	NIWA 	Alexander Wahler NIWA WEB Solutions Niederacher & Wahler OEG Kirchengasse 13/1a A-1070 Wien Email: <a href="mailto:wahler@niwa.at">wahler@niwa.at</a> Tel:+43(0)1 3195843-11
The Open University	OU 	Dr. John Domingue Knowledge Media Institute The Open University, Walton Hall Milton Keynes, MK7 6AA United Kingdom Email: <a href="mailto:j.b.domingue@open.ac.uk">j.b.domingue@open.ac.uk</a> Tel.: +44 1908 655014
SAP AG	SAP 	Dr. Elmar Dorner SAP Research, CEC Karlsruhe SAP AG Vincenz-Priessnitz-Str. 1 76131 Karlsruhe, Germany Email: <a href="mailto:elmar.dorner@sap.com">elmar.dorner@sap.com</a> Tel: +49 721 6902 31

Sirma AI Ltd.	 <p>                     Sirma  <b>Ontotext</b>                      Knowledge and Language                      Engineering Lab of Sirma                 </p>	Atanas Kiryakov, Ontotext Lab, - Sirma AI EAD Office Express IT Centre, 3rd Floor 135 Tzarigradsko Chausse Sofia 1784, Bulgaria Email: <a href="mailto:atanas.kiryakov@sirma.bg">atanas.kiryakov@sirma.bg</a> Tel.: +359 2 9768 303
Unicorn Solution Ltd.	 <p>                     Unicorn                 </p>	Jeff Eisenberg Unicorn Solutions Ltd, Malcha Technology Park 1 Jerusalem 96951 Israel Email: <a href="mailto:Jeff.Eisenberg@unicorn.com">Jeff.Eisenberg@unicorn.com</a> Tel.: +972 2 6491111
Vrije Universiteit Brussel	 <p>                     VUB                      Vrije                      Universiteit                      Brussel                 </p>	Pieter De Leenheer Starlab- VUB Vrije Universiteit Brussel Pleinlaan 2, G-10 1050 Brussel ,Belgium Email: <a href="mailto:Pieter.De.Leenheer@vub.ac.be">Pieter.De.Leenheer@vub.ac.be</a> Tel.: +32 (0) 2 629 3749

---

## LIST OF KEY WORDS/ABBREVIATIONS

SWS Semantic Web Services

CIF "*código de identificación fiscal*", tax identification code, VAT number

WS Web Services

WSDL Web Service Description Language

XML eXtensible Markup Language

WSMO Web Service Modeling Ontology

WSML Web Service Modeling Language

## TABLE OF CONTENTS

<b>EXECUTIVE SUMMARY</b> .....	<b>I</b>
<b>LIST OF KEY WORDS/ABBREVIATIONS</b> .....	<b>VIII</b>
<b>TABLE OF CONTENTS</b> .....	<b>IX</b>
<b>1 INTRODUCTION</b> .....	<b>1</b>
<b>2 ONTOLOGY BUILDING METHODOLOGIES</b> .....	<b>1</b>
<b>3 ONTOLOGY DESIGN PRINCIPLES (YOUR EXPERIENCE OR A CONVENTION?)</b> .....	<b>2</b>
<b>4 SPECIFICATION</b> .....	<b>3</b>
4.1 Goal of the Ontology .....	3
4.2 Scope of ontology .....	3
4.3 Knowledge sources.....	4
4.4 Glossary and terms result of specification phase. ....	6
<b>5 CONCEPTUALISATION</b> .....	<b>10</b>
5.1 Knowledge architecture.....	11
5.2 Stock Market Ontology .....	11
<b>6 IMPLEMENTATION</b> .....	<b>17</b>
<b>7 CONCLUSION</b> .....	<b>19</b>
<b>REFERENCES</b> .....	<b>19</b>
<b>ANNEX I. BANKINTER WEB SERVICES INPUT OUTPUT DATA</b> .....	<b>20</b>
<b>ANNEX II: OWL CURRENCY MEDIATOR:</b> .....	<b>29</b>
<b>ANNEX III: WSML SINTAXIS</b> .....	<b>29</b>

## LIST OF FIGURES

Figure 1: Knowledge architecture of financial ontology.....	i
Figure 1: Invertia web site.....	5
Figure 2: Knowledge architecture of financial ontology.....	11

## LIST OF TABLES

Table 1: bestCotizaciones Input parameter. ....	20
Table 2: bestCotizaciones output parameter.....	20
Table 3: mainIndexes output parameter .....	21
Table 4 cotizacionIndice input parameter. ....	22

---

Table 5: cotizacionIndice output parameter .....	22
Table 6: relevantVariations input parameter .....	22
Table 7: relevantVariations outputparameter .....	23
Table 8: searchValue input parameter .....	23
Table 9: searchValue output parameter .....	23
Table 10: getNews input parameter .....	24
Table 11: getNews output parameter .....	24
Table 12: news Detail input parameter .....	25
Table 13: news Detail output parameter .....	25
Table 14: performBuySell input Parameter .....	26
Table 15: performBuySell output Parameter .....	26
Table 16: getMyPorfolio input parameter .....	26
Table 17: getMyPorfolio output parameter .....	27
Table 18: getPortfolios input parameter .....	27
Table 19: getPortfolios output parameter .....	27
Table 20: sendAlert input parameter .....	27
Table 21: sendAlert input parameter .....	28
Table 22: doCurrencyTransformation input parameter .....	28
Table 23: doCurrencyTransformation output parameter .....	28

## 1 INTRODUCTION

In this deliverable we present a Stock Market Ontology (module/part of financial Ontology) that will be used by the second prototype (Stock Market) developed in WP 10 case study, eBanking as part of the DIP project. The document is structured as follows: Section 2 describes the methodology used in this deliverable. Section 3 specifies ontology design principles. Section 4 describes of specification phase, section 5 discusses conceptualisation of terms captured in specification phase and section 6 describes the implementation of ontology in WSML [6].

The overall objective of WP10 in the DIP Integrated Project is the development of a case study in the eBanking domain. In the context of WP 10 a Stock Market application will be developed in M30 (June 2006). The ontology described in this deliverable model the stock market domain to cover this application.

In this deliverable we have focused on those parts of the ontology that are applicable to our specific case study, as follows:

- 1) We have considered almost all concepts and operations that can be accomplished in Sock Market.
- 2) We have established relationships between all the concepts available from the same point of view, with special attention to the possible combinations of information that a Stock Market user can perform.

The current version of the financial ontology is the result of the analysis and partial reuse of different knowledge sources in the financial domain like existing standards publicly available standards, such as XBRL (eXtensible Bussiness Reporting Language), existing Stock Markets web sites, Stock Market brokers, heterogeneous Stock Market information providers and a representative set of Spanish banks web sites. Also, as recommended by D3.11, specific **WS** (previously developed by Bankinter) are known and listed, and have been used to better focus the scope of the ontology.

From this analysis we have extracted the most representative concepts, unifying the different ways used to express them and removing duplicates.

On the other hand we have tried to cover the most common cases of the European stock markets, trying to cover all the cases and peculiarities that can occur in all the national markets.

The Stock Market ontology has been developed using METHONTOLOGY [ref ], a methodology for ontology construction that has been recommended in DIP deliverable D3.11 for the development of the ontologies needed in all the workpackages.

## 2 ONTOLOGY BUILDING METHODOLOGIES

As has been proposed by WP3 [2] in D3.11 METHONTOLOGY, extended with ideas from other methodologies, like OTK [5], is the recommended methodology to develop ontologies in the DIP project. METHONTOLOGY enables the construction of ontologies at the knowledge level and includes:

- The identification of the ontology development process,

- A life cycle based on evolving prototypes,
- Particular techniques for carrying out each activity.

METHONTOLOGY is supported by main ontology tools as Protégé and WEBODE that can be used to create ontologies with it. In this case we have used wsmstudio<sup>1</sup>[3] and WSMT<sup>2</sup>[4] editors, both created in the context of the DIP project

### 3 ONTOLOGY DESIGN PRINCIPLES

Ontology development is still immature in comparison to software development, where a significant body of design and analysis patterns are available. This section describes the ontology design principles based on our experience and some recommendations described in D 3.11

**Naming conventions:** To achieve a consistent and meaningful ontology it is extremely important to define naming conventions and to obey them throughout the development process:

- **Capitalisation:** Names of concepts in Financial Ontology begin with capital letter. Instance and property names begin with non-capital letters.
- **Delimiters:** For financial ontology we use CamelCase<sup>3</sup> as delimiters. CamelCase eliminates the need for delimiters. Note that CamelCase is also suitable for ontology URIs therefore it is frequently used in web ontologies.
- **Singular or plural:** It is important to decide whether to use singular or plural form of names. It is a common convention - that we have followed- to use the singular form in concept names.
- **Prefix and suffix conventions:** These conventions are mainly important for property names. Our solution is to use “has” and “is” as prefixes for properties, with an additional “Of” suffix for the “is” form. In case of an inverse property pair, one of them should follow the “has” scheme and the other should follow the “is” scheme.
- **Exclude/include superconcepts in names:** The name of superconcepts and superproperties is excluded in the name of subconcepts or subproperties throughout the ontology.

**Documentation:** Ontology entities are documented in natural language. Financial Ontology will be reused; therefore all ontology elements are documented in English and Spanish.

**Modularise:** It is useful and commonly accepted to developed ontologies by splitting them into independent modules. This Financial Ontology is modularized into independent modules representing services, products, channels, users and currencies.

---

<sup>1</sup> <http://www.wsmstudio.org>

<sup>2</sup> [https://sourceforge.net/project/showfiles.php?group\\_id=113321&package\\_id=147251](https://sourceforge.net/project/showfiles.php?group_id=113321&package_id=147251)

<sup>3</sup> <http://en.wikipedia.org/wiki/CamelCase>

## 4 SPECIFICATION

In this section we present the conceptualisation of the ontology according to the intermediate representations proposed by METHONTOLOGY for this conceptualisation phase. To complete this process, as recommended in D3.11 [2], during this process we have also used Kick-Off phase of OKT methodology

### 4.1 Goal of the Ontology

The main purpose of this Ontology is to describe the elements (terms and vocabulary), their relationships and the rules of the given domain, and to describe the knowledge in a generic way.

The Ontology (domain ontology of the Stock Market) will be the backbone of all the systems to be developed. It will be the basis for the rest of the WSMO elements; it will provide the semantic description for the service capabilities, as well as describe the goals and mediators. This way WS are semantically described, and can be discovered, composed and invoked by other services automatically thus transforming WS's into Semantic Web Services (SWS's).

### 4.2 Scope of ontology

It is beyond the scope of this deliverable and of this specific case study the creation of a complete financial ontology, deepening in all the possible branches of products and services offered by a bank. As mentioned in D10.3[1]:

- *Standardisation efforts in the banking domain are very slow and several examples exist of unsuccessful attempts. For instance, we can cite the example of Mobipay in Spain, a micro-payments standardisation initiative, where different interests were involved and no effective market action was finally made.*
- *Innovation-oriented banks like Bankinter prefer creating innovative products on their own, so that they have some competitive advantage during a short period of time (usually around half a year) until the rest of banks implement such innovations too. That is, we strongly believe that a bank like Bankinter will adopt its own conceptual model and then, if successful, this model will be progressively adopted by other banks. This is well documented by a Forrester Research's business report, where the process of ontology adoption in business is explained.*
- *The financial domain is very dynamic: new products appear on a weekly basis and some of them cannot be categorised a priori.*
- *There is a high complexity in the current financial standards, such as IFX (International Financial eXchange Forum), and the agreement between different financial entities is difficult as well, as aforementioned.*

*There are also strong reasons to develop a new ontology (based on existing ontologies and standardisation initiatives) instead of directly reusing already existing ones:*

- *In a mature market, such as the financial one, the only advantage between competitors is the know-how and the technology approach. Therefore, proposals of standardisation usually result in long projects, as aforementioned. In these projects,*

*the strongest banks usually impose their own criteria to the rest, whilst small banks try to find a way to make things slightly (or completely) different in order to compete where the big ones cannot. In that frame, a descriptive but not-too-complex ontology makes the standardization process easier and faster. It also allows each Bank to model its own complexity while maintaining a certain degree of differentiation within a common framework.*

- *Most of the existing ontologies that we have studied model the financial domain from a client point of view, and do not cover sufficiently the internal processes that a bank must follow to deploy a mortgage contract.*

For this reason, we have focused on those parts of the ontology that are applicable to our specific needs, as follows:

- 1) The resulting ontology would cover all the requirements of the second Case Study application to be deployed in M30 (June 2006). The application is focused on the financial domain, specifically in the Stock Market environment and covers almost all the concepts required to describe semantically this market.
- 2) We have established relationships between all the concepts available from the same point of view, with special attention to the possible combinations of information that a Stock Market service can perform.
- 3) XBRL is the closer standard to our needs that we have found. We have analysed it to develop this ontology. XBRL<sup>4</sup> taxonomy focused on “*financial exchange of information in the reception and diffusion of the periodic public information (quarterly and semester information) that the listed societies with shares admitted to quotation must send to the supervisor*”<sup>5</sup>. Thus, this taxonomy is used to report the periodic information of companies to the Stock Market authorities and therefore we have used it for reference and to picked up several useful terms.

### 4.3 Knowledge sources

In order to find what are the most common used terms and pieces of information that banks provide in their web sites, we have studied some of the most representative ones. Not all banks have public information about the Stock Market in their websites, so we have had to discard some of them. To have a broad vision of the market for this ontology and to make it more powerful, we have included in our research some spanish and international independent stock market services (i.e. : Yahoo Finances, Reuters and Invertia), since they are increasingly used by costumers and, usually, offer more detailed information than banks do

The following sources have been used with that focus:

---

<sup>4</sup> <http://www.xbrl.org>

<sup>5</sup> <http://www.xbrl.org/es/english/english.html>

- Financial Web sites :
  - o Banco Santander<sup>6</sup> (Spain)
  - o Invertia<sup>7</sup> (Spain)
  - o Banesto<sup>8</sup> (Spain)
  - o Bankinter<sup>9</sup> (Spain)
  - o Cajamadrid<sup>10</sup> (Spain)
  - o Yahoo Finance<sup>11</sup>s (Worldwide)
  - o Reuters<sup>12</sup> (Worldwide)
- XBRL standard (several terms used)
- Bankinter former Web Services. See Annex I

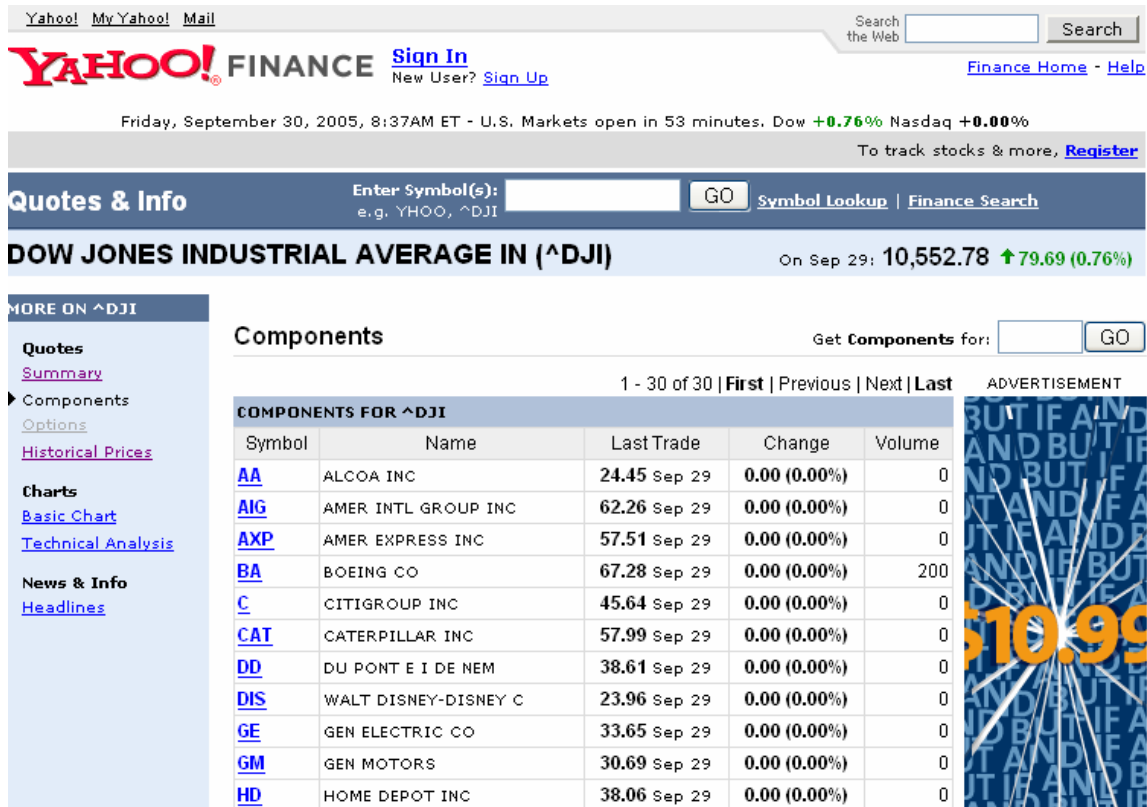


Figure 2: Yahoo Finance web site

<sup>6</sup> [http://www.gruposantander.es/bolsa/inf/index\\_mercados.htm](http://www.gruposantander.es/bolsa/inf/index_mercados.htm)

<sup>7</sup> <http://www.invertia.com>

<sup>8</sup> <http://www.banestobroker.com>

<sup>9</sup> <http://broker.ebankinter.com/>

<sup>10</sup> <http://www.cajamadridbolsa.es/>

<sup>11</sup> <http://finance.yahoo.com/>

<sup>12</sup> <http://today.reuters.com/stocks/Overview.aspx?ticker=bkt.mc>

#### 4.3.1 Terms collected by experts during specification phase.

This section shows the list of all the terms collected by relevant experts during several brainstorming sessions. Accompanying this list, for each term, some comments and considerations of Ontology engineers are included.

As is described in D3.11 [2] in the ontology specification phase we have used the techniques proposed for the kick-off phase of the On-To-Knowledge methodology [5], which recommends obtaining a list of main domain concepts and relations together with their natural language descriptions by means of brainstorming sessions with domain experts.

In this specific case, a second round of sessions took place to fix the goal and scope of the ontology and to obtain a first draft of the terms glossary. Later, a more in-depth work was performed to filter, eliminate duplicates and classify concepts, relations and attributes.

**Term:** Stock, Share: Represents one fraction of one specific company ownership.

*Type:* Concept

*Ontology engineers remarks:* Attributes for this concept

- isParOfCompany
- isContinuous
- hasFaceValue
- hasPiceValue
- hasStockType
- hasBestoffer
- hasBestSellPrice
- hasBestBuyPrice
- hasMaximun
- hasMinimun
- hasDividend
- hasStockMarket

**Term:** Stock Market The market to buy or sell shares.

*Type:* Concept

*Ontology engineers remarks:* Attributes for this concept

- hasStocks
- hasIndexes
- hasName
- hasCurrency

- hasBrokers
- hasCountry
- hasSessions

**Term:** Continuous Market: Part of the global Stock Market that allows electronic transactions.

*Type:* Concept

*Ontology engineers remarks:* Attributes are the same as those of stock market, for this reason we have included isContinuous attribute to the StockMarket concept.

**Term:** Buy Sell order: Single transaction in which one part sells an amount of shares of a company at a price and the other buys.

*Type:* Concept

*Ontology engineers remarks:* We tried to reuse the Purchase Ontology<sup>13</sup> to model this concept. The Purchase Ontology defines general concepts to make a purchase order request. The ontology is a WSML[6] representation of the RosettaNet's PIP3A4<sup>14</sup> "PurchaseOrderRequest" and PI3A4 does not cover properly operations in stock markets. For this reason it is not appropriate to use this ontology to model buy/sell operations in stock markets.

Attributes identified for this concept:

- hasSellBuyDate
- hasNumberOfStocksBuySell
- hasISIN
- isBuyOrSell
- hasCurrency
- isPerformed
- hasDepositary
- hasPortfolioNumber
- hasSession
- hasStockMarket
- hasUserId

**Term:** Price, Market Value: Value of the last transaction of a single share

*Type:* Stock attribute

*Ontology engineers remarks:* **None**

---

<sup>13</sup> <http://www.wsmo.org/ontologies/purchase/>

<sup>14</sup> <http://www.rosettanet.org/RosettaNet/>

---

**Term:** Last price: Value of the last transaction in a session of a single share.

*Type:* Concept

*Ontology engineers remarks:* This concept stores prices of all sessions we model as concept

- hasDate
- hasValue

**Term:** Session: Time unit from the opening to the closure of the market. Usually one working day

*Type:* Concept

*Ontology engineers remarks:* Attributes for this concept

- hasIndexValue
- hasVolume
- hasStocks
- hasDate
- hasOpeningTime
- hasClosingTime

**Term:** Opening session price: Is the last price of the previous session.

*Type:* Stock attribute

*Ontology engineers remarks:* **None**

**Term:** Best Bid/Ask: Best price offered to buy/sell at a specific moment for a single share.

*Type:* Stock Attribute

*Ontology engineers remarks:* **None**

**Term:** Volume: Number of single shares traded in a session

*Type:* Concept

*Ontology engineers remarks:* Attributes for this concept

- hasAmount
- hasVolumeStock
- hasVolumeSession

**Term:** Index: Weighted average of a number of stock prices, one example is IBEX 35. The initial value of any index is 100.

*Type:* Concept

*Ontology engineers remarks:* Attributes for this concept

- hasName
- hasIndexId

- hasMarket
- hasInitialValue
- hasActualValue
- hasStocks

**Term:** Weight:

*Type:* Concept

*Ontology engineers remarks:* **None**

**Term:** Index variation: Variation in value of index in a certain period of time usually a session

*Type:* Function.

*Ontology engineers remarks:* **None**

**Term:** Capital gain: Difference between stocks buy price and stocks sell price

*Type:* Function.

*Ontology engineers remarks:* **None**

**Term:** Maximum/Minimum: Maximum/Minimum value of stock in a certain period of time. This period is usually a year

*Type:* Stock attribute

*Ontology engineers remarks:* **None**

**Term:** Valid period: During this period of time a buy sell order is in force

*Type:* ConditionedBuySellOrder attribute

*Ontology engineers remarks:* **None**

**Term:** OfficialTaxes: Official taxes that could be applicable in a stock market operation

*Type:* OrderExecution attribute

*Ontology engineers remarks:* **None**

**Term:** Commissions: rates that an intermediate apply in an stock market operation

*Type:* BuySellOrdertribute.

*Ontology engineers remarks:* **None**

**Term:** Broker: Society or person that could realize operations in a stock market and apply official taxes to each operation

*Type:* Concept.

*Ontology engineers remarks:* Attributes for this concept.

- CIF
- hasNonOfficialTaxes
- hasSotckMarkets

---

**Term:** Depository: Society or person that holds clients stocks

*Type:* Concept.

*Ontology engineers remarks:* Attributes for this concept

- hasBroker
- hascommisions
- DepositoryID

**Term:** Portfolio: The number (assigned by the Depository) where the stocks are deposited

*Type:* Concept

*Ontology engineers remarks:* Attributes for this concept

- hasStocks
- hasUserID
- hasDepository

**Term:** Associated account: Account in which the sale or purchase of stocks is paid or charged.

*Ontology engineers remarks:* Imported from Financial Ontology

**Term:** Dividend The amount of the last profit distribution that corresponds to a single stock

*Type:* Stock attribute

*Ontology engineers remarks:*

**Term:** Recommendation: Opinion emitted by an expert person or society in the stock market domains

*Type:* Is a service offered for some professionals

*Ontology engineers remarks:* SWS that will be created if is needed for case study.

Note: On the stock market there are many other items to trade with. Since they are less representative for individuals they have not been considered on this ontology. For example: Derivatives, warrants, bonds...

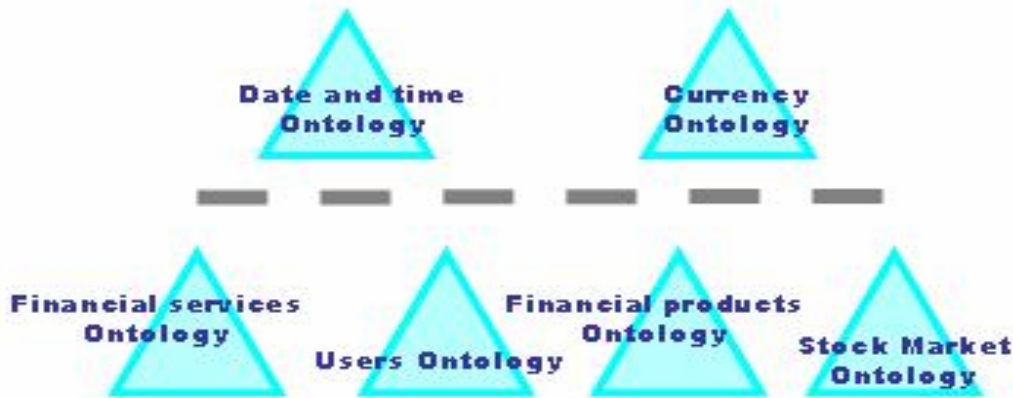
## 5 CONCEPTUALISATION

In this section we introduce the conceptualisation of the ontology according to the intermediate representations proposed by METHONTOLOGY. For this conceptualisation we have used list of all the terms provided by experts during brainstorming sessions (see section 4.4), terms of Financial Web sites analysed, terms of XBRL and Bankinter former Web Services (see Annex I)

## 5.1 Knowledge architecture

As described in D10.3 [1] the financial ontology is composed of several ontologies at different levels of abstraction. Ontologies for the Financial Ontology are structured in a two level architecture. The higher level contains general-purpose ontologies, while the lower level contains more specific ontologies related to financial domain.

Next figure shows this structure



**Figure 3: Knowledge architecture of financial ontology.**

We will use WSMO to describe services of Stock Market Application. In WSMO conceptual framework, OOMediators import other ontologies or into WSMO entities (ontologies, goals, Web Services) and resolve possible terminology mismatches.

WSML [6] is the language used to implement financial ontology, however the Currency ontology has been implemented in OWL, therefore a mediator is needed that allows using the currency ontology into the Financial Ontology, transforming OWL syntax into WSML syntax. Annex II shows ooMediator (WSMO term for ontology to ontology mediator) that translates the OWL description of the currency ontology to WSML

## 5.2 Stock Market Ontology

In the following sections we provide the details of Stock Market ontology.

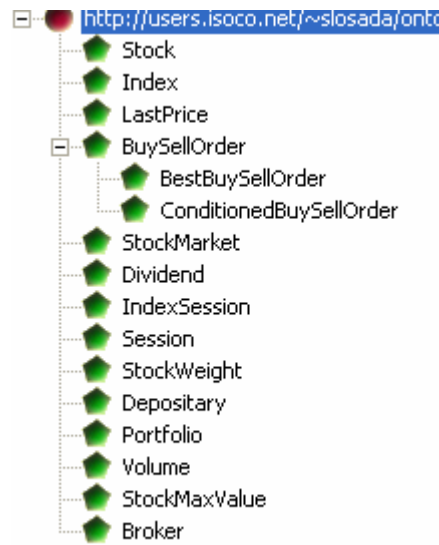
### Terms glossary

Name	Description	Type
Stock	Represents one fraction of one specific company ownership.	Concept
Company	A society owned by several individuals	Attribute
hasStockMarket	The Market where the stock can be sell/bought	Attribute
modoContinuous	Reflects wether or not the stock can be sell/bought in the Continuous Market	Attribute
FaceValue	The amount expressed in currency representing the ownership of the company.	Attribute
priceValue	The price of the last transaction for that stock in an	Attribute

	specific market	
stockType	Specific types of rights in the company	Attribute
bestOffer	The best price that can be found in the market at this point of time. Two figures are available: best selling price and best buying price.	Attribute
hasMaximun	The highest price in an specific period of time	Attribute
hasMinimun	The lowest price in an specific period of time	Attribute
hasDividend	The amount of the last profit distribution that corresponds to a single stock	Attribute
StockExchange	The market to buy or sell shares.	Concept
hasStocks	Stocks that can be sell/bought there.	Attribute
hasIndexes	Figures that reflect the overall trend of the market or part of it.	Attribute
hasName	The name of the market	Attribute
hasCurrency	The currency in wich the stock prices are expressed	Attribute
hasBrokers	The persons/companies allowed to operate in the market	Attribute
hasCountry	The country where the market is established	Attribute
hasSessions	The dates the market has been/will be opened	Attribute
BuySellorder	A client order to buy or sell an amount of shares in the market	Concept
hasBuySellDate	The date and time in which the operation has been performed.	Attribute
hasNumberOfStocksBuySell	Amount of stocks transferred in the operation	Attribute
hasOfficialTaxes	Taxes payed or to be payed	Attribute
hasCommissions	Payed or to be paid to the depositary when it intermediates.	Attribute
hasDepositary	The person/company in charge of keeping safe the stocks	Attribute
hasPortfolioNumber	The account number (defined by the depositary) where the stocks are kept	Attribute
hasSession	The secuencial number of session	Attribute
hasISIN	The identifier of the stocks sell/bought	Attribute
hasStockMarket	The Stock Market where the transaction was or will be performed	Attribute
isBuyOrSell	Determines the type of operation	Attribute
hasUserId	Determines the property of the shares	Attribute
hasCurrency	The currency used to express the amount of the operation. Both the transaction currency and the portfolio currency are required.	Attribute
hasExpirationDate	The date in which the order is cancelled if not performed before	Attribute

isPerformed	Wether the order is performed or not	Attribute
Lastprice	Value of the last transaction in a session of a single share	Concept
hasDate	Date and time of historical values	Attribute
hasValue	Historical values	Attribute
Session	Time unit from the opening to the closure of the market. Usually one working day.	Concept
hasVolume	The amount of shares traded in one period of time	Attribute
hasDate	The date of the session	Attribute
hasOpeningTime	The time at what the market is opened	Attribute
hasClosingTime	The time at what the market is closed	Attribute
Index	Weighted average of a number of stock prices	Concept
hasName	Index name (i.e.; Nasdaq, Dow Jones)	Attribute
hasIndexID	Index identification	Attribute
hasMarket	The market to what the index refers	Attribute
hasInitialValue	The value of the index when a session starts	Attribute
hasActualValue	The latest value of the index	Attribute
hasStocks	The stocks that compose the index	Attribute
Broker	Company or person authorised to operate in the Stock Market	Concept
BrokerID	The identification of the Broker	Attribute
hasBrokerComissions	The comissions that an specific Broker applies	Attribute
hasStockMarkets	The markets in which the broker is authorised to operate.	Attribute
Depository	The person or company in charge of keeping safely the stocks of a client.	Cconcept
hasBroker	The brokers the depopitary works for.	Attribute
hasDepositComissions	The commissions the Depository applies	Attribute
DepositoryID	The identificacion of the Depository	Attribute
PortfolioAccount	The number (assigned by the Depository) where the stocks are deposited	Concept
hasStocks	Identificaction of the Stocks deposited in a Portfolio	Attribute
hasUsedID	Identification of the client owner of the stocks deposited.	Attribute
hasDepository	Identification of the depository	Attribute
BestBuySellOrder	An order to buy shares at the lowest available price in the Barket or sell at eh highest available price in the market	

### Concept classification tree



**Figure 4: Concept classification tree.**

**Concept dictionary**

Concept name	Class attributes	Instance attributes	Relations
Stock	--	isParOfCompany isContinuous hasFaceValue hasPiceValue hasStockType hasBestSellPrice hasBestBuyPrice hasMaximun hasMinimun hasDividend	hasStockMarket hasLastPrices
StockMarket	--	hasName hasCurrency hasCountry	hasStocks hasBrokers hasSessions hasIndexes
BuySellOrder	--	hasBuySellDate hasNumberOfStocksBuySell hasISIN isBuyOrSell hasCurrency isPerformed	hasDepository hasPortfolioNumber hasSession hasStockMarket hasUserId
BestBuySellOrder			executedInSession
ConditionedBuySellOrder		hasExpirationDate hasPrice	
LastPrice	--	hasDate hasValue	
Session	--	hasDate hasOpeningTime hasClosingTime hasIndexValue	hasVolume hasStocks

Concept name	Class attributes	Instance attributes	Relations
Index	--	hasName hasIndexId hasInitialValue hasActualValue	hasStockWeights hasMarket
Volume	--	hasAmount	hasVolumeStock hasVolumeSession
IndexSession	--	hasValue hasIndex	
StockWeight	--	hasWeght	hasStock hasIndex
Depository	--	CIF hasCommisions	hasBroker
Broker	--	CIF hasNonOfficialTaxes	hasSotckMarkets
Portfolio	--	hasPortFololiold	hasStocks hasUserID hasDepository hasAssociatedAccount
PriceInPeriod		hasPeriod hasValue hasStock	

### Imported terms list

Name	Imported from ontology	Type
User	Financial Ontology	User
AssociatedAccount	Financial Ontology	Account
hasStocks	StockMarket	N
hasSessions	StockMarket	N

### Binary relation table

Relation name	Source concept	Cardinality (Max)	Target concept
hasSockMarket	Stock	N	StockMarket
hasLastPrices	Stock	N	LastPrice
hasIndexes	StockMarket	N	Index
hasStocks	StockMarket	N	Stock
hasSessions	StockMarket	N	Session
hasBrokers	StockMarket	N	Broker
hasDepository	BuySellOrder	1	Depositaro

hasPortfolioNumber	BuySellOrder	1	Portfolio
hasSession	BuySellOrder	1	Session
hasStockMarket	BuySellOrder	1	StockMarket
hasUserId	BuySellOrder	1	User
executedInSession	BestBuySellOrder	1	Session
hasVolume	Session	1	Volume
hasStocks	Session	N	Stock
hasStockWeights	Index	N	StockWeight
hasStockMarket	Index	1	StockMarket
hasVolumeStock	Volume	1	Stock
hasVolumeSession	Session	1	Session
hasStock	StockWeight	1	Stock
hasIndex	StockWeight	1	Index
hasBroker	Depositary	1	Broker
hasSotckExchanges	Broker	N	StockMarket
hasStocks	Portfolio	N	Stocks
hasUserID	Portfolio	1	User
hasDepositary	Portfolio	1	Depositary
hasAssociatedAccount	Portfolio	1	Account

### Instance attribute table

Instance attribute name	Concept name	Value type	Cardinality
isParOfCompany	Stock	String	(0, 1)
isContinuous	Stock	String	(1)
hasFaceValue	Stock	String	(0, 1)
hasPiceValue	Stock	String	(0, 1)
hasStockType	Stock	String	(0, 1)
hasBestoffer	Stock	String	(0, 1)
hasBestSellPrice	Stock	Decimal	(0, 1)

Instance attribute name	Concept name	Value type	Cardinality
hasBestBuyPrice	Stock	Decimal	(0, 1)
hasMaximun	Stock	Decimal	(0, 1)
hasMinimun	Stock	Decimal	(0, 1)
hasDividend	Stock	Decimal	(0, 1)
hasName	StockMarket	String	(0, 1)
hasCountry	StockMarket	String	(0, 1)
hasBuySellDate	BuySellOrder	Date	(0, 1)
hasNumberOfStocksBuySell	BuySellOrder	String	(0, 1)
hasISIN	BuySellOrder	String	(0, )
isBuyOrSell	BuySellOrder	String	(0, 1)
isPerformed	BuySellOrder	Boolean	(0, 1)
hasExpirationDate	ConditionedBuySellOrder	Date	(0, 1)
hasPrice	ConditionedBuySellOrder	Decimal	(0, 1)
hasDate	LastPrice	Date	(0, 1)
has Value	LastPrice	Decimal	(0, 1)
hasDate	Session	Date	(0, 1)
hasOpeningTime	Session	DateAndTime	(0, 1)
hasClosingTime	Session	DateAndTime	(0, 1)
hasIndexValue	Session	Decimal	(0, 1)
hasAmount	Volume	Decimal	(0, 1)
hasValue	IndexSession	Decimal	(0, 1)
hasIndex	IndexSession	Decimal	(0, 1)
StockWeight	hasWeght	Decimal	(0, 1)
CIF	Depositary	String	(0, 1)
hasCommisions	Depositary	Decimal	(0, n)
CIF	Broker	String	(0, 1)
hasNonOfficialTaxes	Broker	Decimal	(0, n)
hasPortFololiold	Portfolio	String	(0, n)

## 6 IMPLEMENTATION

The Stock Market ontology has been implemented in WSML [3], and it has been modelled using wsmstudio [4]

Below you can see a fragment of the Stock Market ontology that represents a Stock concept

```
concept _"Stock"
  nonFunctionalProperties
```

```

dc#description hasValue {"Ownership of a corporation indicated by
shares, which represent a piece of the corporation's assets and
earnings.", "Represents one fraction of one specific company
ownership"}
endNonFunctionalProperties
_"isPartOfcompany" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "Represents one fraction of
one specific company ownership"
  endNonFunctionalProperties
_"hasSocksMarket" impliesType (1 1) _"StockMarket"
nonFunctionalProperties
  dc#description hasValue "The Market where the stock can be
sell/bought"
endNonFunctionalProperties
_"isContinuous" impliesType (1 1) _boolean
  nonFunctionalProperties
    dc#description hasValue "Reflects wether or not the
stock can be sell/bought in the Continuous Market"
  endNonFunctionalProperties
_"hasFaceValue" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "The amount expressed in
currency representing the ownership of the company"
  endNonFunctionalProperties
_"hasPriceValue" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "The price of the last
transaction for that stock in an specific market"
  endNonFunctionalProperties
_"hasStockType" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "Specific types of rights in
the company"
  endNonFunctionalProperties
_"hasBestOffer" impliesType (1 1) _"BestOffer"
  nonFunctionalProperties
    dc#description hasValue "The best price that can be
found in the market at this point of time. Two
figures are available: best selling price and best
buying price."
  endNonFunctionalProperties
_"hasBestBuyPrice" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue ""
  endNonFunctionalProperties
_"hasBestSellPrice" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue ""
  endNonFunctionalProperties
_"hasMaximun" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "The highest price in an
specific period of time"
  endNonFunctionalProperties
_"hasMinimun" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "The lowest price in an
specific period of time"

```

```
        endNonFunctionalProperties
    _"hasDividend" impliesType (1 1) _decimal
    nonFunctionalProperties
        dc#description hasValue "The amount of the last
        profit distribution that corresponds to a single
        stock"
    endNonFunctionalProperties
```

Annex III provides a complete implementation of Stock Market ontology in WSMML.

## 7 CONCLUSION

In this deliverable we present the decisions adopted during development process and design of Stock Market Ontology (part of Financial Ontology). This description shows the methodology proposed in D3.11 and also serves as documentation for the Stock Market Ontology. By providing a detailed documentation we hope to foster ontology reuse.

## REFERENCES

- [1] Corcho O, Losada S, Martínez Montes M, Bas JL, Bellido S, (2004) Financial Ontology. DIP deliverable D10.3.
- [2] Gábor Naypál (2005) Methodology for building SWS ontologies in DIP. DIP deliverable D3.11
- [3] Dimitrov, M.Simov A. Momtchev, V. Heep M. et al (2005) WSMO Studio v.1 DIP deliverable D 4.5.
- [4] M.Kerrigan. D9.1v0.2 The Web Services Modelling Toolkit (WSMT). Technical report, 2005. <http://www.wsmo.org/TR/d9/d9.1/v0.2/>
- [5] Staab S, Schnurr HP, Studer R, Sure Y (2001) Knowledge Processes and Ontologies. IEEE Intelligent Systems 16(1):26-34
- [6] De Bruinjn et al The Web Service Modeling Language WSMML, version 0.2 available at <http://www.wsmo.org/TR/d16/d16.1/v0.2/>
- [7] Fernández-López M, Gómez-Pérez A, Pazos A, Pazos J (1999) *Building a Chemical Ontology Using Methontology and the Ontology Design Environment*. IEEE Intelligent Systems & their applications 4(1):37–46

## ANNEX I. BANKINTER WEB SERVICES INPUT OUTPUT DATA

As is recommended in D 3.11 actual Web Services has been used to develop financial ontology. Next paragraphs describes this web services

- **bestCotizaciones.**

- **Description**

Given a value it retrieves the best 5 bids and asks

- **Input Parameter**

**Table 1: bestCotizaciones Input parameter.**

Name	Description	Example
Isdin	Value identificador	ES0113679338
Market	Market identificador	055

- **Output Parameter**

**Table 2: bestCotizaciones output parameter.**

Name	Description	Example
Isin	Value identificador	ES0113679338
Market	Market indentificador	055
Ticket	Ticket	BKT
Name	Value name	Bankinter
Market_desc	Market description	M.Continuo
Last_price	Last price	34,12
Tendency	Indicates if the value is increasing(+) or decreasing(-)	+
Variation	Absolute variation	<No se presenta en pantalla<
%variation	Relative variation	+1,63%
Currency	Currency	EUR, USD
Date	date	9:15 , 26/09
bid_price1	Bid price #1	34,12
bid_price2	Bid price #2	34,11
bid_price3	Bid price #3	34,1
bid_price4	Bid price #4	34,09
bid_price5	Bid price #5	34,08
ask_price1	Ask price #1	34,19
ask_price2	Ask price #2	34,2
ask_price3	Ask price #3	34,22

ask_price4	Ask price #4	34,23
ask_price5	Ask price #5	34,24
Bid_volume1	Bid volume #1	1.646
Bid_volume2	Bid volume #2	198
Bid_volume3	Bid volume #3	8
Bid_volume4	Bid volume #4	1.060
Bid_volume5	Bid volume #5	1.500
Ask_volume1	Ask volume #1	2.823
Ask_volume2	Ask volume #2	783
Ask_volume3	Ask volume #3	1.500
Ask_volume4	Ask volume #4	515
Ask_volume5	Ask volume #5	200

- mainIndexes

- **Description**

Retrieves the most important indexes information

- **Input parameter**

This operation not need input parameter.

- **Output Parameter**

**Table 3: mainIndexes output parameter**

Name	Description	Example
Isin	Index identificador	IBEX
Market	Market identificador	055
Name	Index name	IBEX35
Last_price	Last price	7734,80
Tendency	Indicates if the value is increasing(+) or decreasing(-)	+,-
Variation	Absolute variation	-34,47
%variation	Relative variation	-0,94
Date	date	12:04

- cotizacionIndice

- **Description**

Given an index, retrieves its detailed information

- **Input parameter**

**Table 4 cotizacionIndice input parameter.**

Nombre	Descripción	Example
Action	Type of action to be performed	accion=indice
Isin	Index identificator	IBEX
market	Market identificator	055

- **Output parameter**

**Table 5: cotizacionIndice output parameter**

Name	Description	Example
Isin	Index identificador	ibex
Market	Market identificador	055
Name	Index name	Ibex35
Market_desc	Market description	M.Continuo
Last_price	Last price	7734,80
Tendency	Indicates if the value is increasing(+) or decreasing(-)	+,-
Variation	Absolute variation	<NO SE PRESENTA>
%variation	Relative variation	<NO SE PRESENTA>
Date	Date	9:15 , 26/09
Last_session_price	Last session price	7893,45
Max_session	Session's maximum value	7612,34
Min_session	Session's minimum value	7893,25

- relevantVariations

- **Description**

Retrieves the relevant variations of a given index

Retrieves the 5 bigger increases and the 5 bigger decreases for all the indexes excepting the IBEX35 that will show all the values ordered from the biggest increase to the biggest decrease

- **Input parameter**

**Table 6: relevantVariations input parameter**

Name	Description	Example
Action	Type of action to me performed	accion=valores

Isin	Index identificador	ES0113679338
Market	Market identification	055
Order	Type of order to be performed	

- **Output parameter**

**Table 7: relevantVariations outputparameter**

Name	Description	Example
Name	Index name	EUR
Isin	Index identificador	IBEX
Market	Market identificador	055
Tiker	ticker	ACC
Name	Ticket name	ACCIONA
Market_desc	Market description	M CONTINUO
Last_price	Last price	12,23
Tendency	Indicates if the value is increasing(+) or decreasing(-)	+
Variation	Absolute variation	1,12
%variation	Relative variation	-1,25%
Currency	Currency	EUR, USD
Date	date	9:15 , 26/09

- searchValue

- **Description**

Given a market and an search string it searches all the values that fit the given pattern in the given market. If the market is empty the search is made all over the existing markets.

### Input parameter

**Table 8: searchValue input parameter**

Name	Description	Example
<b>Market</b>	Market code	<b>055</b>
<b>Name</b>	User's input search string	<b>pule</b>

### Output parameter

**Table 9: searchValue output parameter**

Name	Description	Example
Isin	Value identificador	ES0172233118

Market	Market code	055
ticker	Ticket	BIO
Name	Value's name	PULEVA BIOTECH
Market	Market's name	MERCADO CONTINUO
Last_price	Last price	3,11
Tendency	Indicates if the value is increasing(+) or decreasing(-)	-
Variation	Absolute variation	-
%variation	Relative variation	-
Currency	Currency code	EUR
Date	date	07/07

- news

- **Description**

Retrieves the new's headlines of a given value

- **Input parameter**

**Table 10: getNews input parameter**

Name	Description	Example
<u>isin</u>	Value's isin	ES0113679338
<u>mercado</u>	Market code	<u>055</u>

- **Output parameter**

**Table 11: getNews output parameter**

Name	Description	Example
isin	Value's identification	ES0113679338
Market	Market code	055
ticker	Ticker	BKT
Name	Value's name	BANKINTER
Market_desc	Market description	MERCADO CONTINUO
Last_price	Last price	131,70
Tendency	Indicates if the value is increasing(+) or decreasing(-)	+
Variation	Absolute variation	99,67

%variation	Relative variation	311,18
Currency	Value's currency	EUR
Date	Date	06/17
Identificatir	New identificador	T20040625131111
Headline	New headline	Economía/Finanzas.- Bankinter lanza la tarjeta de telepeaje para pagos en autopistas y radiales

- newsDetail

- Description**

Retrieves the detail data of a given new.

- Input parameter**

Table 12: news Detail input parameter

Name	Description	Example
<u>Identificator</u>	Requested new id	20040615114031
<u>Isin</u>	Value's isin	ES0113679338
<u>Market</u>	Market code	<b>055</b>

- Output parameter**

Table 13: news Detail output parameter

Name	Description	Example
isin	Value's isin	ES0113679338
Market	Market code	055
ticker	Ticker	BKT
name	Value's name	BANKINTER
Market_desc	Market's name	MERCADO CONTINUO
Last price	Last price	131,70
Tendency	Indicates if the value is increasing(+) or decreasing(-)	+
variation	Aboslute variation	99,67
%variation	Relative variation	1,18
currency	Value's currency	EUR
date	date	06/17
text	Auxiliar text	EUR

headline	New's headline	"....."
New_detail	New's text	"....."

- performBuySell

- **Description**

This method performs a buy or sell operation of a certain number of stocks.

- **Input parameter**

**Table 14: performBuySell input Parameter**

Name	Description	Example
Isin	Value identifier	BKT
mercado	Market identifier	055
numStocks	Number of stocks to be sold or bought	59
buyOrSell	Parameter indicating if a sell or buy has to be performed	B/S
portfolioId	Portfolio you want to buy/sell stocks from	01280010012548
userId	User identifier	6587469

- **Output Parameter**

**Table 15: performBuySell output Parameter**

Name	Description	Example
ReturnCode	Indicates if the operations has been successful or not	OK/KO
price	Final price of the operation in the market currency	42.00
Currency	currency	EUR
Message	Error message if needed	

- getMyPorfolio

- **Description**

Returns the composition of a given user's portfolio

- **Input parameter**

**Table 16: getMyPorfolio input parameter**

Name	Description	Example
------	-------------	---------

userId	User identifier	6587469
portfolioId	Portfolio you want the composition of	01280010012548

- **Output Parameter**

**Table 17: getMyPorfolio output parameter**

Name	Description	Example
portfolioId	Portfolio you want the composition of	01280010012548
balance	Balance of the portfolio in euros	1256
(table)		
Isin	Value identifier	BKT
numStocks	Number of stocks	300
priceAdquired	Price the costumer bought the stocks	40.00
Currency	Stock currency	EUR

- getPortfolios

- **Description**

Returns all the user's portfolios

- **Input parameter**

**Table 18: getPortfolios input parameter**

Name	Description	Example
userId	User identifier	6587469

- **Output Parameter**

**Table 19: getPortfolios output parameter**

Name	Description	Example
(table)		
portfolioId	Portfolio you want the composition of	01280010012548

- sendAlert

- **Description**

Sends an alert to a given user using a parameterized channel

- **Input parameter**

**Table 20: sendAlert input parameter**

Name	Description	Example
Channel	Channel to be used	Email/sms
contactData	Phone number or email address, depending on the previous parameter	<a href="tel:+34629786364">+34629786364</a> / <a href="mailto:user.dummy@domain.com">user.dummy@domain.com</a>
Message	Message to be sent	Bankinter is over 42.00. Goldman Sachs rating has changed: A+

- **Output Parameter**

**Table 21: sendAlert input parameter**

Name	Description	Example
returnCode	Indicates if the message has been sent successfullt	OK/KO

- doCurrencyTransformation

- **Description**

Given two currencies and an amount it calculates the conversion of the amount between the two currencies

- **Input parameter**

**Table 22: doCurrencyTransformation input parameter**

Name	Description	Example
currencyIn	Currency input	USD
amount	Amount to be converted into currencyOut	1
currencyOut	Currency output	EUR

- **Output Parameter**

**Table 23: doCurrencyTransformation output parameter**

Name	Description	Example
Amount	conversion	1.25
conversionRate	Conversion rate used	1.25

## ANNEX II: OWL CURRENCY MEDIATOR:

This mediator transforms currency ontology in OWL format into WSML format and allows to use this ontology into financial ontology.

```

namespace { _"http://users.isoco.net/~slosada/ontologies/bankinter/owlCurrencyMediator.wsml#",
  dc _"http://purl.org/dc/elements/11#",
  xsd _"http://www.w3c.org/2001/XMLSchema#",
  wsml _"http://www.wsmo.org/2004/wsml#"}

ooMediator _"http://users.isoco.net/~slosada/ontologies/bankinter/owlCurrencyMediator.wsml"
nonFunctionalProperties
  dc#description hasValue "This ooMediator transforms the owl currency
  description into wsml ontology and allows use it in Financial Ontology."
  dc#type hasValue _"http://www.wsmo.org/2004/d2/#ooMediator"
  wsml#version hasValue "$Revision: 1.0 $"

endNonFunctionalProperties
source _"http://www.daml.ecs.soton.ac.uk/ont/currency.daml"
target _"http://users.isoco.net/~slosada/ontologies/bankinter/FinancialOntology.wsml"
usesService http://138.232.65.151:8080/TranslatorService/OWL2WSML/

```

## ANNEX III: WSML SINTAXIS.

```

namespace {
  _"http://users.isoco.net/~slosada/ontologies/bankinter/StockMarket.wsml
  #"
  ,
  xsd _"http://www.w3c.org/2001/XMLSchema#" ,
  dc _"http://purl.org/dc/elements/1.1#" ,
  foaf _"http://xmlns.com/foaf/01/" ,
  wsml _"http://www.wsmo.org/2004/wsml#" ,
  dt _"http://www.wsmo.org/ontologies/dateTime/#" ,
  cu
  _"http://www.wsmo.org/2004/d3/d3.2/v0.1/20040628/resources/owlCurrencyM
  ediator.wsml#" ,
  financial
  _"http://users.isoco.net/~slosada/ontologies/bankinter/FinancialOntolog
  y.wsml#" }

ontology
  _"http://users.isoco.net/~slosada/ontologies/bankinter/StockMarket.wsml
  #"

concept _"Stock"
  nonFunctionalProperties
    dc#description hasValue {"Ownership of a corporation
  indicated by shares, which represent a piece of the corporation's
  assets and earnings.", "Represents one fraction of one specific company
  ownership"}
  endNonFunctionalProperties
  _"isPartOfcompany" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "Represents one fraction of one
  specific company ownership"
  endNonFunctionalProperties
  _"hasSocksMarket" impliesType (1 1) _"StockMarket"

```

```

    nonFunctionalProperties
      dc#description hasValue "The Market where the stock can be
sell/bought"
    endNonFunctionalProperties
    _"isContinuous" impliesType (1 1) _boolean
  nonFunctionalProperties
    dc#description hasValue "Reflects wether or not the stock can
be sell/bought in the Continuous Market"
  endNonFunctionalProperties
  _"hasFaceValue" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "The amount expressed in currency
representing the ownership of the company"
  endNonFunctionalProperties
  _"hasPriceValue" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "The price of the last transaction
for that stock in an specific market"
  endNonFunctionalProperties
  _"hasStockType" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "Specific types of rights in the
company"
  endNonFunctionalProperties
  _"hasBestOffer" impliesType (1 1) _"BestOffer"
  nonFunctionalProperties
    dc#description hasValue "The best price that can be found in
the market at this point of time. Two figures are available: best
selling price and best buying price."
  endNonFunctionalProperties
  _"hasBestBuyPrice" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue ""
  endNonFunctionalProperties
  _"hasBestSellPrice" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue ""
  endNonFunctionalProperties
  _"hasMaximun" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "The highest price in an specific
period of time"
  endNonFunctionalProperties
  _"hasMinimun" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "The lowest price in an specific
period of time"
  endNonFunctionalProperties
  _"hasDividend" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "The amount of the last profit
distribution that corresponds to a single stock"
  endNonFunctionalProperties
concept _"StockMarket"
  nonFunctionalProperties
    dc#description hasValue {"The market to buy or sell shares."}
  endNonFunctionalProperties
  _"hasStoks" impliesType (1 1) _"Stock"

```

```

    nonFunctionalProperties
      dc#description hasValue "Stocks that can be sell/bought
there."
    endNonFunctionalProperties
    _"hasIndexes" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "Figures that reflect the overall
trend of the market or part of it"
  endNonFunctionalProperties
  _"hasName" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "The name of the market"
  endNonFunctionalProperties
  _"hasCurrency" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "The currency in wich the stock
prices are expressed"
  endNonFunctionalProperties
  _"hasBrokers" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "The persons/companies allowed to
operate in the market"
  endNonFunctionalProperties
  _"hasSessions" impliesType (1 *) _string
  nonFunctionalProperties
    dc#description hasValue "The dates the market has been/will
be opened"
  endNonFunctionalProperties
  _"hasCountry" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "The country where the market is
established"
  endNonFunctionalProperties

concept _"BuySellOrder"
  nonFunctionalProperties
    dc#description hasValue "Single transaction in which one part
sells an amount of shares of a company at a price an the other buys"
  endNonFunctionalProperties
  _"hasBuySellDate" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "The date and time in which the
operation has been performed"
  endNonFunctionalProperties
  _"hasNumberOfStockBuySell" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "Amount of stocks dealt in the
operation"
  endNonFunctionalProperties
  _"hasISIN" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "The identifier of the stocks
sell/bought"
  endNonFunctionalProperties
  _"isBuyOrSell" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "Determines the type of operation"
  endNonFunctionalProperties
  _"hasCurrency" impliesType (1 1) _string

```

```

    nonFunctionalProperties
      dc#description hasValue "The currency used to express the
amount of the operation. Both the transaction currency and the
portfolio currency are required"
    endNonFunctionalProperties
    _"hasSession" impliesType (1 1) _"Session"
  nonFunctionalProperties
    dc#description hasValue "The secuencial number of session"
  endNonFunctionalProperties
  _"isPerformed" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "Wether the order is performed or
not"
  endNonFunctionalProperties
  _"hasDepositary" impliesType (1 1) _"Depositary"
  nonFunctionalProperties
    dc#description hasValue "The person/company in charge of
keeping safe the stocks"
  endNonFunctionalProperties
  _"hasPortfolioNumber" impliesType (1 1) _"Porfolio"
  nonFunctionalProperties
    dc#description hasValue "The account number (defined by the
depository) where the stocks are kept"
  endNonFunctionalProperties
  _"hasUserID" impliesType (1 1) _"User"
  nonFunctionalProperties
    dc#description hasValue "Determines the property of the
shares"
  endNonFunctionalProperties

concept _"BestBuySellOrder" subConceptOf _"BuySellOrder"
  _"executedInSession" impliesType (1 1) _"Session"
  nonFunctionalProperties
    dc#description hasValue "The secuencial number of session"
  endNonFunctionalProperties

concept _"ConditionedBuySellOrder" subConceptOf _"BuySellOrder"
  _"hasExpirationDate" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "The date in which the order is
cancelled if not performed before"
  endNonFunctionalProperties

concept _"LastPrice"
  nonFunctionalProperties
    dc#description hasValue "Value of the last transaction in a
session of a single share"
  endNonFunctionalProperties
  _"hasDate" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "Date and time of historical values"
  endNonFunctionalProperties
  _"hasValue" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "Historical values"
  endNonFunctionalProperties

concept _"Session"
  nonFunctionalProperties

```

```

        dc#description hasValue "Time unit from the opening to the
closure of the market. Usually one working day"
    endNonFunctionalProperties
    _"hasDate" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue "The date of the session"
    endNonFunctionalProperties
    _"hasValue" impliesType (1 1) _decimal
    nonFunctionalProperties
        dc#description hasValue "The amount of shares traded in one
period of time"
    endNonFunctionalProperties
    _"hasClosingTime" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue "The time at what the market is
closed"
    endNonFunctionalProperties
    _"hasOpeningTime" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue "The time at what the market is
opened"
    endNonFunctionalProperties
    _"hasVolume" impliesType (1 1) _"Volume"
    nonFunctionalProperties
        dc#description hasValue ""
    endNonFunctionalProperties
    _"hasStocks" impliesType (1 1) _"Stock"
    nonFunctionalProperties
        dc#description hasValue "Weighted average of a number of
stock prices"
    endNonFunctionalProperties

concept _"Index"
    nonFunctionalProperties
        dc#description hasValue "Weighted average of a number of
stock prices"
    endNonFunctionalProperties
    _"hasName" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue "Index name (i.e.;Nasdaq, Dow Jones)"
    endNonFunctionalProperties
    _"hasInitialValue" impliesType (1 1) _decimal
    nonFunctionalProperties
        dc#description hasValue "The value of the index when a
session starts"
    endNonFunctionalProperties
    _"hasActualValue" impliesType (1 1) _decimal
    nonFunctionalProperties
        dc#description hasValue "The latest value of the index"
    endNonFunctionalProperties
    _"hasIndexID" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue "Index identification"
    endNonFunctionalProperties
    _"hasStockWeights" impliesType (1 1) _"StockWeight"
    nonFunctionalProperties
        dc#description hasValue "The stocks that compose the index
and weight in index"

```

```

endNonFunctionalProperties
_"hasMarket" impliesType (1 1) _"StockMarket"
nonFunctionalProperties
  dc#description hasValue "The market to what the index refers"
endNonFunctionalProperties

concept _"Volume"
  _"hasAmount" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue ""
  endNonFunctionalProperties
  _"hasVolumeStock" impliesType (1 1) _"Stock"
  nonFunctionalProperties
    dc#description hasValue ""
  endNonFunctionalProperties
  _"hasVolumeSession" impliesType (1 1) _"Session"
  nonFunctionalProperties
    dc#description hasValue ""
  endNonFunctionalProperties

concept _"IndexSession"
  _"hasIndex" impliesType (1 1) _"Index"
  nonFunctionalProperties
    dc#description hasValue "Weighted average of a number of
stock prices"
  endNonFunctionalProperties
  _"hasValue" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "Value of index in a session"
  endNonFunctionalProperties

concept _"StockWeight"
  _"hasWeight" impliesType (1 1) _decimal
  nonFunctionalProperties
    dc#description hasValue "Weight of stock in index"
  endNonFunctionalProperties
  _"hasIndex" impliesType (1 1) _"Index"
  nonFunctionalProperties
    dc#description hasValue "Weighted average of a number of
stock prices"
  endNonFunctionalProperties
  _"hasStock" impliesType (1 1) _"Stock"
  nonFunctionalProperties
    dc#description hasValue "Stotk that referennces the index"
  endNonFunctionalProperties

concept _"Depositary"
  nonFunctionalProperties
    dc#description hasValue "The person or company in charge of
keeping safely the stocks of a client."
  endNonFunctionalProperties
  _"hasDepositaryID" impliesType (1 1) _string
  nonFunctionalProperties
    dc#description hasValue "The identificacion of the
Depositary"
  endNonFunctionalProperties
  _"CIF" impliesType (1 1) _string
  _"hasCommissions" impliesType (1 1) _string
  nonFunctionalProperties

```

```

        dc#description hasValue "The commissions the Depository
applies"
    endNonFunctionalProperties
    _"hasBroker" impliesType (1 1) _"Broker"
    nonFunctionalProperties
        dc#description hasValue "The brokers the depository works
for"
    endNonFunctionalProperties

concept _"Broker"
    _"CIF" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue ""
    endNonFunctionalProperties
    _"hasNonOfficialTaxes" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue ""
    endNonFunctionalProperties
    _"hasStokcMarkets" impliesType (1 1) _"StockMarket"
    nonFunctionalProperties
        dc#description hasValue ""
    endNonFunctionalProperties
concept _"Portfolio"
    nonFunctionalProperties
        dc#description hasValue "The number (assigned by the
Depository) where the stocks are deposited"
    endNonFunctionalProperties
    _"hasPortFolioID" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue "The number (assigned by the
Depository) where the stocks are deposited"
    endNonFunctionalProperties
    _"hasStocks" impliesType (1 1) _"Stock"
    nonFunctionalProperties
        dc#description hasValue "Identificaction of the Stocks
deposited in a Portfolio"
    endNonFunctionalProperties
    _"hasUserID" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue "Identification of the client owner
of the stocks deposited"
    endNonFunctionalProperties
    _"hasDepository" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue "Identification of the depository"
    endNonFunctionalProperties
    _"hasAssociatedAccount" impliesType (1 1) _string
    nonFunctionalProperties
        dc#description hasValue "Account asociated to this portfolio"
    endNonFunctionalProperties

concept _"StockMaxValue"
    _"initPeriod" impliesType (1 1) _date
    nonFunctionalProperties
        dc#description hasValue ""
    endNonFunctionalProperties
    _"endPeriod" impliesType (1 1) _date
    nonFunctionalProperties
        dc#description hasValue ""

```

```
endNonFunctionalProperties
_"value" impliesType (1 1) _decimal
nonFunctionalProperties
  dc#description hasValue ""
endNonFunctionalProperties

concept _"Dividend"
_"amount" impliesType (1 1) _decimal
nonFunctionalProperties
  dc#description hasValue ""
endNonFunctionalProperties
_"date" impliesType _date
nonFunctionalProperties
  dc#description hasValue ""
endNonFunctionalProperties
```