A XBRL Financial Virtual Assistant

Adalberto Alves Abraão
Programa de Pós-Graduação em Sistemas e Computação
Universidade Salvador - Unifacs
Salvador, Brazil
e-mail: adalbertoabraao@gmail.com,

Paulo Caetano da Silva
Programa de Pós-Graduação em Sistemas e Computação
Universidade Salvador - Unifacs
Salvador, Brazil
paulo.caetano@pro.unifacs.br

***Abstract*— Nowadays, by means of stock exchanges, it is possible to invest in globally located enterprises. It is a market in which it is possible to obtain great profits, but it also conveys great risks of loss. In order to decrease risks, it is important to know the financial strength of the enterprises, through specific indicators, to ask what their values are, for example, "What is the leverage ratio of the company X currently?". Financial consultancy emerges for that need, but it is very expensive. An alternative is to search for the answers by themselves on enterprises websites or from the regulatory agencies. Such tasks require time, calculations and analyses. With the aim at facilitating the obtaining of financial information about the enterprises for any investors, a virtual financial assistant is presented in this paper which answers to the questions of financial type made by the user, interacting by means of a natural language. The assistant utilizes financial information from analytic Online Analytical Processing (OLAP) queries at eXtensible Business Reporting Language (XBRL) database. The architecture of the virtual assistant is presented as well as details a prototype's implementation. As a result, we expect that the assistant correctly answers the user's questions, asked in a natural language and related to financial indexes of given enterprises.**

Keywords- Virtual Assistant; XBRL Financial Virtual Assistant; Financial Virtual Assistant; LDQM application; NLP application; Virtual Financial Assistant Architecture.

I. INTRODUCTION

In a globalized world, electronic interconnections, by means of telecommunication nets and information systems are crucial for the global financial transactions. Stock market is one of the sectors in which the investors act almost entirely online. An environment where, nowadays, it is possible to operate in almost all stock exchanges throughout the world from a computational system with internet access. It is a high-risk market that can provide a high rate of profit over investments but also a great loss. In order to increase success rate, it is vital that the analyses for the decision making are correct and based on authentic and as much as possible updated information. Part of that analysis consists of the identification of the enterprises financial strength, while suggesting typical questionings such as: "What is the leverage ratio of the company X currently?"; "What is the net profit of the company Z in the second quarter of the year Y?" and so on.

Many investors search for the most diverse financial information enterprises by means of reports published online in electronic format Portable Document Format (PDF), HyperText Markup Language (HTML) or text. Many regulatory agencies are currently demanding that those reports be published in XBRL [15], a technology that have features that enable to identifies the fraudulent manipulations more quickly, among other functionalities.

Normally, the investors pay for expensive services of financial assistance in order to obtain information and complete consultancy. However, for retail investors, which mostly do not have assistants, collecting and analyzing information present far greater risks. The available virtual assistants on the web or on smartphones could be alternatives to financial consultancy. However, in general, these software are proprietary and they do not give satisfactory answers to the financial queries.

The non-existence of a virtual financial assistant, capable of being an alternative to a financial consultancy to a retail investor, motivated the creating a virtual financial assistant. A computational system capable of answering questions in a natural language utilizing the financial jargon based on data from reliable financial reports and of high availability.

So, the aim of this paper is to present the architecture and a prototype of a virtual financial assistant which answers to the user's financial questions with XBRL database, interacting by a natural language.

The rest of the article is organized as follows: Section II describes papers and implementations related to virtual assistants. Section III presents the architecture of the proposed virtual financial assistant. Section IV presents the prototype implementation details and its evaluation. Section V shows the conclusion.

II. VIRTUAL ASSISTANTS

The term Virtual Assistant can have many nomenclatures and denominations. For Paraiso et al. [2], Personal Assistants are agents who help people to perform their daily chores. In another work, Zambiasi et al. [9], the authors denominate as

Personal Assistant Software (PAS), software that help people with their daily activities. They affirm that this kind of software has several denominations and characteristics identified by various authors and one can hardly reach a final definition.

The term Embodied Conversational Agent (ECA) is defined by Eisman et al. [3] as an intelligent system represented by a character capable of getting involved in a conversation with a human being. Helping the user to accomplish a given set of tasks is its main function.

The various denominations are directly linked to the actions performed by the assistants. In this regard, Medina et al. [6] affirms that these denominations distinguish the existent assistants as of general purpose and of specific domain. The first ones are useful for providing information about climate, about how to get to a given location or even for performing some actions such as making an appointment, elaborating and sending a message. The second ones serve to answer the questions about a specific domain. Next, works related to this article are cited and analyzed.

The project of a personal assistant with interaction via voice through dialogs in natural language which helps the user in a governmental system was described by Paraiso et al. [2]. They have created a personal assistant that is used as an interface among users and a multi-agent system. Its architecture was divided in three main parts: interaction with the user, performed through Graphic Speech User Interface modules (GSUI); the processing of expressions, performed by the Linguistic Modules (LU) that, supported by ontologies, interprets user interactions by means of syntactic analysis and conversions; dialog management, performed by Agency Modules (AM) which activates the external events execution by means of the agents and controls the dialog and the assistance.

There is a voice conversation interface which allows the user to interact with the software using their own terms in addition to a taxonomy to deal with the users expressions, allowing to analyze if the sentence is well formed and in accordance with the grammatical structure. A personal assistant that can activate service agents locally is dedicated to each user. Those agents, in turn, can delegate sub-tasks to other service agents in order to complete a complex task. The agents are independent and exchange information with each other and with the personal assistant.

The user interface architecture and similar requirements of the proposed personal assistant, like the limited and well-known domain, presented by Paraiso et al. [2] have been contributions to this work.

A framework for projecting virtual assistants of multilingual closed domain that can be integrated to websites was presented by Eisman et al. [3]. The knowledge, i.e., domain, is stored in regular expressions and in an ontology. The regular expressions have the function of construct a syntax to facilitating the acknowledgement of the user queries. Ontology, in turn, supports the choice process of the next action to be performed and the adequate answer creation. That system was designed by using a client-server architecture and it was implemented in three modules The Natural Language Understander (NLU) which recognizes the questions elaborated by the user; The Dialog Manager (DM), which determines what, when and how the assistant is supposed to do; And the Communication Generator (CG), responsible for generating a specific answer that encompasses an action.

The authors presented a virtual assistant implementation based on the framework that supplies information about courses and services offered by a university. Although the framework has been designed to support assistants which help with the navigation in private websites, it has served as an inspiration for the proposed Financial Virtual Assistant.

Zambiasi et al. [9] presented a conceptual model and reference architecture for personal assistance software, which according to them, are inter-agent software and also integrated to corporative business environments, adaptive and inter-operatives with various sub-systems and sources of information and activities. The model and the architecture presented are based on Service Oriented Architecture (SOA).

PAS behaviors are a composite of services available on the web and they are chosen by the user to define the activities which the assistant may execute. The architecture proposed by Zambiasi et al. [9] facilitates the addition of new behaviors to PAS in a dynamic manner. All in all, the assistance that will be performed by PAS it will depend on the behaviors it possesses, and those, in turn, are included through the orchestration or configuration of the available selected services. Although the assistant, focus of this study, does not presume to be a PAS, the study greatly contributes for this work mainly in defining general architectural requirements for the assistant implementation and also with the identification of its main elements. A disadvantage is related to PAS do not process user queries made in natural language.

Within the study field of virtual assistance software there are important academic advances, the same way there are advances on the software development market. Currently, it is possible to perceive the utilization of virtual assistants in various situations, , on social nets, car selling websites, also bookstores and integrated in the smartphones operational systems.

Sys Virtual Assistant [14] by Synthetic Intelligence Network is a virtual assistant software of general purpose, whose main objective is helping users with the interaction man-machine, human-computer interaction (HCI). It works completely in the local machine, without sending data for any server. That feature is relevant in favor of data privacy and the security of the users. Its architecture is based on an Natural Language Processor (NLP) interacting with a collection of modules, or plugins, responsible for the functionalities. It is possible to modify, include or exclude modules, which allows a developer to configure the behavior of the assistant and thus define a new purpose. The utilization of the NLP facilitates the task of mapping the user phrases to perform the assistant software functions.

Although the type of sys virtual assistant is general purpose, one of its great advantages is the possibility of increase its knowledge through the addition of new plugins, which allows the creation and addition of a financial knowledge module. That resource is supported by the platform also developed by SYN network called The Syn Engine [14] that does not require a great computational power to work. It is available through user licenses among which there is a free version. However, there are some disadvantages of the Sys Virtual Assistant such as not being multilingual, and there are only two compatible software environments currently.

The Assistant.ai [13] by Speaktoit Inc is also an assistant of general purpose. It answers not only to questions made by the user but it also suggests assistance based on information from the schedule and its present location. It is multilingual and multiplatform and it can be used in several operational systems or in a given web browser. One of the disadvantages of this assistant is the most advanced features are only available on the charged version. Furthermore, some private user information, such as localization and names on its contacts list are sent to the enterprise servers in order to create the context environments, what may infringe issues related to the user information privacy.

One of the most interesting assistant features is supported by the Application Programming Interface (API) in which it was developed. It is called Api.ai [12] and it is composed by three modularized components: speech recognition, natural language understanding (NLU) and Conversation Management (CM), and user-fulfillment. The NLU and the CM are part of the platform's main component, responsible for knowledge manipulation and part of dialog management. The NLU allows the insertion of queries user patterns in order to generate the domain grammar, as well as a dictionary of terms and their synonyms. A relevant contribution of that platform for this work is the capacity of creating new domains, defining behaviors and thus, generating assistants of specific purpose.

There are several other available assistants on the web or integrated in the main operational mobile systems, but none of them corresponds to the financial scope proposed in this work.

III. VIRTUAL FINANCIAL ASSISTANT ARCHITECTURE

The architecture of the virtual financial assistant was designed for a distributed environment. It is composed by four layers: Presentation; Orchestration; Understanding / Knowledge; and Data.

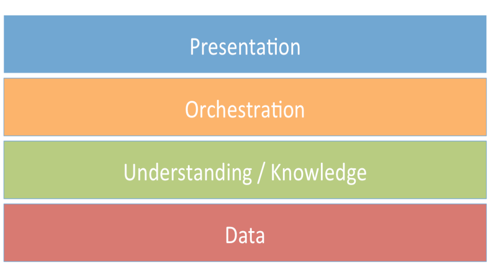

Figure 1. Architecture of the Virtual Financial Assistant in layers.

Figure 1 illustrates the assistant architecture, whose layers are discussed as follows.

Presentation: user layer interface. Responsible for dealing with the users events and for directing them to the Orchestration layer. It also has the responsibility of receiving the information returned from the Orchestration layer and presenting them to the user in an adequate format: text, image, animation, speech.

Orchestration: it is the layer that coordinates the assistant. It accesses the services of the Understanding / Knowledge in order to fulfill the requirements of the Presentation layer.

Understanding / Knowledge: that layer represents the knowledge domain. It is responsible for dealing with the understanding of the questions elaborated by the users and for the knowledge manipulation. It accesses the data layer in order to respond to the requirements of the Orchestration layer.

Data: Layer that represents the data repositories that can be a database or a set of documents, i.e., eXtensible Markup Language (XML), JavaScript Object Notation (JSON). It makes available data for the layer Understanding / Knowledge.

The components of each layer will be described below, according to what has been shown in the Figure 2.

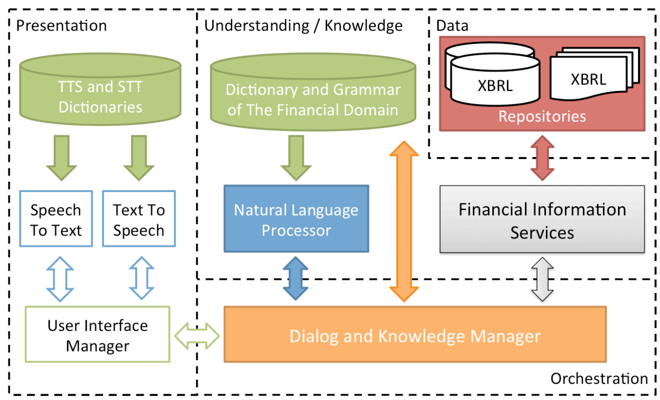

Figure 2. Virtual Financial Assistant Architecture.

A. *Presentation Layer*

It is the layer responsible for the interaction of the assistant with the user. It is formed by the following components: Speech To Text (STT); Text To Speech (TTS), TTS and STT Dictionaries and The User Interface Manager;

1) The Speech To Text: It is the responsible for the user speech recognition (encapsuled in audio streaming) and for text conversion. Normally, it is multilingual, funcionality suportted by dictionaries of specific word/voice for each language. It communicates with The User Interface Manager (UIM). Some of the factors that determine its quality are the precision and the velocity of voice recognition. If it is not obligatory that the assistant has support for speech recognition, the STT is optional.

2) The Text To Speech: it is responsible for converting a text into speech. Similar to STT, it accesses dictionaries of specific word/voice for each language to make multilingual conversions available. It serves the UIM. The TTS is obligatory only when it is pre-established that the assistant interacts with the user by means of voice. Some factors determine its quality, such as pronunciation and intonation in speech performance.

3) The TTS and STT Dictionaries: They make available data in order to facilitate the correct identification of the words embedded in the voice streaming or to facilitate the correct creation of the voice streaming. They are elaborated specifically for each language. The bigger the quantity of available dictionaries, the bigger the multilingual capacity of dealing with the speech of the assistant.

4) The User Interface Manager: It is responsible for collecting the information inserted by the user and directs them to The Dialog and Knowledge Manager; in addition, it is responsible for obtaining its answer and returning it to the user in an adequate format, such as: text, image, animation and speech.

B. Understanding / Knowledge Layer

It is the layer responsible for understanding the questions elaborated by the user, resolving them and directing the answers to the layer Orchestration. The Natural Language Processor, which deals with the sentences in natural language made by the user; Dictionary and Grammar of the financial domain, which define the syntax rules for the sentences of the user and the dictionary of financial terms domain; Financial Information Services, which provide specialized financial information are constituent parts of it.

1) The Natural Language Processor: Its main responsibility is to understand the user sentence, made in a natural language and to extract information from it, which allows a computational system to execute proceedings in response to it. In order to manipulate knowledge, NLP depends on the dictionary and the domain grammar, that is, whenever receiving one expression to be processed, it consults them in order to understand what the user had sentenced.

In case NLP does not find a pattern coherent with the question made, it warns The Dialog and Knowledge Manager (DKM) that it could not recognize the user expression. If it could understand the expression, the process result of conversion is returned to the DKM, by means of pre-established and known parameters.

2) Dictionary of The Financial Domain: It defines which words of the financial domain vocabulary are expected. In addition, it allows the definition of synonyms for the terms, in order to facilitate the process of understanding. Words, names of artifacts, documents, indexes, words related to financial domain must be represented in the dictionary.

3) Grammar of The Financial Domain: it defines the syntax of the sentences, that is which patterns of queries that can be elaborated by the user and which parameters will be extracted from them. The more representative and integrated the dictionary and the grammar are, the more precise it will be the assistant towards understanding what the user is asking in relation to the domain.

4) The Financial Information Services (FIS): They are responsible for dealing with the data and for making available information from the financial domain required by DKM. One example of dealing with data can be accessing data, calculating indicators and analyzing the financial risks.

The quantity, the versatility and the complexity of the information made available by this consulting service are the main responsible features for the level of sophistication of the answers given by the assistant. They can be made available by internet service providers of financial information widespread at any place by means of web service. One of the premises is that they have to be fast, once the consultations need answers within seconds.

C. The XBRL Repositories Layer

The XBRL Repositories is the layer responsible for supplying the financial data. It can be formed by XBRL instances or by XBRL relational database.

These repositories must be made available and administered periodically by reliable institutions, fact that will guarantee that the answers are based in current and authentic data.

D. The Orchestration Layer

It is the layer that coordinates the whole system and it is composed by the Dialog Knowledge Manager, component responsible for keeping a dialog with the user and for managing knowledge. This layer is the behavior manager of the assistant.

1) The Dialog and Knowledge Manager

The Dialog and Knowledge Manager (DKM) is the coordinator of the whole system. It is responsible for keeping a dialog with the user similarly to a conversation between people. A typical transaction of the DKM starts right after receiving a sentence of the user sent by the UIM. It sends the message to the NLP and waits for the parameters of the answer. In accordance with those parameters, it will select and request the available FIS services. After receiving the messages from the selected services, it builds up the answer and forwards it to the UIM, closing the transaction and waiting for the next solicitation.

When the NLP gives a signal that the user sentence was not recognized, the DKM does not activate any services. Only it informs the user about what has occurred via UIM.

DKM allows the assistant to incorporate new pieces of knowledge, as in the case of learning a new term of domain. The addition of new patterns or behaviors can be dynamically done. The questioning patterns for the services that will be activated in order to respond to them are mapped inside the DKM. In short, it defines the assistant behaviors through the management of the grammar, of the domain dictionary and of the services selection which will supply the adequate information. It is the master of knowledge manipulation.

IV. VIRTUAL FINANCIAL ASSISTANT IMPLEMENTATION

Many software companies provides STT and TTS components in various forms, either by services or by browser plugins, and the most of them are free. Two clients were implemented: a mobile application and a web page application. The android mobile operational system was chosen because it provides the STT and TTS components for free. The software development environment for it is free and is installed at the majority of mobile devices. The web page application was developed in Javascript language making the assistant available for any Operational System.

There are some NLPs currently available, many of them proprietary ones. The selection criteria utilized for chose the NLP, as shown in Table I, were:

1. Available functions /API to create new domains;
2. Provides questions grouping and customized parameters as a service;
3. Multilingual;
4. Availability for several platforms;
5. Proprietary solution;
6. Free license charge for evaluation and study;
7. Maintenance of the Grammar and the dictionary without code updates.

For the presented proposition in this job, the API.AI NLP [12] was utilized.

The more typical expressions and terms of financial jargon are entered in grammar and vocabulary, respectively, the greater the NLP ability to understand the financial user queries. Some financial queries patterns were used in the prototype grammar to increase the financial knowledge, for example, “What is the leverage ratio of the company X in 2013”, or “What is the net income of Z company”. These patterns, depending on the NLP, can be configured in various manners, for instance, utilizing regular expressions, specific languages or rules defined by NLP itself for build the syntax.

Some patterns that use the English language structure were registered on the grammar of American version and several patterns that use Portuguese language structure were registered on the grammar of Brazilian version. The construction of the Portuguese syntax patterns requires more effort than the English patterns because of the greater amount of syntax rules of the Portuguese language in relation to the English language.

TABLE I. SELECTION CRITERIAS OF NLP

NLP platform name	Provider	Criterias						
		1	*2*	*3*	*4*	*5*	*6*	*7*
API.AI	Speaktoit	x	x	x	x	x	x	x
Cortana Plataform	Microsoft	o	o	x	x	x	x	o
Google Now Plataform	Google Inc	o	o	x	x	x	x	o
Syn Engine	Synthetic Intelligence Network	x	x	o	x	x	x	o
Siri Plataform	Apple	o	o	x	x	x	x	o

A. Dictionary of terms

Concepts and entities of the Financial Domain have to be in the dictionary of terms. We grouped these terms in a feature of the selected NLP named "Entity". Each entity has a list of terms and each terms has its synonymous. For the prototype, we created following four entities: BalanceSheetsFinancialConcept Company, YearPeriod, CommandExpressions. Objectives and descriptions of each one are described as follows.

1. BalanceSheetsFinancialConcept: Contains the main financial concepts of Company Balance Sheets. We created two dictionaries of financial terms, and the majority of registered terms there were derived from XBRL taxonomies. US-GAAP 2013 [16] for questions in English language and BR-GAAP for questions in Portuguese language. The fragment of the BalanceSheetsFinancialConcept entity is shown in Table II.
2. Company: Contains target company names. Table III shows a fragment of the Company entity.
3. YearPeriod: List of the financial quarter symbols. Example: The first quarter is Q1;
4. CommandExpressions: Entity created for grouping command expressions that can be said by the user, as “What is”, “Show me”, “Inform”. The grammar rules becomes smaller, by the use of it. Table IV lists this command expressions.

When an NLP recognizes a given set of words, it substitutes them for the pattern word defined in the dictionary of terms. For example, in similar fashion, a supposed user expression “What is the debt-to-equity ratio of Petrobras in 2013” after going through the process of substitution, it will be transformed into: “Show me the leverageRatio of Petroleo Brasileiro in 2013".

TABLE II. BALANCESHEETSFINANCIALCONCEPT ENTITY

Financial Concept Name	Synonymous
Assets	*Assets,Total assets*
Cash and cash equivalents	*Cash,Cash and cash equivalents*
EBITDA	*EBITDA,ebitda,“Earning Before Interests, Taxes, Depreciation and Amortization”*
leverageRatio	*leverageRatio,leverage ratio,debt-to-equity, debt to equity, leverage*
Net Income	*netIncome,Net Income,NI*
Short-term investments	*Short-term investments*
Total liabilities	*Total liabilities,Current liabilities*
Total liabilities and stockholders’ equity	*Total liabilities and stockholders’ equity, liabilities and stockholders’ equity*
Total current assets	*Total current assets,Current assets*

TABLE III. COMPANY ENTITY

Company name	Synonymous
Apple Inc	*Apple Inc,Apple*
IBM	*IBM*
Google Inc	*Google Inc,Google*
Microsoft Inc	*Microsoft Inc,Microsoft*
Petroleo Brasileiro	*Petroleo Brasileiro,Petrobras*

B. The rules of grammar

After the process of substitution, the NLP confronts the query with the configured patterns in grammar. For example, in order to make the NLP, used in this proposition, understand the questions cited previously, the following rules of syntax were established:

"@CommandExpressions @BalanceSheetsFinancialConcept:financialConcept of @Company:companyName in @sys.number:year"

or

" @BalanceSheetsFinancialConcept:financialConcept of @Company:companyName [at,in,of,on fiscal year, in fiscal year,of fiscal year,of year, year] @sys.number:year".

The positioning of words, entity names and parameter names defines a pattern of questions expected. The symbol at sign identifies the name of the entity and the name after the colon identifies the parameter expected at that position.. Words or Entity names between square bracket symbols are considered optional.

From that example, supposing the question "What is the leverage ratio of Petrobras in 2013", the result of the NLP, after dealing with the question, will be a set of parameters and values extracted from the sentence.

@action = CompanyRatioService
$financialConcept = leverageRatio
$companyName = Petroleo Brasileiro
$year=2013

The selected NLP offers a resource to organize groups of pattern questions that is called "Intent". In each Intent, the programmer defines the rules and specifies the parameters that must be filled by NLP. In Table V, "What is The Financial Concept of Balance Sheets at Period" intent is presented.

There are also some parameters for each intent that are only defined by the developer, one of them is @Action. In the prototype the parameter @Action was used to identify the name of the class that will handle the parameters sent by each intent. NLP encapsulates all the parameters in a response object in the data format JavaScript Object Notation (JSON).

TABLE IV. COMMAND EXPRESSION ENTITY

Command	Expression
Whatis	*What is,Give me,Give,Would you tell me,Show me,Show*

TABLE V. WHAT IS THE FINANCIAL CONCEPT OF BALANCE SHEETS AT PERIOD INTENT

Id	Rule
1	*@CommandExpressions @BalanceSheetsFinancialConcept:financialConcept of @Company:companyName [in,on] @YearPeriod:yearPeriod [at,in,on fiscal year, in fiscal year,of fiscal year,of year, year] @sys.number:year*
2	*@CommandExpressions @BalanceSheetsFinancialConcept:financialConcept of @Company:companyName [at,in,of,on fiscal year, in fiscal year,of fiscal year,of year, year] @sys.number:year*
3	*@CommandExpressions @Company:companyName @BalanceSheetsFinancialConcept:financialConcept [at,in,of,on fiscal year, in fiscal year,of fiscal year,of year, year] @sys.number:year*
4	*@CommandExpressions @Company:companyName @BalanceSheetsFinancialConcept:financialConcept [in,on] @YearPeriod:yearPeriod [at,in,of,on fiscal year, in fiscal year,of fiscal year,of year, year] @sys.number:year*
5	*[at,in,of,on fiscal year, in fiscal year,of fiscal year,of year, year] @sys.number:year @CommandExpressions @BalanceSheetsFinancialConcept:financialConcept of @Company:companyName*
6	*[in,on] @YearPeriod:yearPeriod [at,in,of,on fiscal year, in fiscal year,of fiscal year,of year, year] @sys.number:year @CommandExpressions @BalanceSheetsFinancialConcept:financialConcept of @Company:companyName*

We created the following four intents for help to keep the state of the dialog with user: "Change Company", "Change Financial Concept", "Change Period", "Change Financial Concept And Period". This feature avoids that users have to make a complete question for each interaction.

C. The services mapping

After the DKM receives the response object of NLP it extracts the @Action parameter and through its built-in mapping forwards to the corresponding class and waits for the response. This mapping is done by a factory pattern class type allowing addition of new classes to handle new services and parameters without quite code. The classes added to the inventory of this factory should extend the standard abstract class called FinancialService. In addition, the code required to access the financial services must include a method for treating the parameters encapsulated in objects of a class named FinancialParametersRequest and other method named getResponse that aim to return the result of the service.

In the cited example the @Action informs that the CompanyRatioService Class must be invoked to treat the rescued parameters.

If it does not find a coherent pattern with the asked question, the NLP will send the empty @action parameter to DKM. When this occurs, a created exception UnknownFinancialServiceException is launched and treated by the DKM which, in turn, sends to the user the following message:

"I'm Sorry! I didn't understand your question!"

For the prototype proposed, a web service of FIS was designed to analyze the financial situation of enterprises. This analysis involves the evaluation of several values and/or variations of the economic indexes of the enterprise within a time period. This web service of FIS is called by the CompanyRatioService Class.

The OLAP queries are the most adequate so that the most complex analyses of risks, performed under the perspective of many dimensions, or business views, typical of economic evaluations may be performed with acceptable flexibility and performance. However, it is complex to perform OLAP queries in XBRL artifacts or database or documents based in XML. In this case, Link Based Multidimensional Query Language (LMDQL) [8][7], can be use to decrease this complexity. It is a specified language for performance of OLAP queries in XBRL documents. Furthermore, there is the definition of specific operators in LMDQL for the financial analysis, considerably decreasing the complexity of the queries and of the orchestration work.

The assistant can answer to typical questions with these operators, such as those ones related to the level of the enterprise debt's using the operator *GrauAlavancagemFinanceira (Financial Leverage Degree)*[7], (available in LMDQL). There are also more complex operators such as those of enterprises evaluation statistical models, which allow the assistant the "capacity" of indicating enterprises with good economic strength, widening the sophistication of financial queries. There is an example of a LMDQL, query (1), to obtain the index of risk exposition.

```
SELECT {#GrauAlavancagemFinanceira#} ON COLUMNS,
{([Exposure Class])} ON ROWS from [Capital Requirements SA]
where [Time].[2013]                                        (1)
```

For this prototype, the *getMeasurementTime* was one of the implemented web services. It has as entry parameters: the enterprise name registered on SEC, the financial measurement name and the period desired. The financial measurement, "leverage ratio" was mapped in order to perform the OLAP query utilizing the operator *GrauAlavancagemFinanceira* in the XBRL repository.

Example (2) is part of the *getMeasurementTime* web service specification:

```
<message name="getMeasurementTimeRequest">
  <part name="measurement" type="xs:string"/>
  <part name="companyName" type="xs:string"/>
  <part name="time" type="xs:string"/>
</message>
<message name="getMeasurementTimeResponse">
  <part name="value" type="xs:integer"/>
</message>
<portType name="financialMeasurement">
  <operation name="getMeasurementTime">
    <input message="getMeasurementTimeRequest"/>
    <output message="getMeasurementTimeResponse"/>
  </operation>
</portType>                                          (2)
```

So, the versatility and simplicity provided by LMDQL queries, encapsulate in the services, allow a high level of sophistication to the financial answers supplied by the assistant.

The Securities and Exchange Commission, from the USA is an institution which makes available in one of its sites several XBRL database containing information from many enterprises registered in the agency. The quantity of available data of these repositories is a limitation of the range of the answers of the assistant. Furthermore, the correctness of the data implicates directly in the correction of the assistant answers.

For a low coupling configuration, the "DKM", that is part of Orchestration layer, must be implemented in a server environment and the Presentation layer in a Client environment. The possibility of modifying the assistant behavior in a single location for all clients is the main reason, i.e., it is easier to perform the maintenance of the mapping in a single code, with the advantages such as intervention speed, lower probability of code update failures, smaller tests quantity, lower alteration costs.

D. *The Evaluation of the Prototype*

In order to validate the prototype, some components of architecture were implemented as described as follows:

- A mobile Android application and a Javascript application to represent UIM.
- Two versions of dictionaries, the American version and the Brazilian version as part of UK.
- A Python Web Service as proxy of NLP.
- Several JAVA Classes as part of Orchestration
- A Java Webservice to access XBRL data.

We loaded 542 not duplicated terms extracted from US-GAAP taxonomy on the American financial dictionary. The mapping of two type of user questions to the corresponding services activation was done. Questions related to the available information about items in the financial statements, reports which display the situation of the organization in a given moment, were mapped to activate a web service called *getMeasurementTime*.

In our experiments we used the following eleven questions (3) for question answering over tree assistant: Our prototype, SIRI [11] and Google Now [4]. The last two are of the most used commercial assistants on the market. We did not compare with other financial virtual assistant because did not found anything similar.

1. *"What is the net income of Petrobras in 2013?"*
2. *"Give me the leverage of Microsoft of fiscal year 2014?"*
3. *"Show me the leverage of Microsoft in the fourth quarter of fiscal year 2014?"*
4. *"What is the leverage of Apple Q4 2014?"*
5. *"What is the Total assets of Petrobras in the second quarter of fiscal year 2014?"*
6. *"Give me the Total assets of Microsoft of fiscal year 2014?"*
7. *"What is the Petrobras Total assets of fiscal year 2014?"*

8. *“Show me the Microsoft' Total assets in the second quarter of fiscal year 2013?”*
9. *“In fiscal year 2013, what is the leverage of Petrobras?”*
10. *“In the second quarter of year 2013, what is the leverage of Apple?”* (3)

All questions were understood and correctly answered. by our prototype. For the first question, its response was presented as follow:

“The net income of Petroleo Brasileiro was U$10,832 million in 2013.”

The responses of Google Now for all question were a list of web links about each query supported by Google Web Search engine [5]. The tree first presented summaries of these lists contained the solicited data in the first lines.

The SIRI answered the questions showing and speaking a following text before present a list of links about the question. e.g., :

“Here’s what I found on the web for ‘What is the net income of Petroleo Brasileiro in 2013’.

The SIRI uses Bing Web Search [1] to supports the answers. The SIRI answers about the 7th, 8th and 10th questions were different than others. In this case, SIRI identified every parameters and showed an answer expression on the screen, similar to our prototype. SIRI acted like an assistant at 30% of the experiment questions. However, the responses about these question were wrong.

Google Now only acted like a web search assistant for 100% of questions.

Also, we evaluated if these assistant keeps the history and the state of the user dialog. After a complete question, e.g., “*Give me the leverage of Microsoft of fiscal year 2014?”,* we used the short questions over the assistants as a sequence. e.g., On the first, "Now show me at 2014". On the second, "Give me the EBITDA". Then, the last short question of the sequence was "And about Apple". These queries have the objective to change the year, the financial concept and the company of the initial complete query, respectively. This feature allows the user ask short questions based on the first complete question. i.e., similar to human conversation, if the user wants to change some attributes of the original question, it is not necessary to ask a new complete question for each interaction.

In our experiment the prototype changed the corresponding parameter of the initial query and answered to the user for each interaction. All responses provided by our prototype were correct. However, Google Now and SIRI understood each interaction as a new query, so, all questions were submitted to their search engines, respectively. All answers, provided by them, had no relationship with the first complete question. Also, the first answers and the third answers had no relationship with the financial theme, respectively. As result of this experiment, we certified that the minimal tracking of state of the dialogue, between a user and an assistant, was only done by our prototype. In this case, it happened because only our prototype answers to the user after it analyzes both the new question and the previous question. This behavior is different from the SIRI and the Google Now, which always process each new question as if it was a new complete question, without any relation with previous question.

Our approach is a positive contribution in this area of research, because it presents an architecture for virtual closed domain assistants based on services, especially the NLP service.

We consider that the prototype of the financial virtual assistant produced based on the proposed architecture is a viable alternative and it proves the effectiveness of architecture. However, there are still some necessary evaluations to do like performance and accuracy test on manipulating a big list of services, datasources, items of dictionaries and items of grammar.

Although this work is a positive contribution, there are some limitations. Firstly, the quantity of utilized companies was little, about ten companies. Secondly, the prototype was not evaluated by investors about effectiveness. Thirdly, the comparison with others similar financial assistants was not done because they do not exist or are unknown for us.

V. CONCLUSION

When an investor gets the right information about the companies in a risky market, e.g., the stock market, quickly and easily, he has a great competitive advantage. For this context, this work presented the details of a virtual financial assistant prototype as an alternative solution to support the small investor.

This virtual financial assistant is a service-oriented computer system whose answers are based on the available financial information of companies in XBRL. The user can interact with it through natural language, supported by a service NLP and a dialog management unit. Its understanding degree of the financial questions and the sophistication level of its answers can be expanded, especially with the use of financial services that encapsulates OLAP analytic queries for XBRL database, as exemplified in this work.

An architecture for implementation of the closed domain virtual assistants, based on domain information services, in which the assistant was based, was also presented.

As evaluation of the prototype, financial questions in natural language were submitted to the prototype, which understood and answered all the questions correctly. The evaluation of the dialog manager feature of the prototype was also done and returned positive results.

We concluded that the architecture has been validated and we recognized enough potential of the prototype to be a viable alternative to traditional existing financial assistance.

In future work, the financial virtual assistant could compare the financial health of companies and, based on the share value of the selected companies, the assistant could notify the user when is a good moment to buy or to sell the shares in a company.

REFERENCES

[1] Bing, Microsoft Corporation. [Online]. Available from: http://www.bing.com. [retrieved: 05, 2016]

[2] E. C. Paraiso, and J-P A. Barthès, "A Voice-enabled Assistant in a Multi-agent System for e-Government Services," Conference: Advanced Distributed Systems: 5th International School and Symposium. Jan. 2005. pp. 495–503. ISSN: 0302-9743, ISBN: 978-3-540-31674-9.

[3] E. M. Eisman, and V. López, "A framework for designing closed domain virtual assistants," Expert Systems with Applications, vol. 39, Issue 3, pp. 3135–3144, Feb. 2012, doi:10.1016/j.eswa.2011.08.177.

[4] Google Now. Google Inc. [Online]. Available from: https://www.google.com/landing/now/. [retrieved: 04, 2016]

[5] Google Web Search. Google Inc. [Online]. Available from: https://www.google.com. [retrieved: 05, 2016]

[6] J. Medina, E. M. Eisman and J. L. Castro, "Virtual Assistants platforms 3.0," Transl. IE Comunicaciones: Revista Iberoamericana de Informática Educativa, n. 18, pp. 41–49, Dez. 2013, [Online] Available from: http://dialnet.unirioja.es/descarga/articulo/4468692.pdf. [retrieved: 04, 2016]

[7] M. A. P. da Silva, and P. C. da Silva, "Analytical Processing for Forensic Analysis," IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), 2014 pp. 364-371, http://dx.doi.org/10.1109/EDOCW.2014.60.

[8] P. C. da Silva, and V. C. Times, "Analytical Processing over XML and XLink. International," Journal of Data Warehousing and Mining, vol. 8, pp. 52, 2012. ISSN: 1548-3924, EISSN: 1548-3932, doi: 10.4018/IJDWM.

[9] S. P. Zambiasi, and R. J. Rabelo, "A Proposal for Reference Architecture for Personal Assistant Software Based on SOA," IEEE Latin America Transactions, vol. 10, pp. 1227–1234, Jan 2012. ISSN 1548-0992, doi: 10.1109/TLA.2012.6142466.

[10] Securities and Exchange Commission U.S. [Online] Available from: http://xbrl.sec.gov. [retrieved: 05, 2016]

[11] SIRI. Apple Inc. [Online]. Available from: http://www.apple.com/ios/siri/. [retrieved: 05, 2016]

[12] Speaktoit. API.AI. [Online]. Available from: https://api.ai. [retrieved: 04, 2016]

[13] Speaktoit. Assistant.ai. [Online]. Available from: https://assistant.ai. [retrieved: 05, 2016]

[14] Synthetic Intelligence Network, SYN: Syn Virtual Assistant, 2015. [Online]. Available from: http://syn.co.in/Syn-Virtual-Assistant.aspx. [retrieved: 05, 2016]

[15] XBRL Consortium. XBRL. [Online]. Available from: https://www.xbrl.org. [retrieved: 05, 2016]

[16] XBRL US. 2013 US-GAAP taxonomy. [Online]. Available from: https://xbrl.us/xbrl-taxonomy/2013-us-gaap/. [retrieved: 05, 2016]