Ontology-driven Natural Language access to Legacy and Web services in the Insurance Domain

Mikhail Simonov
Nomos Sistema S.p.A.
Simonov@nomos.it

Aldo Gangemi
LOA, ISTC-CNR
A.Gangemi@istc.cnr.it

Massimo Soroldoni
Nomos Sistema SpA
Soroldoni@nomos.it

Abstract

Ontologies can play an important role in industrial systems by enabling access to Legacy resources in a transparent and distributed way. Ontologies are developed in order to provide machine-processable semantics of information that is exchanged between different agents, either humans or software. The same mechanism can be used for Legacy service discovery, as well as for automated reasoning about the content of answers obtained from that service. We have created an information system offering a natural language, web-based access to distributed Legacy systems in the insurance domain. The application offers a communication channel that actualizes a new model of business interaction between an end user and a service provider, an insurance company in our case. Our system leverages on the knowledge management framework developed within the Eureka funded research project IKF, which has been adapted to the Insurance Domain by building a dedicated Domain Ontology named IES. The core component, named MetaDiscoverer, implements an ontology-based filtering of user queries in order to discover the intended Legacy service. Our application is able to perform information retrieval, knowledge extraction, automated reasoning, service location and dynamic invocation. The solution is RDF and XML-compliant, according to the recommendations given by the W3C Consortium.

1. Introduction

Current Insurance business still heavily relies on Legacy applications. Back office management activities in the insurance domain are performed by huge and well-developed Legacy systems. Through an intranet, an insurance agent usually deals with a Legacy system in order to obtain needed information, or to answer typical queries asked by insured people that visit the agency or call her by phone. The agent typically deals with the back office when more complex questions are raised or real-time calculations are required. This process is time consuming and automation through semantic capabilities would be helpful. Unfortunately, software with such automated reasoning capabilities is not available in commercial systems.

On the other hand, users typically encounter a web-based application during an insurance product purchasing, or when an "automated" CRM is contacted for FAQs. Managing such relations is not straightforward, because it requires access to Legacy systems, understanding of user queries, composition of insurance regulations and guidelines with user data or even profiles, etc.

An industrial CRM application can provide natural language access if it is able to perform semantic parsing of the user query, and to answer that query by means of automated ontology-based reasoning [e.g. Jarrar et al., 2003 for a review and a proposal concerning Customer Complaint Management].

Conforming to that, IKF-IES is an enterprise Business Information System, including a knowledge management system, and a domain ontology for insurance (IES-Core) that is focused on life insurance and Unit-linked policies.

Semantic parsing techniques are used in order to obtain basic lexical disambiguation capabilities for Italian and English, although other languages can be handled, using appropriate tools e.g. available from the Natural Language Understanding Consortium, known as NLUC[1].

IKF-IES is mainly used for for web-based, self-help purpose. It also supplements human-centered communication, since Call Centers can be coupled with a semantic support coming from the system.

The application offers (Fig. 1):

1 NLUC members are listed on http://www.nluc.com

Witold Abramowicz (ed.), Business Information Systems, Proceedings of BIS 2004, Pozna , Poland

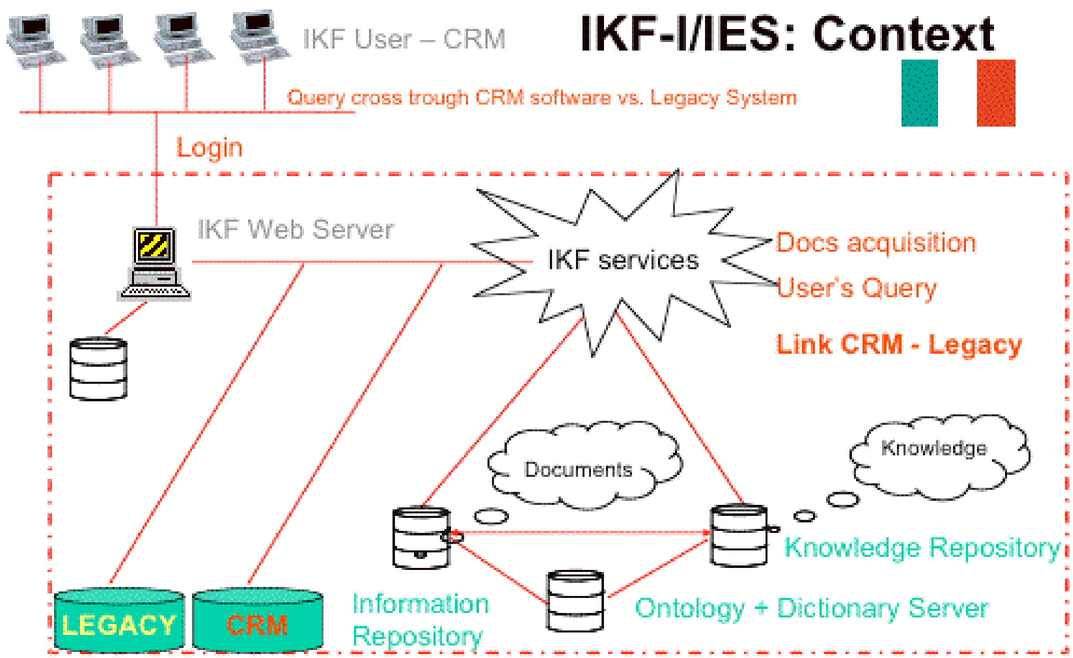


Figure 1. The IKF-IES architecture

- A document management and information fusion system (the IKF framework[2] services)
- A natural language-based access to Legacy insurance systems
- An ontology driven query answering
- An ontology-driven Legacy service discovery
- An ontology-supported Legacy service invocation that employs an intelligent parameter filtering.

In the following, we will briefly describe some of the functionalities and components of IKF-IES.

2. From an insurance business model to the domain ontology

Setting experts' intuition to the minimum, an insurance business could be modeled according to the following simplified schema:

Customer –has– Policy (1..*)
Policy –expects– Receipt (1..*)
Policy –has– Claim (0..*)
Claim –has– Payment (1..*)
Policy –envisages– Payment (1..1)

The schema means that a customer has at least one policy, for which at least one receipt is expected, and at least zero claims can be expected. A claim expects at least one payment. Life insurances also involve a final reimbursement, or the possibility to reclaim the surrender value, as well as many other details, but here we limit the example to the above schema.

Following such schema, user's data are represented in a physical Legacy database, e.g. DB2 is widely adopted in the use case.

Our application relies on the Legacy database, but it also provides other ways to access insurance-related information, and combines Legacy data with that information by integrating IES domain ontology with the DB.

In order to develop the domain ontology, a task force of domain experts, software engineers, and ontology specialists has reached a design agreement. The domain ontology has been built from domain documents and experts' advice according to the ONIONS methodology [Gangemi et al. 1999] provided by one of IKF partners, and the DOLCE-Lite+ foundational ontology [Masolo et al., Gangemi&Mika], developed in the context of the WonderWeb[3] project, and maintained at the Laboratory for Applied Ontology[4] of the Italian National Research

2 IKF is an international EUREKA co-funded project that develops frameworks to allow knowledge management in various business settings; http://www.ikfproject.com

3 WonderWeb is an European research project co-funded by 5th Framework Program; http://wonderweb.semanticweb.org

4 http://www.loa-cnr.it

Council. The resulting design requires the specification of IES ontologies into a highly expressive description logic.[5]

An example of the analyses performed by experts in order to produce IES Core is shown in Table 1.

5 An alternative and possibly complementary approach in building an insurance ontology can be found in [Kietz et al.].

IES Object	DOLCE-Lite+ Class	Informal description	Informal constraints
Policy	Non-physical object (Description context)	A policy is issued by an insurer for an insured and covers certain losses given certain risks	Held_by Insured Issued_by Insurer Covers Insured
Insurer	Non-physical object (Functional role)	An insurer issues a policy to an insured which covers losses given certain risks	Issues policy Pays or disclaims claims
Insured	Non-physical object (Functional role)	An insured is covered by a policy against certain losses given certain risks	Covered by a policy
Claim	Activity	An insured makes a claim against a policy for a loss	Made by insured Paid or disclaimed by Insurer
Risk	Non-physical object (Description context)	Risk is the likelihood of a certain loss for a certain insured	Contemplated by a policy
Loss	Non-physical object (Description context)	A loss exists when an insured suffers some injury to person or property. The loss is said to be covered if it was part of the risk contemplated by the policy	Suffered by insured Covered (or not) by a policy
Disclaimer	Activity	A disclaimer is made by an insurer when a loss is not covered by a policy, usually when not a contemplated risk	Made by insurer

Table 1. A sample table used for experts' drafting of the core ontology

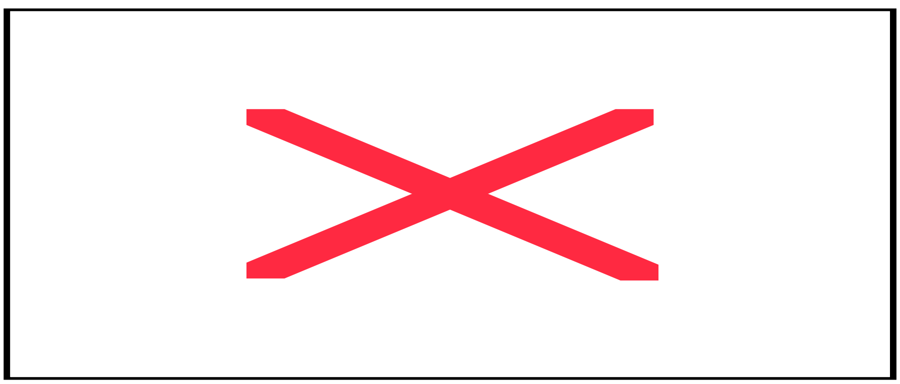


Figure 2. A sketch of the kernel derived from the draft tables

From experts' view, it results that traditional insurance ontology is focused on the notions of *Insurer*, *Policy*, *Insured*, and *Claim*. A further step is to make a sketch of the tabular drafting, distinguishing concepts and relations, and then drafting a preliminary kernel for IES Core (Fig.2).

Next step in IES Core building has been the logical formalization of the preliminary kernel and of the tabular drafts, and their integration within the foundational ontology. DOLCE-Lite+ contains hundreds of concepts and relations organized around typical *ontology design patterns*. The most general ones concern the topmost partition of categories (e.g. *object, event, region*), and their basic relations (e.g. *part, participation, localization*). A more specific design pattern has been developed in order to represent procedural and regulatory knowledge. It is called Descriptions and Situations (D&S [Gangemi&Mika], Fig. 3) because it is defined around the basic relation of *satisfaction* between a description context and its viable realizations (situations). D&S has been extensively used for representing IES Core because of its flexibility. Based on D&S, IES-Core reflects the procedural approach of business modeling widely adopted by Insurers.

Examples of description *contexts* in the insurance domain are policies (as content) and risks (as analytical specifications). Description contexts have a structure:

for example a policy includes roles such as *Insurer* or parameters such as *Minimum age*. Situations are settings in the real world that comply with ("satisfy") a context if the constituents (e.g. a company or an age value) of the setting comply with the components of the context.

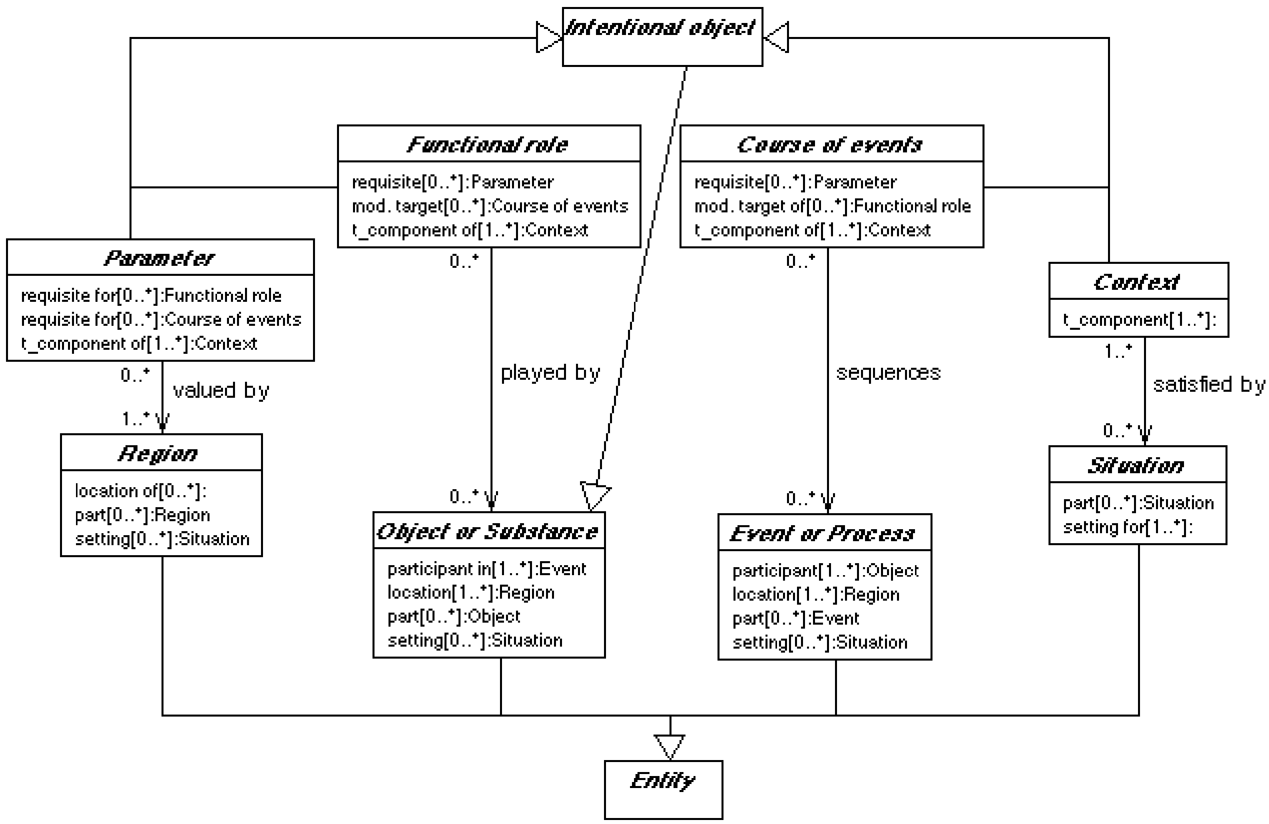


Figure 3. A UML class diagram depicting the *Descriptions and Situations* ontology design pattern within the DOLCE-Lite+ foundational ontology.

3. Tuning the domain ontology with task and application ontologies

Further development of IES Core has been based on ontological requirement analysis, which uses *competency questions* [Grüninger and Fox] to be asked to experts about the use case for the application making use of the ontology. In other words, a domain ontology is considered adequate to the application if it supports the tasks in the use case without overspecifying entity types. Moreover, the existing or expected application components must be taken into account as well. The final result is an application ontology that explicits both domain and tasks according to available software components.

Since the application is a CRM one, customer-oriented query types constitute a basic set of requirements to be fulfilled. For example, in order to obtain answers to FAQs, the following query classification is adopted:

1. Queries that do not require the access to the Legacy system, i.e. that can be answered by the information retrieval component alone; for example "which documents are required to start a new life insurance?"
2. Queries that require access to the Legacy system, when no calculation is needed; for example "who is the insured?"
3. Queries that require an access to the Legacy system associated with a specific calculation; for example "when can I ask the surrendering of my life insurance?" or "I wish to know the surrender value of my life insurance".

The second type of query can be answered easily if the mapping between the query and the business modeling of Legacy database is available (see below).

The last type of query is the most difficult to manage. There are two possible approaches: to store the pre-calculated values in the database, or to invoke dynamically, "on the fly", the needed Legacy service. If the last approach is adopted, the enterprise will be invited to solve problems like: how to discover the correct Legacy service, how to prepare the IN area, or which data from the OUT area must be taken and shown.

Whatever query type we consider, the IR component or the wrapper to the Legacy system must use explicit knowledge about the domain represented in the CRM database, in the query, or in the Legacy schema.

Our business case involves two distinct applications: the Legacy application of the Insurance Company and the CRM application used for support to the call center.

A first (and typical) scenario includes a customer calling an insurance agent, who calls back the customer after examining all needed Legacy documents, and after performing due calculations.

A second scenario includes the background (or batch) pre-calculation of all needed values, for instance once a week or every month, and the immediate communication of results to the customer.

A third scenario enables the new communication channel of Self Web Help, in order to allow the customer to formulate a natural language query.

The main challenge consists in the semantic processing of the query, including its disambiguation. In this procedure, the exact semantic meaning is filtered, and the available Legacy service is located and invoked. Next, the obtained result is semantically enriched through the ontology, and then delivered to the end user.

A stringent temporal constraint must be respected, since the web session has a time out and the remote user is waiting for an answer only for a limited time.

IKF-IES uses a semantic parser in order to process the natural language query, and to extract noun- and verb-phrases.

IES-Core and the insurance Domain Ontology are currently expressed in the Loom-Ontosaurus knowledge representation system [Swartout et al.] that implements a very expressive description logic, but they will be ported to both KIF and OWL (whose versions already exist for DOLCE-Lite+), in order to benefit from both logical programming engines, and Semantic Web inference engines. The assertional part of the knowledge containing textual occurrences has been encoded in a specific knowledge repository specified in a dedicated frame-based knowledge representation system. For details cf. the web portal of the IKF project (http://www.ikfproject.com).

IKF-IES ontologies currently include 506 concepts and 79 relations.

4. The framework

4.1 The Knowledge Management Framework

The Knowledge Management Framework supports the following functionalities:

- the semantic parsing of the query
- the invocation of the inference engine (performing automated reasoning)
- the discovery of the Legacy (or Web) service
- the interaction with the ontology server
- the query tunneling towards the knowledge repository, the information retrieval component, and the Legacy system.

The last process is an asynchronous one, since, as generally recognized, query elaboration times can have different magnitudes: the Legacy call is synchronous, the IR answering is a fast process, and the KR access is a time consuming process.

4.2 NLP tools (Parser and Link Elicitor)

IKF-IES can elicit the knowledge from free text, semi-structured, and structured documents. NLP is performed by means of a commercial parser developed by Expert System, Italy. The commercial solution adopted in the Italian version of IKF-IES uses an embedded lexicon called the Sensigrafo™[6], an enhanced dictionary originally based on WordNet [Miller et al.]. The lexicon includes a taxonomy of *synsets* ("synonym sets", equivalence classes of word senses), and the following supported relationships among synsets:

- noun synset - hypernym (is a)
- noun synset - meronym (part-of)
- verb synset - subject
- verb synset - object
- verb synset entailment
- adjective synset - class
- adjective synset - quality
- synset - antonym
- synset - troponym
- synset disjointness
- synset covering

Sensigrafo™ currently has one of the best coverings of the Italian language.

[6] Sensigrafo™ is a trademark of Expert System S.p.A., Italy: http://www.expertsystem.it/. Expert System is a member of NLUC.

Since the semantic parser is lexicon-driven, but not ontology-driven[7], IKF uses an additional module named Link Elicitor that performs knowledge elicitation driven by the IES Core ontology. The Link Elicitor component also manages uncertain knowledge (by means of fuzzy rules), and produces RDF(S) triples (see next section).

The second processing stage (actually a post-processing) stores the extracted triples in a dedicated Knowledge Repository, represented in an XML/RDF compliant frame-based language.

An analysis of the syntactical structures is adopted in order to establish the degree of plausibility of triples retrieved by means of the ontology. A plausibility=1 is equivalent to the presence of a qualified semantic relation.

4.3 Metadiscovery and Legacy calls

IES Core and the domain ontology of insurance attached to it are not only used for display purposes, but also for automation, integration and reuse of data across applications.

For example, the IES domain ontology is used in order to perform link elicitation, i.e. to discover relations between concepts or individuals underlying terms within documents or records in databases. Link elicitation is performed by deriving a set of relevant RDF(S) triples out of the ontology. Triples constitute a "lightweight" semantic network that is then translated into a suitable Legacy service invocation of the Metadiscovery module. The Metadiscovery module is then ontology-driven and performs a brokering service between the conceptual model of the ontology and the business model adopted in the Legacy database.

The Metadiscovery module performs reasoning and unification between the part of the semantic network coming from the user's query and the object model of the Legacy part of the enterprise application.

After Metadiscovery has resolved the Legacy service, this module physically performs the call and handles marshalling data vs. Legacy and OUT area elaboration. Some special heuristics are used in this process which are protected industrial know how.

Legacy calls are performed after the correct Legacy service has been discovered, and the COMAREA[8] system component has been properly initialized with correct values. This process is also ontology-driven, since the correspondence between slots and values comes from the knowledge frames represented in RDF(S).

4.4 Processes and workflows

Initially, the Information Repository and the Knowledge Repository are empty, but the system needs an insurance-related corpus to be loaded, and some knowledge elicitation has to be performed for each introduced document. Since the relationship between **information objects** and **documents** is maintained during query answering, the knowledge elicitation process creates the appropriate links, and populates the Knowledge Repository with instances of classes describing domain relevant objects.

The query answering adopted by IKF-IES involves three separate answers: the first one comes from a Knowledge Retrieval component, the second one from an Information Retrieval one, and the last, most interesting one is the answer coming from the Legacy system.

Since the enterprise system is constrained by some legal issues, the corpus introduced into the Information Repository must be controlled by a human. To this purpose, a *Document Advisor* role has been introduced.

The IKF-IES corpus consists of:

- *Codice Civile* (Italian Civil Code), *Circolari ISVAP* (specific laws and extensions issued by Italian supervising authority), *Condizioni Generali di polizza* (terms of contract),
- *Condizioni Particolari di polizza* (specific restrictions),*Nota Informativa* (product description), *Fondi e Gestione Separata* (Unit Linked-related), *Performance monitoring* (periodic).

Once the Information Repository is populated by documents, and the extracted knowledge is loaded into the Knowledge Repository, the query can be answered. The process can be summarized as follows:

1. Input query
2. Query expansion and disambiguation (automatic)
3. Meta discovery (Legacy service or web service)
4. Search: information retrieval, knowledge retrieval (unification) and Legacy call
5. Report generation

4.5 User interface and NLP

The system is web-based and the user interface consists of a set of jsp and servlets.

A registered user receives the combination login/password, and this mechanism allows us to build a correct Legacy call in order to retrieve only the **instances of managed objects** that can be retrieved

[7] Although Wordnet are sometimes considered ontologies, we distinguish here on the basis that current wordnets are not formal, axiomatic theories [cf. Gangemi et al. 2002.].

[8] Cobol-written Legacy applications uses the raw data exchange through the shared area known as "common area" or simply "COMAREA" for which the data marshalling and unmarshalling are managed by middleware.

according to the privacy law: only a Customer (*Insured*) of an insurance company registered for online access having an ID assigned can deal with the web based system.

Hence, User = Jack Martin and Pass = *** means UniqueID = 123 and the List of managed contracts = Policy N. = 987. Furthemore, we can retrieve general information about the above mentioned customer in both ways: by Legacy Call or by unification between the the instance of a human having a slot UniqueID = 123, Name = Jack Martin.

Let us assume that the above mentioned customer is 37 years old, the policy N. 987 in the only managed contract (otherwise, we will be invited to disambiguate), and the current Tariff is RXII. We also know that "Today" is 10th December 2003 and the next renewal or surrendering will be at the end of the year, i.e. on 31st December 2003. The remaining duration of the policy can be calculated and will give us the remaining duration = 6 years.

The abovementioned customer can now post a natural language question through the web interface, e.g.: "Vorrei conoscere il valore di riscatto della mia polizza vita", i.e. "I would like to calculate the surrender value of my life insurance policy". We handle noun-, verb-phrases, and the links between them according to the ontology.

Next, a query processing component performs the stages of (1) Semantic parsing, (2) Disambiguation, (3) Query expansion. The next step is service discovery: (1) Semantic Discovery, (2) accessing the service descriptor, (3) Translation vs. Legacy or Web service call, (4) Service Invocation at a correct location.

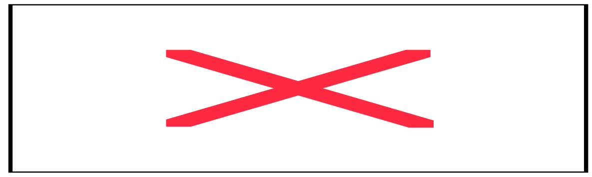
The above mentioned query could look like the following one:

CALC. VAL. RISCATTO Is-a <LEGACY SERVICE>, <Invoked by INV07> <from CICS03> <on server \\urano>. It has 6 parameters:

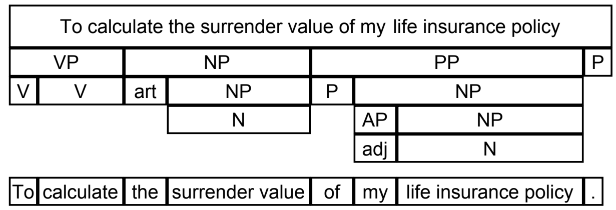
1st Cust.Id, 2nd Age, 3rd PolicyNumber, 4th TariffId, 5th Date, 6th Duration

The binding between actual and formal parameters is performed in order to ensure the correspondence between the service required and the available Legacy service, and so that the correct code base is called by the Invoker. Furthermore, the COMAREA managing component for the Legacy call invocation is run.

Hence the following natural language query:



or the same phrase in English:



is disambiguated and translated into a parse tree in order to be processed by our Link Elicitor.

NLP is assisted by algorithms that establish the fuzzy degree of plausibility of the triples according to the semantic structure of the text fragment.

For example, IES Core allows to infer the following triples including concepts:

- Policy – Surrendering – Surrender_Value
- Insurer – Calculate – Surrender_Value
- Insured – Own – Policy

At this point, we know that someone (actor) wants to know (action) the surrender value (patient), but the above mentioned value will be known only after the calculation (process) to be performed by an Insurer (actor).

Now we have to discover the Legacy service that calculates the surrender value for an instance (my life insurance policy).

Let us assume that the procedure named "LF_GetValRisc()" allowing to calculate the surrender value is available. Once located, the only issue is to replace parameters with effective values. In order to perform this task, we use the information carried in context, i.e. we perform the mapping between concepts and managed objects (a problem similar to the mapping of the OO model to ER model or vice-versa).

Finally, the dynamically generated call has the following syntax:

CALL LF_GetValRisc(Id = 123, Eta = 37, Pol = 987, Tar = RXII, Dal = 31-12-2003, Anni = 6)

Another added value offered by this system is a semantically enriched answer containing the explanation of the procedure, the exact meaning of the concept "surrendering" and the set of links to relevant documents coming from the Information Repository, carefully selected in order to answer the question in the most proper way.

Legacy answers:

The SURRENDER VALUE calculated at the next renewal date, i.e. (31-12-2003) of YOUR Life insurance contract, i.e. (N. 987) will be the sum of (14.000,00) Euro.

Knowledge Retrieval:

SURRENDER VALUE is “The amount an insurance company will repay to a policyholder who wishes to terminate his policy early.” The ontology concept is: #6666; its parent is: #555; its relatives are: #777, #888, #999:

Information Retrieval:

Link to "SURRENDER" is present in docs. (Doc. #222, #333, #444)

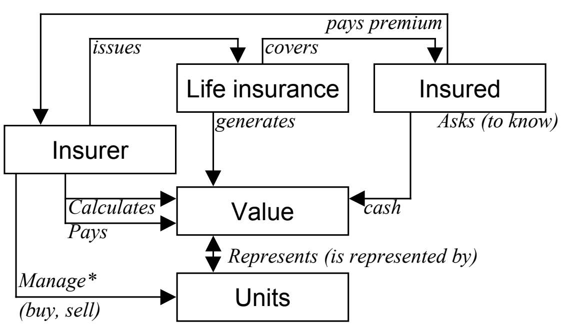
Link to "SURRENDER VALUE" is present in docs: (Doc. #555, #666)

Link to "SURRENDERING" is present in docs. (Docs. #777, #888).

The most important achievement of this system is the seamless, everywhere, everytime, realtime access to the situation stored in the Legacy system, since all calculations are performed on the fly (on demand).

4.6 Business scenario

Life insurance business can be depicted by an additional sketch:[9]



The transition from traditional business in the insurance domain to eBusiness requires a new specific communication channel. We use a web-based communication in order to offer a semantic sensitive eCRM on a self help basis. Registered users receive a combinatorial UserId and a password. Users access the enterprise web portal, which enables to track the user’s behavior and to provide the best service available. An ontology-driven, natural language based Q-A dialog is supported.

A typical user types the UserId and a password to enter as an IES-User. Firstly, the user history is retrieved and an authentication is performed, in order to allow the knowledge retrieval of related objects from the Legacy applications.

The list of managed policies is retrieved and generated. At this point we can answer some typical queries and FAQs. Mostly important, a Legacy call is performed which computes user-related datatype values, without a static batch pre-calculation..

Next, users can formulate a natural language query. The eCRM system manages the query as a free text, i.e. passes the string to the semantic parser that tags and disambiguate relevant textual chunks. The resulting parse tree is passed to the Link Elicitor in order to elicit the knowledge.

The data flow is in XML, a frame based knowledge representation is adopted and the set of data includes:

an Instance of the Classs (Object) “Insured” having following attributes – values:
<Id 999>
<Family Name XXX>
<Name XXX>
<DateOfBirth DDMMYYYY>
<Policy 1234>, <Policy 1235>, <Policy 1236>
<Query: “Phrase in N.L. to be parsed”>

At this point the query is disambiguated and the system tries to discover the needed Legacy (or Web) service to be invoked. In order to facilitate reasoning, disambiguation is aided by some domain-oriented heuristics concerning typical constructions, e.g.

“to know” is interpreted as “to calculate”, “how to do” as “procedure for” or “rules concerning” etc.).

The query is then submitted to the IR component, the Knowledge Retrieval and the Legacy application. The first returns textual chunks, the second returns the ontological data and instances of information objects, such as regulations. The last one returns the required calculation for a given instance.

After the Legacy service is determined, the call must be prepared appropriately. The formal parameters list will be replaced with real data coming from the context.

Finally the IKF Invoker performs the Legacy call. For instance, a swap between two life insurance products involves some expenses, therefore the following Legacy call will be performed:

GET_VAL_SWAP (Id = 999, Age = 37, Policy = 1234, Tarif = RXII, From = 01-01-2004, Years = 6)

[9] We consider the management of gathered funds as an external activity, in order to avoid the inclusion of the whole Stock Exchange dictionary.

Finally, the IKF Demon receives the data from the Legacy, and generates a report as an HTML/XML page.

This cycle enables a user, to know a value calculated "on the fly" by the Legacy, to access to the links returned by the IR component, and to read any regulation contained in the corpus, by following knowledge items and their links. In addition, users can fully understand the operation performed, since the reference conceptual model defined in the ontology is shown as well.

5. Conclusions

We have illustrated a concrete enterprise system developed using innovative techniques and algorithms. Our aim has been to demonstrate the feasibility of an ontology-driven system for intelligent service discovery, Legacy application integration, and dynamic query answering, as opposed to traditional techniques relying on heavyweight, batch pre-processing.

The paper has presented some aspects of the IKF-IES application concerning its architecture, its underlying core and domain ontology, and its functionalities.

The IKF framework has been only briefly described, since IES design is partly independent from it. As a matter of fact, the core functionality of IES is ensured by its ontological component, which is customizable, reusable, and is still growing to cover other subdomains within the insurance domain.

Besides architectural and technical choices, including system components and specification languages, the glue of IES is provided by its formal conceptual model, which enables query reformulation, database integration, and provision of background domain knowledge within a common framework. Formal specification of insurance workflows (by using the D&S design pattern) allows a formal matching between procedural domain knowledge and the system components. This peculiar use of ontologies is exploited in related work carried out within the IKF project for different domains, for example anti-money laundering regulations in the banking domain [Gangemi et al. 2001], and service-level agreements.

Further developments of the insurance ontology and related applications will include the matching between legal insurance regulation and policy enactment.

6. Acknowledgments

Some of achievements described in the article are inherited from Eureka co-funded project named IKF. We thanks the European Eureka Institution for financial help provided. The work described in this paper draws upon the contributions of many people, to whom the authors are indebted. Of course the authors are the sole responsible forany possible mistake. The authors would like to thank the IKF Consortium for its support.

7. References

Gangemi A, Pisanelli DM, Steve G, "An Overview of the ONIONS Project", *Data and Knowledge Engineering*, 31, 1999.

Gangemi A., Pisanelli DM., Steve G., A Formal Ontology Framework to Represent Norm Dynamics. Proc. of *Second International Workshop on Legal Ontologies,* Amsterdam, 2001.

Gangemi A., Guarino N. Masolo C. Oltramari A., Schneider L. "Sweetening Ontologies with DOLCE", Gomez-Perez & Benjamins (eds.), Proceedings of EKAW02, 2002.

Gangemi A., Mika P. 2003, *Understanding the Semantic Web through Descriptions and Situations,* Meersman R, et al. (eds.), Proceedings of ODBASE03, Springer, Berlin, 2003.

Gruninger M. and Fox M.S., "*The Logic of Enterprise Modelling*", in Modelling and Methodologies for Enterprise Integration, pages 140--157, 1996.

Jarrar M, Verlinden R, Meersman R, Ontology-based Customer Complaint Management, Proceedings of WORM03 Workshop at ODBASE03, Springer, Berlin, 2003.

Kietz, J.-U., Maedche, A. and Volz, R., Extracting a Domain-Specific Ontology Learning from a Corporate Intranet, *Conference on Natural Language Learning (CoNLL'2000)*. Lisbon, Portugal, September 13-14, 2000.

Masolo C., Borgo S., Gangemi A, Guarino N, Oltramari A, Schneider L, Deliverable D18: "The WonderWeb Library of Foundational Ontologies", **http://wonderweb.semanticweb.org**, 2003.

Miller G.A., Beckwith R., Fellbaum C., Gross D. & Miller K., "Introduction to WordNet: An On-Line Lexical Database", http://www.cosgi.princeton.edu/~wn; August 1993.

Swartout B., Patil R., Knight K. and Russ T., "Toward Distributed Use of Large-Scale Ontologies", in *Proceedings of the Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Alberta, Canada, 1996.