UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

ON IDENTIFYING KNOWLEDGE PROCESSING REQUIREMENTS

Alain Léger, Lyndon J.B. Nixon and Pavel Shvaiko

May 2005

Technical Report # DIT-05-045

Also, to appear in Proceedings of ISWC'05

On Identifying Knowledge Processing Requirements*

Alain Léger[1], Lyndon J.B. Nixon[2], and Pavel Shvaiko[3]

[1] France Telecom R&D, Rennes, France
alain.leger@francetelecom.com
[2] Free University of Berlin, Berlin, Germany
nixon@inf.fu-berlin.de
[3] University of Trento, Povo, Trento, Italy
pavel@dit.unitn.it

Abstract. The uptake of Semantic Web technology by industry is progressing slowly. One of the problems is that academia is not always aware of the concrete problems that arise in industry. Conversely, industry is not often well informed about the academic developments that can potentially meet its needs. In this paper we present a first step towards a successful transfer of knowledge-based technology from academia to industry. In particular, we present a collection of use cases from enterprises which are interested in Semantic Web technology. We provide a detailed analysis of the use cases, identify their technology locks, discuss the appropriateness of knowledge-based technology and possible solutions. We summarize industrial knowledge processing requirements in the form of a typology of knowledge processing tasks and a library of high level components for realizing those tasks. Eventually these results are intended to focus academia on the development of plausible knowledge-based solutions for concrete industrial problems, and therefore, facilitate the uptake of Semantic Web technology within industry.

1 Introduction

The industrial uptake of Semantic Web technology is still slow. On the one hand, industry is not often well informed about the academic developments that can potentially meet its needs. On the other hand, academia is not always aware of the concrete problems that arise in industry, and therefore, the research agenda and the achievements thereof are not tailored for an easy migration to industrial applications. Thus, in order to increase the industrial uptake of Semantic Web technology, there is a clear need for researchers to have access to a study of industrial requirements, thereby focusing their activities on research challenges arising exactly from those requirements. Simultaneously, industry needs to have access to studies identifying plausible knowledge-based solutions to technological problems in their business scenarios, as well as to success stories which demonstrate the value of adopting knowledge-based technology.

On a large scale, industry awareness of the knowledge-based technology has started only recently, e.g., at the EC level with the IST-FP5 thematic network Ontoweb [1] which had brought together around 50 companies worldwide which are interested in Semantic

* The work described in this paper is supported by the EU Network of Excellence Knowledge Web (FP6-507482).

[1] http://www.ontoweb.org/

Web technology. These companies influenced significantly a global vision of Semantic Web technology developments, provided success stories and guidelines for best practices. Based on this experience, within the IST-FP6 network of excellence Knowledge Web[2], an in-depth analysis of the concrete industry needs in the key economic sectors has been identified as one of the next steps towards stimulating the industrial uptake of Semantic Web technology. To this end, this paper aims at identifying technology locks within the concrete business scenarios, and at discussing plausible knowledge-based solutions to those locks.

The contributions of the paper are:

- a collection of the use cases and their detailed technical analysis used to determine European industry needs with respect to knowledge-based technology;
- a typology of knowledge processing tasks and a library of high level components for realizing those tasks used to focus academia on the current industry requirements.

The rest of the paper is organized as follows. A set of use cases collected from industry and their preliminary analysis are presented in Section 2 and Section 3 respectively. Section 4 describes, via an example, a methodology for identifying technology locks occurring in the use cases and discusses the appropriateness of knowledge-based approaches for resolving those locks. Section 5 summarizes industrial knowledge processing requirements as a typology of knowledge processing tasks and a library of high level components. Section 6 considers the related efforts and industrial experiences with some components of the library proposed. Finally, Section 7 reports some conclusions and discusses the future work.

2 Use Case Collection

A major barrier between industry and research is that the former thinks in terms of problems and solutions and the latter thinks in terms of technologies and research issues. A business use case provides a brief description of a concrete business problem. A technical use case relates a business problem to a solution, and a solution to a technology, which, in turn, may lead to a research issue. Therefore, business use cases and their technical analysis provide an effective means for enterprises to argue and communicate their needs to academia.

In order to enable the collection of use cases we invited companies interested in Semantic Web technology to form an Industry Board (IB). Around 50 companies [3] from 12

[2] http://knowledgeweb.semanticweb.org/

[3] Some examples are Acklin; Amper; Berlecon Research GmbH; Biovista; Bitext; British Telecom, BT; Computas Technology; Daimler Chrysler; Deimos Space; Distributed Thinking; EADS Airbus Industry; France Télécom Division R&D; Green Cacti; HR-XML Consortium Europe; IFP Institut Français du Pétrole; IKV++ Technologies AG; Illycaffè S.p.A.; Labein; Merrall-Ross International Ltd; Neofonie Technology Development and Information Management GmbH; NIWA; Office Line Engineering; QUARTO Software; RISARIS; Robotiker Tecnalia; Semtation GmbH; SNCF; Synergetics; Tecnologia, Información y Finanzas; TSF S.p.A.; Telefonica; Thalés; TXT e-solutions; WTCM, see for details http://knowledgeweb.semanticweb.org/o2i/

Table 1. The use cases collected in 2004.

#	Use case name	IB member	Web page
1	Recruitment	WWJ GmbH	http://www.wwj.de
2	Multimedia content analysis	Motorola	http://www.motorola.com
3	eScience portal	Neofonie	http://www.neofonie.de
4	News aggregation service	Neofonie	http://www.neofonie.de
5	Product lifecycle management	Semtation	http://www.semtation.de
6	Data warehousing in healthcare	Semtation	http://www.semtation.de
7	B2C marketplace for tourism	France Telecom	http://www.francetelecom.com
8	Digital photo album	France Telecom	http://www.francetelecom.com
9	Geosciences project memory	IFP	http://www.ifp.fr
10	R&D support for coffee	Illy Cafe	http://www.illy.com
11	Real estate management	TrenItalia	http://www.trenitalia.com
12	Hospital information system	L&C	http://www.landcglobal.com
13	Agent-based system for insurance	Acklin	http://www.acklin.nl
14	DCVD Semantic Web portal	DaimlerChrysler	http://www.daimlerchrysler.com
15	Specialized web portals	Robotiker	http://www.robotiker.es
16	Integrated access to biology	Robotiker	http://www.robotiker.es

industry sectors[4] have joined the initiative so far. We asked the IB members to describe the actual or hypothetical deployment of Semantic Web technology in their business environments. Thus, in 2004, 16 use cases were provided by 12 companies, see Table 1. The breakdown of the use cases with respect to industrial sectors is shown in Figure 1. For example, the most active sectors (in providing use cases) were *service industry* and *media & communications*. A detailed description of all the use cases can be found in [19].

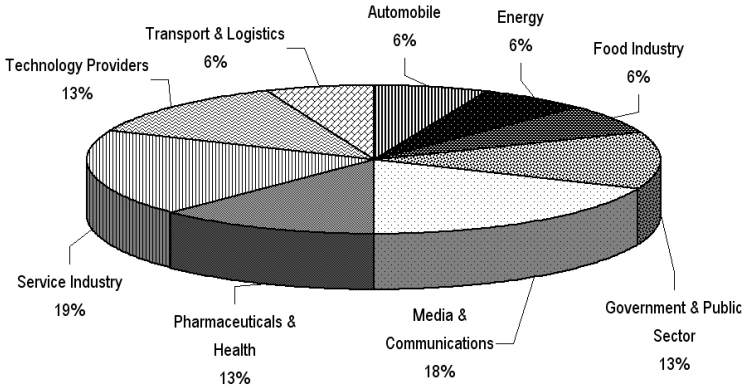

Fig. 1. Breakdown of the use cases by industry sectors

3 Preliminary Analysis of Use Cases

We have performed an initial analysis of the use cases of Table 1 aiming at an overview of the current industrial needs. The IB members were requested to point out technolog-

[4] These are: Health, Telecom, Automotive, Energy, Food, Media, Transport, Space, Publishing, Banking, Manufacturing, Technology sectors.

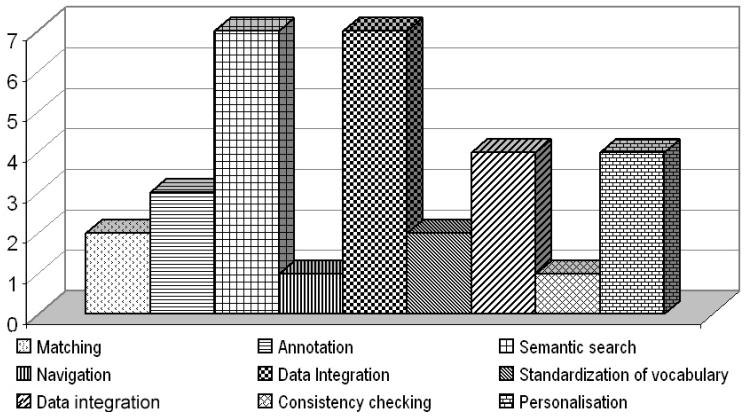

Fig. 2. Preliminary vision for solutions sought in the use cases

ical problems they have encountered in their businesses as well as the knowledge-based approaches they view as plausible solutions to those problems. As a result, we obtained:

- a set of typical business problems for which an industry expert has determined that a plausible solution can come from the knowledge-based technology;
- a set of typical technological issues/locks (and corresponding research challenges) which knowledge-based technology is expected to overcome.

Figure 2 and Figure 3 illustrate the results of this preliminary analysis. The former shows the type of business/market needs for which Semantic Web was considered by the industry as a relevant technological approach. The latter shows the type of technological issues which industry considers that Semantic Web must be able to overcome. Let us discuss them in turn.

Figure 2 shows that in nearly half of the collected use cases industry has identified *data integration* and *semantic search* as typical business problems for which they expect Semantic Web to provide solutions. Below, we illustrate, with the help of two use cases, how a concrete business problem can indicate the need for a knowledge-based solution.

The first use case is taken from the Human Resources field. In this use case, the expert saw a solution needed for the problem of *matching between job offers and job seekers*. The key reason given by the expert for such a need was that ”employee recruitment is increasingly being carried out online. Finding the best suited candidate in a short time should lead to cost cutting and resource sparing”. The second use case is focused on the problem of data warehousing for a healthcare scenario. The solution is seen as being to *introduce a common terminology for healthcare data and wrap all legacy data in this terminology*. The key reason given by the expert for such a need was that ”it reduces the time and cost involved in *data integration* and *consistency checking* of the data coming from different healthcare providers”.

Figure 3 shows the technological issues/locks which industry consider that Semantic Web approaches might overcome. Here, the key issues are: *ontology matching*, i.e., resolving semantic heterogeneity between heterogeneous ontologies; *knowledge extraction*, i.e., populating ontologies by extracting data from legacy systems; *ontology development*, i.e., modeling a business domain, authoring tools, re-using existing ontologies.

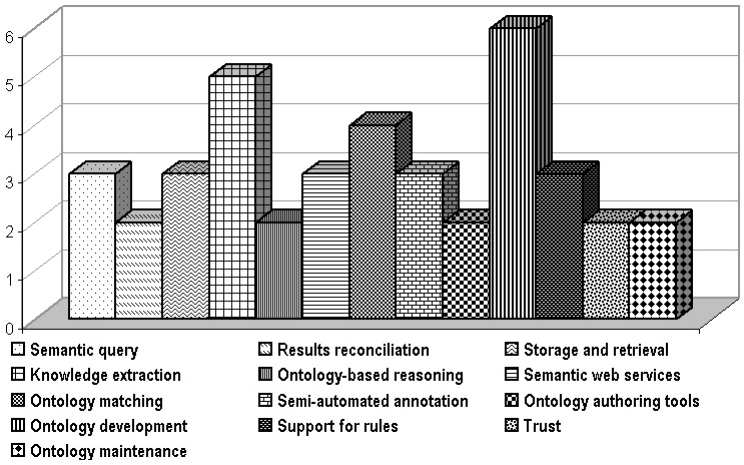

Fig. 3. Preliminary vision of technological issues in the use cases

Let us now illustrate, with the help of yet another use case from our collection, how a concrete business problem can be used to identify such technology locks. The use case deals with the problem of providing unified access to biological repositories on the Internet. The problem is attacked by modeling those repositories (notice they may store their data according to various data/conceptual models) as ontologies. This, in turn, is performed by analyzing the underlying data instances. Finally, since those newly created ontologies will likely use different terminologies, mappings between them must also be established. Hence, in this case the technological issues to overcome are *knowledge extraction* and *ontology matching*.

From the preliminary analysis we can already draw the areas of Semantic Web research which could be of great value to industry (e.g., ontology matching). This analysis (by experts estimations) provides us with a preliminary understanding of scope of the current industrial needs for solutions and concrete technology locks where knowledge-based technology is expected to provide a plausible solution. However, to be able to answer specific industrial requirements, we need to conduct further an in-depth technical analysis of the use cases, thereby associating to each technological issue/lock a concrete knowledge processing task and a component realizing its functionalities.

4 Detailed Analysis of Use Cases

4.1 A methodology

A methodology used for a precise identification of technology locks and knowledge processing tasks they require is based on Rational Unified Process (RUP) [1, 15] which, in turn, extensively exploits Unified Modeling Language (UML) [6]. Out of six standard steps of the RUP approach (i.e., *business modeling*, *service requirements*, *analysis*, *design*, *implementation*, and *validation*) we focus only on three of them, namely, *service requirements*, *analysis*, and *design*:

– *Service Requirements.* These are a set of services available from a system in order to implement a business case. They are determined through analysis of functional

needs, which in turn imply some technical constraints (e.g., time response, number of connected customers) of a system to be developed. Service requirements are expressed via UML technical use cases.

– *Analysis.* This step performs initial system partitioning with respect to its main processing tasks and analyses the use cases in detail. In particular, use cases are refined with the help of UML sequence diagrams, which incorporate the modules for the architecture proposal and the information flow between these modules to fulfill the use case functionality. Notice that during this step we identify the use case's technology locks.
– *Design.* This step refines and homogenizes classes, and drafts the architecture design. It is partially specified in the context of our analysis, namely, it aims only at identifying knowledge processing tasks which resolve technology locks determined in the previous step. We structure knowledge processing tasks as primary and secondary tasks according to their influence on the architecture of a system to be developed. Primary tasks are the common parts for most of actions or parts of actions of the system. Secondary tasks are additional requirements, i.e., extensions of the common parts.

4.2 The methodology by an example

Let us discuss with the help of the B2C marketplace for tourism use case how the above introduced methodology is used for the identification of technology locks and possible knowledge processing tasks resolving them. We first provide a summary of the use case, then we discuss the service requirements, analysis, and design steps.

Use Case Summary. The B2C marketplace for tourism use case considers a scenario where users are offered an one-stop browsing and purchasing of personalized tourism packages by a dynamic combination of various tourism offers (e.g., travel, accommodation, meals) from different providers. A detailed description of this business scenario can be found in [19]. Figure 4 illustrates two primary use cases of the B2C marketplace system.

The first use case, which is called *to plan a nice weekend*, constitutes the entry point inside the marketplace allowing customers to define their personal needs. The platform takes care of identifying potentially useful contents and services, accessing multiple providers and selecting only the relevant ones.

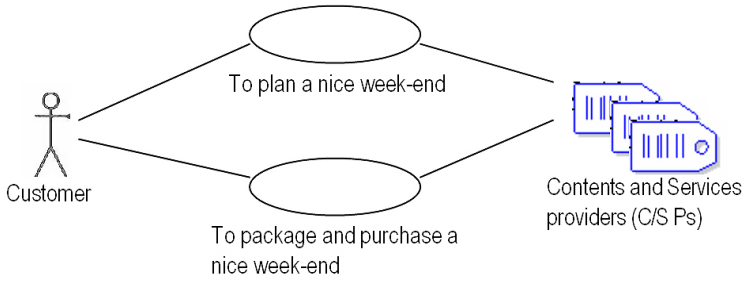

Fig. 4. UML use case diagram for B2C marketplace for tourism

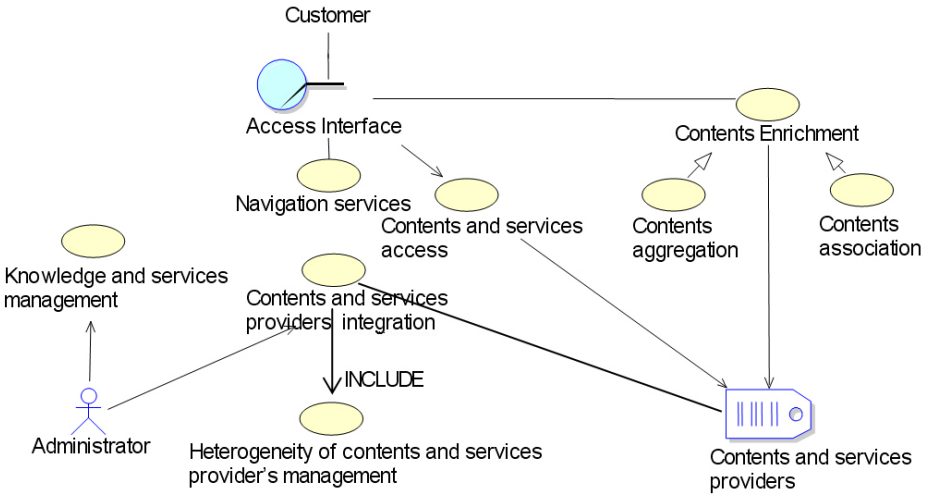

Fig. 5. UML technical use case diagram for B2C marketplace for tourism

The second use case, which is called *to package and purchase a nice weekend*, requires (i) a dynamic aggregation of relevant contents and services (e.g., transport, accommodation, leisure activities), (ii) an automated packaging of week-end proposals, and (iii) facilities for purchasing them on-line.

Service requirements. The technical use case diagram is presented in Figure 5. Let us discuss its actors.
Customer and Access Interface. A customer with the help of its access interface (e.g., mobile phone) accesses services available within the system through the authentication mechanism, personalization, and session management.
Contents and Services providers (C/S Ps). Contents and services providers manage their offers autonomously, i.e., the system does not impose any constraints. Each C/S P has its own rules for structuring information at the protocol, syntactical, and semantic levels. The system adapts itself via an Administrator or automatically.
Administrator performs (i) referencing of new contents and services providers, and (ii) internal knowledge representation and management.

Analysis. During this step, we analyze each technical use case of Figure 5 in detail. In particular, we consider *navigation services*, *contents and services access*, *contents enrichment*, *contents aggregation*, *contents association*, *contents and services provider's integration*, *heterogeneity of contents and services provider's management*, and *knowledge and services management* technical use cases.

For lack of space we discuss here only the *contents aggregation* technical use case. First, we report the actors it involves, then we provide its summary, inputs and outputs, and finally we analyze with the help of sequence diagrams the flow of its events, possible technology locks and potential knowledge-based solutions.

Actors: Customer and Access Interface, C/S Ps.

Summary: The use case *contents aggregation* is inherited from the use case *contents enrichment*. A global schema, which is a model for the data of all the C/S Ps, captures

the knowledge of the domain. The use case performs a fusion of the information issued by different C/S Ps. It aims at providing a user with the result which has the following characteristics:

- No duplication and redundant information;
- Avoid the user having to aggregate the contents issued from different C/S Ps.

Preconditions and inputs:

- The use case *contents and services access* has been executed;

Post-conditions and outputs:

- The aggregated contents are transferred to the access interface.

The flow of events for the *contents aggregation* technical use case is presented in Figure 6. Let us discuss it in detail. The system (ManageContentAggregation component) starts from mapping the data (potentially expressed using different data models) among C/S Ps involved in the processing of the request of a user. This step is essential in order to evaluate the contents of each C/S P, and hence, detect redundancies, complementary information, etc. The flow of events is as follows:

- Identification of the mappings between different data models (requestSchemas);
- Contents aggregation (manageContent): check for duplicated information, fusion of complementary information are operated by the ControlContent component;
- Transformation of the result of contents aggregation into XML formalism;
- The results encoded in XML are transferred to the access interface (loadXmlStream).

Technology locks identification: Technology locks are highlighted in Figure 6. These are the MappingContent and ControlContent components. Let us discuss them in turn.

It is crucial to be able to dynamically discover semantic mappings between the contents of different C/S Ps. The current solution follows the data integration approach which is to create static correspondences between data models [16]. In this case, mappings can be specified in a declarative manner (e.g., manually). However, this solution does not satisfy requirements of the business case. In fact, C/S Ps may appear and disappear over the network, change their contents, schemas, and so on. Thus, the problem

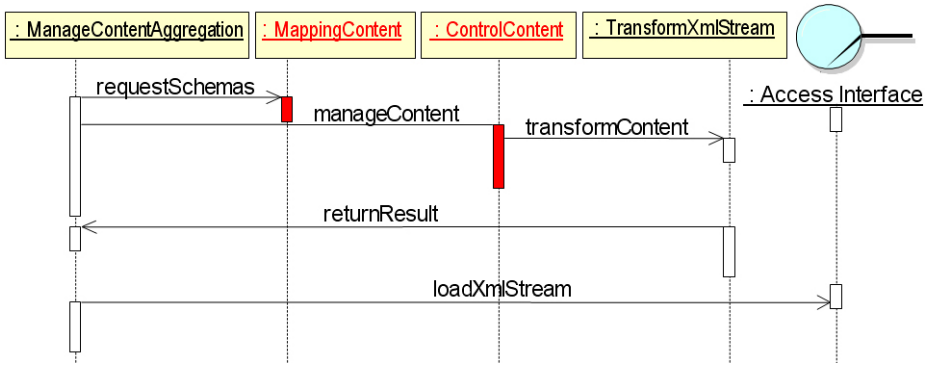

Fig. 6. Flow of events: Contents aggregation technical use case

is to determine those semantic correspondences dynamically. For example, given two XML schemas, suppose in the first schema the *address* element consists of the attributes *name*, *town*, and *postcode* and in the second schema the *address* element is split down into three sub elements $street\_name$, $post\_code$, and *town*. Then, a solution should be developed in order to determine correspondences between the semantically related entities, e.g., the *address* element in the first schema should be mapped to the *address* element in the second schema. A more complex solution is required to determine which attributes of the first schema are to be mapped to the elements of the second schema.

The second technology lock is to execute the correspondences (mappings) produced as output of the MappingContent component. As the use case requires, mapping's execution should not only translate the source data instances under the expected common schema, but also check for duplications, and, if any detected, discard them. This lock can be decomposed into two sub-locks. The first sub-lock is to generate query expressions (out of the correspondences determined in the previous step) that automatically translate data instances of the C/S P's schemas under an integrated schema. For example, [29] provides a standard data translation solution. Such a solution is based on the assumption that correspondences between schema elements are only identified (using a binary choice: a mapping exists or does not exist). However, if the correspondences between schema elements can be determined by providing a more informative specification, e.g., a particular type of the correspondence, namely a logical relation (equivalence, subsumption), then data translation operation could also be performed more accurately. The second sub-lock is to reconcile the data instances. Let us consider one example which deals with duplicates. The current solution interprets data instances as strings and checks if two strings are identical. However, in general, one C/S P may adopt the use of a standard while another C/S P adopts the use of fully expanded descriptions, and so on. For example, ⟨Oro Stube, Restaurant, Trento, TN, NULL⟩; ⟨Oro Stube, ristorante-pizzeria, Povo Trento, TN, I-38050⟩; ⟨Oro Stube, Trento, NULL, 38100⟩ all refer to the same place Oro Stube. Thus, a solution should be developed in order to detect (meaningfully) identical instances and discard the less informative ones.

Design. Having identified technology locks of the B2C tourism marketplace system, we are able to propose knowledge processing tasks required in order to develop plausible solutions to those technology locks. In particular, our technical use case requires the *matching*, *data translation*, and *results reconciliation* knowledge processing tasks:

- *Matching*. This task aims at (on-the-fly and automatic) determining semantic correspondences between the contents of C/S Ps and the global schema. It takes two data/conceptual models (e.g., XML schemas, OWL ontologies) and returns a set of mappings between the entities of those models that correspond semantically to each other. This task is necessary to ensure semantic homogeneity at schema level among C/S Ps, and therefore, it is classified as a primary task in the context of the B2C marketplace system.
- *Data translation*. This task aims at generating query expressions (out of mappings determined as a result of *matching*) that automatically translate data instances between heterogeneous information sources. This task is necessary to ensure semantic homogeneity at the level of data instances, and therefore, it is classified as a primary task in the context of the B2C marketplace system.

– *Results reconciliation*. This task aims at detecting redundancies, duplications, and complements among the data coming from different C/S Ps which are involved in the processing of the request of a user. It takes as input responses of each C/S P involved in the processing of the request, performs all the necessary operations (e.g., cleaning, fusion) and produces a reconciled result. This task is necessary to provide a user with an accurate way of accessing the requested data, and therefore, it is classified as a primary task.

In the above described manner we determine technology locks, discuss appropriateness of the knowledge-based technology, and required knowledge processing tasks for all the technical use cases of the B2C marketplace scenario. Also, we have considered some other business cases (e.g., recruitment of human resources (HR), multimedia content analysis and annotation (MCAA)) and we have analyzed them in detail as demonstrated above, see [26].

5 Knowledge Processing Tasks and Components

Based on the primary and secondary knowledge processing tasks determined during the technical use case analysis (conducted for four business cases, see [26]), we construct a typology of knowledge processing tasks and a library of components for realizing those tasks, see Table 2 and Table 3.

Table 2. Typology of knowledge processing tasks & components. Part 1 - Primary tasks.

#	**Knowledge processing tasks**	**Components**
1	Ontology Management	Ontology Manager
2	Matching	Match Manager
3	Matching Results Analysis	Match Manager
4	Data Translation	Wrapper
5	Results Reconciliation	Results Reconciler
6	Composition of Web Services	Planner
7	Content Annotation	Annotation Manager
8	Reasoning	Reasoner
9	Semantic Query Processing	Query Processor

Table 3. Typology of knowledge processing tasks & components. Part 2 - Secondary tasks.

#	**Knowledge processing tasks**	**Components**
1	Schema/Ontology Merging	Ontology Manager
2	Producing Explanations	Match Manager
3	Personalization	Profiler

Our typology includes 9 primary tasks and 3 secondary tasks. Some tasks are required to be implemented within a single component. For example, (schema/ontology) matching, matching results analysis, and producing explanations of mappings are the functionalities of the match manager component. Thus, the library of high level components contains less components than the number of knowledge processing tasks identified. In particular, it consists of 9 components. Let us discuss knowledge processing tasks and components of Table 2 and Table 3 in more detail.

Ontology Management, Schema/Ontology Merging and Ontology Manager. These aim at (i) ontology maintenance, e.g., editing concepts, resolving name conflicts, browsing ontologies, and (ii) merging (multiple) ontologies, e.g., by taking the union of the axioms, according to evolving business requirements, see [9, 14, 17]. For example, let us consider the HR scenario. It requires exploiting a common HR ontology. Since the job market or some aspects of the recruitment domain such as qualifications may alter, the HR ontology has to be updated. In fact, with a globalization of the job market, recruitment applications might be submitted from new countries which have different educational systems. Therefore, higher level qualifications must be identified within the system and related to existing qualifications. Moreover, in a decentralized distributed environment such as the Web, it is reasonable to expect existence of multiple ontologies, even on the same topic. Thus, some of the relevant ontologies might be useful for extending the HR ontology, and, hence, are need to be merged into it.

Matching, Matching Results Analysis, Producing Explanations and Match Manager. These aim at discovering mappings between the entities of schemas/ontologies which correspond semantically to each other, see [23, 24]. Mappings are typically specified (i) by using coefficients rating match quality in the [0,1] range, see [5, 10, 20, 30], or (ii) by using logical relations (e.g., equivalence, subsumption), see [11, 12]. For example, in the HR scenario, a requirement for Java programming skills may be matched against C++ programming skills as similar with a coefficient of 0.8 or as Java $\sqsubseteq$ C++.

Depending on the application requirements, some further manipulations with mappings (e.g., ordering, pruning) can be performed, see [8]. For example, in the HR scenario, the complexity of qualifications and work experience suggest that exact matches between job requirements and applicants are unlikely to happen; rather a ranking mechanism is required to express the extent to which, for example, the equivalence might be assumed. In fact, when an applicant states that (s)he has a proficiency in C++, how would this rank differently against vacancies requiring persons with skills in Java, Microsoft .NET, or object oriented programming?

State of the art matching systems may produce effective mappings. However, these mappings may not be intuitively obvious to human users, and therefore, they need to be explained, see [7, 25]. In fact, if Semantic Web users are going to trust the fact that two terms may have the same meaning, then they need to understand the reasons leading a matching system to produce such a result. Explanations are also useful when matching (large) applications with thousands of entities (e.g., business catalogs, such as UNSPSC and eCl@ss). In these cases automatic matching solutions will find a number of plausible mappings, hence, some human effort for performing the rationalization of the mapping suggestions is inevitable. Generally, the key issue here is to represent explanations in a simple and clear way to the user. For example, in the HR scenario, explanations should help users of the HR system to make informed decisions on why a job vacancy requirements meet a job applicant request.

Data Translation and Wrapper. These aim at an automatic manipulation (e.g., translation, exchange) of instances between information sources storing their data in different formats (e.g., OWL, XML), see [21, 28]. Usually, for the task under consideration, correspondences between semantically related entities among schemas/ontologies are assumed to be given. They are taken in input, processed according to an application

requirements, and are returned in output as *executable* mappings. For example, in the HR scenario, a wrapper acts as an interface to the input data such that both requests from and responses to the system may be expressed in RDF while the underlying data continues to be stored in its original format.

Results Reconciliation and Results Reconciler. These aim at determining an optimal solution for returning results from the queried information sources. The problem should be considered at least at two levels: (i) contents, e.g., for discarding redundant information, and (ii) routing performance, e.g., for choosing the best (under the given conditions) plan for delivering results to the user, see [22]. In the B2C tourism marketplace scenario, this task prevents customers, for example, from encountering several (identical) responses about the same restaurants or different opening times for the same museum.

Composition of Web Services and Planner. These aim at an automated composition of the pre-existing web services into new (composed) web services, thereby enabling the latter with new functionalities, see [4]. Technically, composition is typically performed by using automated reasoning approaches (e.g., planning, see [27]). In the B2C tourism marketplace scenario, composition of web services is needed when organizing a travel journey. In particular, for the combination of transport and hotel reservation services.

Content Annotation and Annotation Manager. These aim at an automated generation of metadata for different types of contents, such as text, images, audio tracks, etc., see, for example [2]. Usually, an annotation manager has in input the (pre-processed) contents and some sources of explicitly specified domain knowledge and outputs content annotations. For example, in the MCAA scenario, knowledge-based analysis of the audiovisual content should automatically generate semantic metadata, for instance, by extracting the audiovisual features (e.g., color, shape) from visual objects, and by linking them to the semantically equivalent concepts defined in the MCAA ontologies.

Reasoning and Reasoner. These aim at providing a set of logical reasoning services (e.g., subsumption, instance checking tests, see [13]), which are (heavily) tuned to particular application needs. For example, when dealing with multimedia annotations, logical reasoning can be exploited in order to check consistency of the annotations against the set of spatial (e.g., left, right, adjacent, near) and modal (possibility, necessity) constraints. Thus, ensuring that the objects detected in the multimedia content correspond semantically to the concepts defined in domain ontologies. For example, in the football domain, it should be checked whether a goalkeeper is located *near* the goal and potentially holds a ball in his/her hands. The key issue here is in the development of optimizations over the standard reasoning techniques tailored to specific application tasks, because, in general, modal/temporal logic reasoning procedures do not scale well.

Semantic Query Processing and Query Processor. These aim at rewriting queries by exploiting terms from the pre-existing ontologies, thus, enabling a semantics-preserving query answering, see [2, 18]. For example, in the MCAA scenario, query processor should be able to interpret queries by exploiting a set of domain ontologies in order to return relevant multimedia content (e.g., images, videos). Notice that the user should be able to specify queries in different ways, for example, as (i) high level concepts, e.g., *holiday*, *beach*; (ii) natural language expressions , e.g., *give me all the photos of Trento*; (iii) sample images.

Personalization and Profiler. These aim at an adaptation of functionalities available from a system to the needs of groups of users, see [3]. Typical tasks of a profiler include automatic generation and maintenance of user profiles, personalized content management and mining, etc. For example, in the MCAA scenario, users might want to participate in different social networks and to share some annotations over them. Thus, they need a support for new contact's recommendation, adaptive navigation through these new contacts, and so on. In turn, adaptation might be performed along different dimensions, where the use of Semantic Web technology is promising, namely: user' terminal (e.g., PDA, cell phone), external environment (e.g., language, location).

6 Discussion

The IST-FP5 project Ontoweb (2001-2004) has brought (EC) industry awareness of Semantic Web technology on a large scale. In particular, a special interest group on Industrial Applications[5] was formed. It collected over 50 use cases (notice, their majority dealt only with technology producers), which, in turn, provided a good overview of the expectations from Semantic Web technology. Based on those foundations, the subsequent IST-FP6 Network of Excellence Knowledge Web (2004-2007) has deliberately focused on the potential adopters of the technology and an in-depth analysis of the use cases.

In this paper, we report our first results of the business use cases collection and analysis as targeted by Knowledge Web. By a preliminary analysis of the collected use cases we categorized the types of solutions being sought for, and the types of technological locks which arise when realizing those solutions. By a detailed technical analysis of the selected use cases we identified precisely where in the business processes the technology locks occur, described the requirements for technological solutions that overcome those locks, and argued for the appropriateness of knowledge-based solutions. Moreover, a quick analysis of the other business cases of [19] have shown that most of the knowledge processing tasks of Table 2 and Table 3 repeat with some variations/specificity from use case to use case. This observation suggests that the constructed typology is stable, i.e., it contains (most of) the core knowledge processing tasks stipulated by the current industry needs. By drawing from concrete industrial use cases the knowledge processing tasks and components that can provide expected solutions, we link business problems to specific research challenges. We expect the Semantic Web research community to address those challenges. Once knowledge processing components are provided by research, their practical usefulness and contribution to technology transfer from academia to industry can be assessed through an extensive evaluation within different industrial contexts.

Thus, for example in the HR scenario, the sought-for solution is the *semantic* matching between job offers and job applications. By a technical use case analysis we located where in the business process the lock occurs and defined the requirements with respect to the *matching* task and the *match manager* component. Hence, we have already provided (i) a client industry with a clear identification of the place where the system requires knowledge-based solutions and (ii) researchers with a clear definition of

[5] http://sig4.ago.fr

the requirements that must be met by their prototypical implementations of knowledge processing components. In particular, in the HR scenario, some existing implementations of a match manager (e.g., [5, 30]) have been plugged into the business process at the identified location. A prototype has been tested by the client industrial partner, and it had demonstrated a better characteristics (e.g., precision, recall) with respect to the legacy solution. Thus, experience of this use case and some other use cases (e.g., MCAA scenario) gives us a preliminary vision that the proposed approach is able to facilitate the industrial uptake of Semantic Web technology.

7 Conclusions and Future Work

We have presented a set of business cases collected from enterprises which are interested in Semantic Web technology. We discussed via examples a methodology for the identification of technology locks in business cases, appropriateness of the knowledge-based technology, and possible approaches resolving those locks. We summarized industry requirements with respect to the knowledge-based technology as a typology of knowledge processing tasks and a library of high level components for realizing those tasks. We intend our typology as a guide for academic activities, thereby connecting concrete industrial problems with research efforts. Thus, by facilitating the communication of industry requirements to academia and directing research results back to industry, where those results are relevant, we contribute to the process of increasing the industrial uptake of Semantic Web technology.

This work represents only an initial step. In fact, to build the typology presented in this paper we have conducted an in-depth analysis of 4 (out of 16) use cases. Thus, we still have to scrutinize the rest of the use cases and update our typology, although we have a preliminary vision that in those use cases, most of the knowledge processing tasks repeat the current typology. Emerging business cases will also be tracked, as they will likely generate new requirements. For example, future trends such as semantic web services, grid computing, social networking will give rise to knowledge processing components for web service discovery, orchestration; distributed reasoning; and so on.

References

1. Rational software corporation. http://www-306.ibm.com/software/rational/.
2. aceMedia project. Integrating knowledge, semantics and content for user centred intelligent media services. http://www.acemedia.org.
3. G. Antoniou, M. Baldoni, C. Baroglio, R. Baumgartner, F. Bry, T. Eiter, N. Henze, M. Herzog, W. May, V. Patti, R. Schindlauer, H. Tompits, and S. Schaffert. Reasoning methods for personalization on the Semantic Web. *Annals of Mathematics, Computing & Teleinformatics*, 2(1):1–24, 2004.
4. B. Benatallah, M.-S. Hacid, A. Léger, C. Rey, and F. Toumani. On automating web services discovery. *VLDB Journal*, (14(1)):84–96, 2005.
5. A. Billig and K. Sandkuhl. Match-making based on Semantic Nets: The XML-based approach of BaSeWeb. In *Proceedings of the workshop on XML-Technologien für das Semantic Web*, pages 39–51, 2002.
6. G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1997.

7. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. iMAP: Discovering complex semantic matches between database schemas. In *Proceedings of SIGMOD*, 2004.
8. T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello. A system for principled match-making in an electronic marketplace. In *Proceedings of WWW*, pages 321–330, 2003.
9. D. Dou, D. McDermott, and P. Qi. Ontology translation on the Semantic Web. *Journal on Data Semantics*, II:35–57, 2005.
10. J. Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-lite. In *Proceedings of ECAI*, pages 333–337, 2004.
11. F. Giunchiglia and P. Shvaiko. Semantic matching. *The Knowledge Engineering Review Journal*, (18(3)):265–280, 2003.
12. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic schema matching. In *Proceedings of CoopIS*, 2005.
13. V. Haarslev, R. Moller, and M. Wessel. RACER: Semantic middle-ware for industrial projects based on RDF/OWL, a W3C Standard. `http://www.sts.tu-harburg.de/~r.f.moeller/racer/`.
14. Stanford Medical Informatics. Protégé ontology editor and knowldege aquisition system. http://protege.stanford.edu/index.html.
15. I. Jacobson, G. Booch, and J. Rumbaugh, editors. *The unified software development process*. Addisson-Wesley, 1999.
16. M. Lenzerini. Data integration: A theoretical perspective. In *Proceeding of PODS*, pages 233–246, 2002.
17. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proceedings of KR*, pages 483–493, 2000.
18. E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. OBSERVER: An approach for query processing in global information systems based on interoperability between pre-existing ontologies. In *Proceedings of CoopIS*, pages 14–25, 1996.
19. L. Nixon, M. Mochol, A. Léger, F. Paulus, L. Rocuet, M. Bonifacio, R. Cuel, M. Jarrar, P. Verheyden, Y. Kompatsiaris, V. Papastathis, S. Dasiopoulou, and A. Gómez Pérez. D1.1.2 Prototypical Business Use Cases. Technical report, Knowledge Web NoE, 2004.
20. N. Noy and M. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of AAAI*, pages 450–455, 2000.
21. J. Petrini and T. Risch. Processing queries over RDF views of wrapped relational databases. In *Proceedings of the workshop on Wrapper Techniques for Legacy Systems*, 2004.
22. N. Preguica, M. Shapiro, and C. Matheson. Semantics-based reconciliation for collaborative and mobile environments. In *Proccedings of CoopIS*, pages 38–55, 2003.
23. E. Rahm and P. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, (10(4)):334–350, 2001.
24. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV, 2005.
25. P. Shvaiko, F. Giunchiglia, P. Pinheiro da Silva, and D. L. McGuinness. Web explanations for semantic heterogeneity discovery. In *Proceedings of ESWC*, pages 303–317, 2005.
26. P. Shvaiko, A. Léger, F. Paulus, L. Rocuet, L. Nixon, M. Mochol, Y. Kompatsiaris, V. Papastathis, and S. Dasiopoulou. D1.1.3 Knowledge Processing Requirements Analysis. Technical report, Knowledge Web NoE, 2004.
27. P. Traverso and M. Pistore. Automated composition of semantic web services into executable processes. In *Proceedings of ISWC*, pages 380–394, 2004.
28. Y. Velegrakis, R. J. Miller, and J. Mylopoulos. Representing and querying data transformations. In *Proceedings of ICDE*, pages 81–92, 2005.
29. L. Yan, R. Miller, L. Haas, and R. Fagin. Data driven understanding and refinement of schema mappings. *SIGMOD Record*, 30(2):485–496, 2001.
30. J. Zhong, H. Zhu, J. Li, and Y. Yu. Conceptual graph matching for semantic search. In *Proceedings of the ICCS*, pages 92–106, 2002.