An Implementation Model for Managing Data and Service Semantics in Systems Integration

Miklós Kasza, Vilmos Szűcs, Vilmos Bilicki, Gábor Antal, András Bánhalmi
Department of Software Engineering, University of Szeged
Szeged, Hungary
{kaszam,vilo,bilickiv,antalg,banhalmi}@inf.u-szeged.hu

***Abstract*—The ubiquity of data, services and computing devices demands a higher level of understanding of their real nature if one wants to create value-added services based on them. The human brain can understand the concepts behind these artifacts and find appropriate conceptual links easily, however, by applying semantic technologies (ontologies, concept mapping, inferencing, etc.) computer programs can also be taught to behave similarly. Semantic solutions based on these technologies can lead to powerful value-added services for various domains. A generic domain that can be addressed successfully with the help of these technologies is systems integration. In this paper we introduce a generic implementation model that was developed to serve as a basis of integration solutions in various real-life projects.**

***Index Terms*—semantic knowledge representation; ontologies; ontology mapping; ontology merging; systems integration; device management**

I. Introduction

Data and computing is everywhere nowadays. Specialized data providers are collecting and producing data in different domains of each and every aspect of our lives. The bare availability of the enormous amount of data does not make it usable in itself. For providing valuable services to the masses, somebody has to convert it to consumable information (or more importantly knowledge) that can be understood and acted upon by people easily. The ubiquity of computing makes this picture even more complex. A myriad of devices exist on the market possessing various capabilities for creating and accessing data. Data and device providers and service integrators have to work hand in hand to provide the appropriate value-added services, since people demand more and more sophisticated services and want them to be integrated with each other as seamlessly as possible. These facts lead to a scene filled with heterogeneous information sources, channels, consumers and computing devices.

Heterogeneity and diversity lead to a wide-scale interpretation of concepts. Considering two different computer programs dealing with the same domain-specific problems, the representation of common concepts can vary heavily. For instance, an author of a business-related document can be stored explicitly in the document's meta-data section, while in the case of an electronic mail the author can be the sender of the e-mail. Basically, the relation of the author and the sender concepts in this case are not straightforward for computer programs, but can be interpreted easily by humans and thus computers can be "taught" to act similarly. However, the manual interpretation of each domain concept and finding the relationships between each other can use up huge amount of human power and thus is not cost-effective.

Recently the growing availability of computing power created the possibility of advanced information processing. One of the key aspects is the identification and representation of the information's semantic content. The computer programs have to provide the means for capturing the meaning of various pieces of information. Knowledge representation is as old as computer science itself. Multiple approaches exist for representing human knowledge in the form of machine-processible artifacts. Basic information (e.g., the birth date of a person, topic of a university course) can be represented easily; however, the description of the meaning of the information snippets stored in a computer system is a hard problem. Still, it is an important problem, since the attachment of semantic information to the stored knowledge leads to new ways of information management.

As an ideal vision, one can imagine a world of autonomous, co-operating services that have the knowledge of a common concept set and the meaning of various concepts included in them, as well. Studying the field of semantic knowledge representation and processing brings us closer to this idealistic state; where the mapping of various concepts takes place automatically; the information is filtered based on the interest of the target audience and can connect pieces of information based on the meaning.

This paper provides insights on some key problems in the field of semantic information processing and a possible implementation model that can be used while solving the problems. The insights and the model are based on real-life project-based experience. The paper is structured as follows. Section II presents some research projects related to complex integration problems and introduces several R&D projects the authors base their experience on and highlights the key problems that were identified and partially solved during the execution of these projects. Section III describes the problem of semantic information representation and a model for it, which results from the projects that are strongly connected to this field. Section IV explains some practically feasible and effective ways to collect source data for semantic applications. Section V provides a brief description of ontology matching, merging and mapping and their practical applications. Section VI discusses various aspects of the automatization possibilities

 ISBN: 978-1-61208-247-9

of semantic information processing and real-life implementations dealing with automatization. Section VII concludes the paper and identifies some interesting problem areas that are targeted to be further studied and improved in the future.

II. Related Work

There are several research projects which use ontologies and semantics to solve complex integration and communication problems. In the TMTFactory [1] tourism research project [2] the main goal was the integration of several semantic services (e.g., museum, cinema, restaurant searching services) using the ALIVE [3] architecture to create a service collection which can be used by tourists. The project included a Streetbox application which can be used on interactive street displays to find points of interest in a specific area. The improvement of service discovery, system stability and maintainability gave the motivation to use semantic technologies in the application. The Tripcom [4] project aimed the change of Web Service architectures using machine-to-machine communications and ontologies. The TripleSpace technology of the project gives the users a global space of web services to use with the ORDI (Ontology Representation and Data Integration) middleware which can help the modification of the system knowlegde base. The project has an e-Health use-case with the integration of several health services (e.g., data of patients, doctors, hospitals, specialists) to make them easier to use.

The Department of Software Engineering, University of Szeged, took part in several semantic information management-related projects during the course of the recent years. These projects include EU-funded R&D projects, projects funded by the Hungarian Government and projects executed jointly with industrial partners. The common aspect of these projects is the use of semantic technologies in various research and development areas. Out of these projects came several valuable findings that can be used to enhance the practical application of semantic technologies in real-life scenarios. This section covers the scope and overview of these projects.

A. The CONVERGE Project

The electronics industry in Europe faces strong competition not only with companies located in the United States but recently companies in far eastern countries endanger the competitiveness of their European counterparts as well. This challenge can be efficiently targeted by the European industry only if taking the altered circumstances in account and collaborating effectively with each other. Low geological distances and the availability of high-level industrial technology can help them to do so. In order to enhance the efficiency of European-level collaborations, so called non-hierarchical supply chains are being formed. These supply chains are sticked together by decisions made on novel levels; therefore novel approaches are required to address the information sharing issues of the companies. The distinction between sharable and non-sharable information is of key importance in this field. Because of the novelty of the new approaches, the appropriate methodology and tooling is still missing [5], [6]. The CONVERGE project aims at eliminating this imperfection by providing the appropriate methodology and toolkit for supporting decision making on strategic and tactical levels in non-hierarchical supply chains.

In addition to the scientific and technological experts, four industrial partners took part in the consortium executing the project. This fact significantly enhances the acceptance of the project results in the industry, since the industrial partners collaborating in the project provide the field experience that can be exploited to enhance the viability of the emergent methodology and tooling. The solution developed in the project is based on a non-centralized decision support system that enhances the process of production planning and resource optimization by utilizing a novel reference model directly targeted at inter-organizational decision making and existing inter-organizational relationships.

B. Telenor Smart Environment System – Device Integration

The growing availability of ubiquitous computing capabilities can enhance life quality. New smart sensor devices are constantly appearing in various M2M markets. The functions provided by such modern sensor devices enable system developers to create systems that were unimaginable earlier. More and more complex monitoring devices provide functions that can ease the life of humans. However, this rich set of smart devices pose challenges to system developers and specialists as well. To improve the quality and cost effectiveness of smart home systems and services, the necessary devices and sensors must be selected carefully to keep the system available for a low price. However, this can lead to dealing with a diverse set of hardware manufacturers and communication protocols. In addition, the various structures of data coming from the involved devices must be supported by the system. These issues all affect the design and development of the data model and device integration process. Furthermore, the final result of the development process has an impact on the flexibility, the reusability, and the performance of the developed system.

Smart sensor integration processes pose a difficult and complex task for developers. The process of device integration starts at the studying of the protocol used by the given device. The protocol is usually given by the structure of messages (based on a given communication protocol) constructed by the device to send observation or measurement data to a specific server. The structure of the messages is usually defined in a protocol specification document in the development documentation of the device. The messages typically consist of key-value pairs which can be defined by parameter names (keys) as well as data types and possible constraints refer to parameter values. These value sequences provide the exact data that should be forwarded. Besides the parameters some additional information is needed about the place of the data in the message, the type of communication (simple message, acknowledged message, complex communication process) and security. In the project we inspect the device integration processes more deeply and we describe our novel methodology to resolve current issues in the field of smart sensor integration based on semantic protocol and data representations and mappings.

C. R&R Application Service Network

The goal of the Application Service Network project is to create an application service integration platform, in which the integration is not done by IT experts. The end users, who access the services, are to be made capable of assembling integrated services from Internet-enabled service systems that are compatible with the platform. The end users can access the applications (which are assembled directly to meet their demands) as services residing “in the cloud”.

The network under construction can unify the telecommunications networks and the IT-services. This way, novel business applications can emerge based on vertical service integration. The target result of the project is a model that is:

- available as a service in the cloud;
- enables the assemblement and usage of custom applications;
- enables the expansion of application components and
- takes the demands of the players in the service network into consideration.

Besides the model, in the project, a model implementation prototype is provided. Based on this implementation prototype the the model is validated using user scenarios. For these purposes, two main scenarios were selected: the *Medical Attendance* scenario and the *Semantic Map* scenario. A brief description of these scenarios follows.

1) Medical Attendance: A telemedicine system provides functions for the health care industry that make medical attendance faster and more reliable using an appropriate telecommunications environment. These services usually solve emergency or non-emergency problems occurring due to large distances. This way, patients can get medical assistance even while being at their homes with the help of various visualisation or data collection devices. On the other hand, telemedicine services ease the communications between medical staff, and thus urgent consultations and the sharing of medical records become possible.

In emergency cases, when patients are not capable of communicating directly, telemedicine systems can save lives. A use case of such a telemedicine system is providing the emergency units with optimal routes to the nearest appropriate hostpital. An algorithm for this problem requires the availability of parameters that lead to more relevant results. Such relevant parameters are the amount of free space in the hospital, the facilities of the hospital, the patient’s health-related history records, etc. However, acquiring such vital data is not straightforward.

A system of this type can be relied upon by almost the entire health industry. These systems have to support integration with external systems on a high level. Applying semantic information representation seems an ideal solution for these problems.

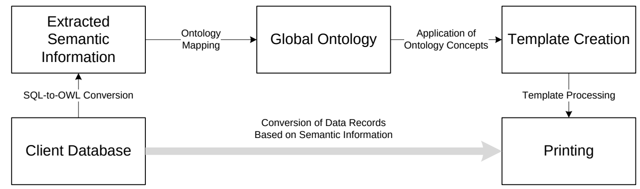

Fig. 1. High level overview of the POS Printing process.

Maintaining semantic context in telemedicine systems can lead to better and safer services.

One part of the project targets the aforementioned scenario in a fully automatized manner. It investigates the possibilities of transferring telemedicine and geo-information services to the world of semantic information management and thus the possibilities for creating a web application that is optimized and automatized more than the existing ones. Therefore, the project’s goal was to build the aforementioned solution from ground up to the highest abstract levels using the ALIVE framework [3].

2) Semantic Map: Another representative scenario for the Application Service Network is the *Semantic Map*. In this scenario, the data set originates from a geo-information database containing places, points of interest and paths. The project investigates the valid scope of integration between geo-information data and semantic web technologies.

D. POS Printing

In large supermarkets, the management of the products and the corresponding product-related marketing material can be a cumbersome task. Usually, the preparation, printing and distribution of the marketing material is done by an appropriate service provider other than the supermarket company. The client base of these printing providers are not limited to one client only and thus they have to deal with product- and service-related data originating from various data sources. The POS (Point Of Sale) Printing project targets a printing provider by applying semantic solutions in order to reduce human work. The main aspect of the project is the semantic annotation of existing data stored in the clients’ databases and mapping it to a common (global) ontology on the provider’s side. This way, the management (design, printing, delivery, etc.) of the marketing material can be based on a common set of product and service store-related concepts, while the integration of different clients’ different database schemas can be done in a semantic way. The process of integrating one client’s information base is depicted on Figure 1.

The goal of the project is to provide methodology and the appropriate tooling for executing the process in an automatized manner (or as much automatized as is possible).

III. Semantic Information Modelling

In recent years, the term of ontology enjoys a growing popularity in the IT world as it promises an appropriate basis to provide the IT tools for human thinking and decision

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-247-9

making ability. The ontology-based interoperability across heterogeneous systems can be achieved to build business logic and complex data flow between processes without the need of human intervention. Ontologies are formal representations of knowledge as sets of concepts within domains, and describe the relationship between them by providing semantic meaning for syntactic terms. The semantic description of data and services allows the automatic understanding and perception of them to achieve collaboration and orchestration between services.

Ontologies can serve as tools for sharing and reusing the existing knowledge in the form of semantically rich structures. Obviously, current computing capacity of modern computers is not enough to store and process information about the world in its entirety. This problem is addressed by partitioning the knowledge into more specific domains. This way, usually ontologies can store knowledge about objects only in several well-defined domains. Despite the differences between various domains, most ontologies provide vocabularies (containing terms that are meaningful in the target domain) and definitions [10].

First of all, the semantic information has to be modeled in some way to be processable by computers. Various models exist for these purpose (RDF, OWL, etc.) In our projects we experimented with some of these models and captured the pros and cons of them (primarily from practical aspects). In the first wave of the projects, the integrated models were semantically annotated by hand (labelling, Java annotations, etc.). This lead to run-time evaluation of the annotated models and the semantic descriptions were generated during run-time. This solution proved to be quite unstable, since it did not make the fine definition of semantic content possible. This way, in the second wave of the projects, we used ontology modelling tools (TopBraid Composer Free Edition [8], Protégé [9]) to define semantics. With the help of these tools the semantic information emerged in the form of OWL-documents. These documents can be stored in generic repositories (file systems, relational databases) or ontology-specific repositories. This solution leads to development-time ontology definitions, however, the ontology mappings are done during run-time.

In the CONVERGE project a given data source with specific metadata was matched to some conventional ontologies, e.g., FOAF [11] and Dublin-Core [12]. In the POS Printing project local ontologies were generated using SQL schemas and matched to a manually annotated global ontology.

IV. Data Collection

By having a model appropriate for our purposes, a well-defined collection methodology had to be developed to populate the model with data. As we found, the collection methodology is a very important area of semantic information management, since the whole semantic ecosystem is viable only if the semantic information can be extracted from existing information sources. Otherwise, the population of the data model itself would take lots of efforts.

First of all, in each project, we had to find the available information repositories that could be used as data sources. We found that various public thematic repositories (accessible via Internet) can serve as bases for several domain-specific aspects. Additionally, other non-public information sources (such as local databases, mail boxes, file systems, etc.) can be used to refine the set of available information.

In the introduced projects we used adapter-based solutions in all cases. As an example, the high-level architecture of CONVERGE's data mediator subsystem is shown on Figure 2. As it can be seen, the architecture is based upon an extensible modular structure that can be extended by introducing new system adapter modules. In the project, adapters have been created for IMAP-based mailboxes, network shares and for CAS Software AG's CASOpen platform. Similar approaches were followed in the case of other projects, as well.

In other projects the data adapter components are called gateways. These gateways mediate the data between the adapted and the target systems by:

- receiving data in the format of the integrated external systems;
- adapting the data to the schema of the target system based on semantic mappings;
- transferring the adapted data to the target system;

In most of the projects HTTP-based gateways were used, however, in CONVERGE, adapters for IMAP-based mail boxes and generic file shares were also developed. In most of the projects the gateway modules are automatically generated from the available semantic information.

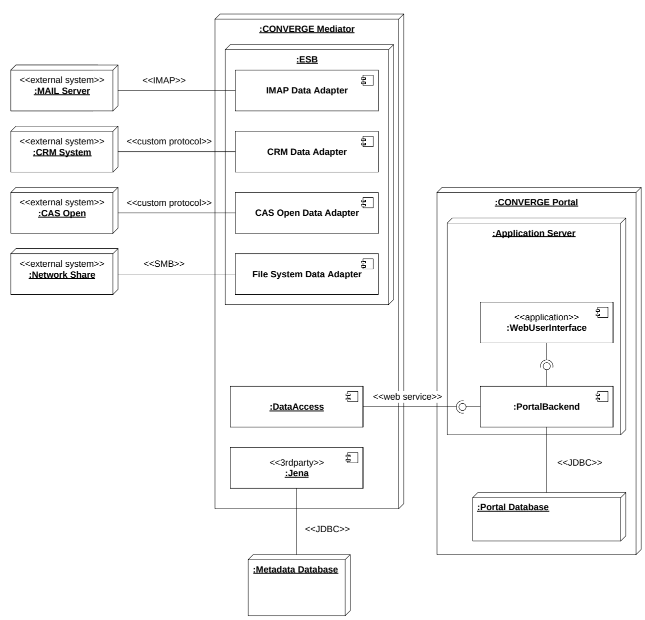

Fig. 2. High-level overview of the CONVERGE's data mediator subsystem.

 ISBN: 978-1-61208-247-9

V. MATCHING, MAPPING AND MERGING

The application of ontologies in knowledge representation is quite straightforward, as long as we do not try to manage multiple ontologies and we do not try to integrate our ontology into a heterogeneous system. In the latter cases we can face incompatibility and heterogeneity issues. This is a common problem, since usually multiple ontologies can be utilized in distributed systems. The most problematic cases appear when domains of the different ontologies are overlapping, i.e., they represent similar types of knowledge but the syntaxes of them are different partially or in their entirety [6]. However, ontology mapping (or matching) solutions can be provided in order to resolve these issues. These tools can find the rules on which concepts in one ontology can be mapped to concepts in other ontologies [13], [14], [15], [16], [17], [18], [19].

Generally, ontology mappings can be classified as follows:

- mapping between a global and multiple local ontologies (*global-local* mappings)
- mapping between multiple local ontologies (*local-local* mappings)
- ontology *merging* and *alignment*

Global-local mappings can be used as the means of ontology integration, i.e., they can describe the rules of mapping various local ontologies to an integrated global ontology [20], [21]. Local-local mappings map the local contents of each ontology on the basis of semantic relationships without the existence of a global ontology. Ontology merging techniques enable the creation of a single, coherent ontology based on multiple existing ontologies dealing with the same domain. The new, merged ontology contains information about each source ontology in a more-or-less unchanged form. Ontology alignment's main purpose is to find a link between two separately stored ontologies when they become inconsistent [22], [23].

After some investigation it was determined that for the purposes of the CONVERGE project (i.e., to create a knowledge model that is capable of storing heterogeneous information available all around in an enterprise), ontology merging was a viable solution to use. However, before being able to merge the available knowledge, first we had to

1) gather the data from external systems and transform it to an ontology-based presentation format
2) find mappings between various concepts used in external systems in order to be able to integrate different concepts originating from different systems but having the same semantic meaning.

During our experimentation we evaluated and compared five different ontology matching tools based on some functional and subjective non-functional metrics: WSMT Mapping [24], COMA++ [25], PROMPT [26], MAFRA toolkit [27] and PyOntoMap [28]. On the functional side we created some similar ontologies and found the optimal mappings by hand. These sets of optimal mappings served as the baseline that was used to evaluate the goodness of the tools' results. Because of the hand-made mappings we used relatively small ontologies, they contained 8.25 concepts and 21.5 attributes on average.

TABLE I
EFFICIENCY OF AUTOMATIC MATCHING TOOLS IN A SAMPLE ONTOLOGY-MAPPING SCENARIO.

Toolkit	S	d_s	L	d_l	C	d_c
WSMT	12	$\frac{12}{20}$	13	$\frac{13}{23}$	7	$\frac{7}{20}$
COMA++	15	$\frac{15}{20}$	17	$\frac{17}{23}$	9	$\frac{9}{20}$
PROMPT	12	$\frac{12}{20}$	5	$\frac{5}{23}$	1	$\frac{1}{20}$
MAFRA	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
PyOntoMap	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*

We used 3 metrics for evaluating the goodness of each tool:

- d_s: the ratio of mappings found by the matching tool when the two ontologies contain only structural differences
- d_l: the ratio of mappings found by the matching tool when the two ontologies contain only lexical differences
- d_c: the ratio of mappings found by the matching tool when the two ontologies contain both lexical and structural differences

In each case, the number of optimal mappings were 20, 23 and 20 for structural, lexical and complex problems, respectively. The results found by the evaluations are summarized in Table I. As it can be seen in the table, the MAFRA Toolkit and PyOntoMap frameworks did not work on the sample ontology set.

With regards to the non-functional metrics, COMA++ proved to be the best choice due to its speed, integrability and automation possibilities. The WSMT Mapping tool seemed to be an accurate tool, however, its functionality is automatizable only partially, because it is built upon the availability of user activity. PROMPT proved to be relatively imprecise, it found only a small part of mappings, and it also lacks the proper automatization functions. The MAFRA Toolkit currently does not include a usable semi-automatic ontology mapping, this way it can not be automatized. PyOntoMap is easy to use, however it is also imprecise and can map only concepts, not attributes. Based on our evaluation, the toolkits under investigation provide API-s for semi-automatic or automatic mappings, however, they are very poorly documented.

After the initial evaluation of the tools, we decided to apply COMA++ to the vocabularies used in our knowledge representation model. Despite the good results in the artificial tests, even COMA++ achieved poor results. It found mappings between totally unrelated concepts and missed almost all the reasonable mappings. In Figure 3 a sample mapping can be seen between the Dublin Core and FOAF vocabularies. It can be seen that it found mappings only on the basis of lexical similarities, however, the lexical mappings were even false (found mapping between *note* and *name* or *Type* and *theme* concepts). Correct mappings were missing in the case of mapping by structural similarities, as well. After some deeper investigation, it turned out that the unsuccessful application of the tool to the real vocabularies can be deducted to some

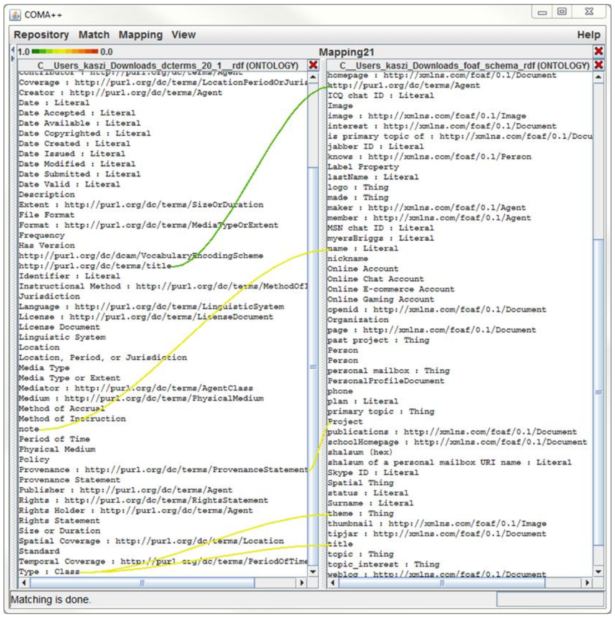

Fig. 3. Sample COMA++ mapping.

important issues:

- the lack of a proper thesaurus worsens the mapping accuracy: as it can be seen, COMA++ can not map differently named concepts having the same meaning automatically. A domain-specific thesaurus could help this issue.
- the "flatness" of the applied ontologies interferes with structural mapping approaches: the mapping could be enhanced by taking structural similarities into account. However, our target ontologies do not have a complex structure, only several subclassings are included, and this fact together with the lexical differences leads to poor mapping results.

Obviously, the matching capabilities of the currently available solutions can be enhanced. Some possible solutions for enhancing ontology matching:

- Refining literal comparisons: resolving abbreviations; learning notational conventions; identifying frequent prefixes and suffixes; word swappings and mixed language notations);
- Enhancing structural comparisons [29], [30]: identifying relations between concept hierarchies and class properties; dealing with transitive and inversible relationships;
- Aggregating similarity metrics: introducing adaptive aggregation, threshold refinement [31] and weighing [32]; introducing methods using new types of discriminative machine learning algorithms or decision trees [32]; shifting to fuzzy aggregation [33], [34];
- Enabling human interventions for refinement.

These solutions would solve problems using methods from other fields (e.g., artificial intelligence, natural language processing, machine learning) and using the given approaches together, taking practical points of view into account. Practical applicability of each enhancement is currently under further investigation and is subject of future work.

VI. Automatization of Processing

When thinking about automatization of information processing tasks, the first question is about the subject material for automatic processing. Since different data repositories and manageable systems usually expose different interfaces for data access and represent data in different formats, the first candidates for automatic processing are data schemas and data access methods. While the data access methods can usually be defined by the appropriate standardized or proprietary protocols and data formats, the interpretation of the data schemas is usually a harder task. The structural adoption of an external data set is therefore based on well-defined rules that are specific to the system to be integrated. The target representations can be formats specially designed to hold semantically enriched information, such as RDF or OWL graphs.

The integration on this level is a well-studied problem and as such, there exist well-elaborated solutions for it.

It must be noted that these adapters can deal only with the data access related differences of the different systems, but do not consider the schema-related differences and specificities. As it was discussed, the automatization of the schema mappings can be donesuccessfully only if the appropriate semantic annotations exist for the adapted schema. In these cases, human interaction is involved only on the data schema/service side. Once they are annotated correctly, human interaction is not required. Unfortunately, the lack of properly annotated data sets and services enforces human interactions in other stages as well.

This way, we identified three models for human interaction involvement:

- No additional human interaction required (Medical Attendance)
- Human interaction is required during development phase (Semantic Map, TSES-DI, CONVERGE, Factory)
- Human interaction is required during run-time

The first approach is the idealistic one, it has been covered earlier. The second approach requires intervention to the processes during development time. In this case, the domain-specific knowledge is inserted into the system during the development of adapter components. For these purposes, the appropriate development tools have to be provided for the developers and domain experts to be able to express their knowledge easily. In the projects we developed tools in the form of Eclipse-plugins that make these tasks easier. Figure 4 shows the wire-frame design of the tool for defining concept matchings.

In most of the projects we used development-time human interactions, since this solution proved to be the most viable at the time. However, the tools used during development time can be elevated to a level, on which end users are made capable of defining semantic properties of the domain. This makes the third approach feasible.

The third option for involving human interaction exposes tools for domain experts during run-time. This way the experts can insert their knowledge into the system without having the need for rebuilding the involved components.

It has to be noted, that the two later cases do not require a priori annotation of the target data sets and services, however the annotation has to be done before the data sets and services can be used by the system. Compared to the first solution, this annotation process takes place later in the process.

Since a primary goal of automatization is enhancing productivity, we made productivity measurements in the Semantic Map scenario of the R&R Application Service Network project. In these measurements the productivity of classic development was compared with semantics-based development (in which automatic code generation can take place with the availability of semantic information). The steps of the compared processes are depicted in Figure 5. As it can be seen, the development was broken up to three individual stages based on the nature of the work required on the stage: design, modelling and implementation. The left side of the picture shows the steps a developer had to take using the semantics-based development, while the right side describes the process of classical development. In both cases a new data source (with the appropriate components) had to be developed.

By looking at the number of steps required using each approaches, one can see that the classic development method requires more human tasks to be done. After the measurements (results shown in Figure 6) we found that the productivity in the design phase does not differ significantly, since in both methods, the developers have to understand the tasks and the domain. The productivity of this phase can be enhanced by assigning tasks to developers familiar with the target domain. The first significant difference shows up in the modelling phase. In this phase, the ontology-based development approach lets the developers concentrate on the important domain-related concepts and their relationships and thus frees the developers from doing manual design in the cumbersome areas. Finally, the results from the implementation phase show

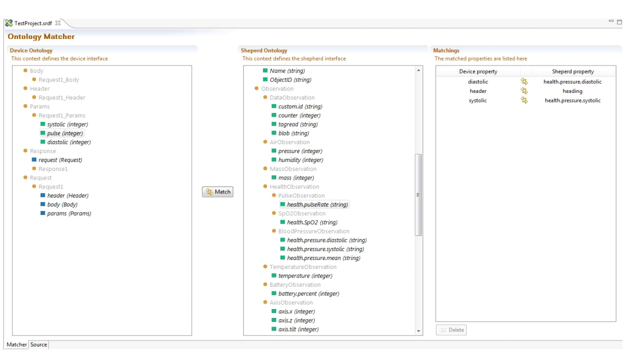

Fig. 4. User interface of an ontology matching development tool prototype

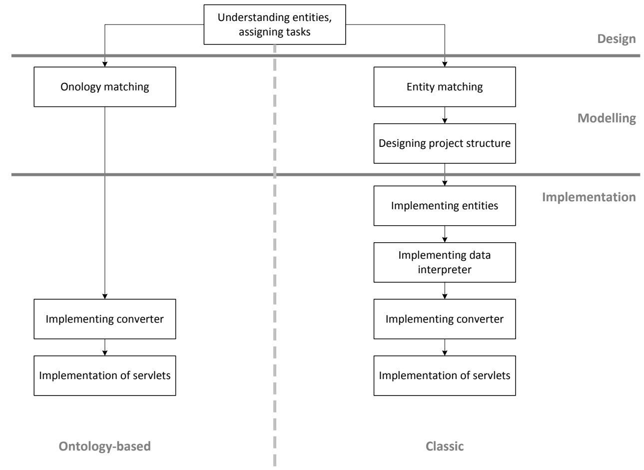

Fig. 5. Processes followed for measuring productivity-related differences between classic and ontology-based development approaches

the real benefit of using the ontology-based approach over the classic one. In this phase the productivity was much higher, since the developers did not have to deal with the details of the program code. Therefore there are much less mistakes caused by the developers and heterogeneities caused by misunderstandings. The generated code is much easier to improve, correct and maintain.

VII. Conclusion and Future Work

During the execution of the projects dealing with semantic information management, we revealed that the information required for successful systems integration can leverage semantic additions. However, we identified some problem areas that can be enhanced in order to make semantic processing more automatic and requiring less human interactions.

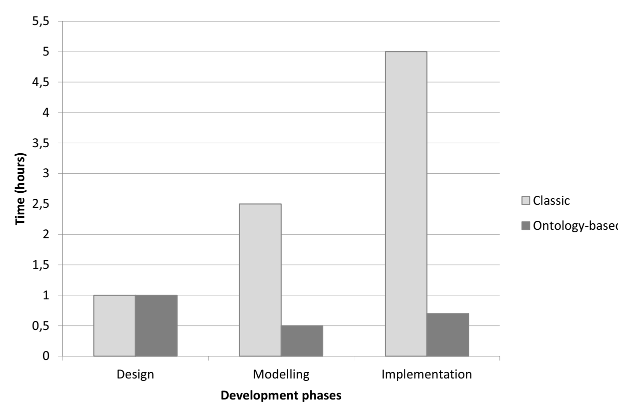

Fig. 6. Results of productivity measurements

A. Availability of a Priori Semantic Annotations

Real-time processing lacks the possibility to find properly annotated services and data sets available. By introducing end-user programming via domain specific languages, this constraint can be released, but in this case the appropriate tooling and generic processing engine must be available.

B. Better Matching Algorithms

The automatic matching of domain concepts and their true relationships requires more flexible matching algorithms. Matching based on simple lexical and structural properties of the concepts can lead to incomplete matching that require human intervention to make it usable. Applying advanced techniques can lower the requirement of human intervention.

C. Productivity

The application of semantics-based modelling and automatic code generation in systems integration makes the productivity of the development tasks more effective. This productivity boost can be introduced in run-time semantics definitions with the help of end-user programming and domain-specific languages.

ACKNOWLEDGMENT

Parts of this work are done in the *CONVERGE - Collaborative Communication Driven Decision Management in Non-Hierarchical Supply Chains of the Electronics Industry* project which is funded by the European Union under grant number 228746 in FP7-NMP.

Parts of this work are funded by the Government of Hungary under grants GOP-2009-1.1.1 and KMOP-2009-1.1.1 and by R&R Software Ltd.

Parts of this work are funded by the Government of Hungary under grants GOP-2011-1.1.1 and by Factory Creative Studio Ltd.

Parts of this work are funded by Telenor Hungary.

REFERENCES

[1] TMT Research & Innovation. Available at: http://research.tmtfactory.com/
[2] K. Alonso, M. Zorrilla, R. Confalonieri, J. Vzquez-Salceda, H. Inan, M. Palau, J. Calle and E. Castro, *Ontology-Based Tourism for All Recommender and Information Retrieval System for Interactive Community Displays*, Information Science and Digital Content Technology (ICIDT), 2012 8th International Conference, 26-28 June 2012.
[3] IST-Alive Project, Available at: http://www.ist-alive.eu/.
[4] The official website of the TripCom project. Available at: http://www.tripcom.org/
[5] B. Scholz-Reiter, J. Heger, C. Meinecke, D. Rippel, M. Zolghadri, R. Rasoulifar, *Supporting Non-Hierarchical Supply Chain Networks in the Electronics Industry*
[6] C. Meinecke and D. Rippel (eds.), *Decision-Making Model, Data Mapping and Integration Roadmap*, Project Deliverable document, CONVERGE, retrieved from http://www.converge-project.eu/images/stories/pd/D2.2-Decision Making Model & Data Mapping.pdf on 2 October, 2011.
[7] K. Kalaboukas (ed.), *System Requirements, Data Sharing concept and System Architecture*, Project Deliverable document, CONVERGE, retrieved from http://www.converge-project.eu/images/stories/pd/D3.1-Specification.pdf on 2 October, 2011.
[8] TopQuadrant TopBraid Composer Free Edition. Available at: http://www.topquadrant.com/products/TB_Composer.html.
[9] The Protg Ontology Editor and Knowledge Acquisition System. Available at: http://protege.stanford.edu/.
[10] J. D. Heflin, *Towards the Semantic Web: Knowledge Representation in a Dyncamic, Distributed Environment*, PhD thesis, Faculty of the Graduate School of the University of Maryland, College Park. 2001. Retrieved from http://www.cs.umd.edu/fs/www/projects/plus/SHOE/pubs/heflin-thesis-orig.pdf in 2 October, 2011.
[11] FOAF Vocabulary Specification 0.98. Available at: http://xmlns.com/foaf/spec/.
[12] DCMI Home: Dublin Core Metadata Initiative (DCMI). Available at: http://dublincore.org/.
[13] E. Rahm and P. A. Bernstein, A survey of approaches to automatic schema matching, In *The VLDB Journal, vol. 10 (2001)*, pp. 334–350.
[14] P. Shvaiko and J. Euzenat, A Survey of Schema-based Matching Approaches, In *Journal on Data Semantics IV (2005)*, pp. 146–171.
[15] A. Gal and P. Shvaiko, Advances in Ontology Matching, In *Advances in Web Semantics I (2009)*, pp. 176–198.
[16] S. M. Falconer, N. F. Noy, and M-A. Storey, Ontology Mapping — A User Survey, In *Proceedings of the Workshop on Ontology Matching (OM 2007)* at ISWC/ASWC 2007, pp. 113–125, Busan, South Korea (2007)
[17] X. Su, *Semantic Enrichment for Ontology Mapping*, PhD thesis, Norwegian University of Science and Technology, October 2004.
[18] P. Shvaiko and J. Euzenat, Ten challenges for ontology matching, In *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part II on On the Move to Meaningful Internet Systems (2008)*, pp. 1164–1182.
[19] M. Sabou, M. d'Aquin, and E. Motta, Using the Semantic Web as Background Knowledge for Ontology Mapping, In *Proceedings of the International Workshop on Ontology Matching (OM-2006)*, pp. 1–12.
[20] H. S. Pinto and J. P. Martins, A methodology for ontology integration. In *K-CAP '01: Proceedings of the 1st international conference on Knowledge capture (2001)*, pp. 131-138
[21] C. M. Keet, *Aspects of Ontology Integration*, Technical report, School of Computing, Napier University, January 2004.
[22] J. de Bruijn, M. Ehrig, C. Feier, F. Martín-Recuerda, F. Scharffe, and M. Weiten, Ontology mediation, merging and aligning, In *Semantic Web Technologies (July 2006)*
[23] T. C. Hughes and B. C. Ashpole, The Semantics of Ontology Alignment, In *I3CON. Information Interpretation and Integration Conference (2004)*.
[24] M. Kerrigan, A. Mocan, M. Tanler, and W. Bliem, *Creating Semantic Web Services with the Web Service Modeling Toolkit (WSMT)*. Available online at http://www.sti-innsbruck.at/fileadmin/documents/papers/creating-semantic-web-services-wsmt.pdf. Accessed on 2 October 2011. The Web Service Modeling Toolkit (WSMT), http://www.sourceforge.net/projects/wsmt
[25] *Schema and Ontology Matching with COMA++*, Retrieved from http://dbs.uni-leipzig.de/Research/coma.html on 2 October 2011.
[26] N. Noy, Prompt, In the *Protégé Community of Practice Wiki*. Retrieved from http://protege.cim3.net/cgi-bin/wiki.pl?Prompt on 2 October, 2011.
[27] N. Silva and J. Rocha, *Semantic Web Complex Ontology Mapping*, Retrieved from http://sourceforge.net/projects/mafra-toolkit/files/mafra-toolkit/0.2/SemanticWebComplexOntologyMapping.pdf/download on 2 October, 2011.
[28] P. Besana, *Using Demster-Shafer for Combining Ontology and Schema Matchers*, Retrieved from http://pyontomap.sourceforge.net/UsingDSforOntoMap.pdf on 2 October, 2011.
[29] J. Euzenat and P. Shvaiko, *Ontology Matching, 1st ed.* Springer Publishing Company, Inc., 2010.
[30] A. K. Alasoud, *A Multi-Matching Technique for Combining Similarity Measures in Ontology Integration*, Phd Thesis, Concordia University Montral, Qubec, Canada, 2009.
[31] T. Kohonen, Learning Vector Quantization, in *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, MA, 1995, o. 537-540.
[32] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 2nd ed.* Prentice-Hall, Englewood Cliffs, NJ, 2003.
[33] J. Dombi, Towards a General Class of Operators for Fuzzy Systems, *IEEE T. Fuzzy Systems, vol. 16,* pp. 477–484, 2008.
[34] M. Detyniecki, *Fundamentals on Aggregation Operators*, University of California, Berkeley, 2001.