Information integration in enterprises is a nightmare due to the differences in software and hardware platforms and due to syntactic and semantic differences in the schemas of the data sources. This is a well-known problem in the area of Enterprise Application Integration (EAI), where many applications have been developed for the purpose of information integration. Most current tools, however, only address the problems of (soft- and hardware) platform and syntactic heterogeneity; they fail to address semantic differences and they only support one-to-one (syntactical) mappings between individual schemas. We will analyze this problem in automobile industry through a specific UML derivate language and a domain ontology to find an adequate solution.

1. INTRODUCTION

The Automotive industry is changing drastically. Many companies are re-organizing, reengineering, downsizing, and above all changing their approach to engineering vehicles. The reasons for these changes are numerous; increased competition, new market requirements, greater customer focus, vastly improved information technology, outdated business practices, etc. To be successful in the coming century the company must challenge and beat the intensifying competition; understand, exceed and drive the new markets; surpass heightened customer expectations; install and fully utilize the most advanced information technology; replace the old business practices with new aggressive global strategies; and, finally, do all this profitably. Automotive companies are focusing on the processes of delivering a new or updated high quality vehicle to the market faster and cheaper. One of the big problems in an OEM (Original Equipment Manufacturing) is the exchange of information among the different applications and with suppliers in a safe way. Most of these applications are domain old proprietary systems and is common to print information from one system, fax it and then a manual input is performed on the other system application. Sometimes there is a slight improvement on this process because a script conversion from one database to the other is outsourced. With this last solution it is spent a big amount of money in consulting services for developing solutions that solve individual business problems. It is common to find hundreds of distinct applications running within an OEM. All this at end result is often costly time delays and the distribution of inaccurate or out-of-date information. Even worst, one often hears that companies carry out procedures in an archaic fashion because “that’s how our computer systems work.”

To assure this goal, domain ontology (Vehicle Corporate Ontology –VCO) was constructed through a new UML2 profile defined specifically for vehicle development and assembly business (VDML) [1]. Ontology creation will benefit of UML classes and visual modeling will decrease syntactic and semantic errors and increases readability. UML itself does not satisfy the needs of the representation of ontology concepts that are imported from descriptive logic and that are include in Semantic Web Ontology Language (e.g. OWL(Web Ontology Language), RDF, RDF Schema, …). The OMG’s MDA (Model Driven Architecture) concept has the ability to create, using appropriate family of languages, a metamodel defined by MOF (Meta-Object Facility) and based on OWL.

Since automobile industry application uses different terminology systems, database information exchange is achieved whose semantics are specifying using logic-based ontologies. A broader terminology is specified by VCO, is used as reference terminologies. Relations among terminology are indirect, once each terminology can be mapped into other using VCO.

2. Semantic Web and OWL

Semantic web architecture is a functional, non-fixed architecture [2]. Barnes-Lee defined three distinct levels that incrementally introduced expressive primitives: metadata layer, schema layer and logical layer [3]. XML and XML schema define syntax, but mean nothing about the data that it describes. That means that some standards must be built on top of XML that will describe semantics of data. This conduct to RDF and a general model in metadata layer RDF schema, which provides mechanisms for defining the relationships between properties declared and others resources. To enable services for the semantic web we have on top the logical layer. This layer introduces ontology languages, that are based on meta-modeling architecture defined in lower layer. This
enables the use of tools with generic reasoning support, independent of the specific problem domain. Examples of these languages are OIL, DAML, OWL and VCO. OWL is a semantic markup language for publishing and sharing ontologies in the web. OWL is developed as a vocabulary extension of RDF and is derived from DAML and OIL [4].

Figure 1: OWL and VCO in the semantic Web Architecture.

3. UML, MOF, MDA AND UML Profiles  
UML has emerged as the software industry’s dominant language and is already an Object Management Group (OMG) standard. It represents a collection of the best engineering practices that have been proved successful in the modeling of large and complex systems. OMG is proposing the UML specification for international standardization for information technology [5]. Wide recognition and acceptance, which typically enlarge the market for products based on it, will be the major benefit. Therefore specific subjects (e.g. vehicle design process) require making UML models more specific and thus more precise. This in turn can be done by using stereotypes (since they are an extension mechanism inherent in second version of UML) as means of adding necessary information to existing model elements. Stereotypes have been given a special attention together with the idea of the Model Driven Architecture (MDA).

UML profiles are UML packages of the stereotype «profile». A profile can extend a metamodel or another profile [6] while preserving the syntax and semantic of existing UML elements. It adds elements which extend existing metaclasses. UML profiles consist of stereotypes, constraints and tagged values. A stereotype is a model element defined by its name and by the base class(es) to which it is assigned. Base classes are usually metaclasses from the UML metamodel, for instance the metaclass «Class», but can also be stereotypes from another profile. A stereotype can have its own notation, e.g. a special icon. Constraints are applied to stereotypes in order to indicate restrictions. They specify pre- or post conditions, invariants, etc., and must comply with the restrictions of the base class [6]. Constraints can be expressed in any language, such as programming languages or natural language. We use the Object Constraint Language (OCL) [7] in our profile, as it is more precise than natural language or pseudo code, and widely used in UML profiles. Tagged values are additional meta attributes assigned to a stereotype, specified as name-value pairs. They have a name and a type and can be used to attach arbitrary information to model elements.

The MOF (Meta Object Facility) [8] defines a simple metamodel (also called MOF model) with enough semantics to describe metamodels in several domains, being the initial focus analysis metamodels for OO projects. The integration between metamodels is done by the interfaces also offered by MOF, being necessary for the integration of tools and applications (for the several phases of the development life cycle) using a common semantics.

The main objective of MOF is to offer a framework that supports any metadata type and that any metadata type can be added when necessary. To do that MOF uses a metadata multilayers architecture based on the traditional metamodeling four layers architecture (very popular inside the standardization communities as ISO and CDIF). The key feature of that architecture is the metamodeling layer that supplies a common language that joins metamodels and models. The highest layer - M3, is composed by the MOF model that is for example instantiated in metamodels for UML or OMG [9]. IDL in the level M2 and these are for its time instantiated in the UML and OMG IDL models respectively in the layer M1. The MOF model is object oriented and its metamodel constructs were defined using UML (and therefore the UML descriptors). The MOF model is self descriptive that is to say the MOF model is formally defined using its own constructs. That characteristic of being self descriptive facilitates the definition of MOF interfaces and behaviors by applying the MOF IDL mapping to the MOF model, offering this way semantic uniformity among computational objects that represent models and metamodels. This also means that when a new mapping technology is defined, APIs for mapping metamodels in that context are also defined implicitly.

MDA (Model Driven Architecture) is a new paradigm in the systems specification [10]. MDA can specify systems at all levels, including middleware levels. This new model of systems architecture, offers us a group of very important advantages for distributed environments: (1) supports the whole development life cycle with more precision; (2) decreases development costs; (3) applicable to all languages, platforms, operating systems, networks and middlewares; (4) most of the MDA patterns is already available: UML, MOF, XMI, CWM; (5) a powerful middleware infrastructure: CORBA, IDL and services; built on a success OMG platform technologies OMG, like CORBA and UML; (6) rigorous mapping in the future for another infrastructures as: XML, SOAP,
Java, .NET; (7) MDA being independent of middleware, languages or systems (language -, vendor - and middleware-neutral) guarantees that following its requirements and good development practices will have an application with scalability and migration potential.

4. VCO and VDML

4.1. VCO

UML methodology, tools, technology and notation are a feasible approach for supporting the creation, development and maintenance of ontologies. This idea is illustrated on figure 2: (1) ODM and UOP are defined in terms of MOF metamodels; (2) UML profiles define a visual notation for OWL ontologies, based on the metamodel. Mappings on both directions are establish; (3) VCO instantiate from ODM and is based on OWL. VCO will allow relations among different terminologies (e.g. design, product engineer, manufacturing, production and deliver) and other derivate from those. This profile will allow in the Activity Diagrams representing business process goals, and their performance measures. In this paper we proposed a UML profile for modeling process in the new vehicle development.

Figure 2: Ontology modeling based on MDA and MOF.

VCO is a semantic markup language for publishing and sharing ontologies in the automobile business and its development as a vocabulary extension of RDF and is derived from DAML+OIL [4]. VCO aggregates other concepts (e.g. classes, proprieties, etc). It groups instances of other concepts that represent similar or related knowledge. A detail description of VCO can be found at [11] All mappings defined are performed in XMI. XMI (XML Metadata Interchange) [12] is a standard for exchange of object oriented systems models that tries to solve the interoperability problem in those two levels through the definition of a generic code pattern, XML [13], and with the definition of a pattern for conceptual schemas, called MOF. Standard XMI was created with the main goal of allowing the interoperability among CASE tools, metadata repositories and development tools, through the metadata exchange in a file or flow (stream) of data based on standard XML. In the conceptual level, XMI is based on other pattern called OMG MOF. MOF as we have just seen is a pattern for defining program interfaces for model repositories, but is also a pattern to describe metamodels. MOF is generic and rich enough to be capable to describe appropriately any object oriented metamodel as for example UML. At the codifying level, XMI is based on XML. But XMI is not a XML language. XMI is a specific ion of XML to generate XML languages adapted for data models as well as to code these metadata in a XML document. Then, the XMI specification is nothing else that a group of rules that normalize the generation of XML starting from MOF.

4.2. UML Profile for New Vehicle Design

The aim of this new profile created specific for new vehicle process development is to capture the specific concepts involved in this activity process and provide an appropriate notation. Based on UML we will propose a meta-model for automobile industry, based on a specific stereotype. New vehicle design main stereotypes are: (1) «Activities», which are performed by actors and operated over resources and information; (2) «Actors», that are someone (a human actor) or something (an automated actor, such as an information system or a production machine) that can perform the actions required by an activity; Actors belong to organization units (departments) (3) «Resource» is the input and output of an activity representing things such as materials, information system operated by human actors; (4) «Information» is also input/output of activity ;(5) «Goal» represents a measurable state that the organization intends to achieve; (6) «Measure»; (7) «Alert». To more information see [1].

5. EAI in automobile business

There is a big diversity of applications and data bases. These databases are often created from the point of view of a particular application and not from a broader company-wide point of view, which makes difficult the reuse across different applications. Owing to those difficulties, developers tend to create new data sources that have some semantic overlap with existing sources. For a particular application this might be an efficient solution – from a company-wide perspective, however, it is the worst solution.

Our aim in the VCO is to create a knowledge space by combining existing data sources from the automotive industry. We applied Semantic Information Integration to solve the problem of integrating heterogeneous data sources in an industrial setting. VCO has created information architecture to support locating and accessing data residing in heterogeneous databases throughout an industrial organization. In order to realize this
architecture we had to integrate the heterogeneous data sources and present the user a unified view of the disparate data schemas for browsing and querying. We chose to create a global schema, integrating all source schemas that act as a mediator [14] between the various data sources. When the mapping between the source schemas and the global schema have been created, the user can browse the global schema and in this way discover what information is presented in the organization and where the information is located. The user can also issue queries to the mediator that is automatically translated to the respective platforms and schemas of the data sources, where they are then executed. Using the mappings from the various source schemas to the global schema, it is also possible to automatically derive transformations of instances between different schemas. We used a semantic approach to the integration problem instead of abstracting a global database schema from the source schemas. The meaning of the data is captured in a central ontology and the data in the sources is given meaning by creating mappings between the sources and the ontology. Mapping operations are performed through three operations: (1) transformation the ontology in XMI. In our case we use Enterprise Architecture (EA); (2) format change to XSL, VCO oriented; (3) transformation is performed based on predefined templates based on VCO language. This means the original platforms and data schemas remain unchanged and are not replaced by the global schemas and the applications using these data sources do not need to be changed when creating the central ontology. We integrate a number of existing heterogeneous data sources, using different platforms and (syntactically and semantically) different data schemas, from the automotive industry by creating a central ontology integrating these data sources. We created mappings between the source schemas and the ontology, thereby creating an integrated unified global virtual view [15] of the information presented in the disparate data sources throughout the enterprise and enabling the querying of the disparate data sources via the central ontology. The data sources used in this approach were academic, but we are trying to use real data. We are aware of the diversity of data schemas and formats. All these sources had to be integrated in the information architecture.

6. Conclusions
Ontologies are very powerful in the sense that they are developed with the human understanding of the domain in mind, instead of taking a very application-oriented approach, as is mostly done with database schemas. Ontologies can help bridge the gap between human understanding and machine understanding of a domain [16]. We developed a mapping process between different database schemas in an organization, map them to a central Information Model or to other database schemas, and in that way enable better understanding of this information by the users in the organization. In automobile industry proprietary data and applications turns difficult standardization and test environments. Other important issue is that the knowledge obtained from the description of process activities using a UML profile (VDML, in this case) can be used in the ontology definition through a mapping between VDML and VCO at M2 layer. We still have a wide work to perform including tests due to the diversity of data and applications and the complexity of the complete process activity.

REFERENCES
[8] OMG. Meta-Object Facility (MOF) 1.4, OMG UML