ISSS/WS-EC-ECIMF/03/0xx

ISSS/WS-EC/03/xxx

CEN/ISSS Electronic Commerce Workshop

E-Commerce Integration Meta-Framework

CEN Workshop Agreement xxxxx:2003

Final draft
February 2003

ECIMF Project Group

1 Introduction

1.1 Background and Goal Statement

There have been many standardization activities in the area of e-commerce communication. The standard bodies and industry groups in multi-national levels have been promoting several standards. Some of these, with long-standing tradition (like EDI variants), have gained significant acceptance, especially among large industry players. However, these standards are often criticized for their complexity, high implementation cost, multitude of local variants, and extensive demand for expertise knowledge. Other frameworks for electronic commerce, defined more recently in the Internet age, try to avoid those mistakes, and they also have seen some acceptance in selected industry sectors (RosettaNet, OAG, cXML, xCBL, upcoming ebXML …).

However, the proliferation of mutually incompatible standards and models for conducting e-commerce resulted in even more increased demand for interoperability and expert knowledge. So, overall, the isolated efforts of industry groups and standard bodies created quite the adverse effect from what was intended, when it comes to wide acceptance of electronic commerce, especially in the SME market.

These issues slow down the spreading of e-commerce applications, and for this reason the industry is looking for methods to meet the exploding demand in the "new economy" to offer increased QoS, reduction of manual labor and cost, and to meet the requirements of nearly real-time reaction to changing market demands. At the same time the industry is aware that existing e-commerce frameworks require costly adjustments in order to fit their business model to that of specific frameworks, with the perspective that similar costs will follow if the business player wants to participate in other frameworks as well.

1.1.1 E-Commerce Integration Meta-Framework scope

In response to these concerns from the industry, this CEN/ISSS project within Workshop for Electronic Commerce proposes the E-Commerce Integration Meta-Framework (ECIMF):

> ***A meta-framework, which offers a methodology, a modeling language and prototype tools for all e-commerce users to achieve secure interoperability of the service regardless of system platforms and without major adjustments of existing systems.***

The most important characteristic of this project is to present a common approach to enable interoperability without enforcing major changes to the existing infrastructure. This is in contrast with many other widely promoted approaches to interoperability, which require from partners to be strictly conformant to a common standard in order to participate in e-commerce.

There are strong reasons for preferring the "enable" instead of (commonly endorsed) "enforce" approach:

- Business partners may have already made significant investments in building interfaces conforming to some standard(s).
- Commonly used integration methodologies are focused on data translation, which results in complex and inflexible solutions. Changing such integration solutions to accommodate new standards is often infeasible.
- There will always be legacy systems that need to be integrated with the "standard of the year" external interfaces. It is simply not realistic to hope that at some point in time all systems will adopt and fully conform to one common standard for every aspect of business communication.

For these reasons, the interoperability-enabling methodologies, such as the ECIMF approach, will play an increasingly vital role in the e-business communication.

The meta-framework, which the project aims to deliver, is understood as a combination of methodology, modeling notation (meta-models) and guidelines for aligning different aspects of e-commerce – hence the name “meta-framework”, because using these artifacts the users will be able to build concrete integration frameworks.

The main purpose of this meta-framework is to facilitate the interoperability by mapping the concepts and contexts between different existing e-commerce frameworks, across multiple architectural layers. An important premise for this project proposal is the following definition of interoperability:

> ***The interoperability, as seen from the business point of view, takes place when the business effects for the two involved enterprises are the same as if each of them conducted a given business process with a partner using the same e-commerce framework.***

As a consequence of this premise, the project proposes using a top-down approach to the comparative analysis of the e-commerce frameworks, which starts from the business context level. The project also reuses the experiences of other projects in the area of enterprise analysis and modeling.

The approach presented here also addresses integration of internal business processes and applications with external e-commerce interfaces required to conduct business electronically, whichever standard they conform to. This is just a special case of interoperability between differing frameworks. However, this case is crucial for companies in adoption of any e-commerce standard.

1.1.2 Benefits

The development and adoption of the ECIMF standard should benefit especially the following groups:

- SME market:
  The small companies no longer will be forced to restructure at all costs their internal systems in order to conform to whatever framework their bigger partners have. The interoperability bridges that conform to ECIMF will allow them to do it gradually, based on the economic principles, while at the same time allowing them to participate in the e-commerce. This should result in more SME-s joining the e-market, even though their internal economy systems may not yet follow any standard e-commerce framework.

- System integrators:
  The system integrators will be able to use a consistent methodology, and a precise framework for defining the integration bridges. The results of their work can be implemented on various conforming platforms, no longer locking them (and their customers) into a single proprietary tool. The overall cost for the implementing the integration solution, its maintenance and amount of manual labor will be reduced.

- Software vendors:
  The software vendors will be able to offer competitive integration products that conform to the standard framework. This means that their products will be more attractive to the customers, who are more likely to choose a solution that guarantees them certain level of independence. At the same time though, the conformance to ECIMF should allow software vendors to offer clearly understood added values, which are now often misunderstood because of the difficulty in comparing proprietary methodologies.

1.1.3 Relationship to various global e-commerce frameworks

The aim of the ECIMF project is not to propose yet another e-commerce framework. We recognize the efforts of various standardization bodies and industry groups to provide global solutions in this area (e.g. ebXML[1], RosettaNet, xCBL, OAGIS framework, Hewlett-Packard's e-Speak[2], Microsoft's BizTalk[3]), as well as other projects offering tailored solutions for specific market or industry sector.

The ECIMF project does not compete with any of these frameworks. We welcome and look forward to cooperate with their representatives in order to enhance the results of this project. The need that the ECIMF wants to address is the interoperability between these frameworks, especially for the transitory periods in

[1] The ebXML project, http://www.ebxml.org/specdrafts/ .
[2] The e-Speak framework, Hewlett-Packard, both as a commercial product http://www.e-speak.hp.com, and an OpenSource free Java implementation of the complete framework at http://www.e-speak.net .
[3] The BizTalk framework, Microsoft, http://www.microsoft.com/biztalk/techinfo/BizTalkFramework20.doc , BizTalk repository at http://www.biztalk.org, and the commercial product BizTalk Server http://www.microsoft.com/biztalk , which additionally contains the mapping and orchestration tools.

SME environment (economic and manpower limitations), which are required for adoption of any of the frameworks.

In our opinion at least two factors will continue to adversely affect the wide-spread adoption of e-commerce: one is the fact that quite a few businesses already made commitments to some of the existing frameworks, in terms of internal expertise, investments, partnerships, and adjustments to the technology and models for business interaction imposed by these frameworks. This situation is combined with the current approach to system integration, which very often locks up the companies to specific system integrator and specific proprietary solutions.

The other limiting factor is that extensive knowledge and experience is still required to adequately understand the differences between the frameworks, and even more to implement some level of interoperability – both between the e-commerce frameworks themselves, and between legacy systems and any given framework. Also, though more and more modern frameworks use UML and UMM to describe parts of their models, there is no general meta-framework that would allow implementing interoperability in a structured way, not to mention the fact that many frameworks are defined using imprecise, natural language descriptions.

It's worth noting a fact that is often overlooked: the differences between e-commerce frameworks are much deeper than just differences in their protocols, scenarios and data formats. There is a need for a unified methodology to compare and align also the semantics of central concepts in order to properly understand these differences.

The development of the ECIMF standard builds on the experiences from projects such as:
- ebXML: specifically Business Process Specification Schema (ebBPSS), Collaboration Protocol Profiles and Agreements (ebCCP),
- UN/CEFACT Unified Modeling Methodology (TMWG-N090),
- RosettaNet Implementation Framework v. 2.0 [4] (RNIF2.0),
- BizTalk 2.0 framework [3] (and BizTalk Server commercial tools),
- OAG Integration Specification (OAGIS 7.1),
- OMG's Model Driven Architecture (MDA),
- eCo framework [5]

and others, in order to provide a sufficiently broad and general model for alignment between the frameworks.

Consequently, we see the ECIMF project as a complementary and necessary part of e-commerce adoption, reducing the cost and amount of labor required to adopt any e-commerce framework.

1.2 Project Details

The following list shortly describes the scope for the ECIMF definitions:

[4] RosettaNet, http://www.rosettanet.org .
[5] The eCo Framework, CommerceOne, http://www.commerce.net/eco .

- **Meta-framework modeling methodology** – an approach to model the interactions and transformations required for mapping between different e-commerce frameworks:
  - Top-down analysis, based on the business process integration
  - Multi-layered modeling approach
  - Calibration of concepts within corresponding contexts (semantic translation)

  This part of the project requires close collaboration with the experts in order to reuse as much as possible the experiences collected by groups like ebXML, RosettaNet, OAG, EDI community and others.

  This part of the documentation is contained in section 2 of this document.

- **Meta-framework modeling language** – a precise notation to describe the concepts of e-commerce frameworks, the contexts in which they occur and interact, and the required transformations between them:
  - Business context correspondence (compatibility of economic goals)
  - Semantics of the base building blocks (actors, messages, transactions), data models
  - Scenarios for message exchange (business processes)
  - Access to external resources (URLs, directories, catalogues, databases, etc…)
  - Messaging models
  - Security models and services, as far as they affect the business process and interoperability on the technical level
  - Transport protocols
  - etc.

  For the business process modeling we suggest substantial reuse of the results of ebXML BP work (cf. ebBPSS), with additions of the modeling notation and language to express the transformations between the business processes on different layers.

  This part of the documentation hasn't been developed, since the previous part – methodology, which provides the basis for notation – hasn't been completed.

- **Proof of Concept** – the project will aim to provide a Proof of Concept implementation of the tool-chain needed for realization of the proposed methodology, demonstrating the interoperability between some concrete e-commerce frameworks. The tools developed by the project will be published under Open Source license, freely available for both private and commercial use.

  This part of the documentation is contained in the Appendix ??? of this document. Additionally, the Open Source application module supporting Semantic Translation with labeling is available on the project's dedicated website.

1.3 Original Project Deliverables and Timescales

The timeframe for this project was set up to be 18 months, in the period of June 2001 – December 2002. The manpower allocated on permanent basis to this project was initially planned as follows (expressed in percentage of time involvement times number of people):

- WebGiro: 50% x 1 person
- KTH: 25% x 2 persons

Furthermore, the list below presents prospective manpower that was likely to be involved on a regular basis:

- KTH: 25% x 1 person
- HP: 50% x 1 person
- Microsoft: 50% x 1 person

Unfortunately, in the course of running the project these resources have never been fully realized, which resulted in parts of this CWA being incomplete or missing.

Assuming the above resources, the originally planned deliverables consisted of the following separate documents (which later have been merged into one CWA):

- **General ECIMF methodology (ECIMF-GM):**
  A document (CWA) describing in detail the multi-layered approach, and the specification of the ECIMF methodology. This part should result from the discussions on the general methodology on how to approach the business process integration. The intention is to keep this part vendor- and tool-independent.

  This document, originally intended as a description of formalized methodology, due to the time and resource constraints was put in a form of general guidelines.

- **ECIMF technical specification (ECIMF-TS):**
  A document (CWA) containing the formal technical specification for modeling notation constructs, and the serialized form for the models (i.e. the ECIML and the MANIFEST specifications).

  This specification hasn't been developed, as explained above.

- **The Proof of Concept implementation (ECIMF-POC):**
  It would include the tools to support the methodology – the ECIMF Navigator for conceptual navigation and calibration, integrated with a ManifestFactory implementation in order to produce the MANIFEST recipes based on the model. It would also contain a Proof of Concept mapping of two business processes from different frameworks. This part should include additional examples of mapping, depending on the contributed resources.

  If the timeframe and the resources available are sufficient, a basic ECIML-compliant agent implementation should be created to support the Proof of Concept mapping.

The following milestones were planned for delivering the results:

1.3.1 Initial Proof of Concept (POC) for the approach

Deliverables:

- Reformulate and elaborate on the FAM CWA material in order to show how Conzilla tool can provide structured and contextualized added value to a textual description.
- Provide an initial description of the methodology for comparing the e-commerce frameworks (this will form the draft of ECIMF-GM document).
- Prepare a simple example of mapping the differences between two e-commerce frameworks (e.g. BizTalk and e-Speak), using the proposed approach.

Timescale: 12 June 2001 (Oslo meeting)
Status: delivered, available as a set of PowerPoint slides.

1.3.2 Initial ECIMF specification and basic integration with tools

Deliverables:

- Initial version of the ECIMF-GM and ECIMF-TS documents, and models of a concrete business process in two selected e-commerce frameworks.
- Customization of the Conzilla tool to support the modeling notation introduced in ECIMF-GM.

Timescale: mid-October 2001
Status: partially delivered ECIMF-GM. Initial models in Conzilla.

1.3.3 Refined ECIMF specifications and extended tool-chain

Deliverables:

- Refinement of the ECIMF specifications based on further comparative modeling of the selected frameworks.
- Extended support for the process in the tool-chain: integration of Conzilla, scripting language and the ECIML code generation to form the ECIMF Navigator tool.

Timescale: 1Q2002
Status: partially delivered. Extended ECIMF-GM and Proof-of-Concept documentation. However, Conzilla support lagging behind.

1.3.4 Further refinements to ECIMF specifications, and a reference ECIML-compliant agent implementation

Deliverables:

- More refined ECIMF specifications, and additions to the tool-chain to support the specification.
- Depending on the support from industry partners, a basic reference implementation of the ECIML-compliant server.

Timescale: 4Q2002
Status: partially delivered. ECIMF-POC documentation completed, but the toolkit only supports basic semantic translation support (Conzilla was replaced with Protégé).

1.4 External Liaisons

The project team coordinated its activities with the following projects:

- Other relevant CEN/ISSS/EC-WS projects
- ebTWG,

- RosettaNet,
- Open Applications Group,
- ISO TC/154 Basic Semantic Register

2 General Methodology[6]

2.1 Overview

The ECIMF project deliverables consist of a recommended methodology, presented in this document, and base tools needed to prepare specific comparisons of concrete frameworks (presented in the section 3 of this document, where you can also find the case studies).

The results of following the ECIMF methodology should be clear implementation guidelines for system integrators and software vendors on how to ensure interoperability and semantic alignment between incompatible e-commerce systems. This generic integration rules should be expressed in an implementation-independent language, providing mapping and transformation descriptions/recipes that can be implemented by ECIMF-compliant agents/intermediaries. This ultimately should allow the e-commerce frameworks to interoperate without extensive manual alignment by the framework experts, and will make the integration logic more understandable and maintainable.

2.1.1 Layered approach

The proposed methodology for analysis and modeling of the transformations between the e-commerce frameworks follows a layered approach.

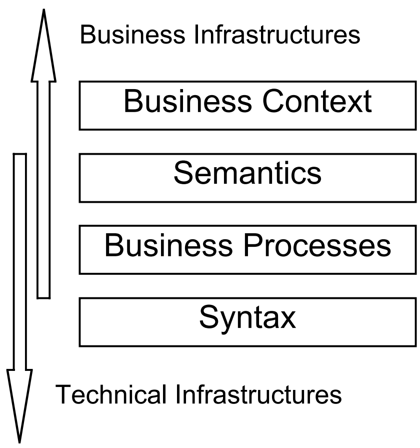


Figure 1 ECIMF layers of integration

This approach means that in order to analyze the problem domain one has to split it into layers of abstraction, applying top-down technique to classify the entities and their mutual relationships:

- First, to establish the scope of the integration task in terms of a **business context** – based on the economic aspects of the partners' interactions,
- Then, to identify the top-level entities and the contexts in which they occur (the data model), and how these contexts affect the **semantic** properties of the concepts,

[6] Editor's note: Originally this section formed a separate document. There may be still some inconsistencies related to this fact.

- Then, to proceed to the next layer in which the interactions (conversation patterns, **business processes**) between the partners are analyzed.
- Then, to go to the lowest, the most detailed level to analyze the messages and data elements **(syntactic level**) in communication between the partners.

Starting from the top-most level, the contexts in which the interactions occur are analyzed and collected, and these contexts affect the semantics of the interactions occurring at the lower layers.

The second dimension of the proposed approach conforms to the Meta-Model Architectures, as described in the MOF standard, introducing the meta-model, model and instance (data) layers. This means that ECIMF will be used to define:

- The modeling notation: a set of modeling concepts with their graphical and XML representation to model the transformations[7],
- The models: concrete transformations between concrete frameworks
- And the model instances of transformations, as realized by an ECIMF-compliant runtime.

Figures 1 and 2 present the ECIMF layers, and how they are applied to define the interoperability model between two incompatible frameworks.

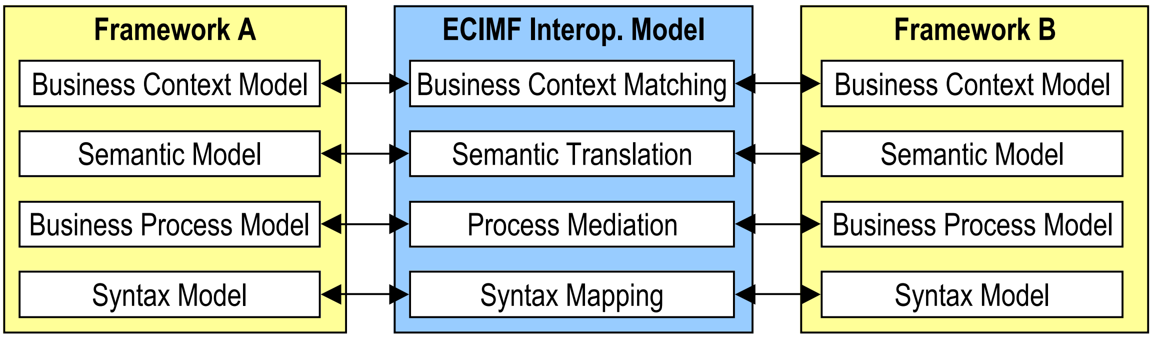


Figure 2 ECIMF methodology – interoperability layers.

Each of these layers is described in detail in the section 2.

2.1.2 Conceptual navigation – ECIMF Navigator

In order to navigate through the framework models and concepts, during the initial stages of the project a prototype tool named Conzilla was introduced, which in later stages of the project was to be augmented with other modules (like data format translating software, automatic generation of interfacing state machines, routing and packaging translators, etc). This extended toolset is called ECIMF Navigator, and its intended use is presented on the Figure 2.

[7] Since the modeling elements regard multiple layers of the ECIMF approach, hence the name "meta-framework", because they will be used to define interoperability frameworks.

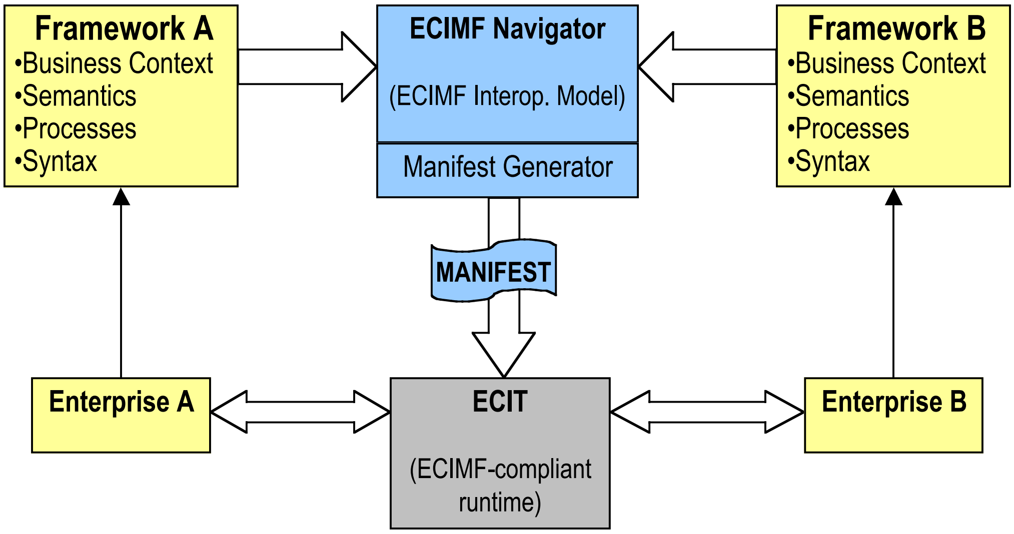


Figure 3 The ECIMF concept of frameworks transformation and alignment.

The ECIMF project used an extension of Conzilla (see http://ww.conzilla.org for more information about the Conzilla project) as a prototype tool for browsing and comparing different e-commerce framework models. One of the goals of the ECIMF project was to extend this tool by necessary backend(s) for producing abstract machine-readable interoperability guides (MANIFEST recipes), expressed in ECIML language.

In later stages, after some limited development and evaluation of future possibilities of the Conzilla platform, the ECIMF project switched to using a well-known knowledge engineering environment Protégé (http://protege.stanford.edu), as it seemed to better match the requirements for extensibility, wider acceptance and sustained maintenance. Concequently, the support for parts of ECIMF methodology has been implemented as Protégé module (so called "tab").

2.1.3 Top-down, iterative process

The ECIMF uses a classic top-down approach for solving the interoperability issues, but combined with an iterative process of refining the higher level models based on the additional information gathered in the process of modeling the lower levels.

This process is described in detail in the Framework Integration Guidelines section.

2.1.4 The modeling notation

The ECIMF project proposes to use an extended UML modeling notation (a UML profile) to express relationships between the semantics and models of the e-commerce frameworks. This E-Commerce Integration Modeling Language ("ECIML"), to be defined as a result of the project, will be a concrete instance of the OMG's MOF meta-meta-model, at the same time re-using as many concepts from standard UML as possible. This puts it in the following relationship to the standard modeling approaches:

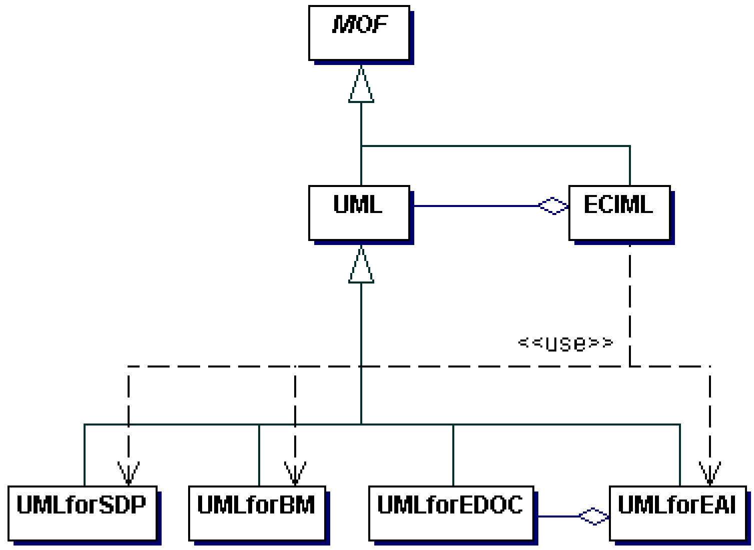


Figure 4 Relationship between the ECIML and other modeling standards.

In other words, the ECIML will be yet another profile of UML 1.4. We will build on the experiences of the projects like pUML (The Precise UML Group), using also the OMG's standards (e.g. CWM, standard UML 1.4 profiles, UML Profile for EAI and UML Profile for EDOC) when appropriate, in order to define a suitable meta-model. We will also reuse as much as possible the specialized concepts developed by the UN/CEFACT Unified Modeling Methodology (UMM), as described in TMWG-N090R10.

One could use the standard UML for modeling the interoperability concepts, but we feel that in its current form it is too generic and lacks necessary precision, and though it's extensible, the way the extensions are specified is often implicit (e.g. stereotyping). In the ECIML meta-model these concepts would be precisely defined. Some of these issues will be addressed in the next major revision of UML standard (2.0), at which point we will evaluate the possibility to use that standard as the sole basis for ECIML.

Consequently, one of the original goals of this project was to define a suitable set of modeling constructs to more adequately address the needs of meta-framework modeling and transformations. However, due to limited resources this part of the project has not been completed.

2.2 Methodology

As mentioned in the overview section, the ECIMF methodology addresses the following four layers of interoperability:

- **Business Context Matching:** this aspect deals with setting up the scope of the integration task – we assume that preparing a complete integration specification for all possible interactions might not be feasible (even if it were possible at all), so the task needs to be limited to the scope needed for solving a concrete business case. This case is identified, the models for each party are prepared, and then it needs to be determined if they match, i.e. if the business partners try to achieve the same business goals.

- **Semantic Translation:** in this step the key concepts and their semantic correspondence is established, so that they can be appropriately transformed whenever they occur in contexts of each of the frameworks (which is also known as “semantic calibration” [CID52]).
- **Business Process Mediation:** in this step the necessary mediation logic is defined, by introducing an intermediary agent that can transform conversation flow from one framework to that of the other, while preserving the business semantics (e.g. the transaction and legal boundaries).
- **Syntax mapping:** in this step the mapping between data elements in messages is defined, based on the already established semantic correspondence and translation rules defined in the first step. Also, the transport protocol and packaging translation is specified.

The following sections describe in detail each of these areas of interoperability.

2.2.1 Business Context Matching

2.2.1.1 Business Context – definition and role

- IT infrastructure exists to support business goals: IT systems don’t exist in a void, but they play specific roles in the business.
- Business context is therefore crucial: information is useful only when considered in the right business context. It is the business context that ultimately determines the meaning of data and information exchange.
- Business flow should therefore be considered before technical flow.
- REA modeling framework can be successfully used as the underlying meta-model

Business Context is a collection of:

- ***Agreements / Contracts defining the Commitments***
- ***Collaboration Patterns (using Business Processes) to execute commitments***
- ***Business Objects with their semantics, lifecycle and state, which encapsulate business data and business rules***

2.2.1.2 Resource-Event-Agent modeling framework

REA Enterprise Ontology has been created by William E. McCarthy, mainly for modeling of accounting systems. However, it proved so useful and intuitive for better understanding of business processes that it became one of the major modeling frameworks for both traditional enterprises and e-commerce systems. Recently, it has been extended to provide concepts useful for understanding the processing aspects (processes, recipes) in addition to the economic aspects (economic exchanges). Please see http://www.msu.edu/user/mccarth4/ for more information.

Some of the REA concepts have been used to model the Business Requirements in UN/CEFACT Modeling Methodology ("UMM", formally known as TMWG N090), and the Business Process Analysis Worksheets in ebXML, and it's use is currently a subject of further study in the Business Collaboration Patterns and Monitored Commitments team of the E-Business Transitionary Working Group (eBTWG) - the successor to ebXML.

2.2.1.2.1 Economic exchange as a central concept

- REA ontology focuses on the idea of economic exchange of resources as the basis of business and trading. In REA models, economic agents exchange economic resources in series of events, which fulfill mutual obligations (called Commitments), as specified in an Agreement between the business partners. See also the detailed definitions in the ECIMF-TS document.
- Economic exchange models define collaborations between partners involved in the process, and these collaborations naturally map to business document exchanges (both in paper and in electronic form).

2.2.1.2.2 Value-chain models (REA Enterprise Scripts)

- REA process diagrams show the high-level flows of economic resources in the enterprise, related to the economic events and collaborations between the agents involved in the exchanges. They are sometimes referred to as value-chain diagrams.
- The resource flows between processes in the value-chain diagrams represent the collective unbalanced stock-flows, consumed and produced by the events belonging to given processes.
- Value-chain model (also known as *REA Enterprise Script*) is a series of processes, consisting of exchanges, where collaborations between agents are realized with recipes (groups of ordered tasks).

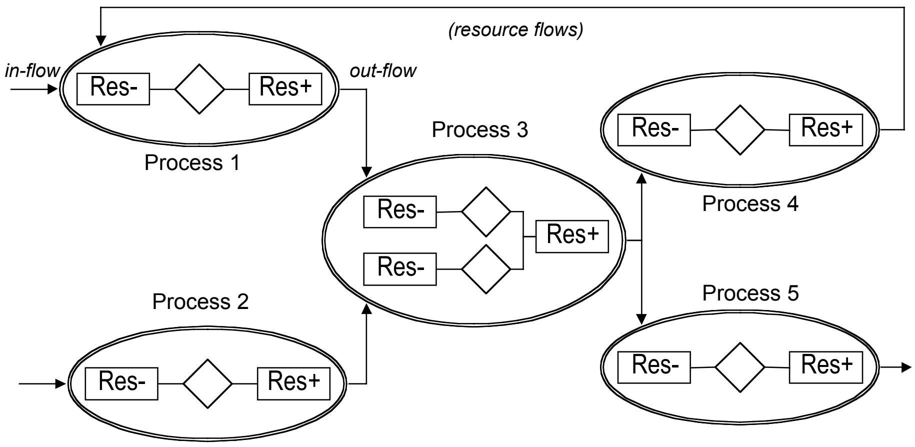


Figure 5 Enterprise value-chain, seen as series of exchanges.

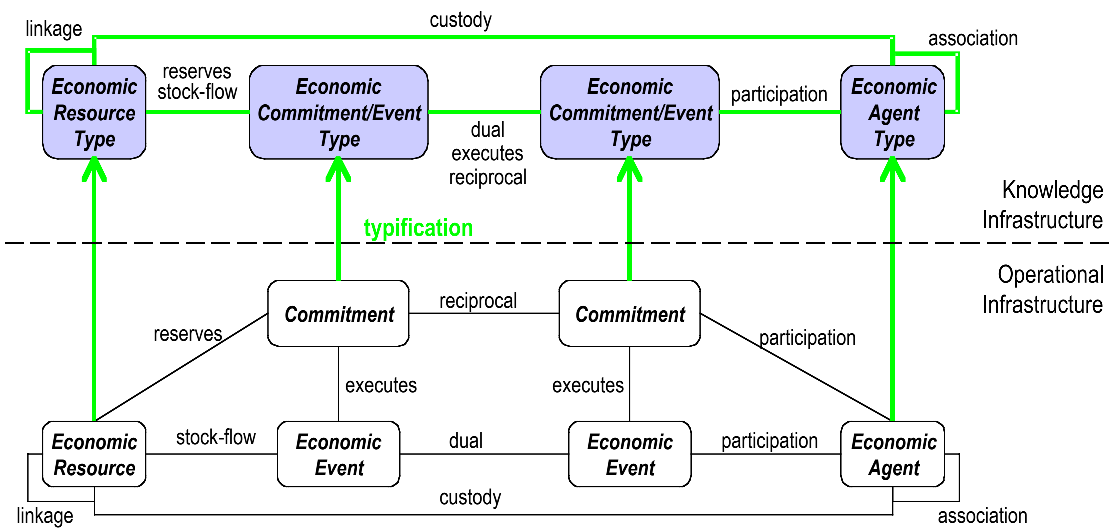


Figure 6 REA meta-model of economic exchanges (simplified).

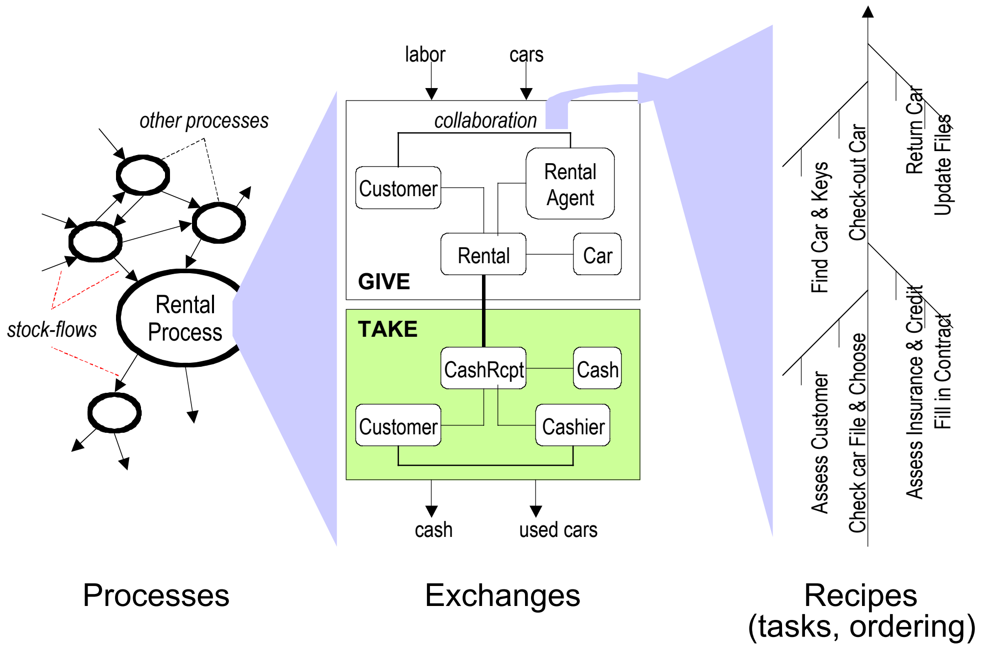


Figure 7 Overview of the processes, exchanges and recipes.

You will find the detailed description of this meta-model in the ECIMF technical specification document (ECIMF-TS).

2.2.1.3 Business Context Matching rules

2.2.1.3.1 Rationale

- Traditional trading partners' agreements: both partners need to agree on:
  - The type of resources exchanged
  - The timing (event sequences/dependencies)
  - The persons/organizations/roles involved

  Also, each of the partners needs to follow the commitments under legal consequences

- Conclusion: in the traditional business, partners achieve common understanding through negotiations, and their results and conditions are then recorded in a formal written contract. In electronic business some standards support creation of electronic TPA's (Trading Partner Agreements). Their formation is a special case of establishing the Business Context Matching described here.

2.2.1.3.2 Matching Rules

Business partners involved in an integration scenario need to consider first whether their business goals and expectations match, before they start solving the technical infrastructure problems. For that purpose, they can create two (or more) business context models, one for each party involved in the integration scenario. The interoperability of the e-commerce scenario, as implemented by two different partners, requires that these models match.

There are several requirements that the models have to meet for them to be considered matching:

2.2.1.3.2.1 #1: Complementary roles

Parties need to play complementary roles (e.g. buyer/seller)

2.2.1.3.2.2 #2: Matching resources

The resources expected in the exchanges need to match to the ones expected by the other partner (e.g. the provided resources could be subtypes of resources requested)

2.2.1.3.2.3 #3: Satisfied timing constraints

The timing constraints on events (commitment specification) need to be mutually satisfiable (e.g. down payment vs. final payment, payment within 24 hours, shipment within 1 week, etc...)

2.2.1.3.2.4 #4: Transaction preservation

The sequence of expected business transactions needs to be the same (even though the individual business activities and resulting conversation patterns may differ). This is especially important for those transactions, which result in legal consequences.

If the above conditions are met, we can declare that the parties follow the same business model to achieve common business goals, and that the differences lie only in the technical infrastructure they use to implement their business model. If any of the above requirements is not met, there is no sufficient business foundation for these parties to cooperate, even in non-electronic form.

A successful completion of this step means that we have established a common business context for both parties. We have also identified the events that need to occur, and the collaborations between agents that support these events. This in turn determines the transactional boundaries for each activity. See example scenario in the Proof-of-Concept section for an illustration of these principles.

(NOTE: this section definitely needs more work…)

This business context model will help us to make decisions in cases when a strict one-to-one mapping on the technical infrastructure level is not possible. It will also help us to decide what kind of compensating actions are needed in case of failures.

2.2.2 Semantic Translation (to be completed)

Figure 8 presents the idea of the semantic translation and the reason why it's a required step in solving the interoperability puzzle. In general, the concepts underlying the foundations on which the IT infrastructures are built, differ between not only the industry sectors, or geographical regions, but even between each company within the same sector. This phenomenon – of different semantics, and different ontologies – causes many complex problems in the area of system integration, and in the area of e-commerce integration specifically.

One of the most common cases that require semantic translation to be performed is when each business party uses a different product catalogue (this situation is sometimes referred to as the "catalog integration", or "catalog merging" problem).

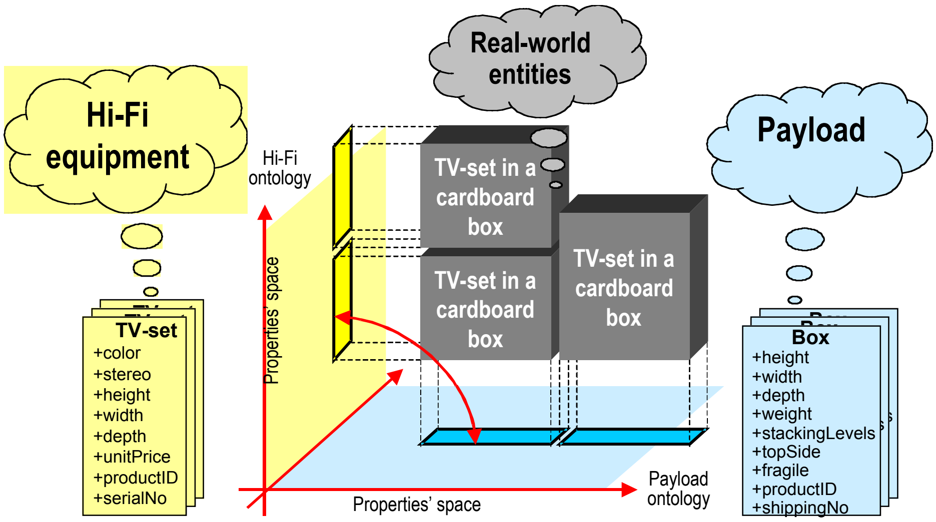


Figure 8 Mapping concepts from different ontologies.

In the example presented on Figure 8, a real-world entity - TV-set in a cardboard box - is represented very differently in two domain ontologies - the ontology of Hi-Fi equipment, and the transportation ontology. Although two representations may refer to the same real entity, in order to communicate that fact to the users of the other ontology we need to perform a semantic enrichment, in order to determine the proper classification of the concept in the other ontology.

What's even worse, we may discover (as is often the case) that the concepts overlap only partially, and the conditions under which they match the concepts from the other ontologies are defined by complex formulas, dependent potentially on several factors such as values from external resources, time, geographical region etc. In this case, the physical dimensions of the TV-set concept are confusingly homonymous to the dimension properties of the Box concept, but in the first case they refer to the TV-set chassis, and in the second case they refer to the cardboard box dimensions. Furthermore, the Box dimensions might be allowed to take only certain discrete values (e.g. according to a normalized cardboard container types), so in order to determine their values based on the information available in the TV-set concept, it is necessary to access some external resource (a cardboard box catalogue).

2.2.2.1 Describing semantic mapping

2.2.2.1.1 Semantic Translation meta-model

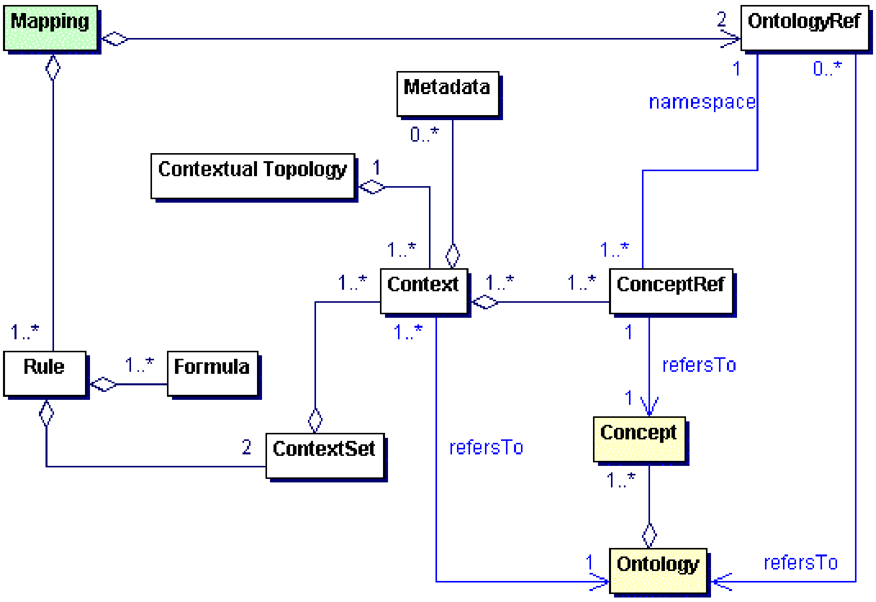


Figure 9 Semantic Translation meta-model

Figure above presents the meta-model for capturing the rules of semantic correspondence between concepts belonging to two different ontologies. This meta-model has been developed based on the principles of contextual navigation, which means that the proper understanding of a concept requires considering the context in which it occurs.

Furthermore, the translation rules (mappings) only refer to the original ontologies and concepts, which means that the original definitions, constraints, relationships and axioms are not recorded in the translation rules, but are only represented by unique identifiers (references). The reason for this is that especially in the e-commerce scenarios these source ontologies are usually completely separate, and maintained by separate organizations. These two concepts (Ontology and Concept) are accordingly marked as “external” in the list below.

- *Ontology*: the original full domain ontology (external)
- *Concept*: concepts defined in the original Ontology (external)
- *Mapping*: a top-level container for the semantic mapping rules, applicable to a pair of ontologies, as specified by the *OntologyRef*-s. (The *Mapping* is marked green in the diagram as the starting point for reading the whole meta-model.)
- *OntologyRef*: a URN uniquely identifying the referred ontology (possibly allowing to access it remotely).

- *ConceptRef*: a namespaced reference to individual *Concept*-s defined in the original *Ontology*. A URN, which possibly allows to access remotely the concept definition in the original ontology.
- *Context*: built on the basis of the original *Ontology* (refersTo), consists of related concepts represented by *ConceptRef*-s, which are considered relevant to the given transformation rule (the exact and full relationship of the *Concept*-s is defined in the original ontology - *Context* captures just the fact that they are related for the purpose of mapping).
- *ContextSet*: a group of one or more *Context*-s referring to the same *Ontology*.
- *Rule*: a rule that defines how to translate between the concepts in a *ContextSet* from one ontology, to the corresponding concepts in a *ContextSet* from the other ontology. A *Rule* consists of exactly two *ContextSet*-s, each one referring to respectively one of the ontologies, and a set of *Formula*-s, which define the valid transformations on these *ContextSet*-s.
- *Formula*: a formal expression defining how translation is performed between concepts from the source *ContextSet* to those in the target *ContextSet*.

The reason for defining the *ContextSet*, in addition to *Context*, is that probably we would like to use concepts from several contexts belonging to a single *Ontology*, and map them to several contexts in the other. But at the same time there is a requirement to state explicitly that we always map between exactly two different ontologies.

2.2.2.1.2 Algorithms for discovering the semantic correspondence

(Many exist, none ideal or fully automatic. There is a need to use several in parallel, plus heuristics…)

2.2.2.1.3 The Formula language

(Needs to be more complex than first-order logic. Probably a full-fledged programming language, e.g. XSLT, JavaScript, XQuery, etc.)

It is yet to be defined what kind of language will be used to describe the transformations between the models. The following is a short list of the requirements that need to be satisfied:

- Preferably Open Source implementations available
- Highly portable
- Well-known: this is needed in order to ease the adoption
- Strongly typed: the transformations need to be precisely defined, and it’s preferred that most logical errors would be discovered during the parsing/compilation, not at the runtime.
- High level (additional tools for manipulation of complex programmatic structures, database and directory access, etc…)

The candidates that we consider at this stage are Java, JavaScript, XSLT, XQuery and Python.

2.2.2.2 Example model

Below is an example of (part of) the model built with the Semantic Translation meta-model.

(NOTE: for now the Formula language is unspecified, and in this example a JavaScript-alike language was used).

```
Rule:rule1
| | |
| | +-- ContextSet:set1 {Ontology 1}
| |       \Context:Party
| |       \Context:Address
| |       \Context:PartyIdentification
| |       \Context:Name
| +--ContextSet:set2 {Ontology 2}
|         \Context:Agent
|         \Context:Location
|         \Context:Name
|         \...
\Formula:formula1
|   \body: "set2.Name = set1.Name"
\Formula:formula2
|   \body: "set2.Location.Address.Street1 =
                  set1.Address.Street;
            set2.Location.Address.Street2 =
                  concat(set1.Address.Zip, set1.Address.City);"
\Formula3:Formula ...
        .....
```

(NOTE2: There is also a working hypothesis that one could use a rule of thumb to treat the ebXML aggregate core components as Contexts, and most primitive core components as concepts - but this needs further research, and discussions with the eBTWG community.)

2.2.3 Business Process Mediation (to be completed)

2.2.3.1 Business Process Models

The elements of Business Process models describe the major steps in the interaction scenario that need to be performed in order to successfully execute the mutual commitments. In this step we identify the business transaction boundaries, and the activities that need to be performed in order to fulfill them, or what kind of activities are needed to rollback (or compensate) for failed transactions.

A *business process* (according to [REA],[ebXML],[UMM]) consists of a sequence of *business activities* performed by one business partner alone, and *business interface activities* performed by two or more business partners. In the ECIMF methodology we will be interested primarily in aligning the *business interface activities*, although in most cases understanding both types of activities is needed in order to understand the business process constraints. These activities realize the collaborations between the involved business Agents, and they also support the economic exchanges identified in the Business Context models. Further, we will use the term BusinessActivity to mean the business interface activity.

In this model, each collaboration task is further decomposed into *business activities*, which may involve one or more *business transactions*, which in turn are executed with help of *business documents* and *business signals*.

2.2.3.1.1 Business Process Meta-model

Here are more detailed descriptions of each of the modeling elements:

- *BusinessProcess*: contains one or more economic exchanges, which in turn contain two or more BusinessCollaborationTasks each.
- *BusinessCollaborationTask*: a logically related group of BusinessActivities, which realizes the collaboration between two Agents in a given Event.
- *BusinessActivity*: a business communication (initiated by a requesting or responding business Agent). BusinessActivities may lead to changes in state of one or both parties.
- *BusinessTransaction*: a set of BusinessDocuments and BusinessSignals exchanges between two parties that must occur in an agreed format, sequence and time period. If any of the agreements are violated then the transaction is terminated and all business information and business signal exchanges must be discarded (possibly some additional compensating actions need to be taken as well).
- *BusinessDocument*: a message sent between partners as a part of information exchange, which contains business data (payload).
- *BusinessSignal*: a message that is transmitted asynchronously back to the partner that initiated the transfer of business process execution control (by sending a BusinessDocument), which doesn't contain any business data, but instead just signifies acknowledgement or error condition.

(NOTE: probably this meta-model needs to be harmonized with UMM or eBTWG, but there is also a need to provide a ***simplified*** *version…)*

2.2.3.1.2 Business Process Models

Business processes are most often modeled using UML activity diagrams (or similar notation), where each diagram represents one of the collaborations. This view relates to the Business Context view in the following way:

- The collaboration links between Agents correspond 1:1 to BusinessCollaborationTasks. This means that for the typical economic exchanges there will always be two BusinessCollaborationTasks – one for the "give" part, and one for the "take" part of the exchange.

In addition to that, the BusinessProcess view enhances the understanding of the Business Context, because it allows us to correlate various Events that are dependent on each other even if they don't belong to the same economic exchange (e.g. consumption of resources, replenishment and sales tasks are dependent on each other, but they are not likely all to be part of the same BusinessCollaborationTask between two specific partners).

2.2.3.1.3 Business Collaboration Tasks and Business Transactions

- The BusinessCollaborationTasks support the execution of the BusinessEvents identified in the previous step. There should be as many Business Tasks as many collaboration links were in the Business Context models.
- BusinessEvents are realized by one or more BusinessTransactions. Consequently, BusinessCollaborationTasks consist of one or more BusinessTransactions
- BusinessCollaborationTasks are represented as UML activity diagrams, showing the activities of both collaborating agents. These diagrams usually contain two parts (swimlanes): one for the requesting (initiating) party, the other for the responding party. The diagrams should also contain the messages passed between the parties.

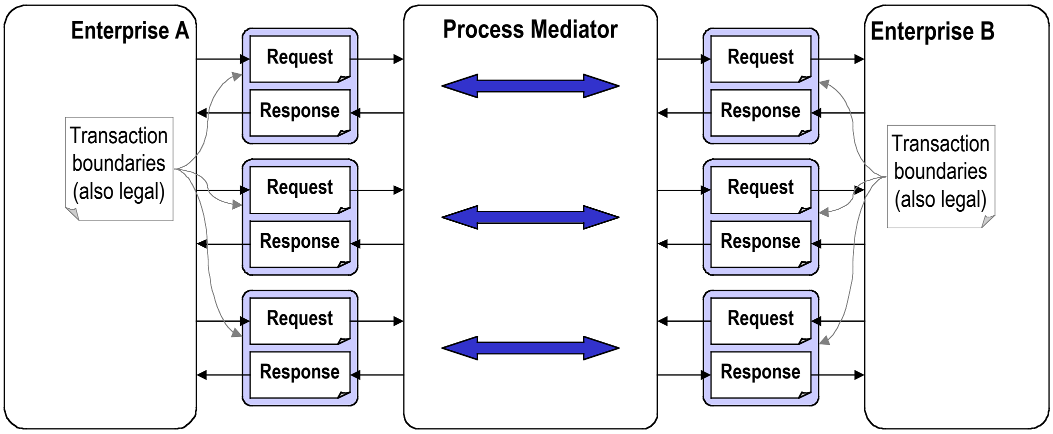


Figure 10 Example scenario that requires Process Mediator.

2.2.3.2 Business Process Mediation Model

The mediation between two different conversation patterns (which may involve different low-level technical transactions) needs to be designed and managed in a Business Process Mediation model.

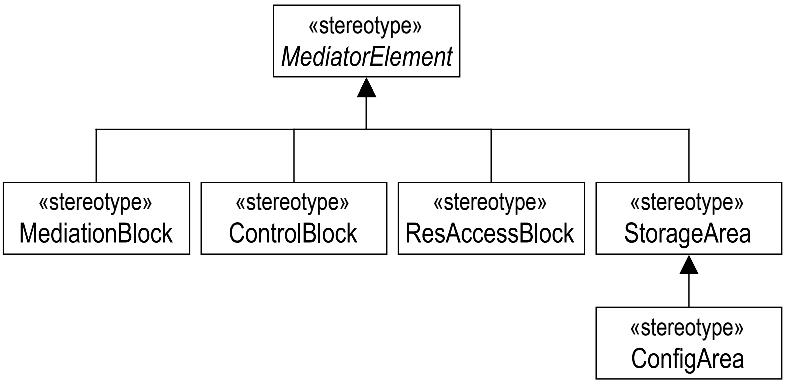
2.2.3.2.1 Business Process Mediation Meta-model

(NOTE: the working hypothesis is that the model elements will be responsible for reconciliation of concrete aspects of conversations. The current idea of the internal structure of the model is as follows:

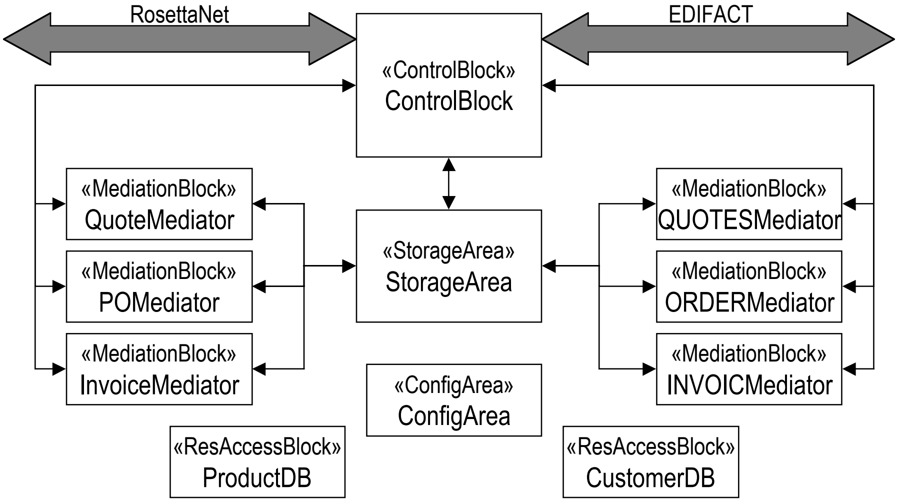
- *there will be mediation blocks handling the flow of each business transaction – totally the number of distinct business transactions on one side plus the number of distinct business transactions on the other side. These mediation blocks will be responsible for handling the details of conversations according to a given framework, within the boundaries of one specific transaction.*
- *there will be resource wrapper blocks, allowing for uniform access to external resources*
- *there will be one controlling block, responsible for managing the overall flow of transactions.*

- *there will be a common storage area, which any mediation block or the controlling block can access in order to store intermediate data – such as previous messages*
- *similar to that, there will be a configuration area accessible to all blocks, containing the configuration parameters.*

To summarize, the following diagram presents the meta-model:



And the diagram below presents a mediation model example:



Again, this is just a working hypothesis – any comments are much welcome!)

2.2.3.2.2 Checking the task alignment
(to be completed…)

2.2.3.2.3 Creating the Mediation elements
(to be completed…)

The process of building this part of the integration model is very closely related to the Semantic Translation, because very often a semantic correspondence needs to be established between the concepts, transactions, messages and information elements.

A Process Mediator is responsible for monitoring the conversation flows between each partner and itself, and according to the mapping rules it should generate appropriate stimuli (in form of message flows) in order to achieve desired state changes in each partner's Business Objects, while preserving the transaction boundaries.

Readers are referred to the Proof-of-Concept section, which illustrates these principles on a real example.

2.2.4 Syntax Mapping (to be completed)

2.2.4.1 Data element mapping

(using the semantic mapping rules. Syntax mapping is often preformed with XSLT, plus optionally the straightforward wrappers for non-XML formats)

2.2.4.2 Message format mapping

(see above. Additionally, it needs to ensure the well-fomedness and validity of messages according to the format specifications.)

2.2.4.3 Message packaging mapping

(ebXML CPP/CPA ?)

2.2.4.4 Transport protocol mapping

(ebXML CPP/CPA ?)

2.2.5 MANIFEST recipes

The meta-framework definitions/recipes for interoperability are named "MANIFEST". The language to be used in these definitions will be called E-Commerce Integration Modeling Language ("ECIML"), and will be based on XML representation of ECIMF models, rules and definitions.

A MANIFEST document consists of a set of interoperability recipes, based on the transformation model prepared using ECIML notation and then expressed in a serialized (XML) format. The MANIFEST-s will be identified by a unique ID, and stored in the repository from which an ECIML-compliant agent can retrieve it. The agent, based on the transformations specified in the MANIFEST recipe, will create necessary processing structures to align the message handling and interactions between the agents belonging to different frameworks. It should also be possible for ECIML-compliant modeling tools to re-use already existing MANIFEST recipes to adjust the interoperability model to specific needs. It is expected that some publicly available repository will store the commonly used templates for inter-framework alignment, so that less experienced or knowledgeable users can leverage the accumulated expertise of framework experts, and by making relatively minor adjustments re-use the templates as their own MANIFEST recipes.

The specifics of the repository need to be further discussed. Initially we suggest possibility of using either ebXML or UDDI to store the MANIFEST recipes.

2.3 The ECIMF-compliant runtime toolkit

The project aimed to provide a simple implementation of the E-Commerce Integration Toolkit ("ECIT"), consisting of the ECIMF Navigator (extending existing toolkits, like

Conzilla or Protégé) and a basic implementation of ECIML-compliant agent, and make these available on an Open Source basis. However, in order to fully leverage the ECIMF approach, we expect the software vendors to follow our initiative and provide complete implementations as proprietary products – still, compatible with the open standard.

The alpha-stage version of this toolkit has been implemented based on the Protégé framework, and is distributed under Mozilla Public License (a non-restrictive, business-friendly open source license).

2.4 Frameworks Integration Guideline

The main objective of the ECIMF project is to provide clear guidelines and methodologies for building interoperability bridges between different incompatible e-commerce standards.

This section presents a general guideline to solving this issue in case of two incompatible e-commerce frameworks F1 and F2. Annex 1 gives additional supporting information.

The guideline has been divided into several steps, to be performed sequentially and iteratively, as needed. The steps follow the methodology described in the previous section – the layers on the top are addressed first, since they give the broadest context necessary for understanding of the lower-level data transformations. The successful completion of all steps will result in a set of interoperability rules, enforced by a framework mediating agent, which will allow parties using different frameworks to cooperate towards common business goals.

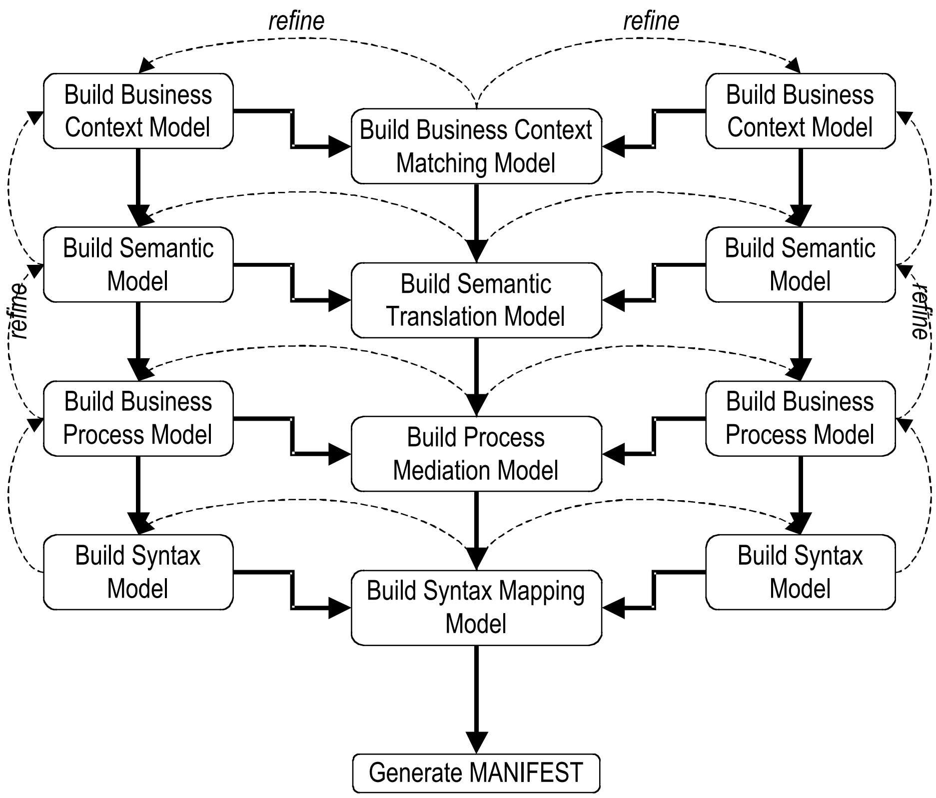


Figure 11 The process of modeling the integration recipes between two e-commerce frameworks.

The guideline has a modular structure, reflected in the fact that in each step several so-called *alternative procedures* have been defined. Each *alternative procedure* refers to a well-defined unit of work that needs to be done (a part of integration step), and allows you to replace or extend the approach suggested for that step with other methods of your choice, as long as they provide you with similar results (artifacts) as the input to the next step. The boundaries of each alternative procedure are clearly marked, and the input/output deliverables are specified.

You can also find a common meta-model defined in each of the steps, which serves as a common vocabulary (shared ontology) for understanding the incompatible frameworks.

One important thing to note here is that the integration modeling between two frameworks is asymmetric, i.e. the integration model will usually contain two elements that refer to the same individual model elements, but defined differently depending on the direction in which the data is traveling.

The subsections below present the details of the guideline.

2.4.1 Analysis of the Business Context Matching

2.4.1.1 Creating Business Context Models

A **business context model** shows a concrete business scenario expressed with the use of economic modeling elements, e.g. those found in the REA meta-model. We suggest using the following standard UML diagrams for that purpose:

- Class diagrams to show the specific types of entities involved.
- Collaboration diagrams to show a specific scenario populated with specific instances of participating entities.
- Value-chain diagrams (REA process diagrams), to clearly define the flows of resources, and how they depend on the collaboration between partners.

For examples of such business context models, please see the ECIMF-POC document.

2.4.1.2 Checking the Business Context Matching Rules

Each of the context matching rules needs to be checked, and any additional requirements or assumptions made need to be recorded, so that they can be used to understand the interactions in the lower layers of the ECIMF model.

Below is a table that summarizes this step of the guideline:

Business Context Matching	
Input	Traditional business knowledge, legal agreements between partners, industry specific rules, legal constraints, specific business goals, common business practices and codes of conduct
Output	Two Business Context Models for the integration scenario, defined in a set of UML diagrams (class, collaboration, activity), and an analysis of their matching (and any additional requirements on which the matching depends).

2.4.2 Creating the Business Process Mediation Model

2.4.2.1 Creating the Business Process models

A **business process model** shows concrete business collaboration, expressed as series of business activities and transactions between the partners. We suggest using the standard UML activity diagrams for that purpose, one diagram for each collaboration.

2.4.2.1.1 Identify the Business Collaboration Tasks

For each collaboration link in the Business Context diagram, a Business Collaboration Task is created.

2.4.2.1.2 Identify the Business Transactions

For each collaboration, and for each Agent, the business transactions are discovered and described. Since the Agents possibly use different frameworks, there might be different transactions expected even for the same collaborations.

For examples of such business process models, please see the ECIMF-POC document.

2.4.2.2 Creating the Mediation model

(NOTE: describe how the process mediation model can be created, using concepts from the Mediation meta-model.)

(NOTE2: the relationship to eBTWG BOT's [Business Object Types] need to be analyzed. BOT's define not only the class (+properties), but also the behavior, state and methods. As such, they are the best candidates to provide the intermediate internal model, and the problem of process mediation could be reduced to the problem of reconciling the state diagrams of the key BOT's. Please see the analysis in PowerPoint slides at http://www.ecimf.org/events/Brussels-20020220/ECIMF-eBTWG.ppt).

Below is a table that summarizes this step of the guideline:

Business Process Mediation	
Input	Business Context models, other information on business processes supporting the business context, semantics of the business processes (obtained in the next step), etc.
Output	Business Process Models, Business Process Mediation Model for the integration scenario, defined in a set of diagrams (activity/business process, ECIMF process mediation diagram)

2.4.3 Creating the Semantic Translation Model

2.4.3.1 Acquiring the source ontologies

(NOTE: describe the process of discovering the ontologies from e-commerce standards, best practices, business rules etc…)

2.4.3.2 Selection of the key concepts

(NOTE: describe how the business context and business process models help to determine the key concepts …)

2.4.3.3 Creating the mapping rules

(NOTE: describe how the mapping rules can be created, based on one of the alternative procedures …)

Below is a table that summarizes this step of the guideline:

Semantic Translation	
Input	Two source ontologies, obtained from formal specifications, UML models, textual descriptions, knowledge of domain experts etc.
Output	Semantic Translation Model, containing rules for equivalence of the key concepts.

2.4.4 Creating Syntax Mapping Model

2.4.4.1 Data element mapping

(NOTE: describe how the external formats can be mapped to internal representation …)

2.4.4.2 Message format mapping

(NOTE: describe how the message well-formedness rules can be satisfied. This may involve proactive "asking" for more information in order to satisfy the demands of a given message format…)

2.4.4.3 Message packaging mapping

(NOTE: describe how the message packaging [encoding, charset, MIME, etc] can be aligned)

2.4.4.4 Transport protocol mapping

(NOTE: describe how the transport protocol parameters need to be defined.)

Below is a table that summarizes this step of the guideline:

Syntax Mapping	
Input	Semantic Translation Model, simple mapping of primitive data types, external resources to be used.
Output	Syntax Mapping Model, containing the exact mapping of data elements, message formats, packaging and transport protocols.

For additional details, and more information on alternative procedures available for each of these steps, please refer to the Annex.

2.5 References

[UMM]: *Unified Modeling Methodology*; UN/CEFACT TMWG N090R9.1; available from: UN/CEFACT TMWG. A copy of the draft can be also found at: http://www.ecimf.org/doc/other/TMWG_N090R9.1.zip

[ebCDDA]: *Core Components Discovery and Analysis*; ebXML, May 2001; available from: http://www.ebxml.org/specs/ebCDDA.PDF

[ccDRIV]: *Catalog of Context Drivers*; ebXML, May 2001; available from: http://www.ebxml.org/specs/ccDRIV.PDF

[CID52]: *Conceptual Navigation and Multiple Scale Narration in a Knowledge Manifold*; Ambjörn Naeve; KTH, 1999; available from: http://cid.nada.kth.se/sv/pdf/cid_52.pdf

[OB00]: *Ontology-Based Integration of Information — A Survey of Existing Approaches*; H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner; University of Bremen, 2000; available from: http://www.tzi.de/buster/papers/SURVEY.pdf

[SAGV00]: *Semantic Translation Based on Approximate Re-Classification*, Heiner Stuckenschmidt, Ubbo Visser; University of Bremen, 2000; available from: http:// www.tzi.de/buster/papers/sagv-00.pdf

[SW00]: *A Layered Approach to Information Modeling and Interoperability on the Web,* Sergey Melnik, Stefan Decker; Stanford University, 2000; available from: http://www-db.stanford.edu/~melnik/pub/sw00/sw00.pdf

3 Proof-of-concept – scenario analysis

3.1 Editor's note

Originally this section formed a separate document. There may be still some inconsistencies related to this fact.

3.2 Purpose and scope

This section presents a step-by-step example of how the ECIMF can be used to prepare a set of recipes for interoperability between two e-commerce partners.

In this scenario, one partner, referred to as a Customer, produces Hi-Fi equipment of various sorts, and needs to ship them to the merchants. The other partner, referred to as Shipping Agency, offers services of shipping goods.

The Customer uses RosettaNet Implementation Framework 2.0 (RNIF) as his e-commerce interface, whereas the Shipping Agency uses EDI (EDIFACT D99.A).

This example follows the steps outlined in the Frameworks Integration Guidelines (in General Methodology section).

3.3 Business Context Matching

In this step, two Business Context models are built and compared, in order to check whether they can match the expectations of the other business partner.

3.3.1 Creating the Business Context Models

The diagrams below have been built using REA modeling elements, here expressed as UML stereotypes.

(NOTE: they present only a subset of the full diagram! E.g. there should be a Resource:Payload and Resource:Labor which is transformed or used by the Events…)

Figure 1 presents the business context diagram for the shipping agency. Here are the key elements of that diagram:

- The agency expects the payment first, and only then delivers the service
- The roles of ShippingAgent and Cashier are split into two different entities (persons, divisions …)
- ShippingAgent and Cashier collaborate with each other in order to satisfy the business rules (payment needs to be fulfilled first, and only then the shipment takes place)
- Both ShippingAgent and Cashier collaborate with the Customer.

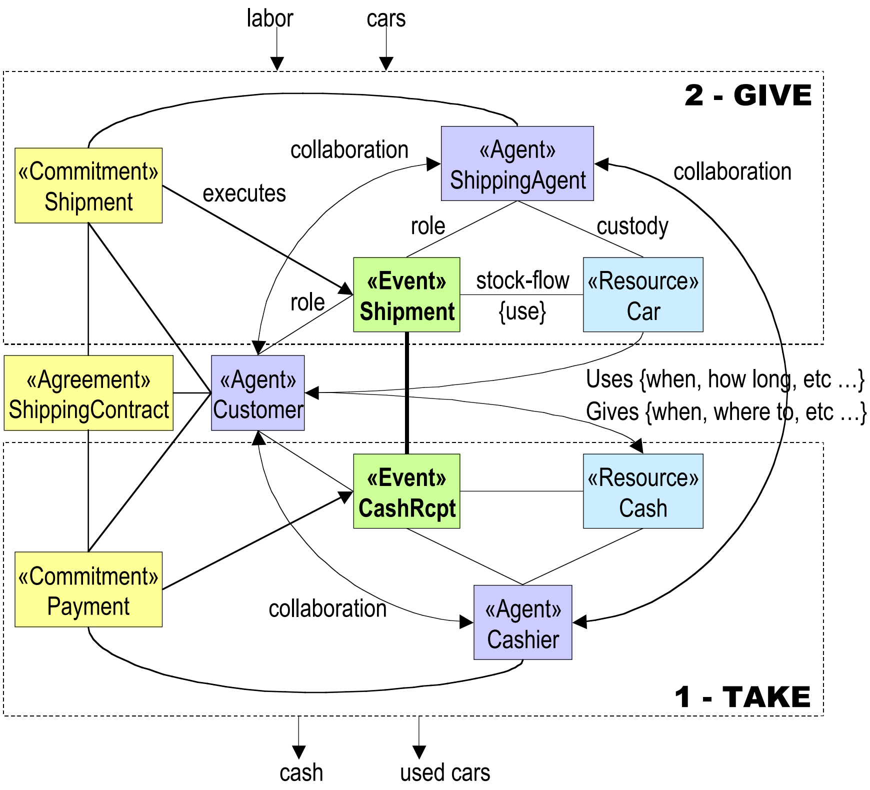


Figure 12 Business Context model as seen by the shipping agency.

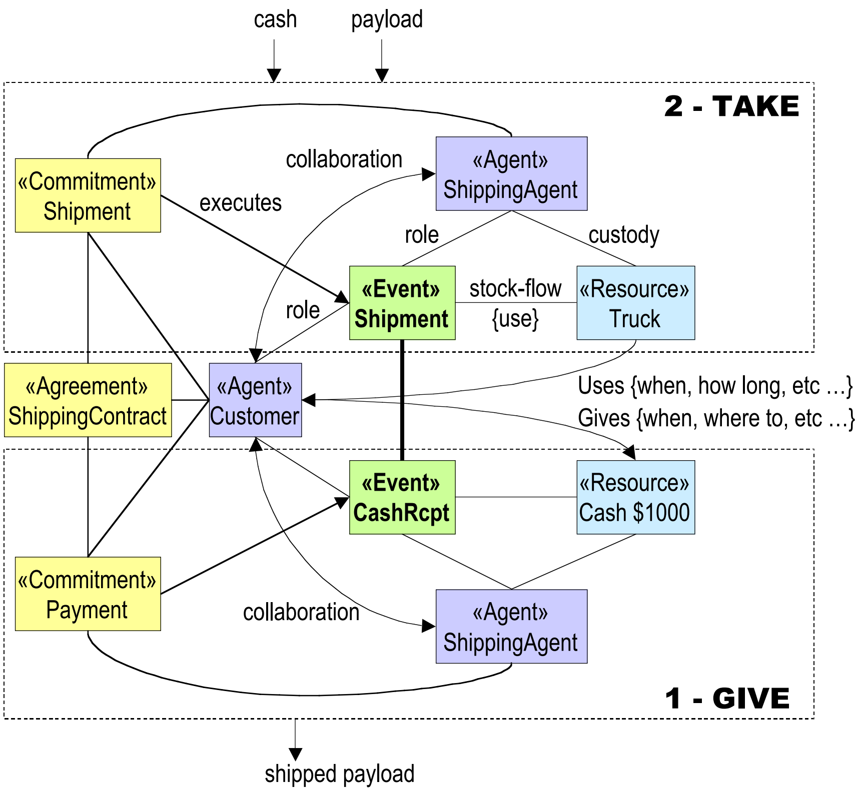


Figure 13 Business Context model as seen by the customer.

Now, for the customer the business context can be represented as shown on the next Figure. The key elements are:

- Customer expects first to give cash, then receive a service
- Customer wants to deal with the same entity for both events
- Customer has some specific demands on the kind of car, and the amount of cash.

3.3.2 Checking the Business Context Matching

From the diagrams above it is clear that in order for these two partners to be able to collaborate – in the traditional or in the electronic way – the following criteria have to be met (which ECIMF calls "business context matching rules"):

- *#1: Partners need to play complementary roles*: which is here the case. Note: although the Customer has a limited view of the Shipping Agency organizational structure (he wants to deal with just the ShippingAgent), it still has to be determined if he is able to deal with two separate persons/entities, which is required by the Shipping Agency (ShippingAgent and Cashier).
- *#2: Expected resources need to be equivalent*: in this case, parties need to agree on the exact kind of transportation used, and the exact amounts of money to be paid. They need to also agree on several additional properties of using the transportation (when, how long, from where, etc …) and providing the payment (when, where to, what currency etc…).
- *#3: Timing constraints need to be mutually satisfiable*: in this case, the Customer is able to satisfy the requirement of the Shipping Agency that he needs first to pay. Further timing constraints may show up when analyzing the collaboration patterns between the parties.
- *#4: Transaction boundaries need to be preserved*: in this case, there are two transactions: payment and shipment, possibly consisting of several lower-level technical transactions. All supporting communication between the partners needs to be aligned in such a way that it preserves these boundaries for each of them.

After additional negotiations, we can state that these two Business Contexts match. These additional requirements identified in this step need to be recorded.

(NOTE: how?)

For the sake of this example, we assume that both parties agreed to follow the model presented on the first Figure.

3.4 Process Mediation

3.4.1 Create Business Process models

Based on the Business Context models, we determined that the collaborations we are interested in are the following:

- **Payment collaboration task**: involving Customer and Cashier
- **Shipment collaboration task**: involving Customer and ShippingAgent.

Based on that, we should be able to identify concrete business processes existing within each organization, which support these collaborations. Also, it should be possible to identify the business transactions, which involve the electronic communication between the partners, and sending of electronic business documents.

3.4.1.1 Identify the Business Transactions

For all collaboration tasks we need to describe two sets of transactions, each according to the framework used by the Agent. As an example, we will analyze in detail the Payment Collaboration Task.

The following table contains the example list of business transactions, together with their business documents, identified for the Customer:

Party	Customer		
Collaboration Task	Payment Collaboration		
Framework	RNIF 2.0		
Transaction name	**Initiator / Responder**	**Request document**	**Response document**
PIP3A1: Request for quote	Initiator	QuoteRequest	QuoteConfirm
PIP3A4: Request Purchase Order	Initiator	PORequest	POConfirm
PIP3C3: Notify of Invoice	Responder	Invoice	
PIP3C6: Notify of remittance advice	*Initiator*	*RemittanceAdvice*	
Message delivery control	*Any*	*Secure Flow*	

In a similar manner, we identify the transactions for the Shipping Agency:

Party	ShippingAgency		
Collaboration Task	Payment Collaboration		
Framework	EDIFACT		
Transaction name	**Initiator / Responder**	**Request document**	**Response document**
Request for quote	Responder	REQUOTE	QUOTES
PIP3A4: Request Purchase Order	Responder	ORDERS	ORDRSP
Notify of Invoice	Initiator	INVOIC	
Notify of remittance advice	Responder	REMADV	
Message delivery control	*Any*	*APERAK, CONTRL*	

However, at this point we discover that the Customer's system doesn't implement the PIP3C6 – in the RosettaNet framework this is optional. We also discover that RosettaNet uses so called SecureFlows for communication control, whereas EDIFACT uses two messages: APERAK and CONTRL. Furthermore, we see that in EDIFACT framework use of these two messages is also sometimes optional. We need to further study their semantics – see the section on Semantic Translation.

It is useful also to picture these collaborations in a common diagram. This is presented on the Figure below. The business transactions are shown here also, as rounded boxes containing the business documents. These transactions change the states of each partner's Business Objects. Areas of potential problems are marked with red color.

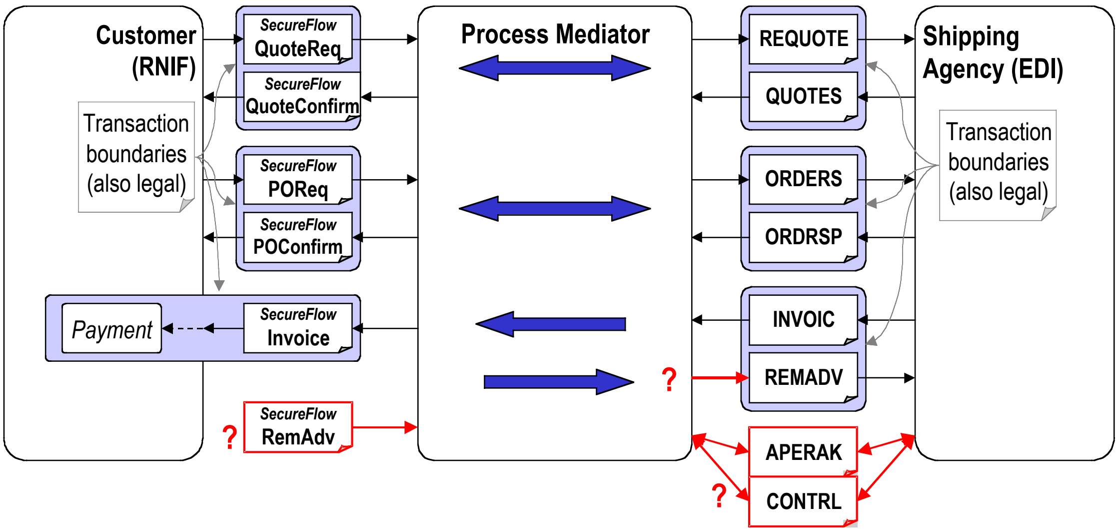


Figure 14 Process Mediation for the Payment Collaboration Task.

3.5 Semantic Translation

This step of integration helps to discover the underlying data model and the differences in meaning of the concepts used by each e-commerce framework. As it will be demonstrated, these differences will affect the design of both the process mediation and the syntax mapping.

For the sake of this example, let's assume that the customer wants to ship TV-sets from the factory to the shops.

This step will make use of the individual ontologies, a shared vocabulary and external resources in order to map between the key concepts in each of the frameworks.

Please note that generally the mappings are not symmetric, i.e. different rules and possibly different external resources need to be used when translating concepts from Customer to Shipping Agency than the other way around. For this reason, two sets of rules will always be present for each concept.

3.5.1 Acquire the source ontologies

For the purpose of this example, we acquired necessary concepts from each of the e-commerce frameworks – RNIF and EDIFACT respectively. We also made quite a few assumptions, which in the real case would have to be obtained from the particular IT system implementation, message implementation guidelines, product catalogues, company's procedures etc.

3.5.2 Select the key concepts

Let's start from the mapping of the two representations of a real-world entity (TV set), which is the subject of the shipment. These representations differ in each framework, because of their different scope.

This entity is represented in the ontology of the Customer as a TV-set – a kind of Hi-Fi equipment, while in the ontology of the Shipping Agency it is represented as a Box – a kind of Payload.

3.5.3 Create the mapping rules

The table below presents the semantics of the two corresponding concepts – TV-set in the Customer ontology, and Box in the Shipping Agency ontology – and the mappings required between the two representations, whenever they occur in the business documents.

Customer: TV-set	**Semantic Translation**	**Shipping Agency: Box**
Properties	**Mapping Rules**	**Properties**

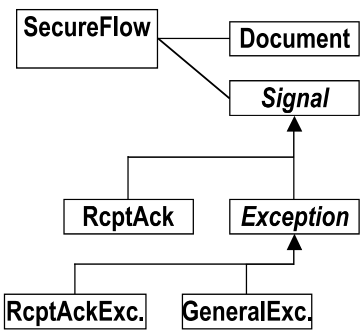
`Height`
`Width`
`Depth`
Represent the physical dimensions of the TV set chassis.
	Tv_set _ Box: dimension values will always be higher, but discrete. Need to be obtained from a cardboard box catalogue (external resource)
	Box _ Tv_set: dimension values will always be lower. Need to be obtained from a TV products catalogue (external resource) using `productID`
		`Height`
		`Width`
		`Depth`
		Represent the physical dimensions of the cardboard box used to ship the electronic equipment of any kind. The values are discrete, because only certain box sizes are available.

Not available (N/A)	Tv_set _ Box: needs to be obtained from a product catalogue (external resource) Box _ Tv_set: not needed	`Weight` Represents the weight of the box with the contents.
N/A	Tv_set _ Box: needs to be obtained from a product catalogue (external resource) Box _ Tv_set: not needed	`StackingLevels` Represents the number of levels the boxes can be stacked, one on top of the other.
N/A	Tv_set _ Box: always set to True. Box _ Tv_set: not needed	`Fragile` Marks the payload as fragile (requiring special care during transportation)
`Color`	Tv_set _ Box: not needed Box _ Tv_set: needs to be obtained from a product catalogue (external resource)	N/A
`Stereo`	Tv_set _ Box: not needed Box _ Tv_set: needs to be obtained from a product catalogue (external resource)	N/A
`UnitPrice`	Tv_set _ Box: not needed Box _ Tv_set: needs to be obtained from a product catalogue (external resource)	N/A
`ProductID` Product identification (type)	Tv_set _ Box: concatenate with the `serialNo` Box _ Tv_set: split into `ProductID` and `serialNo`, based on a required `serialNo` length.	**`ProductID`** **Product identification (type), including serial number. Primary identification data**
`SerialNo` **Serial number. Primary identification data**	Tv_set _ Box: see rule above Box _ Tv_set: see rule above	N/A

There are several interesting observations that can be made based on this example:

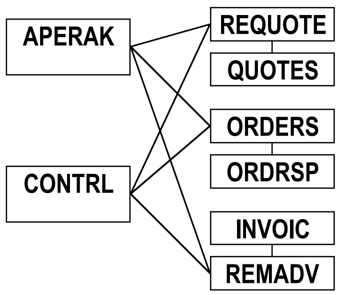
- Several external resources need to be consulted in order to prepare the mapping. It is possible to record the fixed values in the translation rules, but it would be more flexible to be able to query these resources dynamically, during run time.
- However, some of the values can be specified explicitly in the rules, and have fixed value (e.g. the `fragile` Box property).
- The translation rules are definitely not symmetric – e.g. different external resources may need to be consulted in order to supply missing data.
- There is a property, which uniquely identifies the corresponding physical entity (`Tv_set.serialNo` and `Box.productID`), although it is defined differently and requires processing.
- The properties related to physical dimensions are confusingly homonymous, although in reality their relationship is governed by a complex formula (and requires use of external resources).

Before proceeding to the last step (syntax mapping), let’s analyze the message delivery control mechanisms, as these were identified as problematic during the process mediation step.

<table>
<tr><th>Customer (RNIF)</th><th>Semantic Translation</th><th>Shipping Agency (EDI)</th></tr>
<tr><td>

<code>SecureFlow</code> consists of a business document (containing business data), and a responding business signal (acknowledgement).</td><td>The RNIF business documents map 1:1 to EDI business messages, e.g.:

QuoteRequest _ REQUOTE
QuoteConfirm _ QUOTES
PORequest _ ORDERS
POConfirm _ ORDRSP
etc ...

However, individual data elements can be missing, and will have to be collected from the previous messages, or supplied explicitly in the rules, or obtained from external resources.</td><td>

In this particular case, the EDI system uses <code>APERAK</code> and <code>CONTRL</code> messages only to signal exceptions. Acknowledgements are implicit, in the form of response business documents.</td></tr>
<tr><td rowspan="2"><code>ReceiptAck</code>
This signal means that the document business data has been accepted for further processing (which implies also well-formedness)</td><td>RNIF _ EDI: not needed – don’t forward.</td><td rowspan="2">N/A – implementation choice (positive acknowledgements are implicit).</td></tr>
<tr><td>EDI _ RNIF: needs to be synthesized from the response document. Possible problems with timing constraints… (ack. too late)</td></tr>
<tr><td><code>ReceiptAckException</code>
This signal means the document was not well-formed (parsing errors). Business data was not considered at all.</td><td>The semantics of both messages is identical, which means a 1:1 mapping can be applied, both ways.</td><td><code>CONTRL</code>
This message is sent when parsing errors occur. Business data was not considered at all.</td></tr>
<tr><td><code>GeneralException</code>
This signal means that there were errors in the business data</td><td>RNIF _ EDI: always map to <code>APERAK</code></td><td><code>APERAK</code>
In this implementation, this message is sent only when an</td></tr>
</table>

Customer (RNIF)	Semantic Translation	Shipping Agency (EDI)
processing (though it means implicitly the document was well-formed).	EDI _ RNIF: map only if the APERAK message carries an error status.	error occurs when processing business data (though it means implicitly the document was well-formed).

Again, this analysis brings a couple of interesting observations:

- The differences in the semantics of message flow control mechanisms will affect the implementation of the process mediator, because some messages need to be created, removed, or sent at different times than the originating messages. Conclusion: there is no simple 1:1 mapping between messages, and the process mediator is really needed.
- The business documents map 1:1 in this example. However, as shown on the Figure 3, the RNIF side doesn't produce the `RemittanceAdvice` message, which the EDI side needs for completion of the low-level transaction. This message needs to be either synthesized by the process mediator (by accessing an external resource, such as the payee's bank), or the RNIF side needs to implement it.
- The timing constraints for `ReceiptAck` (times defined in RNIF, which define how long the sender has to wait for an acknowledgement before concluding a failure) may be impossible to satisfy in this scenario. The EDI side doesn't produce required `ReceiptAck` signals, and they need to be created based on the response EDI messages – which may be sent too late to satisfy the timing limits defined in RNIF.

After completing this step, we are very well prepared to define the low-level syntax mapping – transformation of the data elements in individual messages.

3.6 Syntax mapping

According to the layered ECIMF model, the syntax mapping – i.e. the translation between the individual data elements – is the lowest layer of interoperability, and it is affected by the rules defined in all the higher layers.

Let's take for example a fragment of mapping between the `PurchaseOrderRequest` and `ORDERS`. Figure 5 shows the fragments of each message and the mapping links between the data elements.

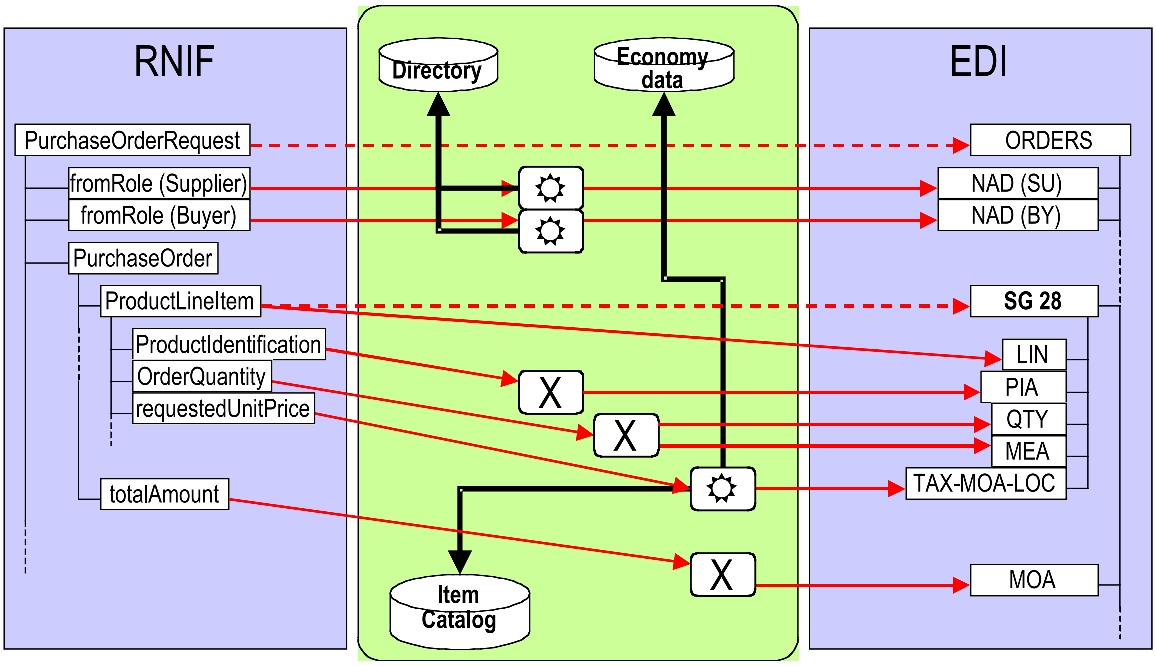


Figure 15 Message syntax mapping.

Again, a few observations can be made based on this example:

- This is a one-way mapping, as the arrows on the red links indicate. This means, that this mapping is valid for translation of PurchaseOrderRequest messages into ORDERS messages, and not necessarily vice versa (in fact, in our example different external resources will be needed to perform the translation in the other direction).
- The dashed lines represent the instance links, i.e. for each instance on one side a corresponding instance on the other side is created. In this case, for one `PurchaseOrderRequest` document one `ORDERS` message is created, and similarly for one `ProductLineItem` one Segment Group 28 (`SG28`) is created. Note, however, that additional limitations need to be considered here, which come from the limitations on the allowed number of the given data elements in a message. In this case, there can be no more than 200000 (according to EDIFACT D99.A) occurrences of `SG28` in a single `ORDERS` message. If there are more `ProductLineItems` than that, they probably need to be divided into two `ORDERS` messages – however, this changes significantly the flow of the low-level transactions, as presented on the Figure 3.
- The boxes with a toothed wheel represent complex processing, with the use of external resources. This is needed e.g. if the identification schemas for parties are different, or in the above-mentioned example of different product classifications.
- The boxes with an "X" represent simple data transformation, like numeric or string operations. E.g. as identified in the Semantic Translation step, the product ID used in EDI (`PIA` element) needs to be a concatenation of the sub-elements of the `ProductIdentification` element in RNIF.

In this step also the differences in the transport protocols and packaging are considered. Some differences (like use of FTP vs. SOAP) will require providing additional protocol parameters, e.g. FTP username and password, SOAP service name, a WSDL file, details of the MIME packaging etc. Some of these parameters can be expressed using ebXML CPP/CPA.

3.7 Generation of MANIFEST

As the final step, based on the models and transformation rules prepared in the steps above, a MANIFEST needs to be generated - an abstract recipe for interoperability between RNIF and EDI, within the given scope.

The example syntax of the MANIFEST document could look like the sample below:

```
<?xml version='1.0'?>
<Manifest>
   <BusinessContextMatching name='Shipment'>
      <BusinessContext id='WidgetsLtd'> ... </BusinessContext>
      <BusinessContext id='JoeShipping'> ... </BusinessContext>
   </BusinessContextMatching>
   <ProcessMediation>
      <Framework id='RNIF' name='WidgetsLtd'>
         <BusinessProcessDefinition location='uddi:...PIP3A4...'/>
         <BusinessProcessDefinition location='uddi:...'/>
         <BusinessProcessDefinition location='uddi:...'/>
      </Framework>
      <Framework id='EDI' name='JoeShipping'>
         <BusinessProcessDefinition>
            ... (here it follows, defined using ebXML BPSS)...
         </BusinessProcessDefinition>
      </Framework>
      <MediationRules>
         ...
      </MediationRules>
   </ProcessMediation>
   <SemanticTranslation>
      <OntologyRef id='RNIF'>urn:ont1 ...</OntologyRef>
      <OntologyRef id='EDI'>urn:ont1 ...</OntologyRef>
      <Rule id='rule1'>
         <SourceCtxSet id='set1'/>
         <TargetCtxSet id='set2'/>
         <formula id='formula1'/>
         <formula id='formula2'/>
      </Rule>
      <ContextSet id='set1'><context id='ctx1'/></ContextSet>
      <ContextSet id='set2'><context id='ctx2'/></ContextSet>
      <Context id='ctx1'>
         <ConceptRef id='tv_set'>urn:...TV-set</ConceptRef>
      </Context>
      <Context id='ctx2'>
         <ConceptRef id='box'>urn:...Box</ConceptRef>
      </Context>
      <Formula id='formula1'>
         <body>
set2.ctx2.box.productID := set1.ctx1.tv_set.productID +
   " " + set1.ctx1.tv_set.serialNo;
         </body>
      </Formula>
      <Formula id='formula1'>
         <body>
set2.ctx2.box.fragile := true;
```

```
         </body>
      </Formula>
   </SemanticTranslation>
   <SyntaxMapping>
         <Mapping>
            <SourceMessage>PurchaseOrderRequest</SourceMessage>
            <TargetMessage>ORDERS</TargetMessage>
            <Rules>
            </Rules>
         </Mapping>
         ...
   </SyntaxMapping>
</Manifest>
```

(This example uses the Semantic Translation ontology, developed for the purpose of this project – see http://www.ecimf.org/contrib/onto/ST/index.html for more details).

Note that for the purpose of configuring the ECIMF-compliant runtime, only the process mediation and syntax translation rules are needed. However, the models of the two other layers are included as well in order to facilitate exchange of the ECIMF models between the modeling tools, and to preserve the knowledge collected during the process of mapping.

In the next step, as presented previously in the Figure 5, the ECIMF-compliant agent receives the MANIFEST and instantiates the necessary adapters. This may involve setting up processing pipelines for messages, creating state machines to keep track of complex interactions, creating translation maps for message elements, reading parameters provided by the communicating parties, etc. This reference environment for execution of the MANIFEST recipe can be provided as a commercial product.

Finally, at this stage it is possible for the parties to successfully establish business interaction, even though they use different e-commerce frameworks to express their activities.

3.8 Implementation: ECIML-compliant agent

(This section is incomplete. Please see Annex 2 for some initial materials)

4 ECIMF Toolkit – description

4.1 Introduction

This software module has been created to illustrate and investigate various methodologies for concept mapping and alignment between different e-commerce standards. These standard e-commerce frameworks are represented as ontologies - shared conceptualizations of a given problem domain as seen by their respective user communities.

In our project we decided to follow a semantic translation approach, which uses an upper-level shared ontology that provides concepts-labels to identify similar concepts in each respective ontology. This approach means that instead of building a full-mesh N*(N-1) collection of translations for each pair of existing e-commerce frameworks, it is sufficient to prepare N translations from that framework by attaching conceptual labels taken from this shared ontology.

Under this approach, the following steps need to be performed:

- Attach labels taken from shared ontology to your concepts
- Find corresponding labels in the foreign ontology
- Apply more steps to refine the relationships:
    - Local context
    - Automated, formal reasoning and inference
    - External context – semantic enrichment
    - Heuristics (best practice and rule of thumb J)
- Define the translation rules in a formal way

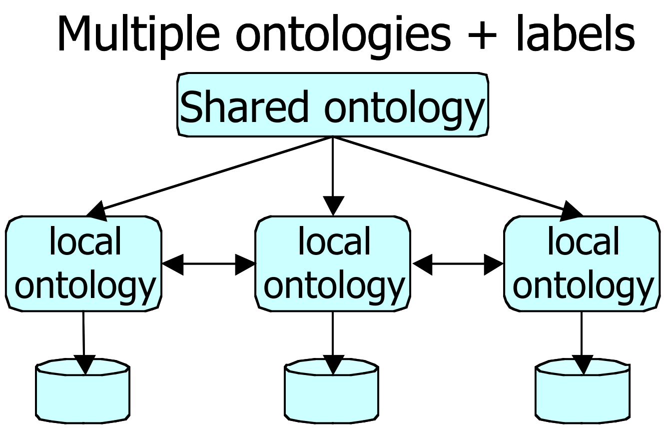


Figure 16 Shared ontology approach to semantic translation.

This software is provided under the terms of Mozilla Public License (see Mozilla site for details).

4.2 Limitations

Originally, this tool was intended as a more or less complete implementation of various modules of ECIMF-compliant agent, as described in the section on project goals. However, due to unexpected shortage of human resources, only the alpha-quality version of semantic translation module has been implemented. So, currently the tool is very limited in its scope and functionality:

- the ontologies that represent e-commerce standards need to be supplied in Protege .pprj format. This usually means that you have to convert them first from some other format, using either Protege built-in import modules or enter the concepts manually... There is a simple EDIFACT import module under development, as well as DTD/XSD import modules.
- the tool currently supports mapping through labeling. It is theoretically possible to use it for other mapping methodologies by using the scripting capabilities, but it would be inconvenient.
- the custom search script function is not supported yet, although the amount of work needed to complete it is small.
- there are numerous layout problems.
- other limitations exist, to be sure...

4.3 Simple usage scenario

First of all, users are strongly advised to first read the introductory material in the presentation http://www.ecimf.org/events/Paris-20020610/Interoperability.ppt - it helps to understand the principles behind the methodology implemented in this tool.

Let's step through a scenario, in which you will map the concepts in the included sample projects:

1. **Download, install, and start the tool.** The exact steps will depend on your platform - under Windows, when the installation process is completed, there will be a new item in your Start/Programs list called ECIMF-ST.
When the tool is started, a console window will also appear, where you can find all sorts of useful debug information.
2. **Labeling:** in this step, you will attach labels to the concepts in each of the individual ontologies
    - Press LOAD button to load SOURCE project. A file selection dialog will open. Go to projects/ subdirectory and select source.pprj project.
    - In a similar way, load the LABELS project from labels.pprj.
    - Highlight one of the concepts in the SOURCE project. The bottom-right panel will show you the details of the concept, including a list of labels attached to it.
    (NOTE: you may want to resize the main window and/or individual panels by dragging the dividers between the panels)
    - Modify the list of labels by creating ("C" button) a label from scratch, attaching ("+" button) a label from the LABELS ontology, editing ("E" button) or removing ("-" button) a label. You will be mostly interested in using the "+" and "-" buttons.

Note that individual concepts can have multiple labels, each of them possibly characterising the concept in a different way. The labeling process helps to fix the SOURCE concept in the conceptual topology that can be described by the LABELS concepts. In other words, the more specific labels are attached to the SOURCE concept, the less ambiguous its definition is according to the LABELS ontology.

- Follow similar steps, but with the TARGET project.

3. **Mapping:** in this step, you will create a formula that describes a mapping between one of the SOURCE concepts and the TARGET concepts:

   - Select the "Mapping" tab. Note that the layout here is different - on the left the SOURCE project is presented, on the right there is the TARGET project, and in the middle you can see the panel where the mapping hints will be presented. You can access also the full MAP project if you want to browse the formulas.
   - Select one of the concepts in the SOURCE project.
   - Press the "Find in TARGET" button to show the hints.
     Note: the "Conf" button shows the various possible algorithms for finding the corresponding labels. Currently, the scripting is not implemented here, but please take a look at various possibilities of searching and matching...
   - If some hints are found, you can select them for use in a formula by checking the "Use?" checkboxes.
   - Create a new formula by clicking on "Create" button. You can also change the name of the formula.
   - A pop-up dialog will appear that lets you edit the formula in your favorite scripting language.
     This panel also shows you what kind of data sources are available to you in this context. A special name called "SOURCE" refers always to the input concept that you selected for mapping. Also, the target concepts that were found are available under their names.
     The properties of each concept are available directly as instance variables, so you can e.g. use a notation "SOURCE.name" to refer to the input concept name.
   - Press OK to save the formula to the MAP project.

   You can review the mapping formulas by clicking on a "Map" tab in the middle panel. Then select the "Formula" concept, and in the bottom-left panel select one of the instances of formulas.

   (Note: currently a layout problem prevents you from viewing the formula body).

4.4 Additional information

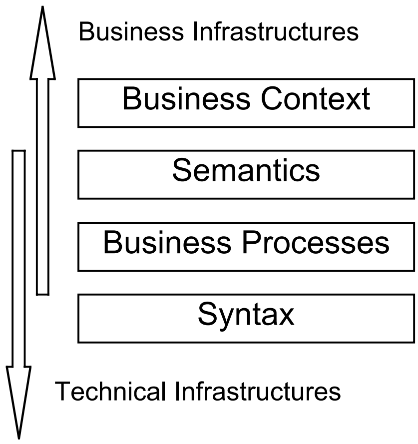
Additional information about the tool can be obtained from the author (Andrzej Bialecki ab@getopt.org). You may also check the PowerPoint slides here: http://www.ecimf.org/events/Paris-20020610/ECIMFToolkit.ppt).

The source code for the tool is included in the installation package downloadable from http://www.ecimf.org/software .

5 Summary and conclusions

The ECIMF project made an attempt to address the interoperability problems by providing a single general and holistic view of all major aspects involved in solving concrete integration scenarios between e-commerce partners.

The most important outcome of the project seems to be the 4-aspect model of interoperability:



We have investigated various existing approaches that address each of these areas, and tried to indicate which of them need further research. Based on this, we present the following conclusions and suggestions for future work.

5.1 Interoperability of Business Contexts

REA models (retro-fitted to virtual organizations) help to understand interoperability issues on the value-chain level. This is because they provide a formal framework to describe contractual commitments and their relationship to partners' collaborations, transactions and processes. They also help to identify differences in local business context.

Recently, REA Enterprise Modeling Framework has been adopted as a central part of business models in ebXML.

5.2 Semantic Interoperability

Today there are islands of well-defined semantics for use in e-commerce, such as universal classification schemas (EAN/UCC, UNSPSC …) and standard e-commerce frameworks (RosettaNet, OAGIS, ebXML, xCBL …).

But there is no generally available, overall and unified business semantics across existing standards. Similar business concepts are being expressed differently, using different semantic depth, which results in ambiguous and overlapping concepts when considered in an integration scenario. This in turn leads to drastic increase in complexity and cost of integration. This also prevents ad-hoc collaboration scenarios between partners using different e-commerce frameworks. Well-established older standards will linger, so that this aspect of integration will not go away any time soon.

The ECIMF project group has identified the need for better and more effective methods for semantic mapping. Some of the most promising methods use upper-level shared ontologies – however, there is no such common unified ontology available at the moment. Readers are encouraged to review Annex 3, where this problem is discussed in depth.

Some of the existing projects are working intensively in this area, specifically:

- ISO TC/154 Basic Semantic Register: provides a cross-linked reference to key concepts across several existing e-commerce standards.
- ECIMF Semantic Mapping Tool: provides a prototype tool to facilitate semantic translation process, with use of shared ontology.
- OntoWeb projects: several projects, e.g. on ontology-based integration of content standards (SIG1), and industrial applications of ontologies (SIG4)

And other similar projects. However, there is still much to be done before the average e-commerce user begins to benefit from this work. The ECIMF project clearly identifies this issue as a fundamental integration problem, and recommends both further basic research into efficient methods of semantic mapping, and a development of upper-level shared e-commerce ontology for the purpose of such mapping.

5.3 Interoperability of Business Processes

The ECIMF project has identified the need to reconcile incompatible definitions of business processes, as specified by different e-commerce frameworks.

Although good and comprehensive models for business process modeling exist (e.g. the one developed by UN/CEFACT ebXML project), there is little or no work being done on process mediation *across* standards. This is a very complex and non-obvious issue, which involves elements like transaction preservation, observing the timing constraints, compensation for failed transactions, legal consequences of failed transactions, partial fulfillment and others.

The ECIMF project suggests that a separate, well-defined module (named Process Mediator) should be responsible for addressing these issues. Initial requirements and suggestions for possible architectures have been presented.

Currently, the project members are aware of just one research project, which tries to address this integration aspect in a systematic way. It is the Process Broker project at Swedish Royal Institute of Technology (http://www.dsv.su.se/~pajo/processbroker/index.html), led by prof. Paul Johannesson.

5.4 Syntactic interoperability

The issue of syntax mapping is the most common aspect of interoperability being addressed today by software vendors. There are many existing software suites which concentrate mainly on this aspect, while offering only very limited functionality in all other integration aspects, as identified above.

Unfortunately, as the ECIMF project concludes, interoperability of message formats and transport protocols is also the last issue to be addressed when implementing integration solutions, and probably the most straightforward – that is, as soon as all other constraints (semantic and dynamic) are well understood. This low-level mapping quickly becomes very complex and difficult to maintain, if it is not driven by underlying higher-level models.

Therefore we recommend that vendors of integration software suites should concentrate on development of model-driven tools for system integration, taking into account the high-level e-commerce models being developed by recognized standard bodies and industry forums (such as UN/CEFACT ebXML, RosettaNet, OMG, OAG, UBL and others).

6 Acknowledgments

The ECIMF project group gratefully acknowledges valuable input and support received from the following individuals and organizations:

- CEN/ISSS WS-EC members
- Bob Haugen, Logistical Software LLC
- William McCarthy, Michigan State University
- Ambjörn Naeve, Mikael Nilsson and Matthias Palmér from Royal Institute of Technology (KTH), Centre for User Oriented IT Design (CID / NADA), Sweden
- Paul Johannesson from IT University, Department of Computer and Systems Sciences, Sweden
- WebGiro AB, Sweden