E-Commerce Frameworks Integration Guideline

Draft 0.2
August 6, 2001

Introduction

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 document describes an experimental step-by-step guideline to solving this issue in case of two incompatible e-commerce frameworks F1 and F2.

This scenario has been prepared based on the research in the following areas:

- 3-layer integration methodology proposed in [ECIMF-GM]
- UN/CEFACT Unified Modeling Methodology [UMM]
- ebXML Core Components Discovery and Analysis [ebCDDA]
- ebXML Catalog of Context Drivers [ccDRIV]
- principles of conceptual navigation, presented in [CID52]
- semantic translation approach based on approximate context transformation, presented in [SAGV00]
- survey of ontology-based approaches to information integration [OB00]
- layered approach to information modeling [SW00]

The guideline has been divided into several steps, to be performed sequentially and iteratively, as needed. The result of their successful completion will be a set of interoperability rules, which allows parties using different frameworks to cooperate towards common business goals.

These steps are listed below, and explained in detail in further sections:

- **Initialization**: this stage 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 stage roughly corresponds to the Inception phase in [UMM].
- **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 F1 and F2 (which is also known as "semantic calibration" [CID52]).
- **Process mediation**: in this step the necessary mediation logic is defined, by introducing an intermediary agent that can transform conversation flow from F1 to that expected by F2.
  These two steps (Semantic translation and process mediation) can be documented in a way prescribed for the Business Modeling and Analysis phases of [UMM]
- **Syntax translation**: 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 protocol and packaging translation is specified.
  This final step corresponds to the Design phase in [UMM]

The value of this document is a work-in-progress research, for discussion among project members – no final conclusions should be drawn from it. Please provide your comments and ideas to the project participants, or directly to Andrzej Bialecki (abial@webgiro.com).

NOTE: when preparing this guideline I came to conclusion that this process needs to be further analyzed, as I couldn't definitely pinpoint from which layer to start. Generally, there are two scenarios, which seem to make a lot of sense, but each with some serious problem:

1. *Scenario presented in the following sections: start with Semantics, then Dynamics and Syntax. But there is a bootstrapping stage required (called here Initialization) which defines the scope of business processes and the messages exchanged between parties, and this fits better in the Dynamics layer…*
2. *So, the other scenario would be to start from Dynamics (merge the Initialization and Process Mediation), then Semantics and then Syntax. But we need to know the meaning of the data that messages contain, otherwise we can't say for sure whether this or that difference is significant…*

We are investigating now possibility to use REA to identify the similar business events as an introductory phase to the scenario 2. above.

1. Initialization

- **Define the scope**: select the business processes from Framework 1 (F1) and Framework 2 (F2), which you want to integrate. This step may need to be repeated if more business processes are involved than what it initially seemed. Let's represent these processes as follows:

  $$F_1(B_x) \rightarrow F_2(B_y, B_z, \ldots)$$

  The arrow means that there is a relationship between a business process B_x in framework F_1 and processes B_y, B_z, … in framework F_2. This relationship in general is probably not symmetric, meaning that from the above we cannot conclude that

  $$F_2(B_y, B_z, \ldots) \rightarrow F_1(B_x)$$

  in the same way.
- **Identify the messages** exchanged in conversations between parties participating in these business processes:

  $$F_1(B_x(M_1, M_2, \ldots, M_i)), F_2(B_y(M_1, M_2, \ldots, M_i), \ldots)$$

  This step will be further elaborated in the Process Mediation section.
  (NOTE: the BOV model [UMM], if available, is a good point of start here).

2. Semantic translation

- **Identify the key concepts** in use for message exchanges conducted according to each framework, within the context of the selected business processes:
  - **For each message** in B_i identify the key indispensable information elements that decide about the success of the information exchange from the business point of view in each of the frameworks:

    $$M_i(E_1, E_2, \ldots, E_n)$$
  - **For each message** M_i in B_i, based on the framework model, identify the key concepts that these information elements represent. In terms of OO and UML modeling, use the information collected in the previous step to build an object diagram, where instances of classes represent the key concepts (perhaps already identified in the

formal framework description) and properties take the values from the message elements:

$$M_i(C_1(E_1, E_2, \ldots), C_2(E_m, E_n, \ldots), \ldots, C_n(E_x, E_y, \ldots))$$

This notation means that each message M_i contains a set of key concepts (classes) – information elements, which decide the meaning of the message.

- **Collect the key concepts** in a unique set:

$$F_1(C_1, C_2, \ldots, C_n, \ldots, C_x, \ldots, C_z)$$

(NOTE 1: should we at this stage suggest building an abstracted conceptual model of separately F1 and F2?)
(NOTE 2: this step corresponds to the process of building conceptual topology of frameworks F1 and F2, which are sets of conceptual neighborhoods [CID52]).

- **Collect more semantic data** about each concept, as expressed by each framework's specifications, in a form of properties and constraints:

$$C_i(p_1, p_2, \ldots, p_m, c_1, c_2, \ldots, c_x)$$

We introduce the notation P_i to denote a property with its accompanying constraints. Therefore we may express the above as follows:

$$C_i(P_1, P_2, \ldots, P_m, c_n, \ldots, c_x)$$

These additional semantic data will probably point to some obvious generalizations, which in turn may lead to reduction of the set of unique concepts.
(NOTE 1: The steps detailed above lead to creation of framework ontologies – or, in the language of [UMM], Lexicons with core components. Similarly, the process described below corresponds to finding a translation between ontologies [OB00] – although, since the ontologies are built from scratch here, the approach to use shared vocabulary may provide useful reduction in complexity (cf. [OB00]). The latter approach is similar to the process described in [ebCDDA] for discovery of domain components and context drivers).
(NOTE 2: the Business Operational View [UMM] model of the frameworks, if available, is a very appropriate source for this kind of information)
(NOTE 3: two concepts $F_1(C_x)$ and $F_2(C_y)$ may in fact represent one real entity – however, due to the different contexts in which they are described they may appear to be non-equal. Such cases will be resolved in the following steps)

- **Generate hypotheses about corresponding concepts** in the other framework:
  - Concepts are likely to correspond if they:
    - have similar properties
    - are similarly classified
    - play similar roles (similar relationships with other concepts, occur in similar contexts)
- **Test each hypothesis:**
  - Check the constraints on the properties, describe the differences in property specifications (such as scale, allowed values, code lists, classification) and check the correctness of classification based on the following criteria:
    - The **necessary conditions** for concept $F_i(C_x)$ is set of values/ranges of some of its properties that are true for all instances of that concept. Therefore, if a concept C_y doesn't display them, it cannot be classified as C_x. Necessary conditions help to rule out false correspondence hypotheses.
    - The **sufficient conditions** for concept $F_i(C_x)$ is a set of properties and constraints, when met automatically determine the concept classification. Sufficient conditions help us to identify the concepts that surely correspond because they show all sufficient conditions.

Example: "TV-set" meets sufficient conditions for being a "house appliance". However, it fails to meet the necessary conditions for a "cleaning house appliance".

- **Approximate classification**: if the above steps result in well-defined rules of correspondence for most cases of the observed concept occurrence, the hypothesis can be considered basically true. It is probably not feasible to strive for exact solution in 100% cases – we may allow certain exceptions. There are several ways to determine the level of proximity:
  - **Rough classification:** the concept definition can be treated as having its upper and lower bounds. The upper bound (the most precise) is necessary conditions, and the lower bound (the most general) is the sufficient conditions. We may declare that $F_1(C_x) \rightarrow F_2(C_y)$ even when necessary conditions are not met, but sufficient ones are.
  - **Probabilistic classification:** we can determine (based on e.g. available pre-classified data sets) the significance of each property on the result of classification, and so calculate the probability of entity belonging to a specific class.
  - **Fuzzy classification:** for each property we define a fuzzy rule, which describes the level of similarity of the tested property. Then, the best match is defined when maximum number of rules gives positive results.
- **Other hypotheses**: if the hypothesis cannot be proven with a sufficient degree of certainty, other hypotheses need to be formulated and tested.
- **Possible difficulties** that may arise:
  - There is **no corresponding** concept: may be there are too many unknown properties to determine the corresponding concept in F_2, because in the context of F_1 they were irrelevant. In this case, the information required to find $F_2(M_x(C_y))$ needs to be supplied from elsewhere, based on properties of the real entities that $F_1(M_i(C_j))$ and $F_2(M_x(C_y))$ refer to - we need to provide more semantics about the concepts than what is found in the framework specifications (usually from a human expert).
  - There are **many corresponding** concepts, depending on which property we choose: we could arbitrarily choose the one that plays the most vital role from the business point of view – and choose which properties decide that an instance of a concept in F1 could be classified as an instance of corresponding concept in F2:

    $$F_1(C_x(P_i)) \rightarrow F_2(C_y(P_j))$$

    See also the section above on probabilistic classification.
  - The **conflicts in property** constraints cannot be easily resolved. This case calls for help from the domain expert.

- **Describe the rules and exceptions** (if any), and in what contexts they occur.
  (NOTE: how to describe the exceptions? Well, for that matter, how to describe the rules? ☺)
  (NOTE 2: there are three ways to address this problem, according to [OB00]:
  - *Create a single global ontology, which will include concepts from both frameworks. Not feasible for even moderately complex cases.*
  - *Create mappings between concepts in ontologies (this is the approach suggested above, although [OB00] warns again that it leads to very complex mappings)*
  - *Using shared vocabulary, re-build the ontologies from scratch – the result will be somewhat automatically aligned. Then, prepare the translation rules, which should be now much simpler.)*

3. Process mediation

- **Analyze process specifications**:
    - **For each** business process in each framework:
        - Identify **request and response messages**. We suggest also building a more complete diagram containing two activity diagrams: one for requesting party, other for responding party. The diagram should also contain the messages passed between the parties.
          (NOTE: this step will benefit from information collected in BOV and FSV models, if available (cf. [UMM]))
        - Determine **legal obligations** boundaries: which interactions and messages bring what legal and economical consequences.
        - Determine the **transaction boundaries**, rollback/compensation activities and messages for failed transactions
- **Identify differences** in message flow, by comparing message flow between requesting/responding parties in $F_1(B_x)$, and similarly for $F_2(B_y, B_z, \ldots)$:
    - **Missing messages/elements**: are those that are present in e.g. $F_1(B_x)$, but don't occur in $F_2(B_y, B_z, \ldots)$. This is also true about the individual data elements, which may become available only after certain steps in the conversations, different for each framework. These messages and data elements will have to be created by the mediator, based on already available data from various sources, such as:
        - previous messages
        - configuration parameters
        - external resources

      and sent according to the expected conversation pattern.
    - **Superfluous or misplaced messages/elements**: are those that don't correspond directly to any of the required/expected messages as specified in the other framework. Also, they may be required to arrive in different order. The mediator should collect them (for possible use of information elements they contain at some later stage) without sending them to the other party, or change the order in which they are sent.
    - **Different constraints** (time, transactional, legal…): this issue is similar in complexity to resolving the semantic conflicts (see below), and a similar approach could be taken.
      (NOTE: namely???)

4. Syntax translation (to be completed)

- **Message format** translation
    - **For each** data element E_i in M_i define the translation rules, based on the context of:
        - **Semantic differences**: identified in the Semantic Translation step
        - **Dynamic differences**: identified in the Process Mediation step
- **Message transport** translation
    - **Align packaging and transport** protocols, based on the specifications in each framework.
- *(to be continued…)*

References

[ECIMF-GM]: *E-Commerce Integration Meta-Framework, General Methodology*, CEN/ISSS/WS-EC project, 2001; available from: http://www.ecimf.org/doc/CWA/GM/ECIMF-GM.pdf

[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