

DYNAMIC MODEL HARMONIZATION BETWEEN UNKNOWN eBUSINESS SYSTEMS

Makoto Oya and Masumi Ito

*Graduate School of Information Science and Technology, Hokkaido University, N14, W9,
Kita-ku, Sapporo, 060-0814, Japan*

Abstract: It is usually assumed that systems share an agreed model of business protocol to implement consistent exchange of series of business messages between eBusiness processes. R&Ds based on this approach are making progress such as model repository technology, business process modeling and business process description languages. This assumption, however, is not always satisfied when considering flexible and free dealings among independent and autonomous eBusiness systems demanded in future eBusiness environment, where systems having their own way of feasible interactions without an agreed business protocol model unexpectedly encounter each other, and want to execute business conversation using their best efforts.

We propose a solution to the above challenge in this paper, taking an approach: Each system exposes a model specifying possible interfaces and behavior; Systems exchange exposed models; Each system reduces its exposed model adjusting to the opponent system; Then, systems begin business conversation if both models were successfully reduced. The paper provides a formal definition of exposed models and an algorithm to reduce exposed models, and presents results using experimental implementation showing appropriateness of the proposed definition and algorithm. It also discusses matching methods of business resource definitions used in the algorithm. The results firstly target eBusiness application but are also applicable to dynamic protocol generations in other areas including robots, mobile devices and ubiquitous environments.

Key words: model harmonization; business protocol; exposed model; interface; behavior; business resource definition; model reduction algorithm.

1. INTRODUCTION

As Web Services expand from simple request-response technology to flexible message-based bidirectional conversation technology, alignment of business protocol among eBusiness systems becomes an urgent key issue.

Web Services basic protocol stacks, consisting of SOAP¹, WSDL² and relevant specifications, are standardized, implemented by many products, and already used widely. Messaging protocols such as ebXML messaging⁴ and WS-Reliability⁵ are standardized and expected to be supported soon by many infrastructure products with relevant technologies including WS-Transaction and BTP⁶. Based on these technologies, researches and standardizations about business process modeling are underway including description languages such as WS-BPEL^{7,8} and WS-CDL⁹, and business process modeling notations such as BPNM¹⁴ and BPML¹⁵.

It is usually assumed that systems share an agreed model of business protocol to execute consistent exchange of series of business messages. Sharing or standardization of business protocol specification is generally possible in an enterprise, an enterprise group, a particular domain group, or a community with a well-defined scope such as RossetaNet. Most of current studies including the above business process modeling studies are based on this assumption. Repository technology to share models or meta-models is researched and standardized (ebXML¹⁰, MOF¹²). Sharing technology of reusable assets including models at development stage is also advancing (e.g., RAS¹³) and being implemented by IDEs such as Eclipse.

The assumption that systems can share pre-agreed model of business protocol, however, is not always satisfied. eBusiness systems scattered over the world in the Internet environment do not always share a complete agreed business process model even if they may share portions of model, meta-model and meta-data. It is desired that such independent and autonomous systems are able to communicate using their best efforts when they encounter. In addition to the current R&D based on the shared model assumption, dynamic harmonization of different business process models must be researched at the same time, to realize flexible and free dealing required in the future eBusiness systems. This requirement is come out not only from eBusiness application field but also from other application fields such as robot communication¹⁶ and ubiquitous application.

In this paper, we propose a solution to this challenge, called *model harmonization*, extending dynamic business protocol generation^{17,18} from a model harmonization view point. The approach is as follows:

- Each eBusiness system exposes a model specifying feasible interfaces and behaviors.
- Systems exchange exposed models when encountered.

- Each system reduces its exposed model adjusting to the opponent system.
- Systems begin business conversation if both reduced models are not null.

The main proposal of this paper is to formalize exposed models under this approach and propose the *model reduction algorithm* used in the third step. The model reduction algorithm uses matching of business resource definitions appeared in exposed models. We also propose an early stage solution for their matching method using Web ontology. Note that, however, the model reduction algorithm does not assume full solution for business resource definitions matching, and enough effective with rather primitive matching methods.

The structure of this paper is: Section 2 explains the concepts and defines terminology. Section 3 is the main section and provides a formal definition of models exposed by systems and an algorithm to reduce exposed models. Section 4 provides some solution for business resource definitions matching required during model harmonization. Section 5 shows results of experimentation using experimental implementation. Section 6 discusses the whole results and future issues. Section 7 gives short conclusions.

2. MODEL HARMONIZATION

Model harmonization is based on the following concepts regarding eBusiness systems in open network such as the Internet/Web:

1. A system is modeled as an *internal model* and an *exposed model*. An internal model is a design model of the system, describing functionality used inside of the system or in its internal operational environment, and having a key role in its implementation process. An exposed model is to externally declare the system's functionality to other systems that may be encountered some time. Other systems interact with the system through its exposed model. Model harmonization concerns about exposed models. An exposed model may be simply called a model in this paper.
2. An exposed model consists of an *interface model* and a *behavior model*. An interface model includes a set of operation signature specifications and relating attributes like as IDL or WSDL portType. A behavior model describes behavioral specification of operations.
3. *Business resource definitions* (or *BRDs*) are used in an exposed model to specify types of business elements in messages such as product id, price, invoice, and shipment notification. A business resource definition is an object defining business resource, and has a unique identifier called a *business resource definition identifier* (or *BRD identifier*). By specifying business resource definition identifiers in an exposed model, the

definition bodies of business resource and its reference are separated. A business resource definition identifier typically has a form of URI.

4. *Model harmonization* is defined as a process to automatically adjust exposed models between encountering systems.

3. MODEL DESCRIPTION AND HARMONIZATION ALGORITHM

This section is the main part of this paper, proposing a way to describe exposed models and an algorithm to dynamically harmonize exposed models.

3.1 Model Description

Figure 1 shows a formal definition of exposed models. It is a kind of meta-model and also definable using other meta-model languages such as MOF, UML profile and XML schema. We here use a traditional style of formal definition to concentrate on essential arguments.

$M = (IM, BM)$ <p style="margin-left: 20px;"><i>IM</i>: An interface model defined below. <i>BM</i>: A behavior model defined below.</p>
$IM = (U, T, D)$ <p style="margin-left: 20px;"><i>U</i>: A finite set. (Operations) <i>T</i>: $U \rightarrow \{I, O, (I, O), (O, I)\}$ <i>D</i>: $U \times \{I, O\} \rightarrow \{(BRD, \dots, BRD)\}$</p>
$BM = (B)$ <p style="margin-left: 20px;"><i>B</i>: A regular expression on <i>U</i>.</p>

Figure 1. Definition of Exposed Models

An exposed model *M* consists of an interface model *IM* and a behavior model *BM*. *U* is a set of operations. *T* is a mapping from each operation to its *message exchange pattern*, specifying *I* (input-only), *O* (output-only), (*I, O*) (input-output) or (*O, I*) (output-input), abstracted form WSDL2.0 Part 2³. *D* specifies a sequence of business resource definitions defining elements of each message. Each *BRD* is given by a business resource definition identifier. *BM* consists of a *behavior pattern* *B*, expressed by a regular expression of operations.

Figure 2 shows an example, optionally querying price, then, requesting an order and receiving a confirmation of order acceptance, if the order is accepted, receiving a bill and notifying payment, then receiving a delivery notice and sending an acceptance notification. Business resource definitions are specified by URIs in a name space ns = 'http://www.examp.org/brd008/'.

```

U = { PriceQuery, GetPrice, Order, Response, Bill, PaymentNotice, Delivery, Acceptance }
T(PriceQuery)=O, T(GetPrice)=I, T(Order)=O, T(Response)=I,
T(Bill)=I, T(PaymentNotice)=O, T(Delivery)=I, T(Acceptance)=O
D(PriceQuery,O)= 'ns:price.ask', D(GetPrice,I)=ns:price.ans',
D(Order,O)='ns:form.order', T(Response,I)='ns:form.order_response',
D(Bill,I)='ns:form.bill', D(PaymentNotice,O)='ns:notice.conf',
D(Delivery,I)='ns:notice.shipment', D(Acceptance,O)='ns:notice.conf'
B = ( SEQ,
      ( OPT, ( SEQ, PriceQuery, GetPrice) ),
      Order, Response,
      ( OPT, ( SEQ, Bill, PaymentNotice, Delivery, Acceptance ) ) )
    
```

Figure 2. Example of Exposed Model

Note that this formal definition is designed considering application to Web Services. Fig. 3 is an example of a mapping result from the model described in Fig. 2 to WSDL style.

```

<definitions name= "Model PS" xmlns:ns="http://www.examp.org/brd008/">
  <message name="Message1">
    <part name="pt_1" element="ns:price.ask"/>
  </message>
  <message name="Message2">
    <part name="pt_1" element="ns:price.ans"/>
  </message>
  <portType name="Model PS interface">
    <operation name="PriceQuery">
      <output message="Message1">
    </operation>
    <operation name="GetPrice">
      <input message="Message2">
    </operation>
  </portType>
  <behavior>
    <SEQ>
      <OPT>
        <SEQ> "PriceQuery" "GetPrice" </SEQ>
      </OPT>
      "Order" "Response"
    </OPT>
    <SEQ> "Bill" "PaymentNotice" "Delivery" "Acceptance"
  </SEQ>
  </OPT>
</SEQ>
</behavior>
</definitions>
    
```

Figure 3. Example Mapping to WSDL Style

3.2 Model Harmonization Algorithm

Suppose two systems S and T have encountered. Assume S exposes a model P with an interface model IM_P and a behavior model BM_P , and T does Q with IM_Q and BM_Q (see Fig. 4). Model harmonization algorithm, the main result of this paper, is explained below. To simplify the explanation without losing generality, we limit a message exchange pattern is always I or O , because (I, O) and (O, I) are equivalent to successive two operations whose message patterns are I and O . We also limit only one business resource definition is specified for each operation in D .

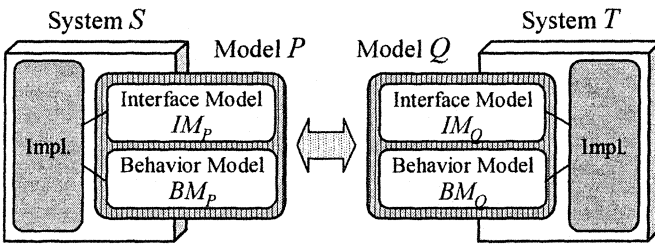


Figure 4. Model Harmonization

3.2.1 Top-level Algorithm

- Step a :
exchange P and Q between S and T
- Step b :
At S ,
reduce P adjusting to Q , using the model reduction algorithm
suspend execution if the resulted model $newP$ is ϕ
- Step c :
At T ,
reduce Q adjusting to P , using the model reduction algorithm
suspend execution if the resulted model $newQ$ is ϕ
- Step d :
start conversation between S and T using the reduced models

Figure 5. Top-level Algorithm

The top-level algorithm is shown in Fig. 5. The model reduction algorithm used in step b and c is explained in the next subsection. Note that steps b and c may be executed in parallel at each system.

3.2.2 Model Reduction Algorithm

The proposed algorithm, called *model reduction algorithm*, is shown in Fig. 6. Note that Fig. 6 describes the process executed at S , i.e., in Step b in Fig. 5. The process executed at T , i.e., in Step c, is similar except that P and Q are swapped. The following notations are used:

$P = (IM_P, BM_P)$, $IM_P = (U, T_P, D_P)$, $BM_P = (B_P)$, $Q = (IM_Q, BM_Q)$, $IM_Q = (U, T_Q, D_Q)$, $BM_Q = (B_Q)$. ϕ is an empty set, \cap denotes intersection, and $(A)^*$ means a power set of A . ε is a null string, $+$ is concatenation of strings, and $|$ denotes a 'choice' operator in regular expressions. $L(R)$ denotes a set of all strings matching to a regular expression R .

A function *match()*, called a *business resource definition matching function*, is used in the algorithm. *match(from, to)* examines, in its execution context, a business resource definitions BRD_f specified by *from* and BRD_t , specified by *to*, and detects inclusion relationship between $I(BRD_f)$ and $I(BRD_t)$ as follows, where $I(brd)$ is a set of all instances of a business resource definition *brd*:

$$\begin{aligned} match(from, to) &= True && \text{(if } I(BRD_f) \subset I(BRD_t) \text{ is detected),} \\ &False && \text{(if } I(BRD_f) \not\subset I(BRD_t) \text{ is detected),} \\ &Unknown && \text{(otherwise, i.e., neither is detected)} \end{aligned}$$

The algorithm has two versions, *strict* and *optimistic*. The algorithm in Fig. 6 is the strict version. In the optimistic version, four invocations of *match()* in Step 1 are replaced by

$$\begin{aligned} match(D_P(u, O), D_Q(v, I)) &= True \text{ or } Unknown, \\ match(D_Q(v, O), D_P(u, I)) &= True \text{ or } Unknown, \\ match(D_Q(v, O), D_P(u, I)) &= True \text{ or } Unknown, \text{ and} \\ match(D_P(u, O), D_Q(v, I)) &= True \text{ or } Unknown. \end{aligned}$$

Step 1 compares U and V , and decides possible pairs of operations whose business resource definitions are matched (or not not matched in the optimistic version). $\Phi_P(u)$ is a set of operations in V that match to u in U , and $\Phi_Q(v)$ is a set of operations in U that match to v in V . Step 2 removes unmatched operations from original behavior patterns B_P and B_Q , and computes revised behavior patterns B'_P and B'_Q . Step 3 is the main part of this algorithm. Operations in the behavior pattern exposed by the opponent system T is replaced by corresponding possible operations in S , and an intersection with the own behavior pattern is taken. $G(v)$ is a regular expression created by connecting all operations in $\Phi_Q(v)$ using choice ($|$). After replacing all V operations in B'_Q by $G(v)$, a regular expression *newB_P* giving an intersection of $L(B'_P) \cap L(B'_Q)$ is calculated. Step 4 finally generates a reduced model *newP*.

Step 1 :**for each u in U :**

$$\Phi_P(u) = \{ v \mid T_P(u)=O \text{ and } T_Q(v)=I \text{ and } \text{match}(D_P(u,O), D_Q(v,I)) \\ \text{or } T_P(u)=I \text{ and } T_Q(v)=O \text{ and } \text{match}(D_Q(v,O), D_P(u,I)) \}$$
for each v in V :

$$\Phi_Q(v) = \{ u \mid T_Q(v)=O \text{ and } T_P(u)=I \text{ and } \text{match}(D_Q(v,O), D_P(u,I)) \\ \text{or } T_Q(v)=I \text{ and } T_P(u)=O \text{ and } \text{match}(D_P(u,O), D_Q(v,I)) \}$$

$$U' = \{ u \mid u \in U \text{ and } \Phi_P(u) \neq \phi \}; V' = \{ v \mid v \in V \text{ and } \Phi_Q(v) \neq \phi \}$$
if $U' = \phi$ or $V' = \phi$ then exit (suspend)**Step 2 :****obtain B'_P satisfying $L(B'_P) = L(B_P) \cap (U')^*$** **obtain B'_Q satisfying $L(B'_Q) = L(B_Q) \cap (V')^*$** **if $L(B'_P) = \phi$ or $L(B'_Q) = \phi$ then exit (suspend)****Step 3 :****for each v in V' :** $G(v) = \varepsilon$ **for each u in $\Phi_Q(v)$: $G(v) = G(v) + "|" + u$** **replace all v in B'_Q by $G(v)$** **obtain $newB_P$ satisfying $L(newB_P) = L(B'_P) \cap L(B'_Q)$** **if $L(newB_P) = \phi$ then exit (suspend)****Step 4 :** $newU = \{ u \mid u \in U' \text{ and } (u \text{ in } newB_P) \}$ **if $newU = \phi$ then exit (suspend)** $newT_P = T_P$ **restricted in $newU$; $newD_P = D_P$ restricted in $newU$** $newIM_P = (newU, newT_P, newD_P)$; $newBM_P = (newB_P)$ $newP = (newIM_P, newBM_P)$ **return $newP$**

Figure 6. Model Reduction Algorithm

4. BUSINESS RESOURCE DEFINITION MATCHING

$match(from, to)$ is implemented in various ways. A trivial implementation is to return T if $from=to$, F if $from \neq to$ and both are in the same name space, and U otherwise. The model harmonization algorithm effectively works even using such a trivial matching method as shown in

Section 5. This section discusses a more improved method based on the following approach.

Definitions of business resource cannot always be processed by machines because they are defined in various forms, sometimes in natural language. We associate information describing a BRD (business resource definition) as its proxy, called a BRD description, having an identifier of the BRD. A BRD description is expressed using Web ontology, which may exist at some place in the Internet, and maintains binary relations between the BRD and other BRDs. In the proposed method, *match(from, to)* is calculated by inspecting paths of BRDs from *from* to *to*.

4.1 BRD Description

A BRD description includes descriptions of binary relation and structure. *Fig. 7* shows an example. A binary relation description specifies relationship between the concerning BRD and another BRD such as equivalence, subsumption (inclusion) and disjointness (exclusion). A structure description expresses structural characteristics of the BRD. Both are described using OWL²¹ in RDF/XML²² Syntax. Binary relations are expressed using *owl:equivalentClass*, *owl:subClassOf* and *owl:disjointWith*. The algorithm mentioned below focuses on these three binary relations.

<pre> <owl:Class rdf:about="http://www.examp.org/brd008/form.bill"> <owl:equivalentClass rdf:resource="http://www.examp.org/brd008/form.seikyusho"/> <rdfs:subClassOf rdf:resource="http://www.mkik.org/form.request"/> <owl:disjointWith rdf:resource="http://www.examp.org/brd008/notice.shipment"/> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="http://www.struex.com/form.name"/> </owl:Restriction> </rdfs:subClassOf> </owl:Class> </pre>	<p>Binary Relations</p> <p>Structure</p>
--	--

Figure 7. Example of BRD Description

4.2 BRD Matching Algorithm

Figure 8 shows the algorithm. BRD_f and BRD_t are BRDs specified respectively by *from* and *to*. B_{fe} , B_{fs} , B_{fd} , B_{te} , B_{ts} and B_{td} are sets of BRDs. BRDs determined as equivalent with BRD_f , as subsumed by BRD_f and as disjoint with BRD_f are respectively stored in B_{fe} , B_{fs} , and B_{fd} in Step 1. BRDs determined as equivalent with BRD_t , as subsumed by BRD_t and as disjoint with BRD_t are respectively stored in B_{te} , B_{ts} , and B_{td} in Step 2. *visible(brd)* is a function that determines whether a BRD description *brd* is referable in this execution context. *equivalentClass(brd)*, *subClassOf(brd)* and

disjointWith(brd) are functions returning sets of BRDs specified in owl:equivalentClass, owl:subClassOf and owl:disjointWith in *brd*.

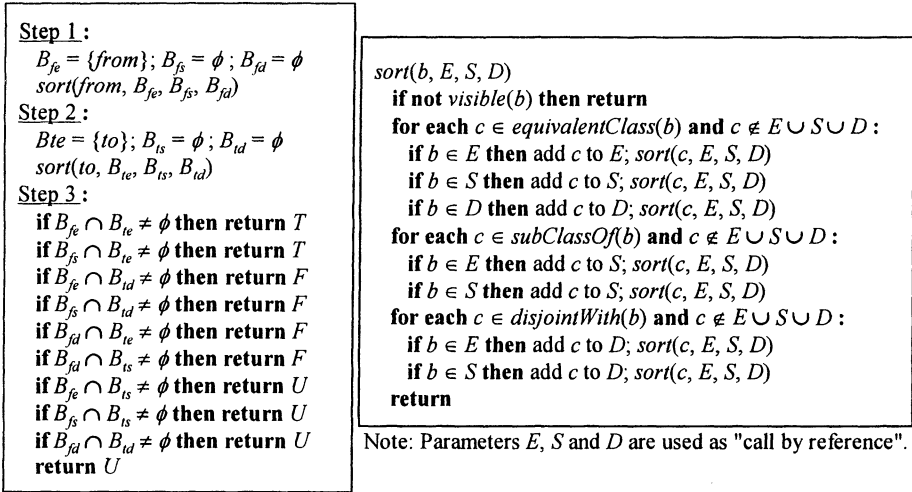


Figure 8. BRD Matching Algorithm

In Step 1 and 2, *sort()* recursively searches BRDs started from BRD_f or BRD_t and stores them into appropriate sets $B_{fe}, B_{fs}, B_{fd}, B_{te}, B_{ts}$ or B_{td} . Step 3 makes a final decision inspecting these six sets.

Note that the above is to show the essential algorithm. Practically, it is not necessary to fully complete Step 1 and 2 before execution of Step 3. Step 1 and 2 may incrementally run in parallel, and Step 3 may be executed each time when a new BRD is added.

5. EXPERIMENT

This section shows result of experiment using MD_REDUCE, an experimental implementation of the model reduction algorithm. MD_REDUCE is developed using Python and the whole algorithm mentioned in Section 3.2 is implemented with necessary background processes including set operations and automata transformations. Models are expressed by Python objects, making independent of middleware platforms and transformable to specific platforms such as XML/WSDL.

Figure 9 and 10 shows an example case. A buyer's system and a provider's system having unadjusted exposed models P and Q try to start conversation. P has nine operations, PriceQuery to ask price, GetPrice to receive price, Order to order, Response to know result of order, AnsQuestion

to answer a question from provider during order process, Delivery to get a delivery notification, Acceptance to answer completion of acceptance, Bill to receive a bill, and PaymentNotice to notify payment. Business resource definition identifiers are given to each message as in Fig. 10.

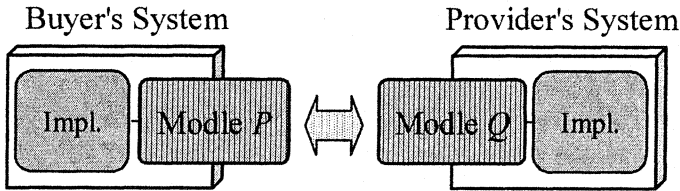


Figure 9. Example Case

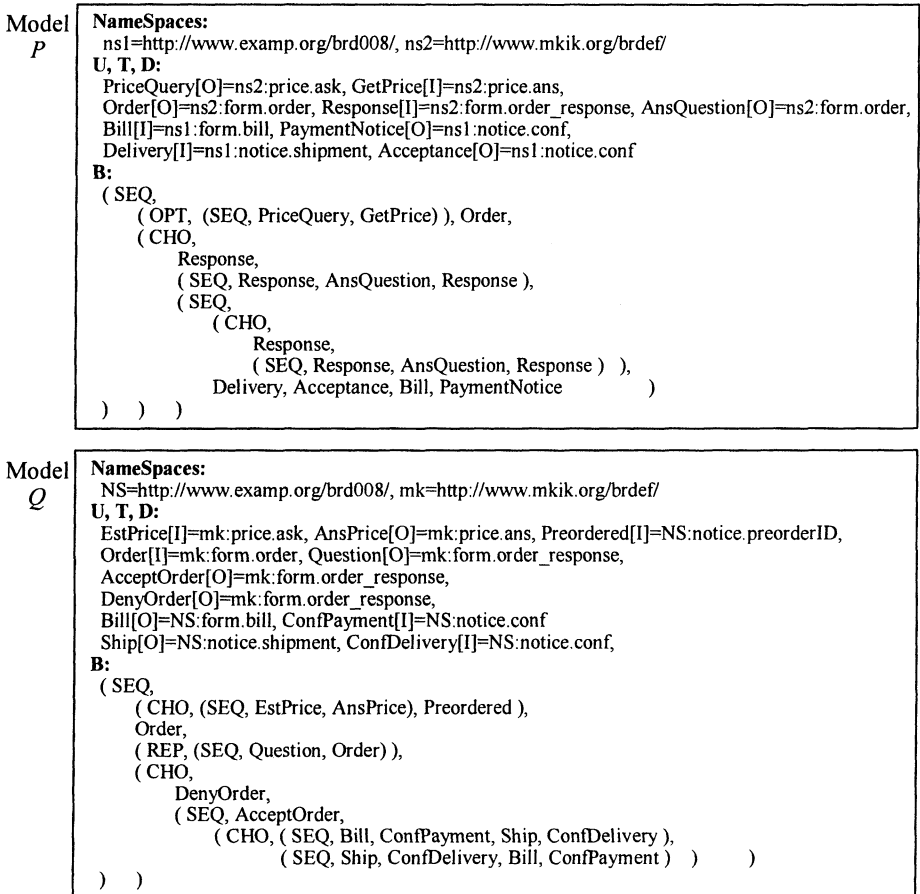


Figure 10. Model P and Q (before reduction)

Note that *U*, *T* and *D* in the formal definition are expressed in a short form in Fig. 10 as "uuuu[t]=dddd" where uuuu is an operation name, t is a message exchange pattern, I or O, and dddd is a business resource definition identifier. Q has ten operations whose names are not compatible with P. Behavior of P and Q is not aligned. Q always requires price estimation or pre-order ID before starting order, may ask back questions more than zero times to close order, and may ship after billing or bill after shipment. Whereas, P may skip price query process before ordering, accept question only once, and requires shipment before billing.

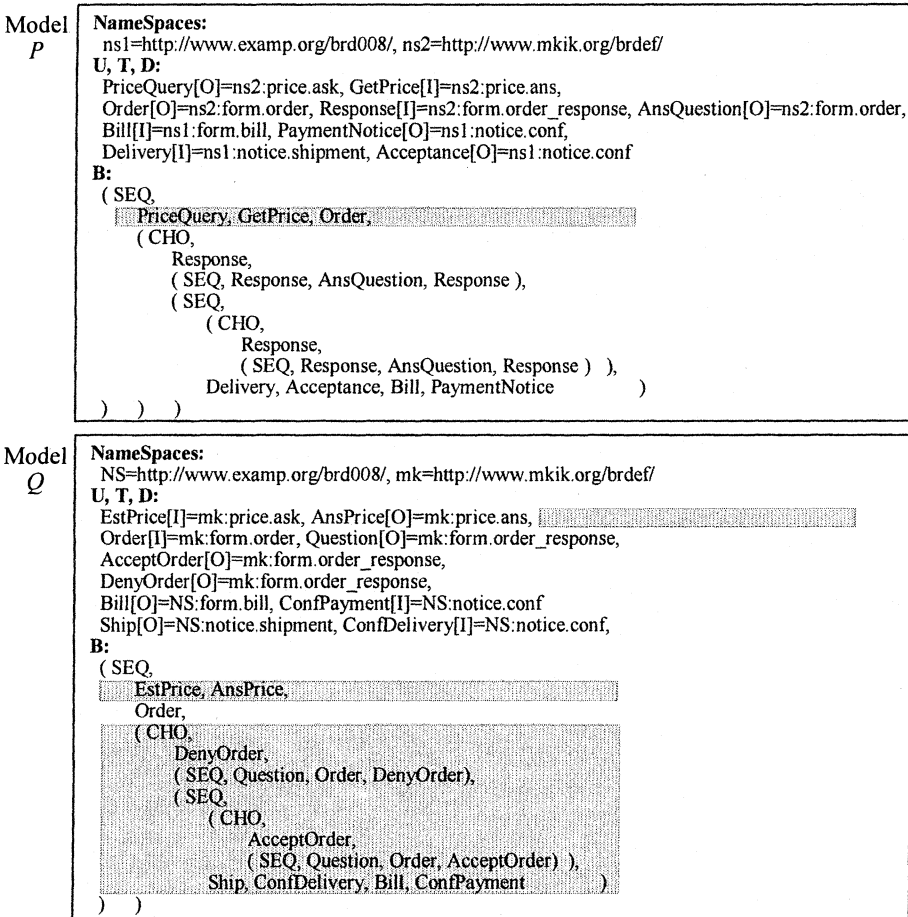


Figure 11. Model P and Q (after reduction)

Figure 11 is the execution result of MD_REDUCE. Hatchings show modified parts by the model harmonization algorithm. Price query process in *P* has been mandated. Preorder option in *Q* has been removed. Repetition of

questions in Q has been limited. These results show the algorithm effectively reduce both models.

In this experiment, a trivial *match()* mentioned in Section 4 is used, which indicates that the model harmonization algorithm is effective even without refined business resource definition matching algorithms.

6. DISCUSSION

The appropriateness of the model description definition and the model harmonization algorithm mentioned in Section 3 is verified through experiment using MD_REDUCE.

Table 1 is a simulation result showing the model harmonization algorithm effectively removes invalid protocol sequences in several sample cases. "Total" is a total number of possible message sequences produced by System 1 or 2. "Errors" is a number of message sequences within the "Total" causing intermediate suspension because one side receives a message in a format it cannot process. Internal errors in implementation are not counted. The trivial business resource matching is applied and the strict version is used. As shown, the model reduction algorithm effectively removes unnecessary protocol errors.

Table 1. Simulation Result

	without model reduction				after model reduction			
	System 1		System 2		System 1		System 2	
	Total	Errors	Total	Errors	Total	Errors	Total	Errors
Case 1	72	48	24	0	24	0	24	0
Case 2	12	8	4	0	4	0	4	0
Case 3	18	14	24	20	4	0	4	0
Case 4	16	14	18	16	2	0	2	0
Case 5	4	4	4	4	0	0	0	0

Future issues on model harmonization algorithm include the followings: 1) Expansion of expression of behavior patterns. The model description in this paper limits expression of behavior patterns as regular expression, i.e. automata, in order to ensure decidability of the algorithm. Even in such limitation, model harmonization takes effect as discussed. To make it further effective, state machine expressions with appropriate restriction is to be introduced. 2) Internal error handling: In this algorithm internal errors that occur during actual process in implementation are ignored. Exception mechanism needs to be studied.

Business resource definition matching in Section 4 is still at an early stage. Though refined business resource matching is not a prerequisite of the model harmonization algorithm, improvement of business resource

definition matching needs further study. A major issue is to find an effective and practical matching algorithm traversing through BRD descriptions scattered in the Internet space. Reliability of each BRD description may need to be evaluated during execution of the algorithm. Another issue is to expand BRD descriptions beyond three types of binary relations as mentioned in Section 4.1. A certain subset of OWL DL¹⁹ may need to be introduced considering appropriate balance between expressiveness and complexity of reasoning.

The model harmonization proposed in this paper may be positioned as an extension of OMG/MDA¹¹. An exposed model is a PIM (Platform Independent Model) able to map to PSMs (Platform Specific Models), and, at the same time, is a PSM over a specific business process platform. The model harmonization is an approach to dynamically generate a PIM common to concerning business process platforms. The further study should be done from this point.

7. CONCLUSIONS

We have shown a method to dynamically align business protocol with best effort between eBusiness systems not sharing a pre-agreed business process model, by exchanging and harmonizing models exposed by unexpectedly encountering systems. Definition and algorithm have been given, and verified through experimental implementation. Business resource definition matching has discussed and an early stage solution has been proposed. Future eBusiness communication must allow both cases when pre-agreed business protocol models exist and when do not or are partially given. The results of this paper give an initial step for its realization.

REFERENCES

1. M. Gudgin, M. Hadley, N. Mendelsohn, J. Moreau and H. Nielsen, SOAP Version 1.2 Part1: Messaging Framework, W3C Recommendation, 2003.
2. R. Chinnici, J. Moreau, A. Ryman and S. Weerawarana, Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Working Draft, 2005.
3. M. Gudgin, A.Lewis and J. Shlimmer, Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts, W3C Working Draft, 2005.
4. ebXML, Message Service Specification Version 2.0, OASIS, 2002.
5. K. Iwasa (ed.), et al, WS-Reliability 1.1, OASIS, 2004.
6. S. Dalal, T. Fletcher, A. Green, B. Haugen, A. Ceponkus and B. Pope, Business Transaction Protocol Version 1.1.0, OASIS, 2004.

7. A. Arkin, S. Askary, B. Bloch, F. Curbera, Y. Goland, N. Kartha, C. Kevin, S. Thatte, P. Yendluri and A. Yiu, Web Services Business Process Execution Language Version 2.0, OASIS Working Draft, 2005.
8. T. Andrews, Francisco Curbera, et al, Business Process Execution Language for Web Services Version 1.1, 2003, <ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>
9. N. Kavantzias, D. Burdett, G. Ritzinger, T. Fletcher and Y. Lafon, Web Services Choreography Description Language Version 1.0, W3C Working Draft, 2004.
10. ebXML, Collaboration-Protocol Profile and Agreement Specification Version 2.0, OASIS, 2002.
11. J. Miller, J. Mukerji, et al, "MDA guide version 1.0", OMG omg/2003-05-01, May 2003.
12. OMG, Meta Object Facility (MOF) 2.0 Core Specification, OMG, ptc/03-10-04, 2003
13. OMG, Reusable Asset Specification, OMG, ptc/04-06-06, 2004.
14. S. White (ed.), et al, Business Process Modeling Notation (BPMN), BPMI.org, 2004.
15. A. Arkin, Business Process Modeling Language, BPMI.org, 2002.
16. M. Oya, K. Naruse, M. Narita, T. Okuno, M. Kinoshita and Y. Kakazu., Loose Robot Communication over the Internet, Journal of Robotics and Mechatronics, Vol.16, No.6, pp. 626-634, 2004.
17. M. Oya, and Y. Kakazu., Automatic Business Process Generation in the Autonomous Web Services Environment, IAS-8, pp.112-119, 2004.
18. M. Oya, M. Kinoshita, Y. Kakazu, "On Dynamic Generation of Business Protocols in Autonomous Web Services", The IEICE Transaction on Information and Systems, Vol. J87-D-I, No.8, pp.824-832, 2004. (in Japanese)
19. T. Berners-Lee, J. Hendler and O. Lassila, The semantic web, Scientific American, 2001.
20. F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P. Patel-Schneider, The Description Logic Handbook, Cambridge University Press, 2003.
21. P. Patel-Schneider, P. Hayes and I. Horrocks, OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation, 2004.
22. G. Klyne and J. Carroll, Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation, 2004.