VISUALIZATION OF WEB-BASED AUSTRALIAN EXPORT TRADING

Lisa Soon and Phoebe Chen
School of Information Systems
Queensland University of Technology, Brisbane, Australia
{l.soon, p.chen}@qut.edu.au

Abstract: E-commerce applications tend to be used by a non-homogenous user population, requiring special attention to the modelling of underlying business processes, in order to make the execution of an activity as conspicuous as possible. Current modelling languages do not provide for concepts and symbols to represent all communicative aspects of a business transaction. Speech act theory offers categories that may be used to supplement the concepts and notations of current modelling languages. We are developing a Visual Business Modelling Language (VBML), for which we propose various speech acts in addition to the common symbols in a modelling language. To explore and demonstrate the expressiveness and logic of VBML, we apply it to the web-based Australian export trade. In particular, we show how an export trader can make a cargo declaration through the web to facilitate a document exchange with the Australian Customs Service. This model benefits application designers, software developers, Australian Customs Service, and the export traders since it delivers a clear view of the trade and the corresponding software application processes.

Key words: Visual Business Modelling Language (VBML), Speech Act Theory (SAT), Electronic Commerce, Electronic Data Interchange (EDI), traditional EDI, web-based EDI, XML/EDI

1. INTRODUCTION

It is essential that a model of a system is perceived as reflecting a piece of reality and the user's requirement (Bergholtz & Johannesson, 2000). While models help to understand the system by simplifying (Conallen, 1999), they should also capture systems and business details sufficiently.

X. Zhou et al. (eds.), *Visual and Multimedia Information Management*

There have been various modelling languages used to present software architectures, business systems, and an increasing emphasis on speech act theory in this context. We develop a Visual Business Modelling Language (VBML) that can break down the web-based business document information exchange and visually represent it by reflecting the essential user requirements and processes in Australian export trading activities. Using VBML, we built a business model for electronic document exchange between Australian export traders and the Australian Customs Services, using web technology and a traditional EDI (Electronic Data Interchange) network.

In the following section, we investigate some of the features of common modelling languages and their objectives. Then, we examine how Speech Act Theory (SAT) has been integrated into some modelling languages. Although it has been highlighted that SAT can address problems of a range of modelling languages, its categories have not been widely used as components, and they are not visually presented in models. The advantages of using the Visual Business Modelling Language are demonstrated in section 5. In concluding, we expect that VBML may be a valuable tool beyond the context of our case.

2. BUSINESS MODELLING OF BUSINESS APPLICATIONS

Many business models exist to guide businesses to enhance their conduct of commercial activities (Lawrence, 2001). Due to the common use of business applications over internet, intranet and extranet, there is a constant need of business modelling languages. Models help us to understand the system by reducing complexity (Conallen, 1999). It is critical that any proposed model is able to spell out the business requirements and components of system architecture through underlying business-modelling language. Therefore, many existing process description languages and business process languages (being modelling tools) place a great emphasis on business needs and requirements.

Examining process description languages (Johannesson & Perjons, 2000, Swenson, 1993, Guerin, 2001, Pitt, 2001, Scheer, 1999, Weske, 1999), process definition languages (Cass, 2000), and the Unified Modeling Language (D'Sourza, 1999, Tanuan, 1998, Conallen, 1999), shows some common pitfalls. Many languages highlight the importance of consideration of social acts or speech acts in the application design and in the business model. While there is an emphasis on the use of speech act categories in models, there are no visual representations for speech acts in any modelling

language. A particular language uses only three speech act elements, namely belief, desire and intention (BDI) (Pitt, 2001). While BDI are critical in most business applications, the same applies to other speech act categories too. Because of such shortfalls, we have examined the types of speech act elements and the presentations of speech act symbols that are related to business activities.

We are testing the capabilities of the proposed components for modelling electronic commerce. Our objective is to represent visually web-based Australian export trade in a model. In section 3 we will present how we apply SAT to the business communication.

3. SPEECH ACT THEORY

As already mentioned above, various modelling languages apply speech act theory (SAT). SAT helps to clarify that messages are not just pieces of data, but have some social effects (Hasselbring & Weigand, 2001). Speech Act Theory is referred to in various disciplines such as philosophy, linguistics, communication, and sociology (Smith, 1990, Tien, 2000). A number of researchers in computer and communication technologies have already considered SAT in the design of applications.

The basic tenet of SAT is that speech acts imply a function of meaning (morpheme) in a communicative sentence. Three basic categories of speech acts are distinguished: A message can be locutionary to deliver a meaning; this act is to say something (e.g. love, hate, desire, etc). The illocutionary message (e.g. warning, convince, persuade, etc) has a force; it informs with a tone, attitude, feeling, motive or intention. The third is a perlocutionary act (e.g. promise, producing consequential effects).

Speech acts can be assertive (statement characterised as either true or false, e.g. The database is used over two networks), directive (get the hearer to do something, e.g. Please change that.), comissive (commits the speaker to some future course of actions, e.g. I promise to be there), expressive (expresses something to do with the speaker's psychological state, e.g. I hate this), declarative (makes the state of affair for the statement, e.g. The project is completed.). These speech acts provide us with the social and pragmatic aspects of communication needed for software architecture design. We apply SAT to communication as follows:

· Assertive: The users state what is true or false in the software application. E.g. The permit is issued when approved is granted.
In our example of VBML model, we use: *acknowledgment, intention.*

· Directive: The users set a directive or instruct to have something done.
E.g. The multimedia message must contain fonts, colours, data, text, graphics, images, sound and video.
In our example of VBML model, we use: *request.*

· Comissive: The users want the application to achieve some kinds of end results.
E.g. the software must be developed as an inventory system.
In our example of VBML model, we use: *commitment, obligation, responsibility,* and *promise.*

· Expressive: The users state the cognitive preference of design features such as like/ dislike in the application.
E.g. Software caters for user-friendliness and ease of use with simple keyboard buttons and mouse clicks.
In our example of VBML model, we use: *belief* and *desire.*

· Declarative: The users require the state of work to be demonstrated.
E.g. The software development has to commence next month and end in six months period.
In our example of VBML model, we use: *trust.*

In our research, we focus on the more frequently used actions in general business applications, namely *belief, desire, intention, request, acknowledgment, commitment, obligation, responsibility, promise and trust.* The speech acts should be used in any model in order to cater for communicative, institutional, and deontic notion (Jayaweera et al., 2001). SAT reflects how language, understanding and communication work, and stresses the role of inference and context (Smith, 1990, Kimbrough, 1997). Speech act concepts are thus employed in developing and describing systems for information retrieval and computer-mediated communication (Johannesson & Perjons, 2000, Kimbrough & Moore, 1997, Winograd & Flores, 1987, Hassel, 1998). When deciding how to model something, determining the correct level of abstraction and detail is critical to providing something that will be of benefit to the users of the model (Conallen, 1999).

Considering the social effects qualifies the implementation of SAT as a soft approach in modelling. In contrast to that, most application systems are still built using a hard approach. The hard approach designs a system with the software application detailing the application structure for a particular type of hardware and operating system platform. Adopting a soft approach to supplement the hard approach, the design and development of an application system will capture in the critical business requirements. Therefore, there

should also be an emphasis on the soft approach by incorporating speech acts as additional business details into application design and development. The speech acts we are proposing are limited to the generic ones and thus valid for most business applications, especially for electronic commerce activities over the web. Their usefulness is shown by the following examples:

- A user *believes* that a program task is performed to achieve a result.
- A user *desires* an instant response to an e-mail reply.
- A business client has an *intention* to book an air-ticket.
- The vendor *promises* to deliver the service.
- In electronic commerce, pressing the submit button is to *request* for a product with a buyer's *commitment*, *obligation*, or *responsibility* to pay for the product.
- After pressing the submit button, the browser user expects an *acknowledgment* message.
- The customer *trusts* that his *request* for product he sent and his payment *committed* will help him obtain the product finally.

The ten speech act categories are selected due to their anticipated frequency of use in all business application scenarios. The adoption of speech acts into a business model makes the system architecture salient with a more pertinent business requirements and user needs specification.

4. VISUAL BUSINESS MODELLING LANGUAGE

In modelling, the visual representation of actual knowledge on the application structure and business needs is essential; this is particularly true for complex systems. A useful visual language should specify, visualise, construct and document the artefacts of software systems. It should also be used to develop a business model and non-software systems by the systems designer and application developers. In a clear and well-explained model, any hidden facts and ideas will be made explicit with graphical symbols and simple textual notations.

We are therefore developing a Visual Business Modelling Language (VBML). VBML contains the essential graphical symbols for speech act elements (*belief, desire, intention, request, acknowledgment, commitment, obligation, responsibility, promise and trust*) and for process description elements. For simplicity reasons, we allocate to desire and intention one symbol since both acts are too close in semantic social effects. The same applies to commitment, obligation, and responsibility. Therefore, the ten speech acts are presented by seven graphical symbols in VBML. Other than symbols, VBML allows simplified textual notations to explain ideas within a

model. The graphical symbols we develop aim at demonstrating both the business process symbols and the essential speech act elements that have not been adopted yet by many modelling languages. All proposed BML symbols are shown in figure 1, followed by an explanation of their semantics.

4.1 Visual Business Modelling Language (VBML) and Its Graphical Symbols

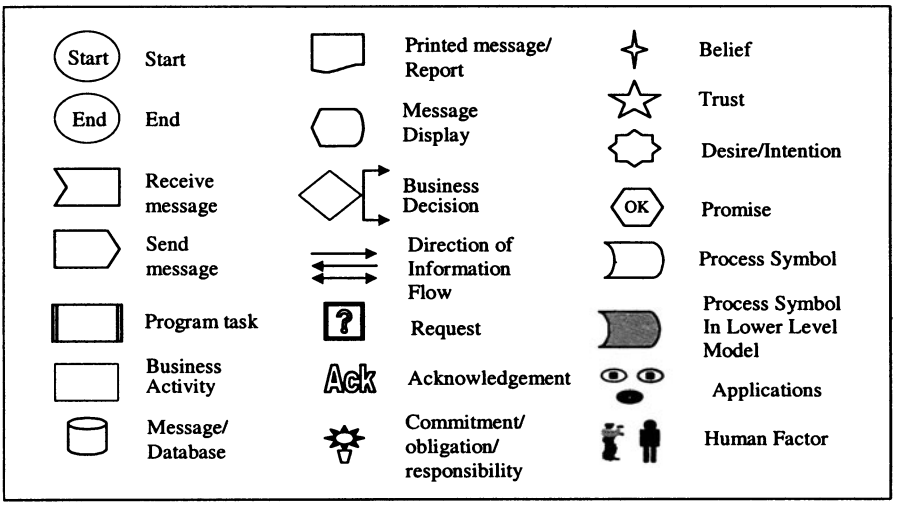

Figure 1. Symbols used in VBML

The symbols in figure 1 are defined as follows:

Start	Indicates the start of any process instance.
End	Describes the end of a message flow of the process instance.
Receive Message	Describes the acceptance of a message from the input queue.
Send Message	Describes the sending of a message.
Program task	It is a predefined software program to perform a designated task.
Business Activity	Describes any business operation(s) to be performed on the process instance.
Message/ Database	This is the database either used in any business process or within a software system.
Printed Message/ Report	It is the output available in a hardcopy information format.
Message Display	It is the output available as softcopy information for display on computer screen.

Business Decision	The business decision is made here based on certain rules.
Direction of Information Flow	There are symbols to show the direction the information flows into the next process or activity.
Request, Acknowledgement, Commitment/obligation/ Responsibility, Belief, Trust, Desire/Intention, Promise	These are the speech acts shown in any related processes, program tasks, or at any point deemed appropriate in the model.
Process Symbol	At a higher level of model, the process symbol is used to indicate some tasks to be performed in a specific process in order to achieve a business activity.
Process Symbol in Lower Level Model	In a lower level model, two process symbols encapsulating one or more software applications are used to elaborate the program tasks needed in order to carry out a process not expanded at a higher level model.
Applications	These are the software applications used in various situations to perform specific program tasks.
Human Factor	Both are symbols showing actors taking part in the business activity or the use of software application.

5. MODELLING OF AN INTERNET BUSINESS APPLICATION WITH SPEECH ACT ELEMENTS – PROCESSING OF AUTHORISATIONS TO EXPORT BY THE AUSTRALIAN CUSTOMS SERVICE

Before the application of any visual aid, it is essential to know the background of application systems, so the use of visual aid can show the functions and application processes effectively. Hence, it is crucial for us to expound the conceptual understanding on the web-based export cargo declaration.

We began the requirements engineering with a case study (Yin, 1994) in the Australian Customs Service Information Centre. Data were obtained by interviewing customs officers, and by collecting government literature, descriptions of work processes and manuals on trade regulations and computer systems architecture. All of the available documentation was subjected to analysis (Weske et al., 1999).

The procedures an export trader has to follow through before the exportation of cargo are explained in figure 2. The web-based Australian export trade procedures involve two essential parties, namely Australian Customs Service (ACS) and export traders.

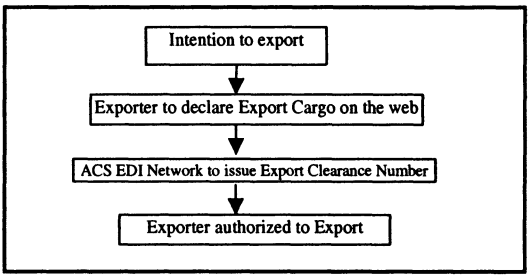

Figure 2. The Procedures an export trader has to follow through before any exportation

Figure 2 shows how an export trader with the *intention* to export has to declare his cargo with ACS as a *request.* Exporters of all goods need to obtain a legal authority (a *promise* from government) to export with an export clearance number from Australian Customs Service before any exportation (a *commitment, obligation* and *responsibility* to export).

5.1 Technological Architecture of Australian Export Trade

The Web-based declaration involves the use of a-browser to lodge an export cargo declaration/reporting. The message sent via the web-browser will have to be picked up by a traditional Electronic Data Interchange (EDI) network in ACS for processing. EDI is a form of paperless exchange of business documents over the telecommunication lines and private networks in standardised electronic format between the computer of an organisation and its customers and/or suppliers (Swindells & Henderson, 1998, Ketler et al., 1997, Clarke, 1999).

EDI is used in Australian Customs Service's two computerised EXIT systems for exporting purposes (Australian Customs Service, 1994). EXIT systems are the application systems that process the messages the export traders send via the web-browser. In other words, the Internet front end does not replace EDI (Hasselbring & Weigand, 2001). The web technology XML (eXtensible Markup Language) works together with an existing EDI network. Export traders use web-browser to communicate their message with ACS's EDI network. The community of export traders declares cargo through the web-browser. XML technology produces web-based EDI that is a common format for communication over the web and over the legacy EDI network. The export traders do not have pay to use the EDI network service but pay for Internet service to use the browser.

Using the graphical symbols designed above, we can simply provide a bird-eye's view on the business document processes involved in web-based Australian export trade.

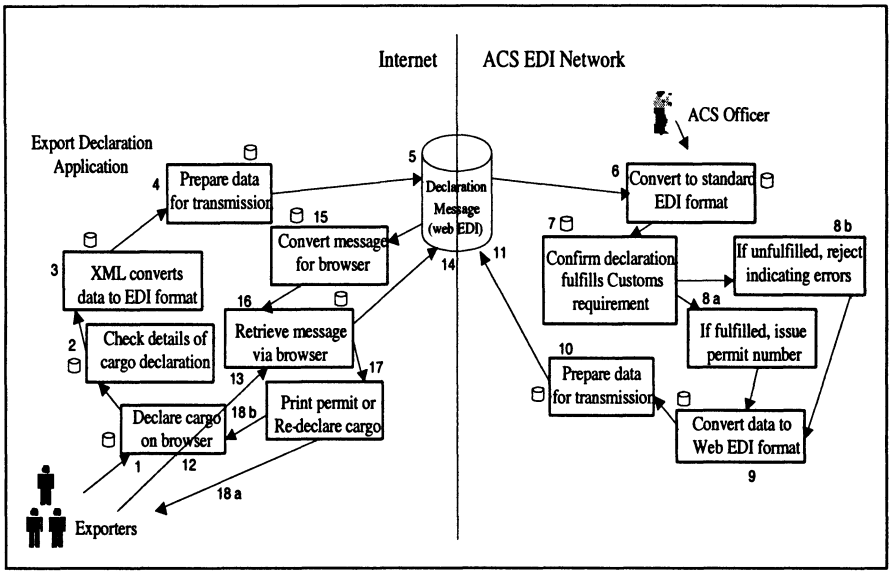

Figure 3. An Overview of Web-based Business Document Exchange in the Australian export trade

The internetworking by connection involving Internet and ACS EDI network is shown in figure 3. The vertical line indicates a data transmission required for the web-based message to be transferred to an EDI network for processing. The two types of human symbols represent both the export traders and the ACS officer handling and overseeing the activities. The message/database symbol represents a common web-based EDI format required for the web technology and the EDI network in ACS. On the left-hand-side of the vertical line, the activities taking place within a web export declaration application are shown. On the right-hand-side of the vertical line, the activities in the application over EXIT systems in the ACS EDI network are shown. The document exchange starts when the exporters use the web-based application to declare a cargo (box 1). Both the exporter and the application program will check the details of the cargo declaration (box 2). With XML, it converts the message into web-based EDI format (box 3). The application will then prepare the data for transmission (box 4). After the message is transmitted (via cylinder 5) into the ACS EDI network, the value-added network will convert such message into the required standard EDI format (box 6) for processing. The EXIT system application will check whether a declaration message contains all information required and fulfils customs regulations (box 7). If fulfilled, customs will issue an export

clearance number (box 8a). If unfulfilled, customs will reject the declaration producing an error message (box 8b). Such information together with the original message will be converted into web-based EDI again (box 9). The system will then prepare (box 10) and store the message in the network mailbox (box 11). The exporter will retrieve the returned message over the internet (box 12). The exporter actually accesses the ACS EDI network mailbox picking up the processed message through Internet facility (box 13). When the web export declaration application picks up the returned processed message (box 14), it converts the message (box 15) for display on the browser (box 16). At the point when the message is displayed (box 17), the exporter either can print the issued permit (approval) to export the cargo (arrow 18a); or has to re-declare the cargo providing more details (disapproval hence error message, after arrow 18b). That starts the business document exchange cycle again.

5.2 Speech Acts in Australian Export Trade

We will examine the speech acts elements concerned throughout web-based business documents exchange between ACS and export traders before we can visually present them in our model. Speech act symbols are used in the model to demonstrate the export and ACS requirements on the software systems. With the knowledge of the business requirements from our case study, we have developed a model for web-based application systems.

Figure 4 shows that an exporter initiates an export declaration with a *belief* that the exporter will obtain an export clearance permit for the exportation of cargo. The exporter then submits a cargo declaration *request* to ACS for processing receiving an *acknowledgement*. There is a cargo reporting process in ACS to handle the message. After processing, the exporter will try to retrieve the declaration message with a *belief* that the request is answered. The retrieved message is a sign of *acknowledgement* of the message being received with either an approval or rejection with some error messages. If the reply message is an approval (*promise*), the exporter can print out the export clearance permit *trusting* with a *commitment* to export his physical cargo. Otherwise, the disapproval shown as a form of error message(s) will be displayed on browser. The exporter will now have an *intention* or *desire* to re-declare (*request*) his cargo again with a *belief* that his revised declaration will be reprocessed and issued with a permit.

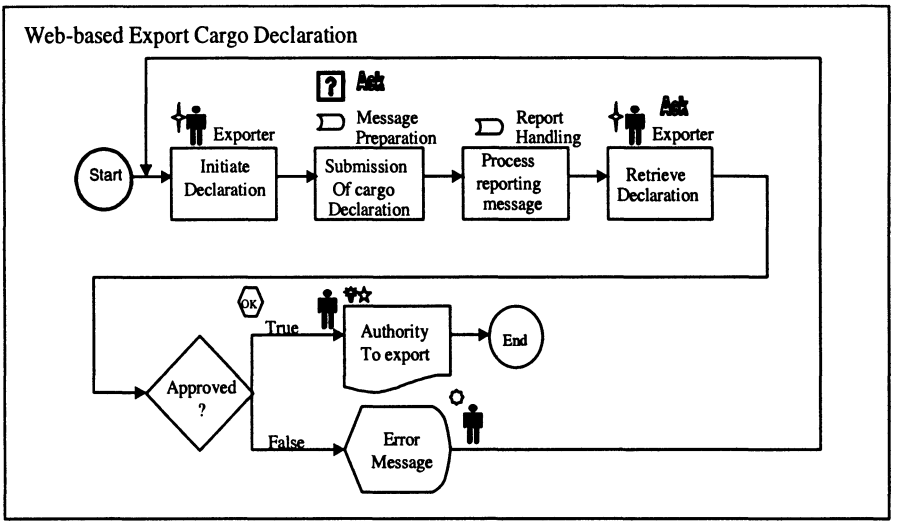

Figure 4. Export Cargo Declaration over the Internet

5.3 A Model with Further Details in Processes Involved in Web-based Australian Export Trading

Notice that there are two process symbols in figure 4 that need to be further illustrated. Therefore, the aims of figure 5 are to pursue the details not covered in figure 4.

We will first examine the details of the first *process* symbol (with the textual notation of message preparation above it) in figure 4. Note that this *process* aims at the submission of a cargo declaration. To show how the submission message can be prepared, we break down the program tasks involved in the export declaration software application. Over the web application, it first obtains the message entered by the exporter. The application will then validate the format of data entered. It also checks whether the data fulfil the general export requirements. The next task is to convert the message into the web EDI format. After conversion, it will send the message out as a *request* for permit over the Internet to the ACS. Once the transmission is performed, the application will prompt out an *acknowledgement* message indicating whether the transmission is done successfully.

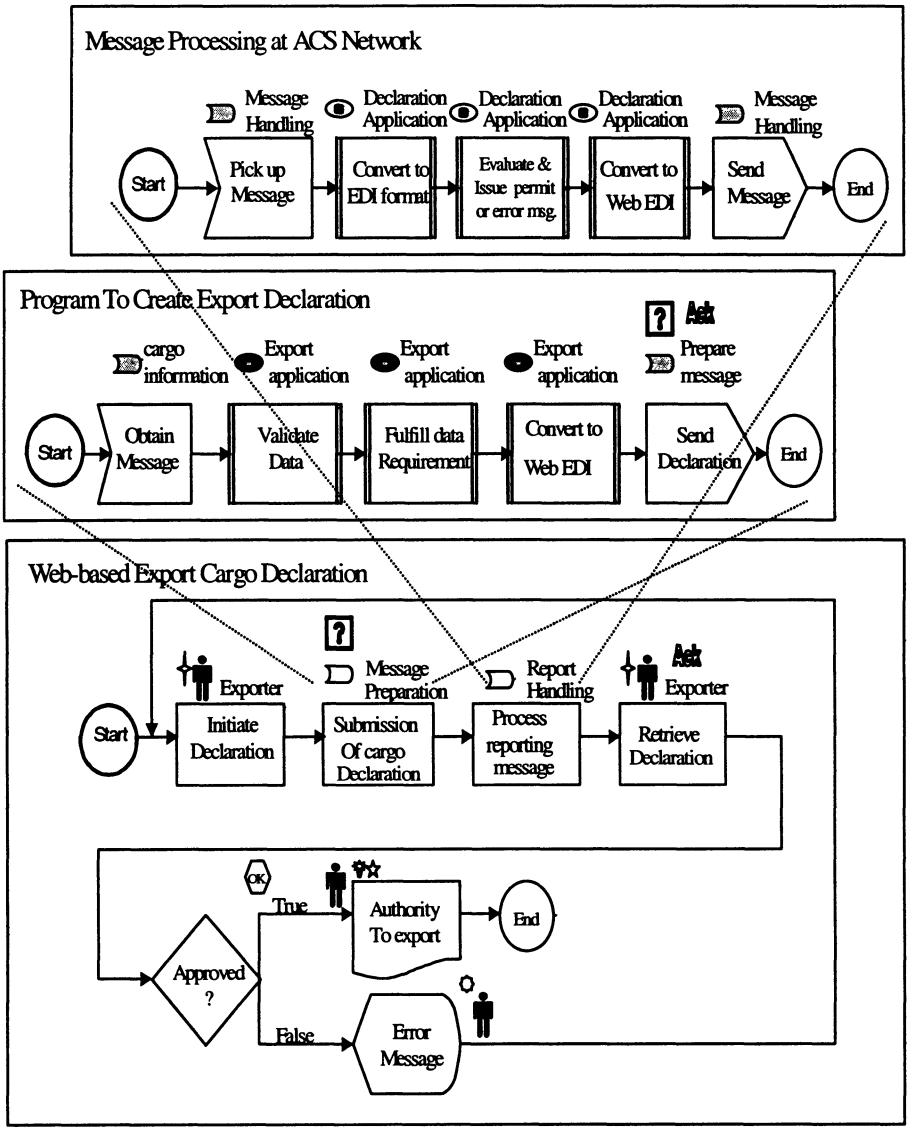

Figure 5. Elaboration of Processes Involved at ACS's End in Australian Export Trade

The details of the second *process* symbol (with the textual notation of report handling above it) in figure 4 are as follows: once the ACS network picks up the message sent over, the EXIT system application will convert the message into the standard EDI format (UN/EDIFACT) suitable for processing in the EDI network. The application will also evaluate the message for information required. It will issue the export clearance number for any message fulfilling the ACS rules and export requirements. Likewise, it will reject any message not fulfilling the rules and requirements producing

one or more error messages for information omitted, erroneous, or incomplete in cargo declaration. The message will then be converted into web EDI format. Such a message is sent out from the ACS network and returned to the exporter via internetworking. In conclusion, figure 6 illustrates how various business activities and processes with the required speech acts can be represented visually in a model for Australian export trade.

6. CONCLUSION AND FUTURE DIRECTION

While the previous modelling languages have not described speech act elements and developed corresponding modelling symbols, VBML aims at fixing theses pitfalls. The use of all symbols in VBML allows for a more comprehensive coverage of aspects of requirements in the business model based on it. VBML implies also the proposition of the use of a soft system approach to supplement the traditional hard system approach in modelling. The advantages of the VBML models are that web-application designers, software developers, and business users can have much clearer views of the overall picture, the embedded processes and the application requirements of the integration work done. This has been shown by an application for the web-based Australian export trade.

We envisage that VBML can be further improved and modified. Currently, we regard it as a proposition awaiting more future development work. In addition, it will require more testing to ensure its applicability in other business contexts and to explore its acceptance within the potential VBML user communities.

REFERENCES

Australian Customs Service. (1994). EXIT Procedural Manual. *Canberra, ACT: Australian Customs Service.*

Bergholtz, M., & Johannesson, P. (2000). Validating Conceptual Models – Utilising Analysis Patterns as an Instrument for Explanation Generation. *Paper presented at the 5th International Conference on Applications of Natural Language to Information Systems.*

Cass, A. G., Lerner, B.S., McCall, E.K., Osterweil, L.J., Sutton, S.M., & Wise, A. (2000). Little-JIL/Juliette: A Process Definition Language and Interpreter. *Paper presented at the 22nd International Conference on Software Engineering, Limerick, Ireland.*

Clarke, R. (1999). Electronic Data Interchange (EDI): An Introduction. *Available: http://www.anu.edu.au/people/Roger.Clarke/EC/EDIIntro.htm [2001, 20th April].*

Conallen, J. (1999). Modelling Web Application Architectures with UML. Communications of the ACM, 42*(10), 63-70.*

D'Sourza, D. F., Wills, A.C. (1999). *Objects, Components, and Frameworks with UML: The Catalysis Approach.* Mass.: Addison-Wesley Longman.

Guerin, F., & Pitt, J. (2001). Denotational Semantics for Agent Communication Languages. *Paper presented at the 5fth International Conference on Autonomous Agents, Montreal, Quebec.*

Hassel, L. (1998). Media, Speech Act Theory and Computer Supported Cooperative Work. *Drexel University, U.S.A.*

Hasselbring, W., & Weigand, H. (2001). Languages for Electronic Business Communication: State of the Art. Industrial Management & Data Systems, 101*(5), 217-226.*

Jayaweera, P., Johnnesson, P., & Wohed, P. (2001). From Business Model to Process Pattern in E-Commerce. *Paper presented at the 6th International Workshop on the Language-Action Perspective on Communication Modelling (LAP2001), Montreal, Quebec.*

Johannesson, P., & Perjons, E. (2000). Design Principles for Application Integration. *Paper presented at the 12th Conference on Advanced Information Systems Engineering, Stockholm.*

Johnson, L. (1998). *Meaning and Speech Act Theory.* A Marvin Farber Conference on Law and Institutions in Society, SUNY Buffalo University.

Ketler, K., Willems, J., & Hampton, V. (1997). The EDI Implementation Decision: A Small Business Perspective. *Paper presented at the 1997 Conference on Computer Personnel, San Francisco, CA.*

Kimbrough, S. O., & Moore, S. A. (1997). On Automated Message Processing in Electronic Commerce and Work Support Systems: Speech Act Theory and Expressive Felicity. ACM Transactions on Information Systems, 15*(4), 321-367.*

Lawrence, E., Corbitt, B., Fisher, J., Lawrence, J., & Tidwell, A. (2001). Internet Commerce: Digital Models for Business *(2nd ed.). Brisbane, Australia: Wiley.*

Pitt, J., Kamara, L., & Artikis, A. (2001). Interaction Patterns and Observable Commitments in a Multi-Agent Trading Scenario. *Paper presented at the 5th International Conference on Autonomous Agents, Montreal, Quebec.*

Scheer, A. W. (1999). ARIS- Business Process Modelling. *Berlin: Springer.*

Smith, B. (1990). Towards a History of Speech Act Theory. In A. Burkhardt (Ed.), Speech Acts, Meanings and Intentions: Critical Approaches to the Philosophy of John R. Searle *(pp. 29-61). Berlin, New York: de Gruyter.*

Swenson, K. D. (1993). Visual Support for Reengineering Work Processes. *Paper presented at the Conference on Organizational Computing Systems, Milpitas, CA.*

Swindells, C., & Henderson, K. (1998). Legal Regulation of Electronic Commerce. The Journal of Information, Law and Technology (JILT), *3.(3), Available: http://elj.warwick.ac.uk/jilt/98-93/swindells.html [2001, 20th April]..*

Tanuan, M. (1998). Software Architecture in the Business Software Domain: The Descartes Experience. *Paper presented at the 3rd International Workshop on Software Architecture, Orlando, Florida.*

Tien, L. (2000). Publishing Software as a Speech Act. Berkeley Technology Law Journal, 14*(1), 1-57.*

Weske, M., Goesmann, T., Holten, R., & Striemer, R. (1999). A Reference Model for Workflow Application Development Processes. *Paper presented at the International Joint Conference on Work Activities Coordination and Collaboration, San Francisco, CA.*

Winograd, T., & Flores, F. (1987). *Understanding Computers and Cognition: A New Foundation for Design.* Reading, Mass: Addison-Wesley.

Yin, R. K. (1994). Case Study Research: Design and Methods. *Thousand Oaks, CA: Sage.*

BIOGRAPHIES

Lisa Soon is a research student in Information Technology at the Queensland University of Technology, Australia. Her current research interests include E-Commerce, E-Business, export trading and information management.

Phoebe Y. Chen is currently the Lecturer in the Faculty of Information Technology at the Queensland University of Technology, Brisbane, Australia. She completed her PhD in the field of Information Systems and Computer Science at the University of Queensland. Her research interests include scientific visualization, multimedia, bioinformatics, database, E-Commerce and web-based intelligent systems.