Business Software Specifications for Consumers: Toward a Standard Format

Shouhong Wang, University of Massachusetts, Dartmouth, USA

ABSTRACT

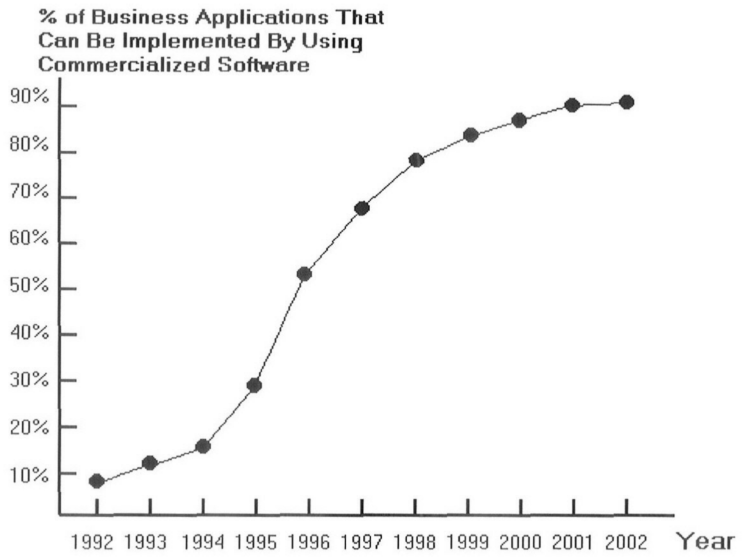
Commercialized business application software packages have been widely used to implement business information systems. In order to determine whether a software package meets the system needs, consumers must check the software specifications against the target system requirements. Since the commercial software industry does not have a standard format of software specifications for consumers, free-formatted descriptions of application software and ad hoc demos are commonly used in marketing software products, but are often too ambiguous for consumers to uncover the implemented capacity. This paper proposes a model of commercialized business software specifications for consumers. It suggests that software packages need to provide specifications for consumers in four aspects: business operations, user-computer interfaces, user-perceived inputs and outputs, and business rules. Using an example, the paper demonstrates the implementation of the model.

Keywords: information analysis techniques; information requirements analysis; information technology adoption; requirement specification; system analysis methods; system documentation; user orientation; user requirements

INTRODUCTION

Information systems analysis and design lies in the core of the information systems discipline. The techniques and approaches of information systems analysis and design are continually renovated. About 15 years ago, systems analysis and design projects were more likely to place the focal point on the use of databases and fourth-generation languages to implement real business information systems. Gradually, systems users and consultants found that commercialized business application software packages were readily available in the software market. According to the author's observations over the past decade in supervising 428 real-world MIS (management information systems) analysis and design projects, the percentage of business applications that can be implemented by using commercialized software packages has dramatically increased since 1994 (Figure 1). Clearly,

Figure 1. Increasing commercialized business software

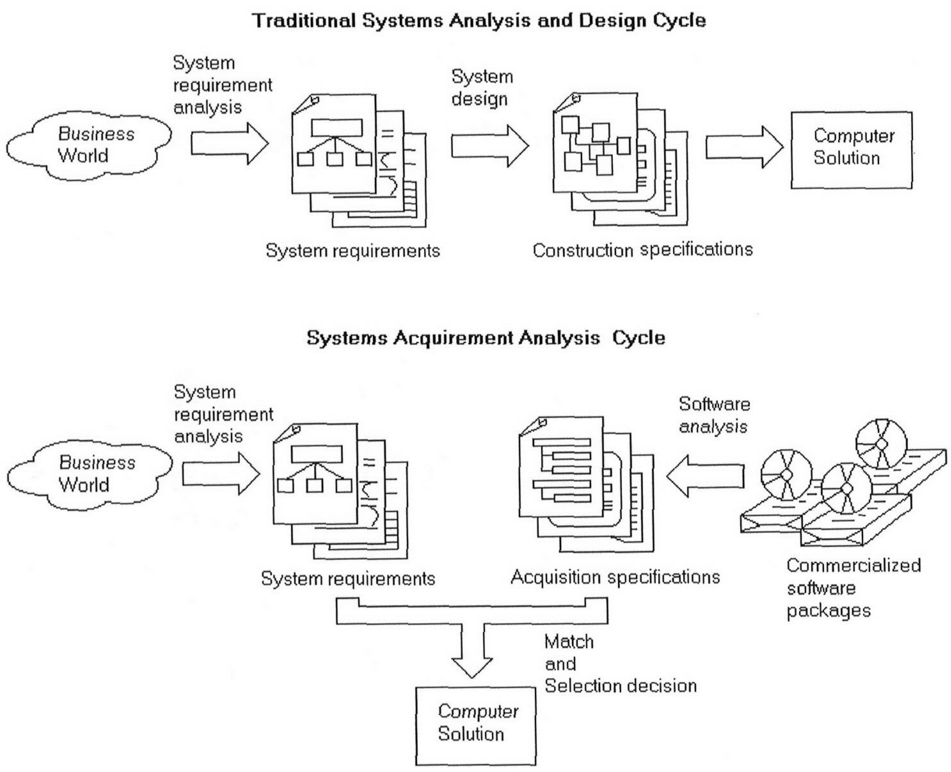


the phenomenon and the trend observed are based solely on the author's personal experience, and the claim may not be generally valid. Nevertheless, the observed real-world cases indicate that about 90% of small or middle-size business applications can be implemented by using off-the-shelf software packages. One can shop online to find a low-price and well-designed laundry management system, salon management system, and flower-delivery management system, to name a few.

As a result of the proliferation of commercialized business application software, for most business information technology professionals, the tasks of system design and implementation have been shifted from software construction to software system adoption. Nowadays, assessment of strategic values of software systems has become the central issue of systems development (Jurison, 2000). In this view, the theme of systems analysis and design for business enterprises has been shifted from system *construction* to system *acquisition*. In the traditional systems analysis and design cycle, system specifications are used for software development. However, in the system acquirement analysis cycle, specifications of software systems are needed for software consumers in choosing commercialized software packages that best match their system requirements. Accordingly, specifications for software construction and specifications for consumers, which are so called *acquisition specifications*, play different roles, as depicted in Figure 2.

The software industry has various specification instruments with de facto standards for business software development, such as data flow diagrams (DeMarco, 1978; Gane & Sarson, 1979), UML (unified modeling language;

Figure 2. Comparison of the two systems analysis cycles


Traditional Systems Analysis and Design Cycle
Business World
System requirement analysis
System requirements
System design
Construction specifications
Computer Solution
Systems Acquirement Analysis Cycle
Business World
System requirement analysis
System requirements
Acquisition specifications
Software analysis
Commercialized software packages
Match and Selection decision
Computer Solution

Rumbaugh, Jacobson, & Booch, 1999), and entity-relation diagrams (Chen, 1976). These instruments are used to describe the deep structures and system components for software construction, but are not meant for software feature description. Consumers of a commercialized business software package would like to have explicit specifications about the business process that can be carried out by the software package rather than the specifications for the construction of the software system (Wang, 2002). This is similar to the fact that consumers of computer hardware or cars never want to review the manufacturing blueprints in making a purchase decision. Since nowadays there is no commonly applied format of business software specifications that can be used for conveying the software features to consumers, sellers of commercialized business software packages use free-format descriptions and ad-hoc-style demos to market their products. Consumers have few guidelines for examination of software system utilities. Although this issue has been standing for some time (Trauth & Cole, 1992), few practical techniques of software specifications for consumers have been reported in the literature.

To facilitate consumer-centered system acquirement, a standard structure of acquisition specifications is imperatively needed. In this paper, we propose a model of business application software specifications for consumers to meet this challenge. This model formalizes the major aspects of business application software that have been identified in the literature as the most important factors for consumers. To focus on the primary issue of software specifications, this model excludes environmental requirements, such as minimal requirements for hardware and operating systems. The rest of the paper is organized as follows. The next section describes the major aspects of business software packages that are important to consumers and the structure of these aspects. "Implementation of the specification model: A case study" presents the implementation of this model structure through the use of XML (extensible markup language). Finally, the last section summarizes this study.

MAJOR ASPECTS OF BUSINESS APPLICATION SOFTWARE

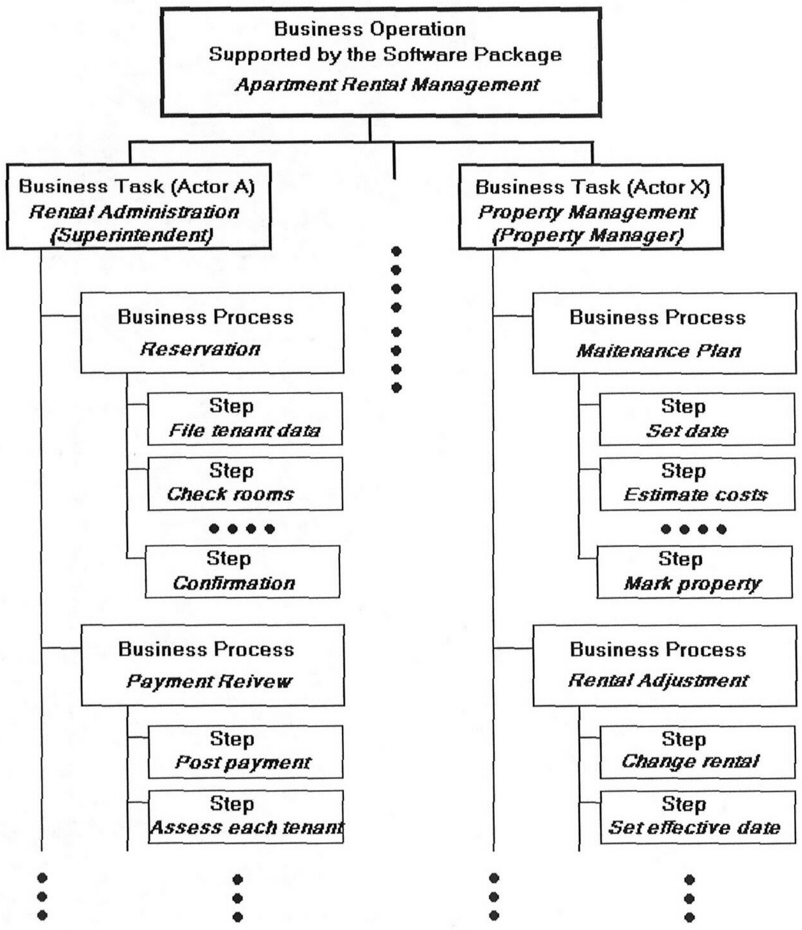
In formalizing the structure of business application software specifications for consumers, the first step is to identify the aspects of software systems that are most important for consumers in choosing packaged software. The second step is to integrate these major aspects into a structure that software consumers can easily understand and analyze. The third step is to implement the structure using the uniform data exchange language XML so that consumers can readily access acquisition specifications.

Business Operations: Tasks, Processes, and Steps

Business operations implemented by the application software package are the major concern for organizations (Kendall, 1994). In the view of the traditional structured modeling approach (e.g., DeMarco, 1978), which is simple and commonly adopted, a business operation is a hierarchy of suboperations. Using simple terminology for consumers, a business operation supported by a software system can be decomposed into three levels of suboperations: tasks, processes, and steps, which are corresponding to group and individual activities in business, as proposed.

1. *Task.* A business task is a set of business processes performed by a group of actors through the interaction between the actors and the software system to accomplish a specific outcome. The specifications of a task shall describe the business functionality of the system in accomplishing the task and the type of the actors. Here, actor is referred to as a particular role played by the user(s).
2. *Process.* A business process is a set of steps performed by a single actor through the interaction between the actor and the software system to carry out the associated task. The specifications of a process shall describe the accomplished functionality for the particular actor.

Figure 3. Specifications of business operation



3. *Step.* A step is a specific action performed by an actor through the interaction between the actor and the software system in carrying out the associated process.

Business operation specifications can be organized in a tree, as shown in Figure 3. As a practical example, segments (in italics) of an apartment-rental management system are included in the figure.

Note that the tree structure represents the hierarchical relationships between the tasks, processes, and steps, but not necessarily the sequence of the business operation. Here, the tasks, processes, and steps in a business operation tree may not be independent. Many real-world operations would involve complex interactions resulting in network structures. The use of dependent trees to describe network structures usually brings about redundant descriptions. Nevertheless, the tree structure of business operation specifications is easy to understand and is in compliance with XML that is used to implement the model.

User-Computer Interfaces

As application software has become widespread, the human-computer interaction is one of the most important aspects in software specifications (Diaper & Addison, 1992). It is so important because systems development specifications, such as the data flow diagram and UML, emphasize descriptions of functional and data requirements within the context of software engineering, but not within the context of software usability (Sutcliffe, 1989; Wang, 1995). To determine whether an application software package fits the business task requirements, one must specify user-centered cognitive aspects of usability (Anonymous, 1993; Benyon, 1992; Diaper, 1989; Harrison & Monk, 1986). Accordingly, the second aspect of acquisition specifications for commercial software packages is the user-computer interfaces.

A user-computer interface is the part of the software system that allows the user to interact with the system in carrying out the tasks. It includes the screen displays that provide navigation through the software system, as well as the screen displays that capture or generate data (Wang, 1995).

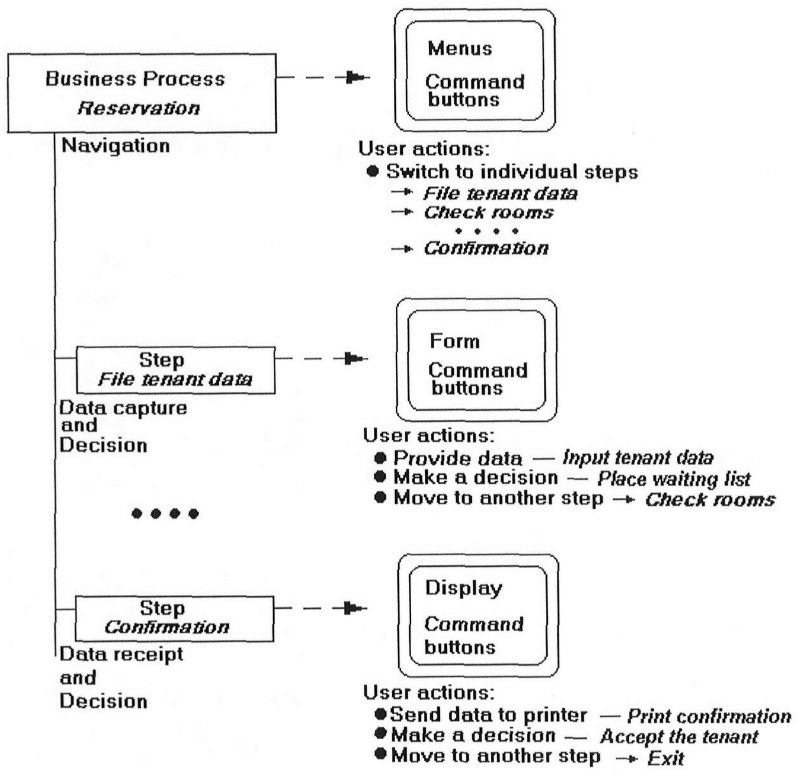
In specifying a software system, a user-computer interface is associated with a process or a step, as illustrated in Figure 4. Again, a practical example in italics is included in the figure.

Conceptually, there are three basic types of user-computer interfaces: navigation, data capture coupled with decision, and data receipt coupled with decision (Dennis & Wixom, 2003). A navigation interface is associated with a business process. It provides menus and command buttons that allow the user to proceed through the subsidiary steps. An interface for data capture coupled with decision provides forms and command buttons that allow the user to input data and make a business decision or move to another step. An interface for data receipt coupled with decision displays or prints data for the user and allows the user to make a decision or move to another step.

User-Perceived Inputs and Outputs

Originally, inputs and outputs of processes were considered to be central components of systems analysis and design in almost every systems analysis and design approach (Ballou & Pazer, 1985; Carey & McLeod, 1988). HIPO (hierarchy plus input, process, output; Stay, 1976) is a typical example of input- and output-driven approaches. Later, research (Srinivasan, 1985) indicated that user-perceived inputs and outputs, not system internal inputs and outputs, are the major measures of effectiveness of systems. For example, in evaluating a payment system, the user is interested in the receipt for a payment rather than the payment history data written to the disk. Research of contemporary object-oriented systems analy-

Figure 4. Specifications of user-computer interfaces


Business Process
Reservation
Navigation
Menus
Command buttons
User actions:
● Switch to individual steps
→ File tenant data
→ Check rooms
• • • •
→ Confirmation
Step
File tenant data
Data capture and Decision
Form
Command buttons
User actions:
● Provide data — Input tenant data
● Make a decision — Place waiting list
● Move to another step → Check rooms
● ● ● ●
Step
Confirmation
Data receipt and Decision
Display
Command buttons
User actions:
●Send data to printer — Print confirmation
●Make a decision — Accept the tenant
●Move to another step → Exit

sis (Wang, 1996) and information system planning (Li & Chen, 2001) has also confirmed this finding.

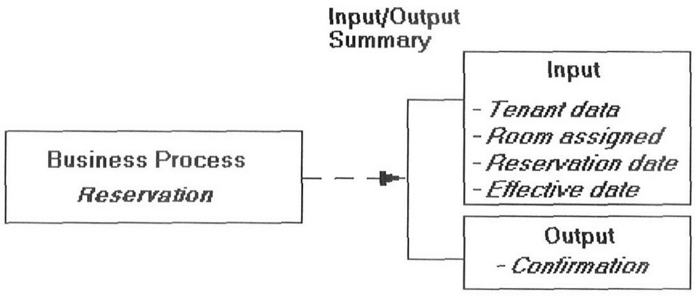
User-perceived inputs and outputs are associated with a business process, as illustrated in Figure 5. The specifications of a user-perceived input or output shall provide summarized information of the input and output contents.

Business Rules

Business rules are constraints or guidelines for business operations. They specify the relationships between an anticipated condition and expected actions or outcomes. Using the currently available computer techniques, business rules are implemented through coded decision procedures or data models. Yet, there is a lack of systematic techniques for mapping business rules and the software system onto each other (Amghar, Meziane, & Flory, 2000). As the perspectives of business rules are crucial for software systems acquirement, important business rules implemented in the software system must be described in an explicit way (Hale, Sharpe, & Hale, 1999).

In this proposed model, work flows are specified in the business operation and interface specifications, and are not con-

Figure 5. Specifications of user-perceived inputs and outputs



sidered to be business rules. Here, business rules are referred to as those constraints associated to a task or a process. Software specifications for consumers shall describe two types of business rules implemented in the software system: user-defined and built in.

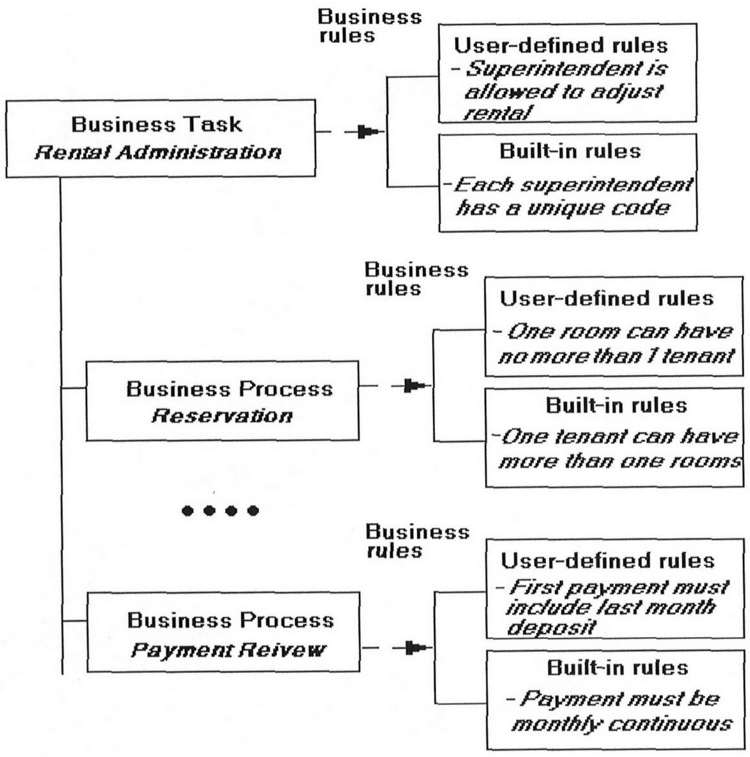
1. *User-defined business rules:* The specifications of a user-defined business rule shall describe the general formula of the rule and how the user can define his or her own parameters during the installation of the software system. The specifications shall also indicate the default settings for those user-defined business rules.
2. *Built-in business rules.* The specifications of a built-in business rule shall describe the formula of the rule that has been implemented in the software system.

Discussion

The above model of application software specifications for consumers de-emphasizes several aspects of design specifications for software construction, as discussed below.

1. *System decomposition:* In systems analysis and design, system decomposition is one of the important issues. This is because the system modules are the construction units for the software development. On the other hand, the software users are not particularly concerned with these construction units, as long as the system supports the required business operation. For instance, in object-oriented systems analysis and design, objects are the system modules. However, few software companies specify the objects of their systems for consumers.
   In this proposed model, the structure of specifications is generally not a system decomposition solution, although it could be a start point for structured systems analysis and design. Specifically, the structure of business operations provides a user's view of the system instead of the designer's view of the system. The definitions of business tasks, processes, and steps, as well as

Figure 6. Specifications of business rules


Business Task
Rental Administration
Business rules
User-defined rules
- Superintendent is allowed to adjust rental
Built-in rules
- Each superintendent has a unique code
Business Process
Reservation
Business rules
User-defined rules
- One room can have no more than 1 tenant
Built-in rules
- One tenant can have more than one rooms
Business Process
Payment Reivew
Business rules
User-defined rules
- First payment must include last month deposit
Built-in rules
- Payment must be monthly continuous

the relationships between them, are not necessarily corresponding to the system modules. The issues of module coupling (how modules are interrelated) and cohesion (how the lines of programming code are related to each other) in the structured systems analysis and design are not relevant here.

2. *Data modeling:* Data models are certainly important for systems construction and software implementation. However, reviewing data models would involve excessive efforts for a consumer. Also, conceptual data modeling techniques vary depending upon systems development tools (Topi & Ramesh, 2002). For instance, the traditional entity-relationship model and object-oriented models use different semantics at the conceptual level. Consumers may not be familiar with the particular data modeling method used for a software system. In fact, few software producers provide data models for their consumers while marketing their products.
3. *Internal states of processes or objects:* In designing a software system, states of a process or an object are used to describe the dynamic aspects of the process or object and specify the instances that evolve over the time. However, compared with industrial

control engineering systems, business information systems use less states descriptions because business rules and operation structures describe system states in an intuitive way (Wang, 2002).

IMPLEMENTATION OF THE SPECIFICATION MODEL: A CASE STUDY

To experience a practical application of the proposed model, a precommercialized software system with a reasonable scale was specified using the method. It is the Apartment Rental Management System developed by a small real estate and software company in southern New England, where the author was a project consultant for the software development. One of the company's founders was a fairly sophisticated computer end user, but was not a system developer. He was leading the software development team. The other founder was a real estate and rental businessman and was familiar with the rental-management software market, but had little computing background. He played the role of marketing manager and facilitated the software development team to fully understand consumers' requirements.

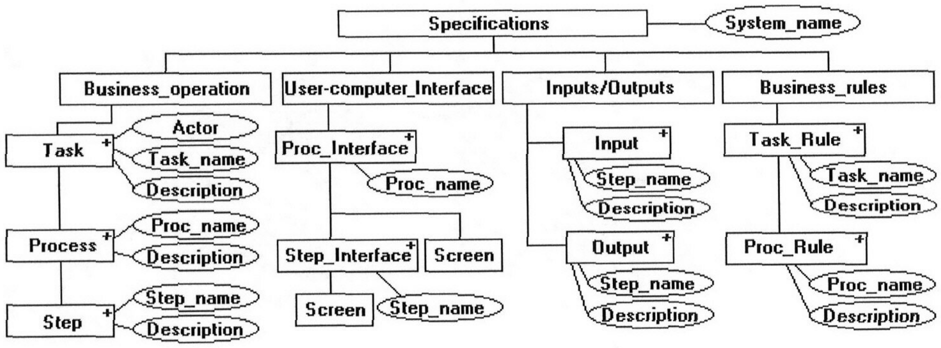
The system was implemented in Microsoft Access. Using it, the user was allowed to perform various tasks on apartment rentals, including property management, tenant record maintenance, reservation, billing and payment, damage and property maintenance, and cash flow management. The company reviewed several competitors' software products and found that the current diversified forms of specifications do not describe their products to consumers adequately. The work team felt that a formalized structure was needed to specify the software system, and decided to use the proposed model.

One guiding principle of the proposed user-centered acquisition specification model is to standardize the specification forms for software consumers. XML (W3C, 2003) enables software companies to develop consistent specification documents with a common format across the application domain. XML documents are flexible for customization. Software specifications encoded in XML make it easy for consumers to retrieve specification information and compare components of various software systems. Thus, XML was used for documenting the acquisition specifications for the Apartment Rental Management System.

The XML document of the software system acquisition specifications presents a structure tree with actualized data elements and hyperlinks between these elements. For an intuitive presentation, the data tree represented by the XML document, instead of the XML Schema or DTD (document type definition), is depicted in Figure 7 using the common convention of notations in the XML field. Here, a rectangle represents an element, an ellipse represents the attribute of the element, and a plus sign symbolizes multiple entries of the element.

In Figure 7, each of the four aspects of acquisition specifications has its branch in the data tree, and can be easily presented to the user through XSLT (extensible style language transformation). As discussed in the previous section, there are

Figure 7. Data tree represented by the XML document of acquisition specifications



connections between those elements in different aspects that are not shown explicitly in the figure. For instance, a process can have its user-computer interface and its business rules. These connections are presented in Figure 7 implicitly through common attribute names (e.g., process name) for the associated elements (e.g., process and its user-computer interface). XLink was used to implement these implicit connections through hyperlinks. Figure 8 shows a small portion of XML segments that illustrate the hyperlinks between the payment review process and its associated user-computer interface and business rules. The XML document of the software specifications allows consumers to download from the Internet and is more flexible to use than documents with other formats. To compare two or more software systems, one can develop an alternative matrix for these systems and evaluate them based on the hierarchical specifications. Using the well-established analytic hierarchy process (AHP; Saaty, 1980) method, for instance, an adoption decision can be derived based on the conceived alternative matrix.

The proposed model provides a template for the company in specifying the features of the Apartment Rental Management System for potential consumers. It helps the company in two aspects. First, this specification structure is used as a marketing tool that clearly documents the product for consumers. It is also used as an internal communication tool that bonds the marketing side and the development-team side in the company. As a result, the company now has a clear vision of its product. Second, the company is now using this specification model to compare the product with its competitors' products. This allows the company to have better understanding of the strength and weakness of the product in order to penetrate the market. The company has found that the software system is well documented for consumers using the organized acquisition specifications. It has highlighted the features of the software system that are unique to other competitors' products. The

Figure 8. XLink implements the hyperlink connections

```
<?xml version="1.0"  ?>
...
<Process     ProcessName = "PaymentReview"
             ProcessDescription =
              "1. For each unit
                1.1. Review outstanding payment.
                1.2. Specify an extension if there is a reason
                     of the late payment.
                1.3. Print a late payment notice to the tenant.
               2. Print all records of outstanding payment."  >
   <Link  xlink:type = "simple"
          xlink:href = "#UCI_PaymentReview" >
          User-computer interface for this process
   </Link>
   <Link  xlink:type = "simple"
          xlink:href = "#BR_PaymentReview"  >
          Business rules for this process
   </Link>
   ...
   <Step ... >
   ...
</Process>
...
<UserInterface  UCI_ProcessName = "UCI_PaymentReview"
                Screen = "UCI_PaymentReview.gif"   >
   <StepUCI ...>
   ...
</UserInterface>
...
<ProcessRule   BR_ProcessName = "BR_PaymentReview"
               Description =
                "1. A due date is set for monthly payment.
                    The default due date is 15th of the month.
                 2. Payment for the final month of the lease
                    is the one-month deposit."    >
</ProcessRule>
...
```

specification model is easy to implement. However, the company has found that there are needs for integration of the acquisition specifications and users' manuals, which would make it easier for consumers to learn the system. Currently, the company is undertaking further analysis on this issue.

It is our experience that this model offers a pragmatic and useful structure of software specifications for consumers. Clearly, more experiments need to follow in order to make further validation.

SUMMARY

The proliferation of business application software packages has introduced new tasks of acquirement analysis for the systems analysis and design field. To facilitate the communication between software consumers and software builders, application software specifications for consumers must be user-centered instead of builder-centered. This paper proposes a model of software acquisition specifications for consumers. It suggests that business operations, user-computer interfaces, user-perceived inputs and outputs, and business rules are the four essential aspects of acquisition specifications. The four aspects are organized into a static tree, and can be hyperlinked into a complex network for the consumer to examine.

In the history of management information systems, methods for systems specifications have been dominated by computer software builder-centered approaches. On the other hand, the fast growth of the commercialized business software packages on the market demands concise and precise consumer-centered specifications of business software. The proposed acquisition specification model bridges the gap between the business requirement definitions and the software implementation descriptions for consumers. Compared with the current system specification techniques such as data flow diagrams and UML, the proposed acquisition specification model is easy to understand for consumers; yet, it mirrors the utilities of application software packages explicitly for consumers. Once the structure of acquisition specifications becomes standardized, it will be possible to investigate the general mapping relationship between acquisition specifications and software construction specifications.

ACKNOWLEDGMENTS

The author is indebted to the associate editor and three anonymous referees for their valuable comments for the revision of this paper.

REFERENCES

Amghar, Y., Meziane, M., & Flory, A. (2000). Using business rules within a design process of active databases. *Journal of Database Management, 11*(3), 3-15.

Anonymous. (1993). Two communities, two languages. *Communications of the ACM, 36*(4), 113.

Ballou, D. P., & Pazer, H. L. (1985). Modeling data and process quality in multi-input, multi-output information. *Management Science, 31*(2), 150-162.

Benyon, D. (1992). The role of task analysis in systems design. *Interacting with Computers, 4*(1), 102-123.

Carey, J. M., & McLeod, R., Jr. (1988). Use of system development methodology and tools. *Journal of Systems Management, 39*(3), 30-35.

Chen, P. P. (1976). The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems, 1*(1), 9-36.

DeMarco, T. (1978). *Structured systems analysis and design.* New York: Yourdon.

Dennis, A., & Wixom, B. H. (2003). *Systems analysis and design* (2nd ed.). New York: John Wiley & Sons.

Diaper, D. (1989). The discipline of HCI. *Interacting with Computers, 1*(1), 3-5.

Diaper, D., & Addison, M. (1992). Task analysis and systems analysis for software development. *Interacting with Computers, 4*(1), 124-139.

Gane, C., & Sarson, T. (1979). *Structured systems analysis: Tools and* *techniques*. Englewood Cliffs, NJ: Prentice-Hall.

Hale, D., Sharpe, S., & Hale, J. E. (1999). Business-information systems professional differences: Bridging the business rule gap. *Information Resources Management Journal, 12*(2), 16-25.

Harrison, M. D., & Monk, A. F. (Eds.). (1986). *People and computers: Designing for usability*. UK: Cambridge University Press.

Jurison, J. (2000). Perceived value and technology adoption across four end user groups. *Journal of End User Computing, 12*(4), 21-28.

Kendall, J. E. (1994). End user reengineering: Breaking the rules systems developers wrote. *Journal of End User Computing, 6*(2), 24-26.

Li, E. Y., & Chen, H. G. (2001). Output-driven information system planning: A case study. *Information & Management, 38*(3), 185-199.

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The unified modeling language reference manual*. Boston: Addison-Wesley.

Saaty, T. L. (1980). *The analytic hierarchy process: Planning, priority setting*. New York: McGraw Hill.

Srinivasan, A. (1985). Alternative measures of system effectiveness: Associations and implications. *MIS Quarterly, 9*(3), 243-253.

Stay, J. F. (1976). HIPO and integrated program design. *IBM Systems Journal, 15*(2), 143-154.

Sutcliffe, A. (1989). Task analysis, systems analysis and design: Symbiosis or synthesis? *Interacting with Computers, 1*(1), 6-12.

Topi, H., & Ramesh, V. (2002). Human factors research on data modelling: A review of prior research, an extended framework and future research directions. *Journal of Database Management, 13*(2), 3-19.

Trauth, E. M., & Cole, E. (1992). The organizational interface: A method for supporting end-user of packaged software. *MIS Quarterly, 16*(1), 35-53.

Wang, S. (1995). Object-oriented task analysis. *Information & Management, 29*(6), 331-341.

Wang, S. (1996). Toward formalized object-oriented management information systems analysis. *Journal of Management Information Systems, 12*(4), 117-141.

Wang, S. (2002). Teaching the object-oriented approach for business information systems analysis and design. *Journal of Informatics Education and Research, 4*(1), 17-26.

W3C, Extensible Markup Language (XML). (2003). Retrieved November 2, 2003, from http://www.w3c.org/XML/

Shouhong Wang is a professor of business information systems at the University of Massachusetts, Dartmouth (USA). He received his PhD (1990) in information systems from McMaster University, Canada. His research interests include systems analysis and design, artificial intelligence in business, and electronic commerce. He has published more than 60 papers in academic journals, including Information Resources Management Journal, Journal of Management Information Systems, Information & Management, International Journal of Information Management, Human Systems Management, IEEE Transactions on Systems, Man, and Cybernetics, Computers & Operations Research, Management Science, European Journal of Operational Research, OMEGA, Decision Sciences, IEEE Transactions on Patter Analysis and Machine Intelligence, Journal of The Operational Research Society, INFORMS Journal on Computing, Computers & Industrial Engineering, Knowledge and Information Systems, International Journal of Intelligent Systems in Accounting, Finance, and Management, Neural Computing & Applications, Journal of Electronic Commerce Research, Industrial Management and Data Systems, *and others. He has been a biographee of* Canadian Who's Who *since 1995, a biographee of* Marquis Who's Who in America *since 2000, and a biographee of* Marquis Who's Who in American Education 2004.