Electronic Preprint for Journal of Information Processing Vol.23 No.4

Regular Paper

ReVAMP: Requirements Validation Approach using Models and Prototyping — Practical Cases of Requirements Engineering in End-User Computing

Shinobu Saito[1,2,a)] Jun Hagiwara[1] Tomoki Yagasaki[1] Katsuyuki Natsukawa[2]

Received: September 16, 2014, Accepted: March 4, 2015

Abstract: Prototyping practices are widely used. Requirements engineers develop screen prototypes with paper or HTML. However, feedback on the prototypes has limited effectiveness. Screen prototypes are mainly useful for reviewing only user interface requirements. To cope with this situation, we propose a requirements validation approach using models and prototyping (ReVAMP). This approach provides customers with a set of requirement models and a system prototype generation tool for trial use. A generated system prototype is implemented with both business application features and access control features. Thus, customers could give requirements engineers more practical feedback on requirements for not only a user interface but also other aspects of a target system. To evaluate the proposed models and tool, we introduce two business information system development projects in which the proposed approach was applied.

Keywords: requirements models, prototyping, business application feature, access control feature

1. Introduction

The validity of the prototyping approach has been widely discussed in the requirements engineering domain [1], [2]. This approach could contribute to reducing requirements errors, namely, misunderstandings between customers and requirements engineers, ambiguous requirements, and conflicts among requirements. Customers can review system screens of a development system by using a prototype such as paper or HTML. Their feedback on the prototype could help requirements engineers to understand precisely the requirements requested by customers. A lot of commercial and open source software (OSS) prototyping tools have been introduced [3], [4], [5]. A number of business information system (BIS) development projects have been using these prototyping tools for requirements review in the requirements definition phase.

However, feedback on prototypes has only limited effectiveness. Screen prototypes with paper or HTML are mainly useful for reviewing only requirements related to user interfaces; display items of the screen, layout of the screen, and screen transitions. If a prototype becomes advanced enough for customers to be able to review requirements related to business application features executed by a system, customers could give requirements engineers more practical feedback on requirements for not only a user interface but also other aspects of a target system. We call this a "system prototype."

As this is seen as a system prototype, we took care of the access control features that need to be considered. Generally, access control features are composed of two functions: user authentication and access authentication [6]. First, a user authentication function is usually implemented as a user log-on screen that authenticates the validity of user information, namely, user account name and password. Second, an access authorization function restricts system users to the access content, for examples, system screen and system resource of the BIS on the basis of the roles of the system users. Generally, there are several roles (e.g., general user, manager, and system administrator). If log-in users who have no authority try to access restricted content, the access authorization function prohibits them from accessing it. To take this approach with a system prototype, the prototype has to include not only business application features but also these access control features.

In our prior research [7], [19], we studied a prototype generation tool for requirements review in the requirements definition phase. This tool generates a system prototype from three types of requirements models: business process, business rule, and user interface. We examined the effectiveness of our tool to aid customers in reviewing requirements with the generated system at review meetings. However, a limited number of customers had a short meeting-time, that is to say, on a few hours, to use the system prototype for reviewing the requirements. It remains to be researched whether and how the system prototype would help customers with multiple roles to review requirements if they could

[1] NTT DATA CORPORATION, Koto, Tokyo 135–8671, Japan
[2] NTT CORPORATION, Minato, Tokyo 108–0075, Japan
[a)] saitousnb@nttdata.co.jp

operate the system prototype as a trial use for a certain period (e.g., several weeks). The system prototype needs to become sophisticated in order to provide appropriate functions to each role. As mentioned before, our view is that the system prototype needs to include both features.

In this paper, we propose a requirements validation approach using models and prototyping (ReVAMP). This approach provides customers with a set of requirements models and a system prototype generation tool for trial use. We define five types of requirements models. The five types are categorized into two groups. One group for business application features includes three models: business process, business rule, and user interface. The other group for access control features includes two models: user authentication and access authorization. Our tool can transform the requirements models into a system prototype, which is a web application system. The generated system prototype is implemented with both business application features and access control features. We also introduce two projects in which the proposed tool has been applied.

As shown in **Table 1**, we treat seven requirements categories as a validation target. Each category is included in either business application features or access control features. Two rightmost columns describe two types of prototype: Screen Prototype and Proposed System Prototype. Requirements categories 1-3 (Display Items of Screen, Display Layout of Screen, and Screen Transitions) are related to user interfaces of a system. Requirements category 4 (Search Results of Input Item) and category 5 (Decision Results of Input Items) respectively correspond to the search function and the decision function of a system. Requirements category 6-7 (User Authentication and Access Authentication) comprise the access control features of a system. In the table, the marking "X" at the cross point means that the corresponding requirements category could be reviewed by the corresponding prototype. For example, categories 1-3 could be reviewed by the screen prototype. On the other hand, categories 4-7 could be reviewed only using the system prototype. So, in this paper we aim to review the entire category utilizing the proposed system prototype.

The structure of this article is as follows. In Section 2, we describe related works. In Section 3, we describe our approach, ReVAMP. We define the requirements models in Section 4. We also present an interrelation of these models. In Section 5, our system prototype generation tool is described. In Section 6, case studies are introduced. We discuss the results in Section 7. Finally, we give a conclusion in Section 8.

Table 1 Requirements category to be reviewed by prototype.

Requirements Category	Screen Prototype	Proposed System Prototype
Business Application Features		
1. Display Items of Screen	X	X
2. Display Layout of Screen	X	X
3. Screen Transitions	X	X
4. Search Results of Input Items		X
5. Decision Results of Input Items		X
Access Control Features		
6. User Authentication		X
7. Access Authentication		X

2. Related Works

An evolutionary prototyping approach was proposed [1], [2]. In the approach, a prototype created for requirements review was continuously improved and adding-implemented for the final BIS even after completing the requirements definition phase. The evolutionary prototyping approach does not necessarily aim to confirm customer's requirements within a requirements definition phase. On the other hand, our approach focuses on customer's requirements validation until the completion of the requirements definition phase.

Chusho et al. [11] proposed a web application development environment that includes the functions of a visual modeling tool, model transformation, and source code generation. Making use of the development environment, customers could create a business application system on their own behalf. In the meantime, customers have to be well found in programming knowledge in order to reflect their needs and intentions in the developed application. In comparison with the environment, our approach and tool enable customers without programming knowledge to generate system prototypes which have both business application and access control features.

Ogata et al. [12] proposed a prototyping generation tool. The tool generates a prototype including a UI (HTML) from UML diagrams (activity, class, and object diagrams). The generated prototype provides a function to create the perpetuating data. Thus, customers could see the data that they entered in the previous screens. Therefore, during the requirements review, they could check the interactions between system user and system. The prototype enables customers to understand and review the requirements of the developing system. However, the generated system provides only limited functions. Customers cannot review search or decision functions.

Mibe et al. [13] analyzed the use cases and categorized them into several use case patterns. They also created system screen patterns and program components with respect to the screen patterns. A prototype is generated from the use case patterns, the system screen patterns, and corresponding program components. This prototype contributes to requirements elicitation. In contrast, our tool is used to specifically generate a system from requirements models resulting from requirements elicitation. Our system prototype contributes to requirements validation.

Some researchers [14], [15] have proposed methodologies and tools that generate a prototype from a state-chart UML diagram. In these methodologies, a state-chart diagram is not created at the first step. It is generated from different artifacts during an incremental process. These artifacts are other UML diagrams (sequence, collaboration, and class diagrams), object constraint language (OCL), and an original model. These methodologies focus on UML models in the system design phase. However, we use a set of requirements models to review requirements in the preceding (i.e., requirements definition) phase.

Kariyuki et al. [16] proposed an automation tool for generating

test scenarios and a skeleton code. This tool supports requirements engineers for evaluating completeness of use cases. However, this tool only helps them review use cases which are related to a limited aspect of business application features.

Additionally, the above mentioned research has given little consideration to the access controls feature in the context of a prototyping approach. Prototypes of them are mainly used for review on the business application feature.

3. ReVAMP

3.1 System Prototype for Requirements Validation

For requirements verification and validation, as shown in **Fig. 1**, we define four requirements engineering processes that use our models and system prototype. In the first process, requirements engineers elicit requirements from customers.

They create requirements models on the basis of the customer's requirements in the second process. Using the models, the tool generates a system prototype in the third process. After that, in the final process, customers do a trial use on the generated system prototype. They give feedback to requirements engineers. Throughout these processes, requirements engineers try to understand customer's requirements correctly and completely.

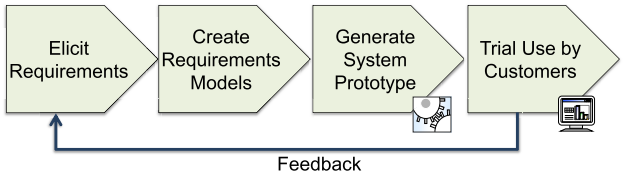

Fig. 1 Requirements engineering process using models and system prototype.

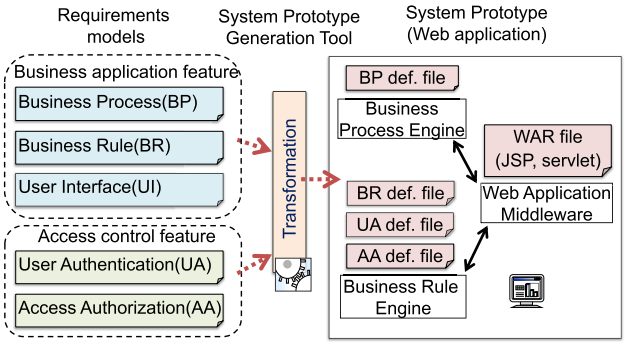

Fig. 2 Requirements models and system prototype generation tool.

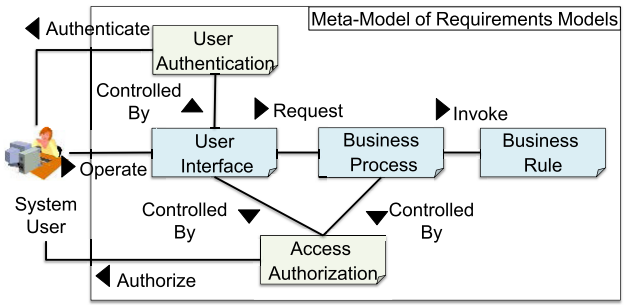

Fig. 3 Meta-model of requirements models.

3.2 Requirements Models and System Prototype Generation Tool

In this approach, we define five types of requirements models. These models are inputs of our system prototype generation tool. **Figure 2** shows an overview of the requirements models and system prototype generated by the tool. Our tool transforms the models into multiple definition files and a WAR file. Each file is deployed on the business process engine, business rule engine, or web application middleware. In this way, a web application can be executable as a system prototype.

3.3 Meta-Model of Requirements Models

Figure 3 shows a meta-model of the five types of requirements models. A set of the five types represents both business application and access control features.

User interface represents "Display Items of Screen," "Display Layout of Screen," and "Screen Transitions" which are requirements categories 1-3 of Table 1, respectively. Business process represents search function requested by user interface, which corresponds to requirements category 4 "Search Results of Input Items." Business rule represents decision function invoked by business process, which corresponds to category 5 "Decision Results of Input Items." User authentication and access authorization represent categories 6-7 of access control features respectively.

4. Requirements Models

4.1 Business Process

Figure 4 shows a sample business process based on the requirements analysis method [10], which is related to the business process engine in the tool. The columns of the business process are No (number of process), Process name, Function type, Data (name of data table), Data item (name of column of table), Business rule ID, and Transition destination.

In this figure, the process name of the first process is "Search Reservation." The first function type, "Search," is the search action in the Data "Reservation" table. The setting of a Data Item ("Reservation_No," "Reservation_Name," "Staff_Name," "Reservation_Date") is the search key. The second Function type, Output, represents the result of the above search action. The next process, "Search Credit_Status," defines the invocation of the busi-

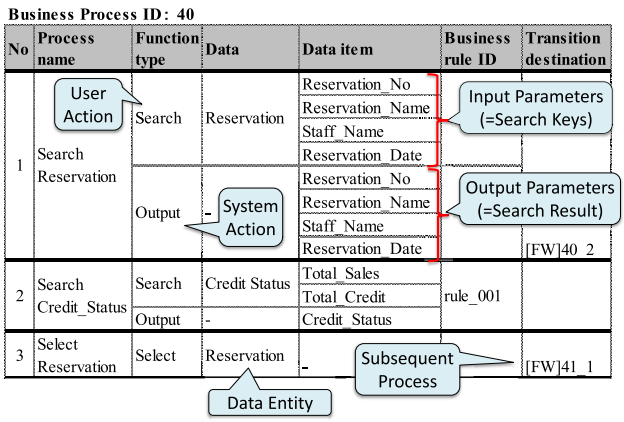

Fig. 4 Business process.

Table 2 List of function types.

No	Function type	Description
1	Create	Create new record
2	Search	Search record
3	Output	Display record list of the research
4	Select	Select record
5	Edit	Edit contents of record
6	Commit	Commit edited record
7	Display	Display records and then quit

Table 3 List of composite patterns of function types.

No	Composite pattern	Remarks
1	(Search→Output) →Select	"(Search → Output)" can be described repeatedly.
2	(Search→Output)→Display	"(Search → Output)" can be described repeatedly.
3	(Search→Output)→Edit→Commit	"(Search → Output)" can be described repeatedly.
4	Create → Edit → Commit	"Search" can be described in the sentence.

"Output" is optional.

Business rule ID: rule_001

Credit_Status	Condition	Decision
	Comparison between Total_Credit and Total_Sales	Credit Status
	Total_Credit = 0	No Credit
	Total_Sales > Total_Credit	Partial Credit
	Total_Sales = Total_Credit	Complete Credit
	Total_Sales < Total_Credit	Over Credit

Fig. 5 Business rule.

ness rule. This process includes a business rule identifier (business rule ID = "rule_001") to invoke the corresponding business rule. The first function type, Search, of the process defines the condition of the business rule. The setting of the Data item ("Total_Sales" and "Total_Credit") is the input parameter for the business rule. The second function type, Output, represents the result from executing the business rule. The output parameter of this is the Data item "Credit_Status." After completion of the second process, five outputs from the two processes are forwarded to the corresponding system screen. The screen shows a record list for reservation and Credit_Status. The third process, Select Reservation, represents the selection from the record list on the screen made by the system user.

Table 2 shows a list of the function types for the tool. There are seven types: create, search, output, select, edit, commit, and display. As shown in Fig. 4, one business process includes more than one function type. **Table 3** shows a list of the composite patterns of the function types. We define four types of composite patterns for the tool. Figure 4 corresponds to the No.1 composite pattern.

4.2 Business Rule

Figure 5 shows a sample of a business rule that is composed of the condition and decision columns. The rule is invoked by a process of the business process. Four expressions are described in the condition column (comparison between Total_Credit and Total_Sales). In the decision column (Credit Status), four decisions ("No Credit," "Partial Credit," "Complete Credit," and "Over Credit") are described. The decisions correspond to each expression in the condition column. After completion, the selected decision value is returned to the process that invokes the business rule.

Business Process ID: 40

No	Process name	Function type	Data	Data item	Business rule ID	Transition destination
1	Search Reservation	Search	Reservation	Reservation_No Reservation_Name Staff_Name Reservation_Date		
		Output	-	Reservation_No Reservation_Name Staff_Name Reservation_Date		[FW]40_2
2	Search Credit_Status	Search	Credit Status	Total_Sales Total_Credit	rule_001	
		Output	-	Credit_Status		
3	Select Reservation	Select	Reservation	-		[FW]41_1

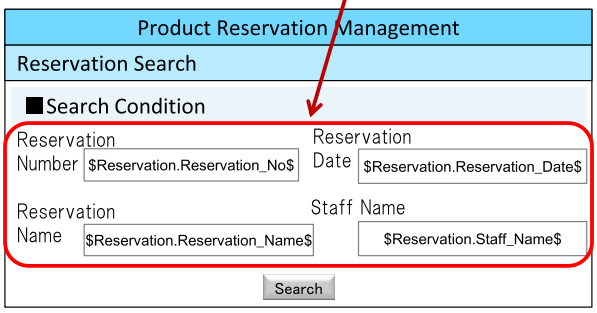

Fig. 6 User interface and business process.

User table

User name	Password	Role_ID
"user01"	"XXXX"	"1"
"mngr01"	"YYYY"	"2"
"admn01"	"ZZZZ"	"9"

Role-Initial Screen Mapping rule: UA_001

Role-Initial Screen Mapping	Condition	Decision
	Role_ID	URL
	"1"	"/Search_40.jsp"
	"2"	"/Search_40.jsp"
	"9"	"/Admin_menu.jsp"

Fig. 7 User table and role-initial screen mapping rule table.

4.3 User Interface

The user interface defines the layout of the input and output forms. The file format used is HTML. As shown in **Fig. 6**, each data item of the business process is mapped into the corresponding form in the user interface. "$Data.Data Item$" is described in the corresponding form as the mapping relation between the user interface and the business process. With reference to the description, the tool generates Java Server Pages (JSP) for the web system from the HTML file.

4.4 User Authentication

As shown in **Fig. 7**, this model is comprised of two tables: User table and Role-Initial Screen Mapping rule table.

User table contains user information for system log-on. User information (user name and password) and Role ID are described in the user table. The second table indicates the initial screen for each "Role_ID." The role-initial screen mapping rule table represents the relationship between the role of a user and the

URL of the initial screen. When a user logs in to the web system, the initial screen is forwarded to the web browser of the client of the log-in user with reference to the table. For example, when "user01" logs on the system, the system displays the screen "/Search_40.jsp."

4.5 Access Authorization

Figure 8 shows a sample model of access authorization. In the decision table, the labels in the condition column are URL and Role_ID, and the label in the decision column is Permission. The generated web system performs access authorization with reference to the rule of the model.

Access Authorization rule: AA_001

Access Authorization	Condition		Decision
	URL	Role_ID	Permission
	"/Search_40.jsp"	1	TRUE
	"/Search_40.jsp"	2	TRUE
	"/Search_40.jsp"	9	TRUE
	"/Admin_menu.jsp"	1	FALSE
	"/Admin_menu.jsp"	2	FALSE
	"/Admin_menu.jsp"	9	TRUE

Fig. 8 Access authorization rule.

4.6 Interrelation among Requirements Models

Figure 9 shows the interaction of the requirements models. This is a sample scenario for product reservation management. The sequence of the scenario is as follows.

(1) From the log-in screen, user information (Username = "user01" and Password = "XXXX") is posted.

(2) The log-in user is authenticated on the basis of the user table for User Authentication. After that, the role (Role_ID = "1") is given to the log-in user ("user01").

(3) In reference to the Role-Initial Screen Mapping rule table of User Authentication, the URL of the initial screen (URL = "/Search_40.jsp") is selected on the basis of the value of the condition (Role_ID = "1").

(4) User interface 1 is a screen for doing a reservation search. From it, the search condition (Staff Name = "Mitaka Saburo") is submitted to the first process of the business process (Business process ID = "40"). The first process is mapped to the corresponding servlet (URL = "/40_1").

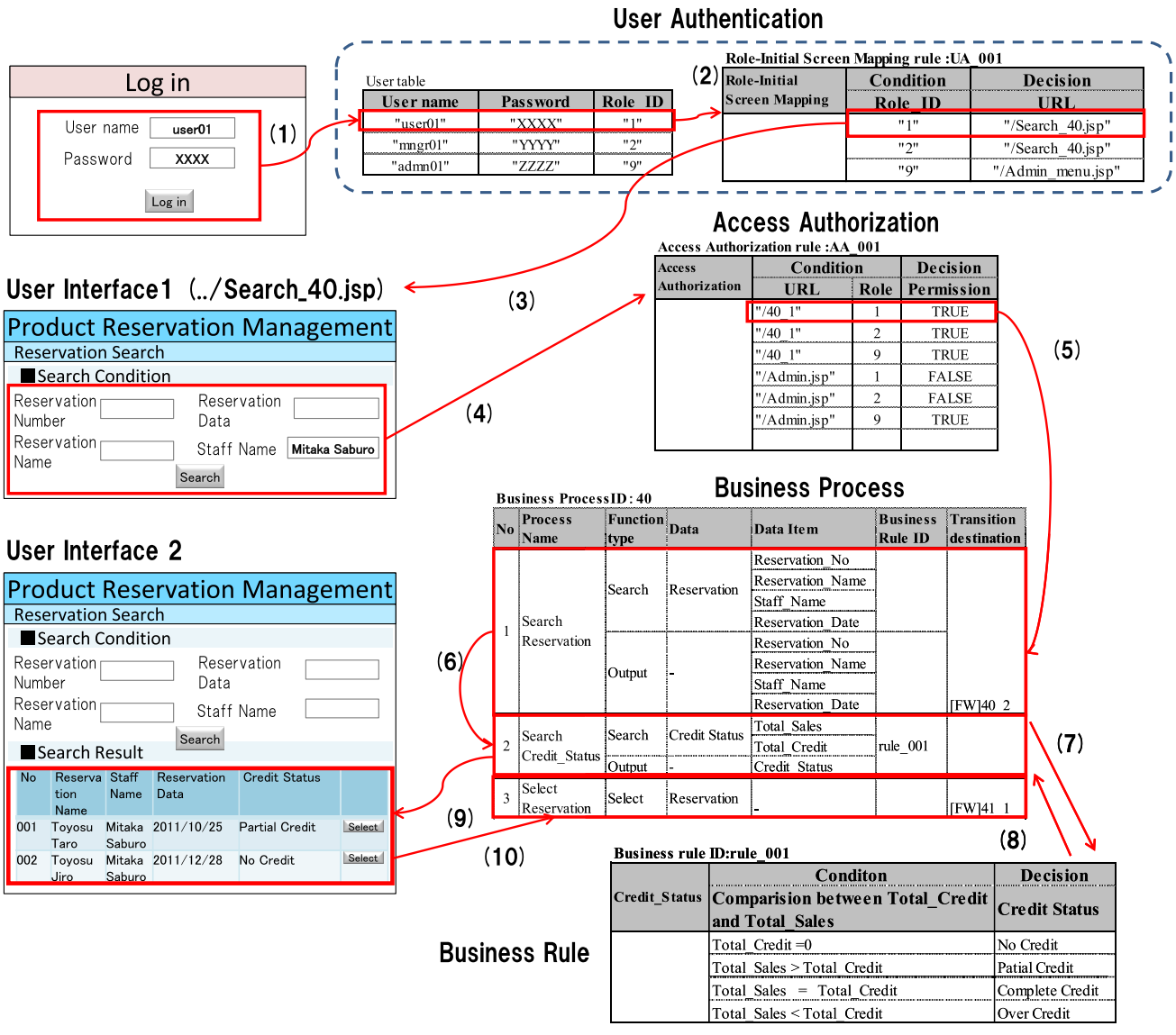

Fig. 9 Requirements models and their interrelation.

(5) On the basis of the requested URL and the role of the requester, the access authorization rule is invoked. If the decision is "TRUE," the corresponding process is activated. Because the permission of the user ("user01") is "TRUE," the first process (40_1 process) is activated.

(6) The 40_1 process is the "Search Reservation" action. After it is conducted, the executing process transits from the first process to the second process (40_2 process).

(7) The 40_2 process invokes a business rule. Business rule ID ("rule_001") is described in the "Business Rule" column for the 40_2 process. In the process, a set of data items ("Total_Sales," "Total_Credit") is the parameter input to the business rule. These parameters are submitted as values of the condition of the rule. Similarly, the data item "Credit_Status" is the parameter output from the business rule. This parameter corresponds to the value of the decision of the business rule.

(8) The business rule is executed by the 40_2 process. This rule, "rule_001," receives the condition data from the 40_2 process and executes it. After that, the business rule returns the decision data to the 40_2 process.

(9) After receiving the output (Credit Status) from the business rule, the 40_2 process forwards the screen (User Interface 2), which is created by the outputs from the result of both the search action for the 40_1 process and the rule execution for the 40_2 process.

(10) User Interface 2 shows the two reservations as a result of the reservation search on screen 1. When one reservation is selected on the screen by the system user, this screen activates the No.3 process of the business process.

5. System Prototype Generation Tool

Our system prototype generation tool generates a system prototype from requirements models. **Figure 10** shows the architecture of the generated system prototype. The user interface is transformed into JSP. The business process is transformed into a BP def. file, servlet, and two integration modules (adapter and wrapper). The BR def. file is transformed from the business rule. The WAR file includes the JSP, servlet, adapter, and wrapper modules. The UA and AA def. files are transformed from the user authentication model and access authentication model, respectively.

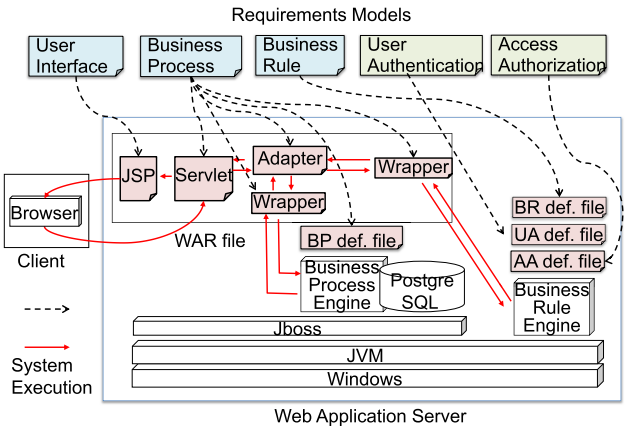

Fig. 10 Architecture of generated system prototype.

The BP def. file is interpreted by the business process engine [8]. Similarly, the BR def. file is interpreted by the business rule engine [9]. Both the UA def. file and the AA def. file are executed by the business rule engine. The adapter module performs the role of intermediary between the business process engine and the business rule engine. The two wrappers are the application interface modules for both the business process engine and business rule engine. This architecture enables the generated web system to behave on the basis of the description of the above mentioned five requirements models.

6. Case Study

6.1 Two Practical Cases

We adapted the proposed approach to two projects in our company. Both were new development projects for different industrial customers. Before each case, we gave a lecture on how to create the models and how to use the tool to all requirements engineers for the two projects. Each engineer was given a three-hour lecture. **Figure 11** shows profiles of the two cases. We also introduced the purposes of the case study for customers. To get practical feedback from the trial use, we requested customers use the generated system prototype as if it were being used for actual operation.

In the project of case 1, a service reservation management system was developed. The system had three roles: sales officer, account officer, and system administrator. The sales officer used the system at their branch office. The sales officer listened to the visitor's needs, and searched a service that fits the needs using the system. If visitors wanted to buy the services, the sales officer reserved them with the system. The account officer calculated the billing amount on the basis of the reservation information of each visitor and charged them as payment of services. The system administrator was in charge of the whole task of system maintenance, i.e., managing the master database. **Figure 12** shows one of the system screens for the role of account officer.

The total number of requirements models for business application features is 84 (= No. of UI: 36 + No. of BP and BR: 48). On the basis of these models, we estimated that the number of function points (FPs) of the target system in case 1 was 198. Next, this value (196 FPs) was converted into the work volume. We used the average project performance metrics of the new software development [17], the value of which was 15.6 (FPs developed by one software engineer for a month). We calculated 12.7 (= 198/15.6) man-months as the estimated work volume.

In the project of case 2, an office supplies management system

Case No.	System	Stakeholder (System User)	Number of Screens	Estimated Function Points [FPs]	Estimated Work Volume [man-month]
1	Service Reservation Management System	• Sales officer • Account officer • System administrator	36	196	12.7
2	Office Supplies Management System	• Office supplies manager • General user • System administrator	19	87	5.6

Fig. 11 Profiles of two cases.

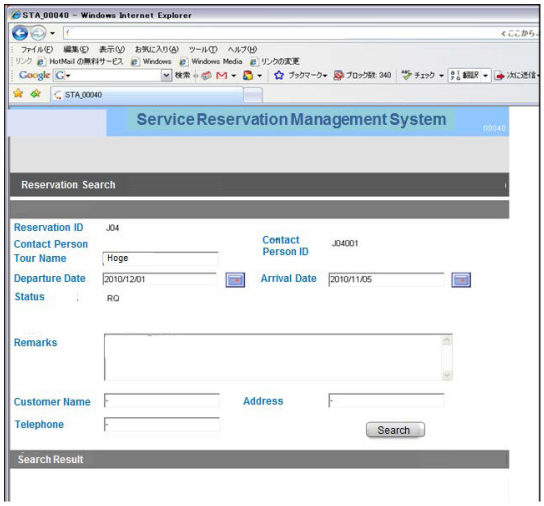

Fig. 12 Screen of generated system prototype.

Sub-task	Work Volume [man-hour]	
	Case 1	Case 2
Creating Models + *Screen Prototype Reviews*	233.0	105.0
Preparing for the Tool	2.0	7.5
Adding Implementations	21.0	0
Total work volumes	256 (=1.6 man-month)	112.5 (=0.7 man-month)

Fig. 13 Work volumes for modeling and system prototype generation.

was developed. The system had three roles: office supplies manager, general user, and system administrator. The office supplies manager operated the system in order to update the information on office supplies, i.e., the name of the supplies, serial number, purchaser name, and space for storage. A visitor can access this information. However, a general user cannot update the information. The system administrator maintains the master data base.

The total number of requirements models for business application features was 31 (= No. of UI: 19 + No. of BP and BR: 12). Similarly, we evaluated the FPs and work volume for the target system.

6.2 Work Volumes for Modeling and System Prototype Generation

Figure 13 shows the results of the work volume for modeling and system prototype generation. We identified three sub tasks: creation of models + screen prototype review, preparation for tool, and adding implementation.

Only the case 1 project had a task of adding implementation to the generated system prototype. In case 1, to address the customer's needs, we implemented functions for printing on-screen information and exporting the CSV files on reservations.

The important thing is that just soon after creating the user interface, customers reviewed these screens. In other words, we conducted the requirements review by using the screen prototype before generating the system prototype.

6.3 Result

6.3.1 Number of Change Requirements

In the case studies, the trial periods of cases 1 and 2 were 3 and 4 weeks, respectively. We got feedback from customers. **Figure 14** shows the number of change requirements obtained from this feedback. The figure indicates that these stakeholders requested not only modification but also new adding requirements. Since completion of the case study, these change requirements have not been changed throughout the whole period of the each project.

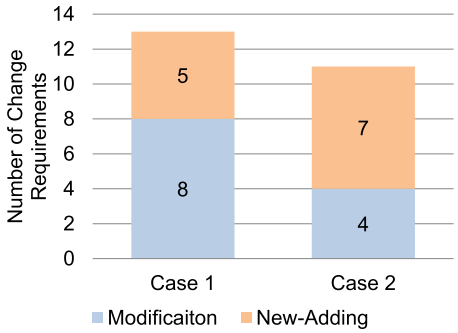

Fig. 14 Number of change requirements from feedback.

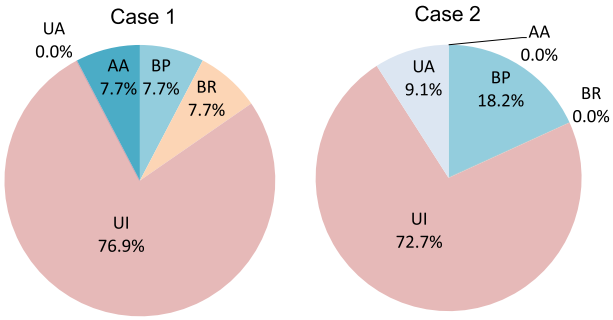

Fig. 15 Proportion of models changed by feedback.

6.3.2 Feedback to Requirements Models

Figure 15 shows the proportion of models changed by feedback in both case. Importantly, in both projects, about 25% of feedback affected four requirements models (i.e., BP, BR, UA, and AA). Sums of values of the four models in case 1 and case 2 are respectively 23.1% (= 7.7 + 7.7 + 0.0 + 7.7) and 26.3% (= 18.2 + 0.0 + 9.1 + 0.0). We found that about one-quarter of change requirements were identified only by generated system prototype.

In both projects, over 70% of feedback was related to the user interface as shown in Fig. 15. These facts indicate that even the customers who had completed the screen prototype review had different user interface requirements than they had before using the system prototype. Our system prototype helped customers identify hidden requirements.

7. Discussion

7.1 Reducing the Project Risk

The result of the case study shows that customers, who had already completed the screen prototype review, requested both change and adding requirements soon after trial use on the system prototype. This data indicates that if a project uses only a screen prototype for requirements review, the project might fail to validate any other requirements than user interface for the target system. It is probable that the project risk (i.e., cost overrun, re-work) is increasing. The system prototype is valid for reviewing requirements related to not only business application features,

including user interface, but also access control features. Moreover, these change requirements have not been changed for the rest of the project period. We claim that our system prototype enables requirements engineers to grasp "stable" requirements. We can therefore state that our approach lowers project risk.

7.2 Effectiveness and Cost of Requirements Modeling

Our tool directly generates a system prototype from requirements models. When a customer requests requirements changes, requirements engineers could react quickly by modifying the corresponding requirements models and generating the system prototype to reflect the change requirements. On the basis of feedback on the generated system prototype, a requirement engineers could get change requirements from customers more quickly. On the other hand, to use the tool requirements engineers have to describe the requirements models strictly and properly. They must elicit clear requirements from customers in the earlier requirements definition phase. As a result, it takes a lot of time to describe the requirements models. Moreover, it also takes much more time to do a walkthrough review of the requirements models.

7.3 Tool Limitation

Our tool cannot generate the web pages dynamically. For example, if one screen (HTML) is added to the system prototype, requirements engineers have to define the relation between the new screen and corresponding business process, and also add the access authorization rule on each role in the model of access authorization. After that, the system prototype can be re-generated from entire model which includes a new one and the modified models. In the case study, we recognized that requirements engineers were subject to the minor restrictions of the re-generation task.

8. Summary and Future Work

We propose an approach to trial use of a system prototype for requirements validation. We defined five requirements models and a system prototype generation tool that uses the models. We used two case studies to evaluate the approach and discussed feedback from customers. Importantly, we found that the system prototype can enable customers to identify hidden requirements that are not elicited by paper or screen prototypes.

As the next step, we plan to develop a model checker to support requirements engineers in detecting model defects. The model checker could help reduce human errors during modeling.

Acknowledgments This authors are grateful to Messrs. T. Hoshino and Y. Iimura of NTT, T. Kitani, T. Kobashi, M. Hiraoka, T. Hishiki, and S. Sato of NTT DATA, and A. Shitomi for their assistance in the case study.

References

[1] Word: help and how-to, http://office.microsoft.com/en-us/word-help/
[2] Snyder, C.: *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*, Morgan Kaufmann (2003).
[3] Axure, http://www.axure.com/
[4] DENIM, http://dub.washington.edu:2007/denim/
[5] iRise, http://www.irise.com/
[6] IPA, Security Center: SECURE PROGRAMMING COURSE, http://www.ipa.go.jp/security/awareness/vendor/programmingv2/
[7] Saito, S. and Hagiwara, J.: System Prototype Generation Tool for Requirements Review, *Proc. 10th Joint Conference on Knowledge Based Software Engineering*, pp.81–87 (2012).
[8] PEXA Engine, http://www.atrris.com/
[9] Drools, http://www.jboss.org/drools/
[10] Ouch, T. et al.: Requirement Analysis Method for Enterprise Business System Development by using Use case Descriptions and Business Process Flow Charts which are Optimized to Business Process with Ledger Sheets Handling, *IPSG, SIG Technical Reports*, Vol.2011-SE-171, No.31, pp.1–8 (2011) (in Japanese).
[11] Chusho, T. and Yagi, N.: Visual Modeling and Program Generation for End-User-Initiative Development, *Proc. 9th Joint Conference on Knowledge-Based Software Engineering*, pp.102–115 (2012).
[12] Ogata, S. et al.: The Method of an Automatic Web UI Prototype Generation Based on Requirements Analysis Model in UML, *Computer Software*, Vol.27, No.3, pp.14–32, Japan Society for Software Science and Technology (2010) (in Japanese).
[13] Mibe, R. et al.: A Method for Requirement Acquisition for Web Applications by Usecase Driven Prototyping, *Journal of Information Processing Society of Japan*, Vol.49, No.4, pp.1669–1679 (2008) (in Japanese).
[14] Elkoutbi, M., Khriss, I. and Keller, R.K.: Automated Prototyping of User Interfaces Based on UML Scenarios, Automated Software Engineering, Vol.13, No.1, pp.5–40 (2006).
[15] Sánchez Díaz, J., Pastor, O. and Fons, J.J.: From User Requirements to User Interfaces: A Methodological Approach, *Proc. 13th International Conference on Advanced Information Systems Engineering*, pp.60–75 (2001).
[16] Kariyuki, S. et al.: A Technique for Supporting Comprehensive Acceptance-test Based on the Inter-use Case Reletions, *Journal of Information Processing Society of Japan*, Vol.50, No.12, pp.3052–3073 (2009) (in Japanese).
[17] Economic Research Association, http://www.zai-keicho.com/pdf/02_h22_jutaku_no02.pdf
[18] JBoss, http://www.jboss.org/
[19] Saito, S. et al.: System Prototype Generation Tool for Requirements Review and Its Application, *IEICE Trans. Inf. Syst.*, Vol.J96-D, No.10, pp.2204–2213 (2013) (in Japanese).

Shinobu Saito received B.S., M.S., and Ph.D. degrees in engineering from Keio University, Kanagawa in 1999, 2001, and 2007, respectively. He joined NTT DATA in 2001. He works in requirements engineering. He is now with R&D Headquarters at NTT DATA.

Jun Hagiwara received B.S. and M.S. degrees from Chiba University, Chiba in 2006 and 2008, respectively. He joined NTT DATA in 2008. He works in requirements engineering. He is now with R&D Headquarters at NTT DATA.

Tomoki Yagasaki received B.S. degree from Chiba University, Chiba in 2000. He joined NTT DATA in 2000. He works in requirements engineering. He is now with NCIT.

Katsuyuki Natsukawa received M.S. degrees in Engineering from NARA Institute of Science and Technology, Nara in 1996. He joined NTT in 1996. From 2009 to 2012, he worked at NTT DATA. He is now with the Software Innovation Center.