

# A Study on Design and Development of User-Centric Business Applications using Role-Stage Model

Venkataratinam, Kathambari

---

(出版者 / Publisher)

法政大学大学院情報科学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 情報科学研究科編 / 法政大学大学院紀要. 情報科学研究科編

(巻 / Volume)

9

(開始ページ / Start Page)

167

(終了ページ / End Page)

172

(発行年 / Year)

2014-03

(URL)

<https://doi.org/10.15002/00010540>

# A Study on Design and Development of User-Centric Business Applications using Role-Stage Model

V. Kathambari

Graduate School of Computer and Information Sciences, Hosei University,  
Tokyo, Japan  
Kathambari.venkataratinam.8c@stu.hosei.ac.jp

**Abstract**— Role-Stage model is a programming model, initially discovered in the context of development of agent-based language for simulation of complex social systems. This research aims at integrating the various facets of business operations using the proposed Role-Stage model for the design and development of flexible, adaptive, user-centric business applications. The technique involves the three essential building blocks called Stages, Roles and Agents, to describe reason and conceptualize business processes at model description time as well as at runtime. The effectiveness of the proposed model is illustrated by a sample enterprise application, designed and implemented based on the model.

**Keywords**— *Role-Stage model, Business Process, Agent-Based Technique*

## I. INTRODUCTION

Object-oriented analysis and design methodologies have a high modeling capability, as objects are representations of real-world concepts. Also, the concepts of roles and role modeling have been widely applied in information systems to describe the collaborations of the objects and separate the contextual behavior of objects from core behavior. As emphasized by the survey on role-based information systems [7], roles can be taken as underlying mechanism to system analysis, design and construction.

The emerging trend in business software development is to build software with a design focusing on reflecting the user-experience. Roles have improved the conceptualization of processes in business application systems [8]. An Enterprise Resource Planning Software[ERP] that integrates all aspects of business operations when tailored with roles in its presentation layer can improve the effectiveness of business by directly involving the end-users who work on the application. However the current approaches focus on developing an application centered on the role involved and the business activities that the role participates in. The basic purpose of introducing roles is to contextually define the behavior of the objects. The problem with the products that are based on user-models is that there are no simpler means for users to conceptualize the contexts, the relative interactions and dependencies between roles that are essential to complete the business process.

The main objective of this research is to model, design and implement the business processes based on a programming model that provides improved conceptualization of the

participating agents, their context based behavior and the contribution of agents toward completion of the process. The model must be simple to describe and reason business processes with support for description of the collaboration at both modeling and programming levels.

Agent-based simulation language called SOARS-Spot Oriented Agent Role Simulator [9, 13] simulates complex social systems by decomposing the multi-agent interactions. The framework uses the notion of spots, stages to provide a spatial and temporal decomposition of interaction among agents. This property of the agent-based approach can be effectively used in development of business software where the interactions among agents assuming roles are decomposed and their collaborations are visualized at both agent-level and process-level.

The remainder of this paper is structured as follows. The next section describes the background works and motivations to the proposed approach. The section entitled Role-Stage Model details the domain of business process management, elaborates the rationale for an agent-based solution and introduces the structure of the proposed model. It is followed by the description of a business process with roles and stages and features of our approach. We then present a case study that demonstrates the benefits of the approach with a sample enterprise system. The study describes how the concepts introduced herein have been applied to a real-world organizational framework with real-world business processes and resources. The application of our approach to different phases of development such as specification, design and implementation is presented. The extended benefits of our approach in monitoring business processes and managing workload of roles are discussed with examples. The final section discusses on the relevancy and differences of the proposed approach with related works and highlights the open issues that need to be further addressed.

## II. BACKGROUND AND MOTIVATION

### A. Role Modeling

Roles have been used in modeling for several years. Figure 1 represents the bureaucracy role model discussed from the work of Riehle D on composite patterns [2]. In Figure 1 the rounded boxes are roles and the triangle represents role specialization. The collaboration path is shown by the arrows and filled circle shows that more than one entity can play the

role. The roles in the design are assigned to objects in the application, thus enabling the objects to play multiple roles. Several other research works on design and implementation methodologies of role models help us understand the benefits and application scope of role models.

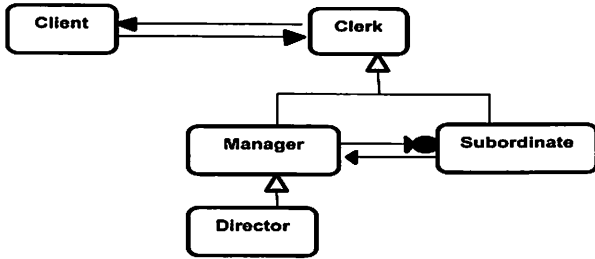


Fig. 1. Role Diagram for Bureaucracy pattern [2]

### B. Agent Based Modeling

Roles and role models have been found very effective in agent based modeling. The Agent Based Modeling framework –SOARS [13] represents the agent processes with parallel and sequential decomposition. The base reference for our work is the SOARS model, where the agents in the dynamic systems take roles based on the place of their interaction that is modeled as spots. Figure 2 represents the overview of the SOARS model. The model introduces several notions such as spots, agents, roles, stages, rules and actions for describing an agent based dynamic system.

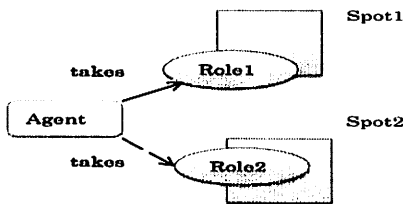


Fig. 2. Overview of the SOARS model

### C. Business Process Modeling

One of the primary challenges in development of business software is the need to enable the users make sense of the information they have. Roles in business process modeling have been widely used to represent the structure of business objects, their relationships and collaborations. A number of ERP software developers focus on research and development of role-centered approaches to improve the business productivity. The Customer model of Microsoft Dynamics [12] introduces a software strategy of tailoring roles in to the User-interface to improve the user-experience. NetSuite ERP [11] provides a personalized dashboard for employees centered on an organizational model. Such products have been our sources of reference for understanding the benefits of using role concept to model business processes.

The major motivation for our research is to devise a model for design and development of flexible, user-centric business applications supporting user-level transparency interactions and process-flow. The model is a means to improve knowledge on the capabilities of the role in a process.

## III. ROLE-STAGE MODEL

### A. Model Description

A conceptual representation of a business process in figure 3 shows a process as a sequential execution of activities which in detail can be represented using activity diagram [15]. The initiation and termination of a process are represented by a circle and filled circle respectively.

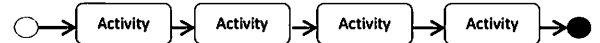


Fig. 3. Conceptual representation of a business process flow

The Role-Stage model defines a process in terms of agents, roles and stages providing a simpler but combined structure to represent the agent interactions and their position in the execution of a process. Roles can be played by individuals or agents to whom the tasks were delegated.

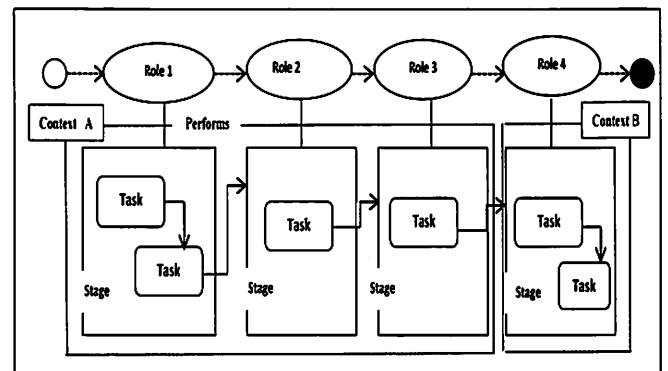


Fig. 4. Representation of Business Process with Roles and Stages

Figure 4 shows the representation of a process flow using the Role-Stage model. In the model devised, the behaviors of all the roles defined in a system are described in blocks called stages and each stage encapsulates the details of the role's actions. The interaction between roles is in terms of stages and the entire process is represented as sequence of stages. The dashed block is the context within which the roles perform the operation. Stages are rectangular blocks whose notifications to the succeeding stage are illustrated with solid arrows. The roles involved in the process perform the tasks in the corresponding stages and the entire process is given as interactive interdependent actions of the business participants to complete the process. In various scenarios, certain chain of tasks can be automated within the stage. A role invokes the initial task which automatically invokes the task that follows it. For example: When the sales order has been approved after verification by Role Manager in the context of a financial group, an invoice is generated and forwarded to the customer. The model achieves the temporal decomposition of the interaction among roles within a process. Also, the dependencies of the roles to achieve the completion of a process are captured.

The main components of our model's framework are interacting participants and autonomous agents that are responsible for performing particular activities. Here we define "stages" as set of logically related activities that the role-player will manage. Examples of stages include *Decision* –activities whose end result is to establish a set of standards for business entities like shipment carriers, payment methods, *Approval*–providing approval for quotes, invoices etc.

We define the terminologies used in our model as follows:

- **Context**  
Field of collaboration of objects in a system.
- **Agent**  
Real-time users and possible software agents defined during the analysis phase
- **Roles**  
Properties and behaviors played or assumed by the agents within a context
- **Stage**  
Blocks of actions defined for each role, executed in an order for completion of a process

The main idea of the model structure is to provide an information model that reflects the organizational structure. Any participant, human or autonomous, can avail information about the overall process flow, the relative dependencies between the organizational elements and manage the workflow more efficiently. The roles assumed by the human users do not have the compulsion to understand the complex activity diagrams. They may interpret the process at stage-level and carry out business operations delegated at each stage.

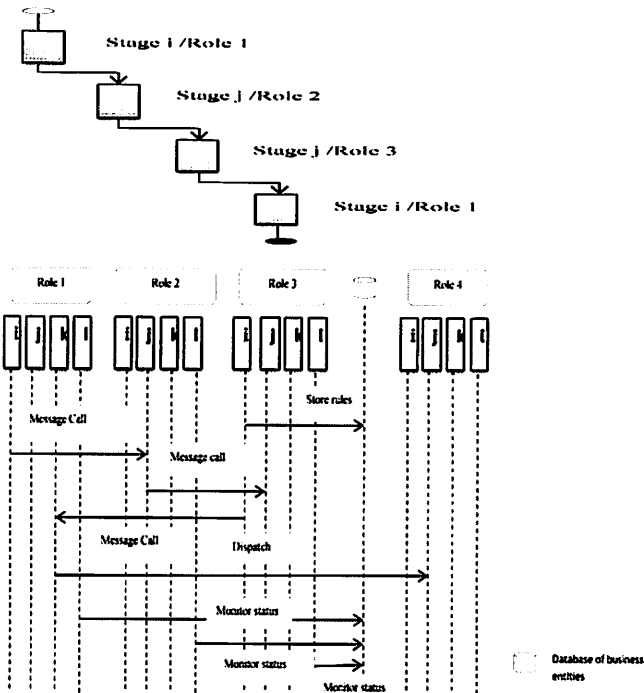


Fig. 5. (a) Process Description (b) Interaction Diagram

A process can be described as sequential blocks of stages with each role performing the operation illustrated alongside

the stage as shown in figure 5(a). In our example of process description, we consider roles Role1, Role2, Role3 inside a context C and Role4 outside C. Considering the stages of operation as i, j, k, l, at each stage, the participating role is captured while the behavior of the role is encapsulated within the stage. Each stage notifies the role of the next stage and the process terminates after the execution of the final stage. The detailed activities are represented in the interaction diagram with the stage-based timeline in figure 5(b).

## B. Model Features

The features of our model based on the description, structure and interaction diagram given above can be summarized as follows:

- **Temporal Decomposition of agent interactions**

From figure 5(a), for a role Role3, its participation in a process is after the completion of stage j by Role2. With our model, each role understands its current actions in each stage as it is notified by other roles that performed their actions in the previous stages.

- **Improved usability at model time and runtime**

A system built based on our model has a greater usability due to the following reasons:

- A user can easily understand its participation in a process without the need to understand the details of its actions. Activity diagrams used in modeling are complex but with our proposed model, users of the system can benefit from the stage-based design.
- The model is reflected in the software and hence a role personalized view combined with stages in the user-interface level simplifies the work of the user. This enables development of software, designed for people. We mention details in our case study.

- **Reusable Constructs**

We implement the model in an object oriented environment. Roles and stages are defined as first class constructs and hence the stages can be used as reusable components. Also the structure of stages is the same for all contexts and hence can be made as a unified approach for designing a large application.

- **Flexibility**

There are several patterns identified for the implementation of role models. We choose the decorator pattern [14] for implementation and all design principles of the pattern apply for our development. We can add additional behavior to roles without directly modifying their properties. The core classes are open for extension and closed for modification. Stages add flexibility to the design as new behaviors for a role in a context can be added without modifying the role class itself.

#### IV. CASE STUDY

A well-designed business process management system has several advantages over the traditional counterparts. Business applications designed based on our proposed agent-based approach has the competitive edge over other traditional approaches as they improve business productivity with better co-ordination, planning, execution and monitoring of business activities. In our case study, we have implemented a running business application designed and developed based on our proposed model. To help demonstrate the benefits of our approach, we describe the implementation of our software in this section.

We consider a business environment with three major functional groups- Sales, Finance and Purchase. According to the role-stage model, the three groups are visualized as three contexts in which agents assume roles to perform business functions. We introduce roles like buyer, supervisor, manager, representative, accountant and controller. The business activities performed within each context is encapsulated within stages – Decision, Approval, Execution and Monitor. The primary goal of the research work is to develop business process applications that reflect a company's organizational structure when modelling and describing the behavior of its components. We reflect the nature of behavior of business objects using the structure of context, roles and stages as shown in the figure 6. S defines the stages, R represents roles and the arrows represent the interactions among stages during business operations. The diamond represents the interactions between the context and external business participants such as vendors and customers. Suppose that this structure reflects a real-world organizational scenario, given a business process like payment involving payables and receivables. The buyer receives invoice from the vendor for purchase, validates and forwards it to the purchase group manager for approval. The manager interacts with finance accountant to process the payables. The accountant upon approval from controller pays the vendor for the purchase. The accountant also confirms any receivable payment from customers and notifies the sales representative to process and dispatch invoices. Although the process flow is complex, the roles performing the actions need only a stage-level interpretation of the business process. Furthermore, the negotiations, notifications and interactions between roles in different contexts are facilitated via the stages and hence the agents playing the roles are aware of the overall process flow alongside with their responsibilities and interdependencies.

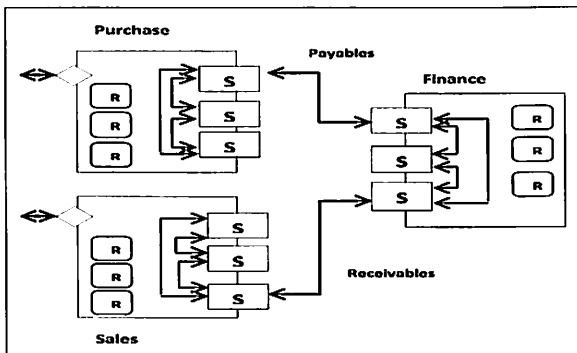


Fig. 6. Overview of an organization structure with roles and stages

#### A. Specification

The effectiveness of the approach is that the model can be realized at all phases of development. In our case study we begin with specification of the tasks to be performed followed by the design and implementation. The tasks delegated to each role during the specification phase define the actions performed by the roles in the context. During specification of a context, such assignment is made feasible using the role-stage-responsibility chart. A sample for such specification is provided in table 1. Here we consider the purchasing group as context.

TABLE I. SAMPLE ROLE-STAGE RESPONSIBILITY CHART FOR PURCHASE DEPARTMENT

Roles / Stages	Buyer	Supervisor	Manager
<b>Approval</b>	Validate invoices	<i>Validate; Approve orders</i>	Approve new vendors
<b>Decision</b>		Decide new supplies	<i>Set rules for validation</i>
<b>Execution</b>	<i>Create, Modify Release orders</i>	Draft plans	Set individual targets
<b>Monitor</b>	Status track	Status track	Monitor performance

#### B. Design

Model interaction diagrams are repositories of information for the designers of the application which help in encapsulating the business functions within appropriate stages. A process-specific interaction diagram describes the temporal interaction between roles. A system may also include autonomous agents that perform simple tasks like a sample computation involving a database-lookup to complex tasks like being a moderator between the organization and external business objects. All roles played by human users or autonomous agents are collectively captured in the interaction diagram.

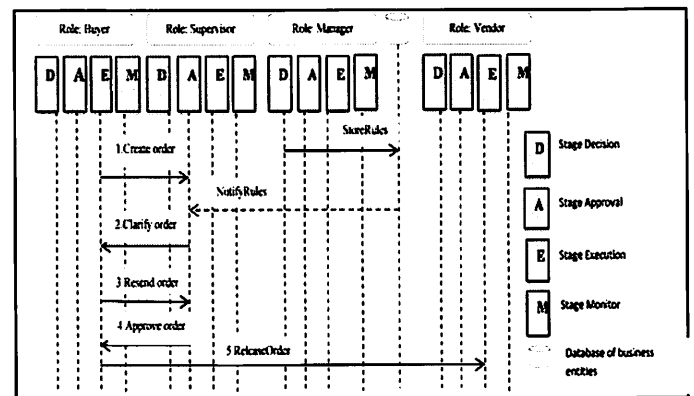


Fig. 7. Interaction diagram for procurement process with roles and stages.

The illustration in figure 7 describes one of the main business processes within the purchase context- *procurement* of materials italicized in the specification chart given in table 1.

The order of operation is represented along the stage-based timeline. 1. Buyer begins the process by creating the purchase order. 2. This order is evaluated by the supervisor based on predefined rules. 3, 4. Evaluation is done as to whether the order is approved, rejected or modified. 5. Upon approval, the order is forwarded to the vendor.

### C. Designing the User-Interface

Our design of user-interface focuses primarily on involving the participants in the software. We tailor the roles and stages to the user-interface and reflect model description and design in the software with a personalized screen-view for the roles. Each role manages its actions using stages. This design helps the role to focus on its tasks and operate effectively based on priorities. When tasks are performed, the stages notify the next responsible role. The resources accessible to the roles are accessed from corresponding stages. The information and tasks presented to the users are based on the role-stage model, enabling collaboration, thereby driving efficient and consistent designs. A sample personalized dashboard of a user playing the role of a Purchase Manager is shown in the figure 8. The tasks are arrayed within the stages and the notifications correspond to incoming tasks in the respective stages.

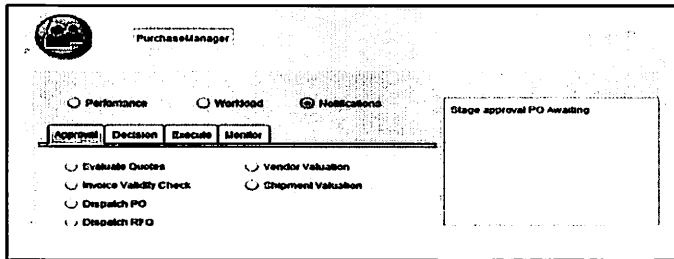


Fig. 8. Screenshot of the model-based user-interface

### D. Building Business Processes with Role-Stage Model

Several approaches have been used for implementing roles in object-oriented languages. We use the decorator pattern for designing the interface to the role. Both Role and Stage implement the same AgentCore interface. A concrete stage object using the instance of AgentCore will have the knowledge of the role object it acts upon and adds behavior corresponding to the role at run-time. This enables each role to have a stage-specific behavior and the resources handled by the role during the process are accessed from the stages. The structure of the class diagram is shown in figure 9.

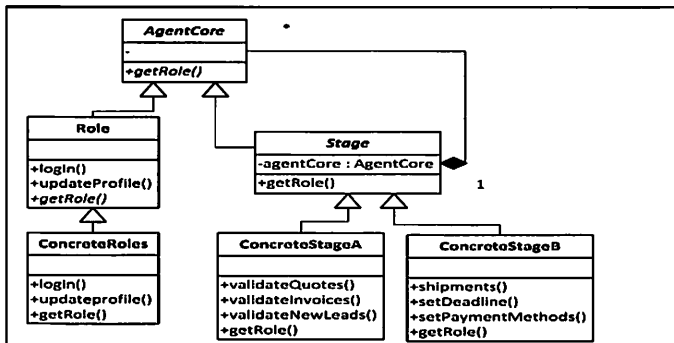


Fig. 9. Role-Stage based class diagram for business process design

## V. REAL-TIME MANAGEMENT SUPPORT USING ROLES AND STAGES

In this section, two features of operational support essential for a business system are described using our methodology: a) Monitoring the progress of a business process b) Managing the workload of the users playing the roles. These are implemented additionally on the system described in the previous section without compromising the user-level transparency of the base system. This demonstrates that our proposed methodology can also be applied over a broader range while keeping the application open for extension. We describe the process monitoring and workload management extensions that enable business management support in real-time, in detail with examples below.

### A. Monitoring the progress of a business process

Monitoring the business process is essential for the roles to collaborate with one another and achieve business process management. We use a separate stage called monitor in the design that helps us to track the status of business entities. The monitor log updates each time a role performs an action on the entity and helps the roles to follow up the activities for completion within deadlines. This simplifies task management and improves collaboration among roles within and between contexts. A sample monitor log for a customer order in the sales context is shown in the table II. Each time the roles perform actions with respect to the order entity, the log is updated with timestamp. This enables the managers to identify lapses in task accomplishment. The manager can negotiate with the corresponding role to meet the completion. Thus programming with roles and stages also achieves simple in-built status tracking for the application.

TABLE II. MONITOR LOG VIEW FOR STATUS TRACKING

Entity	PR#134				
Role/Stage	Representative	Supervisor	Customer-Liaison	Manager	Accountant
Approval			New order Received @timestamp1	Order Approved @timestamp4	
Decision		Deadline set for dispatch @timestamp3			
Execution	Order validated @timestamp2				Awaiting payment processing..

### B. Managing workload of the roles

One of the other important advantages of using roles and stages for building business processes is that the workload of each role can be analyzed and compared using graphs with entities and number of tasks handled in each stage, thereby managing the distribution of workload equally and efficiently among the users playing roles. We illustrate the number of tasks assigned to the role Purchase Manager with respect to the business entities, the role operates upon in figure 10. Such stage-based allocation of work among roles develops responsibility for provisioning and managing the business.

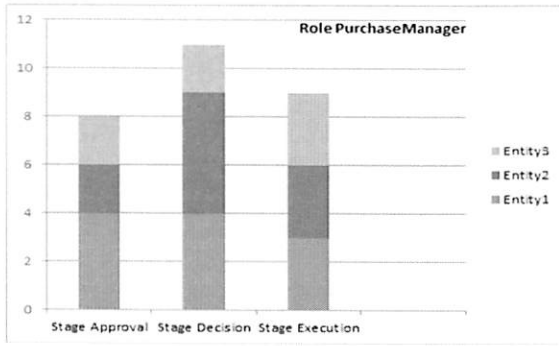


Fig. 10. Sample graph for stage-wise task allocation

## VI. DISCUSSIONS

We have implemented our concept of the role-stage model across different contexts in a business application. The user-interface of our application has an information architecture based on the Role-Stage model. The information and tasks presented to the users are based on the model design. The users playing the roles also receive alerts in their interface based on updates in each stage. The roles can process information with improved access to the flow of information. Our model is easy to interpret and can be used with compliance across an organization. As our model takes motivation from an agent based technique for simulation, it is easier to simulate business processes in complex systems when the application is developed based on the Role-Stage model. We can further simulate the amount of work done by a role and derive ideas for changes in the processes and task assignment to roles. As the model is based on a flexible design, changes in behaviors are updated to the stages without complex modifications. There have been many works on different implementation methodologies of role models. The current experiments are built based on the decorator pattern. We also plan to work on other implementation methodologies to include more features in the model such as dynamic adaptations.

## VII. RELATED WORKS

There are many works in the past on programming with roles [1, 4]. Our model uses the role concept introduced in such works. The properties and characteristic features of roles [4] defined by Gottlob, Schrefl, and Röck introduces the benefits of using roles to handle the evolution of objects. Riehle [2] introduced a role type to describe the view an object holds on another object. A set of roles and their relationships are used to describe the collaboration of objects. Kristensen and Kendall [3, 5] summarize the properties of roles which are used in building role-based systems in an object oriented environment. Several programming languages have been developed as an extension of object oriented languages to support role models. The Epsilon model [6] provides dynamic role binding to realize adaptation of an object to environments. Environments in the Epsilon model allow explicit separation of concerns. Roles can also be implemented using aspect-oriented programming. Kendall presents different options along with a hybrid approach to implement roles in programming [5]. Aspects, like role models are orthogonal to object definitions. The main

objective of writing aspects is to deal with cross-cutting concerns. In our model we achieve separation of concerns using contexts and stages. Stages separate collaboration of objects across multiple contexts.

The above approaches focus on the roles and methods for implementing the roles in software. For development of business applications, we need to improve specialization in such models to enhance productivity. Stage helps us divide and conquer the role relationships within and outside the contexts. The number of language tools that support role based programming help us design systems with objects, roles and stages and provide more reusable components with improved maintainability.

## VIII. CONCLUSION

In this paper, we have introduced the Role-Stage model for design and development of effective user-centric business applications. We have illustrated the features of our model with example. The runtime implementations of the model constructs effectively reflected on the presentation layer of the application enables the users to make sense of the information they process. The dependencies between users and interactions are captured at description time and implementation time. We have prototyped the approach with different contexts of an application to evaluate the feasibility and effectiveness of our model.

## REFERENCES

- [1] Andersen, E. (Egil), Conceptual Modeling of Objects: A Role Modelling Approach, PhD Thesis, University of Oslo, 1997.
- [2] D. Riehle. Composite design patterns. In OOPSLA '97, October 1997, pp 218-228
- [3] B. B. Kristensen, "Object-Oriented Modeling with Roles", Proceedings of the 2nd International Conference on Object-Oriented Information Systems (OOIS95), Dublin, Ireland, 1995
- [4] Gottlob, G., Schrefl, M. and Röck, B., "Extending Object-Oriented Systems with Roles", ACM Trans. on Information Systems, July 1996, pp 268-296.
- [5] Kendall, E. A., "Agent Roles and Aspects", ECOOP'98 Workshop on Aspect Oriented Programming, July, 1998.
- [6] T. Tamai, N. Ubayashi, and R. Ichiyama, "An Adaptive Object Model with Dynamic Role Binding", Proceedings of the International Conference on Software Engineering (ICSE2005), St. Louis, Missouri, USA, May, 2005, pp.166-175.
- [7] Zhu, H., Zhou, M.: Roles in Information Systems: A Survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38(3), May 2008, pp. 377-396
- [8] Zhu, H. "Role Mechanisms in Collaborative Systems", International Journal of Production Research, 41, 1 (Jan. 2006), pp181-193.
- [9] H. Deguchi, H. Tanuma and T. Shimizu: SOARS: Spot Oriented Agent Role Simulator -Design and Agent Based Dynamical System in Proc. AESCS04, 2004, pp.49-56
- [10] A. Scheer ARIS - Business Process Modeling, 2nd edition, Springer, 1999
- [11] Netsuite ERP Products: <http://netsuite.com>
- [12] Microsoft ERP Products :<http://www.microsoft.com/dynamics>
- [13] SOARS project: <http://www.soars.jp/>
- [14] Gamma, E.R., R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-oriented Software. Addison-Wesley, 1994.
- [15] OMG, "OMG Unified Modeling Language Specification, v 1.3," Version 1.3, ed, (1999).