Development of Information-Task-Presentation Linked Model for Systematic UI Design

*Wan Chul Yoon, **Seung Hun Yoo

**Dept. of Industrial Engineering, KAIST, 373-1, Kusong dong, Yusong Gu, Daejeon 305-701 KOREA, wcyoon@kaist.ac.kr*

***Dept. of Industrial Engineering, KAIST, 373-1, Kusong dong, Yusong Gu, Daejeon 305-701 KOREA, cityboy@kaist.ac.kr*

Abstract: The aims of this research are to establish a systematic UI design model that can support designers' UI development/ evaluation process, and to develop a computer-based UIMS (User Interface Management System). UI developers have widely accepted UIMSs as efficient and systematic methods for the system UI design as systems have become more complicated and multi-functioned. Current UIMSs adopt not only a single UI model, but many kinds of UI models to provide the designers with more broad view on the system, from the low to high level abstraction models. However, most of them are not widely used by designers in the field of HCI because they lack of the understanding of intended practical design process. In this study, practical design process and task modeling methods are analyzed. The results of the analysis are generalized to elicit the requirements for a system modeling framework that can support the communication between UI experts and designers. Concurrently, three kinds of task models for the system UI design are selected based on the system engineering theory: Information hierarchy model, Task procedure model and Presentation model. These models make up a unified and systematic task model, Linked-Task Model. The process of the Linked-Task Model is arranged according to the findings from the analysis of the practical design process. The Linked-Task Model provides designers the ability to support the systematic and intuitive design concurrently through a flexible process. It also provides the designers. wider and robust view of the system.

Key words*: *Systematic UI Design, Task model, UIMS, Information hierarchy, Task procedure, Presentation model

1. Introduction

In multi-disciplinary environment of system UI development, UIMSs can function as efficient tools for system UI design between designers and developers. In most cases, UIMSs have focused only on designing icon, visual expression of information, layout and programming-at the feasible and low abstraction models-and lack of understanding mental operation of human. In order to support UI modeling of high abstraction level, many of UI tools and UIMSs have been proposed but they are not widely accepted in practice. Reasons are following 3 issues. First, the process and knowledge of all experts in multi disciplinary team are very different from each other. Second, most UIMS are developed by engineers or programmers and force a inflexible process which doesn’t assist or allow the flexible and intuitive idea generation and development process of designers. Third, each UIMS doesn’t supports the characters of various industry domain: for example, UIMSs are usually focused on development of office-ware or industrial application in which the design problems and the goals of tasks are well defined. Therefore there is no UIMS accepted to develop and evaluate the quality of system UI in such a domain like online games or multi user systems where design problems are ill defined.

This research aims to establish Linked Task Model(LTM), a systematic UI design framework functioning as a communication platform. LTM assists flexible and intuitive idea generation of designers and systematic solution of UI design problems from the early stage of UI design process. For this purpose, LTM links information hierarchy model, task procedure model and presentation model into one model. The main interaction model for LTM is based on the Seeheim model and it will supports UI iterative design-analysis pattern for designers to develop a design and predictive analysis of system UI.

2. Previous works and Research Scope

2.1 Design Process

Seeheim model, the basic interaction model of system engineering, defines HCI as an information transition from user to computer and decompose computer interaction into three elements; presentation components, dialogue and application interaction model. Tools which are frequently used to support for Seeheim model are transition network, formal language and event -based diagram. (Fig.1)

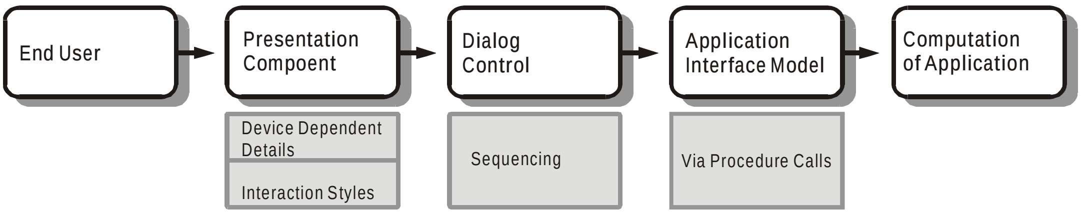


Fig.1 Seeheim model for HCI process

Vicente(2001) introduces system design process as an iterative pattern between design and analysis to make the abstract relationship between purpose and physical form of a system. [2] Rassmussen (1989) has explained the design process with the framework of abstraction-decomposition space and insists that the purpose of a system is the reason for the design of interface. (Fig.2) [3]

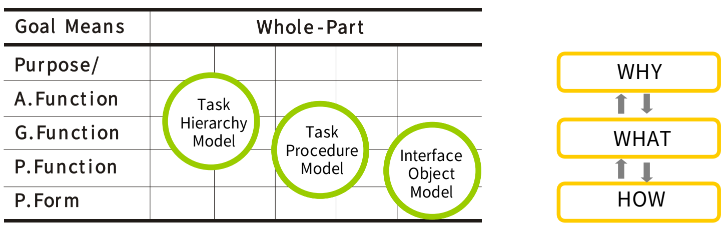


Fig.2 Abstraction-decomposition design space

According to Vicente and Rassmussen, design process is an iterative feedback between design and analysis in the abstraction-decomposition space, not a inflexible one-way procedure from analysis to design. Designers can generate an idea first and evaluate it later, opposite from the usual systematic process. This framework explains the intuitive idea generation process of designers who make the innovative and detailed design concept at the early stage of design process. But no UI tool supports such a flexible process of designers and regards it as an unsystematic output, and no UI tool is adopted by designers.

2.2 Interaction Modeling

UI design methods can be classified into two categories: Methods for system-centered design and user-centered design. User-centered design is a method to design a system UI based on the task of users, which is appropriate to reflect and analyze users' knowledge but requires a lot of time and efforts and lacks of the ability to predict the real feature and character of a system at the early stage of design. System-centered design is to describe interaction based on the system state; that is very efficient to design a system itself but the UI problems of a system which are caused in the interaction between system and user cannot be evaluated before the whole design process is over. [4]

Model-based interface design method is to design a system UI based on a specific interaction model of a system that means the simplified and generalized feature of the most important architecture or inner-relation of a system. A system can be modeled into so many kinds of models according to the various viewpoints of designers that it is a reliable approach to adopt multiple models for a system design because a single model usually doesn't express the whole features of a system. It is also recommended to use mutually exclusive and independent models from each other, but the optimal set of models for the system design is not defined yet. [5] (Silva, 1999) As a result, most of system designs require a revision phase after the final result is accomplished because of the UI problems which are not predicted through the design process. In the other hand, designers need a prototypical model which can predict the structure, relation and sequence of system interaction and enables the systematic design of UI. [6] According to Puerta(1999), present implication-oriented tools don't support integration of design process and most desirable model for system UI design is user task model, because user-task model is based on the user 's knowledge. Many researchers insist that user task based design is a solution for this problem, but task based design also has been criticized because it requires so much time and doesn't satisfy the fast process of practical design process. Hix(1993) and Lee(2001) classified modeling methods into formal method, diagrammatic methods, and prototyping methods by the output of methods. [4] Limburg(2001) compares and classifies UI models by their semantics. [7]

Table1. Classification of UI models by the semantics of model

	Features	GOMS	GTA	CTT	MAD	TKS
Information Structure	Task Planning	Operator	Constructor (Parent)	Operator	Constructor (Parent)	Plans
Task Procedure	Operation Description	Method		Scenario	Task	Procedure
	Task Tree Leaves	Unit Task	Basic Task	Basic Task		Action
	Operational Level	Operator	Actions/ System Operation	Action		
Presentation	Specification of Object Used	Cognitive Objects	Object Hierarchy	Objects	Objects	Object Hierarchy
	Cognitive Aspects	User Performance	User Performance			
	Co-operative Aspect		Roles/ Agents	Roles Task Trees		Roles/ Agents

2.3 UIMS(User Interface Management System)

UIMS is a computer aided systematic UI developing environment based on a specific interaction model and inherits the whole attributes and character of the basic UI model. [9] The basic requirements for prototypical UIMS are as follows; First, it should be easier and more systematic than the previous tools. Second, the output of

UIMS should be a declarative model for system. [10] Most of UIMSs are too complicated to be educated and provide only small parts of solutions for system UI problems. [5] (Silva(2001)) UIMS usually adopts bottom-up design process; define the attributes of interface object first and design the interaction last. It is not matched with top-down process of design. Another problem of UIMS is its difficulty to reuse the UI data of a system design next time. Although it is one of the most important objectives of UIMS, it is not fully in practice and whole analysis is repeated in every project.

There is another difficulty in model based design process that the optimal model and process for design and analysis are totally different. In design phase, system based models- information hierarchy, interface, dialogue or state transition diagrams-are proper. In contrast, task based models are usually used in evaluation or task analysis of a system. Therefore, process or tool to evaluate the design concepts systematically in both of system-based and task-based process is required to support the iterative design process. Scogings(2001) insisted a task model that combines menu tree and task procedure model through which designers can check the duplication of unnecessary information or the omission of required information. [11] In this model, the number of task is very critical problems because if task exceeds 20 or 30, the complexity of the model increases so high that it is a burden for designer to manage these analysis data and elicit a design of a system. Lim(2001) insisted that design information should be extracted from the user activity data and analyzed by the sequence of user behavior. [12]

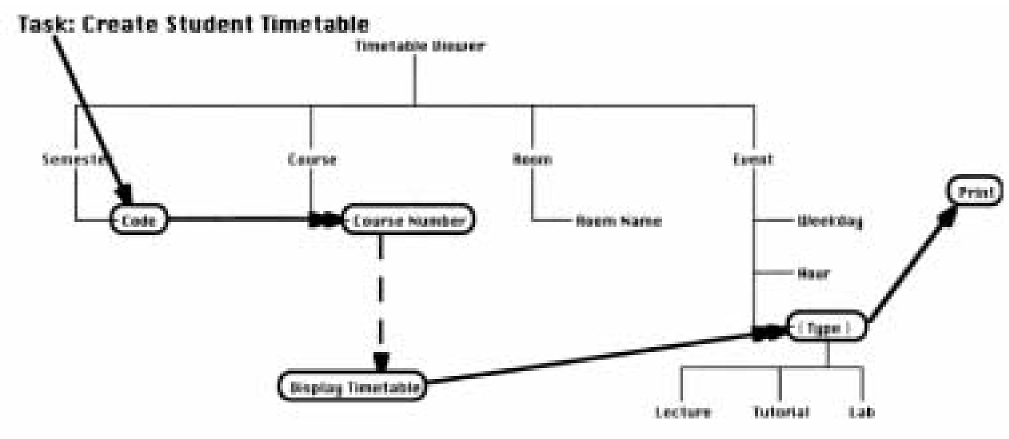


Fig.3 Linking task procedure models and menu structure

3. Frame for the model of linked UIMS

3.1 Hypothesis for Linked Model

Linked-Task Model (LTM) supports a system design process through linking information hierarchy model, task procedure model, and the low-level presentation model of a system, which are the most widely adopted and required models for a system description. Although these three models are widely adapted, they are seldom used together in a system interaction design. Developers can use any of these models to design UI of a system at first, and can convert it into other models. LTM enables designers to choose any of three interaction models and process that is frequently adopted in system design. Designers can predict and evaluate whether the interaction of a system follows the mental models of users on the system. The flexibility of LTM also assists designers to evaluate the final design of UI even at the early stage of a design process.

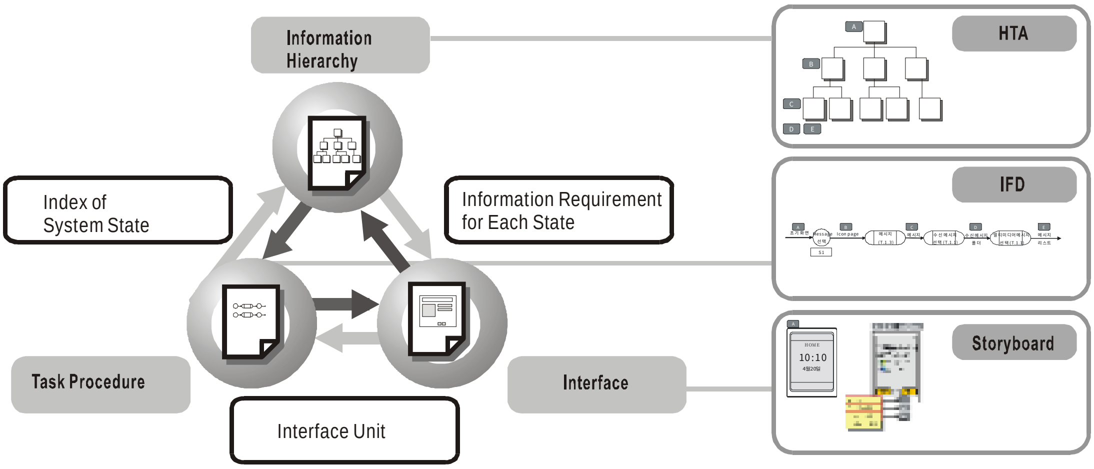


Fig.4 LTM framework as a systematic UI design tool

The main interaction model adopted in LTM is IFD(Interaction Flow Diagram), a task-procedure model suggested by Yoon(1996). LTM also adopts a mid-level expression tool, Interface unit, to reinforce the weakness of a task-procedure model: lacks of ability to express the feasible state of a system. An interface unit is a simple sketch or description of system states, and helps designers to predict and evaluate possible UI problems of a system which are caused not by the task structure or task procedure but by the visual components of a system interface. The use of interface unit in a task procedure model can be converted into the nodes or cards of an information hierarchy model of a system. It can be also used to check the duplication or omission of information.

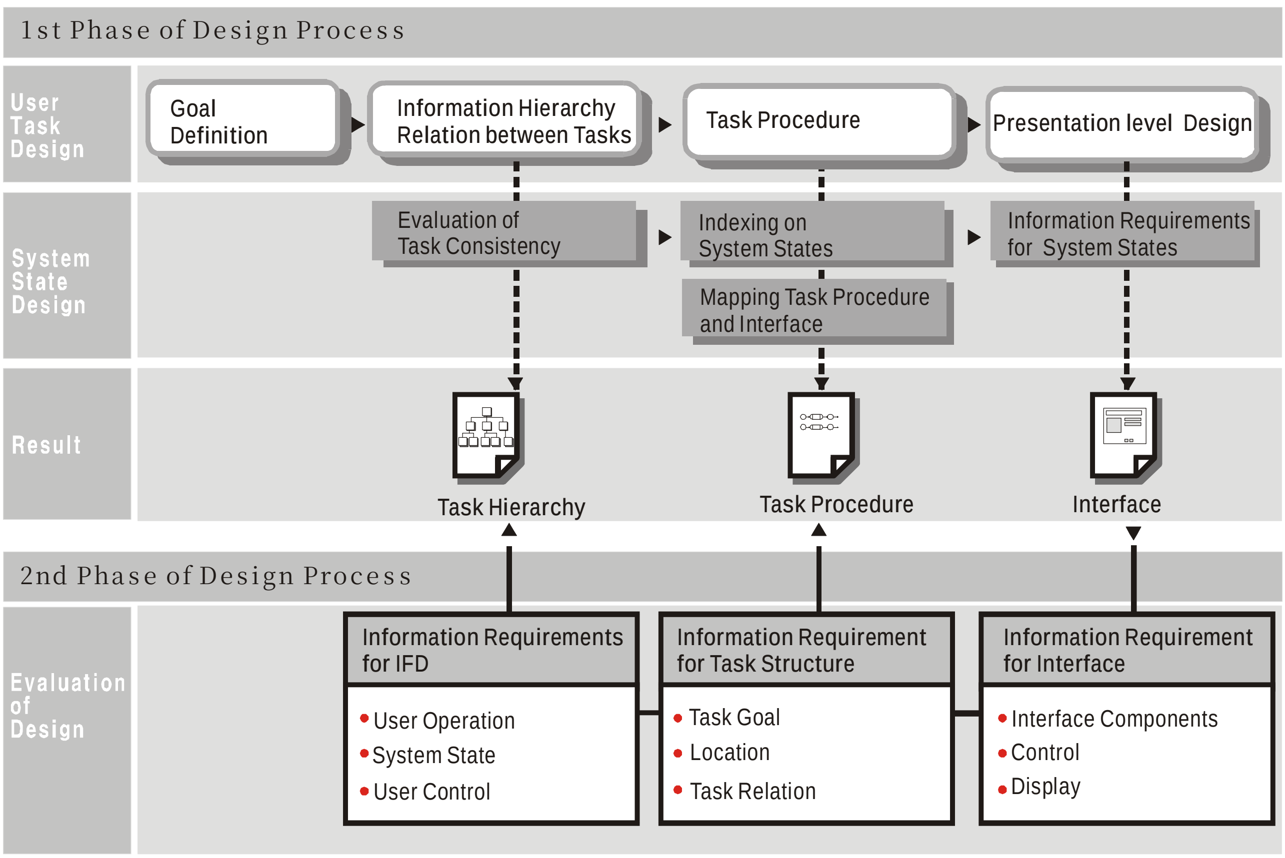


Fig.5 Iterative design process supported by LTM

In the system design process, designers set the goal of a system and elicit the structure of tasks and information of the system first. Secondly, designer set the procedure of each task and the interface of the system. During this process, it is usually happened for designers to make an interface design not based on the systematic analysis, but based on the intuition even at the first stage of design process. But those intuition- based design can be a useful tool to evaluate the possible alternatives of a system interface and to make a predictive analysis process. Designers can use such a rough interface design as an Interface unit in LTM. Interface unit helps designers to regard the real feature of a system during the task procedure design phase. LTM also adopted the concept of SIR(System Information Requirement) that is defined during the task procedure modeling on every task procedure. SIR can function as a checklist to evaluate Interface Unit. SIR helps designers to deal with the UI problems which happen in the presentation level of a system design. The interface unit and SIR can support the concurrent development of 3 models and help to reduce the time and efforts. Those two tools are the main means to link three different interaction models in LTM and to share the design output as a communication tool for multi-disciplinary researchers.

Target Element / Source Element		Information Hierarchy	Task Procedure	Interface
Information Hierarchy	▶		▪ Task Structure ▪ Relations between Tasks ▪ Consistency of Task Procedures	▪ Location of Interface in Information Structure ▪ Omission of States of System
Task Procedure	▶	▪ Redundancy of States ▪ Orphan Links		▪ Consistency of Task Procedures
Interface	▶	▪ Feature of a Real System ▪ Information Requirement for Each State	▪ Real Features of Task Procedures ▪ Information Requirement for Each Phase	

Fig.5 Liking three task models in LTM with SIR and interface unit

3.2 Process of LTM

To descript more that 2 tasks with a task procedure model, same numbers of task procedure models are required. In contrast, the relationship between those tasks is defined in the information hierarchy of a system and users can perform those 2 tasks in a same dialogue box (state of a system). [Fig.6]

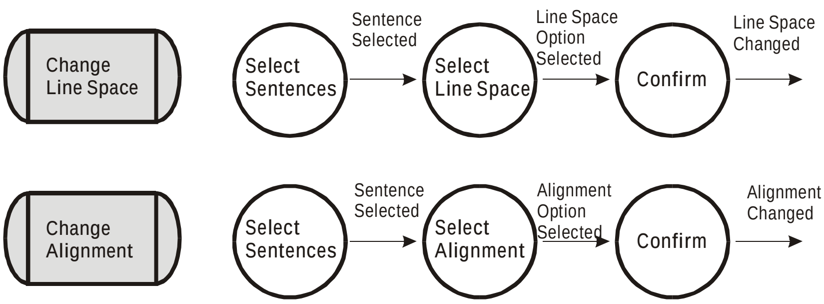


Fig.6 Example of task description with task procedure model

In this case, it is almost impossible to predict the UI problems of a system caused by interface of system. Task procedure model can express users' knowledge on the task of the system, but not the detailed information of system that can be used as clues for users to make their decisions. Interface unit reinforces this problem. The interface unit is a kind of prediction for the system state and doesn't have to be defined into elaborated level. Instead, SIR, the description of required information to perform a task is used to evaluate the interface unit.

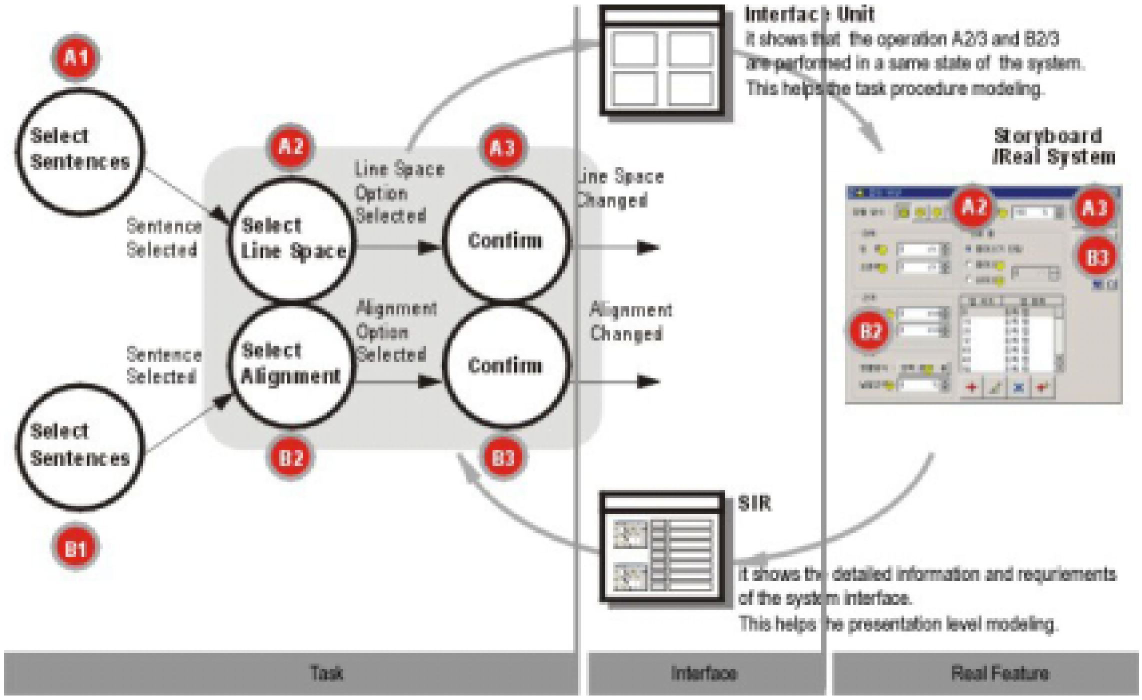


Fig.7 Setting relationship between task procedure and system state with Interface unit

Linking task procedure knowledge and the interface enables engineers, programmers, and designers to use their own process and knowledge for the successful communication and reduce the burden to learn the knowledge of other team.

4. Case Study

LTM was adapted to develop a UI of a multi functioned cellular phone for a Korean cellular-phone company. HTA (Hierarchical Task Analysis) method was used to set the task structure of the cellular phone first. About 200 tasks were categorized into several groups by the similarities of their goals, and revised through a card sorting technique. The information required to perform each task were recorded as SIR.

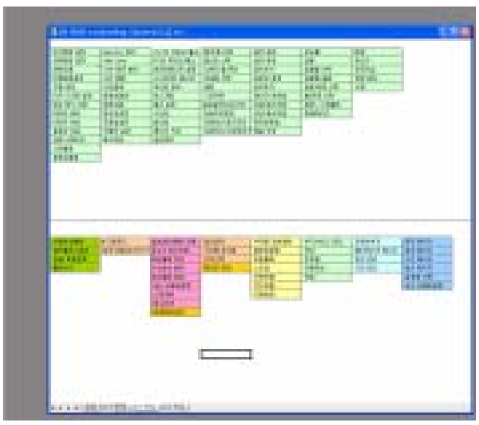


Fig.8 Information hierarchy of system based on HTA

The procedure of each task was described in an IFD format; the diagram for the flow of user' operation. It includes the information on users' operation, states of the system and buttons or controls that users operate to perform tasks. All of states were indexed to figure out their sequences and relations. All tasks were analyzed based on IFD and decomposed into several basic tasks. Finally, 20 basic tasks were elicited and they could make up more than 200 high level tasks. Through this constructive process, all of task could be designed with consistency. These task procedures could be used not only in design phase, but also in analysis phase as a guide for the task analysis.

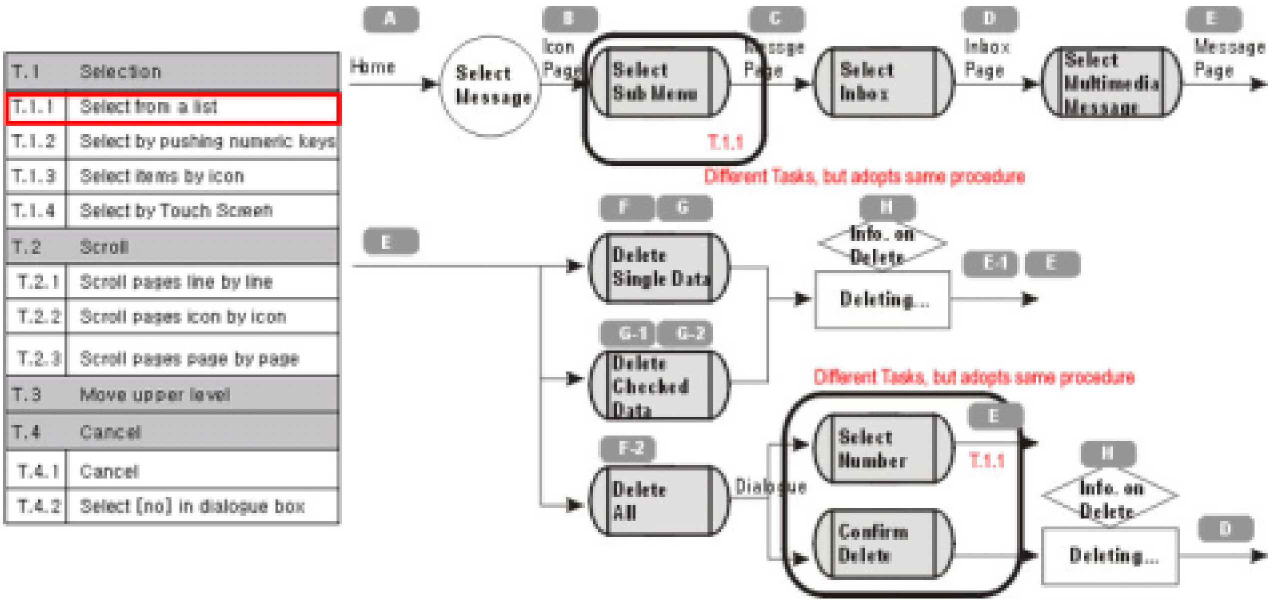


Fig.9 Task procedure model with the indexing on system state

Next, a set of storyboards was designed and allocated to the every system states of IFD to figure out the information requirements at the presentation level and the information requirements are added to SIR. Every storyboard has the same index with that of task procedure and could be developed from the interface unit. Unlikely interface unit, storyboard requires detailed information requirement for each state of task. When more than two phase of user task were performed in one state, task procedures could be revised due to the SIR. Converting interface unit and task procedure into storyboard is a support for a designer's intuitive idea into systematic one.

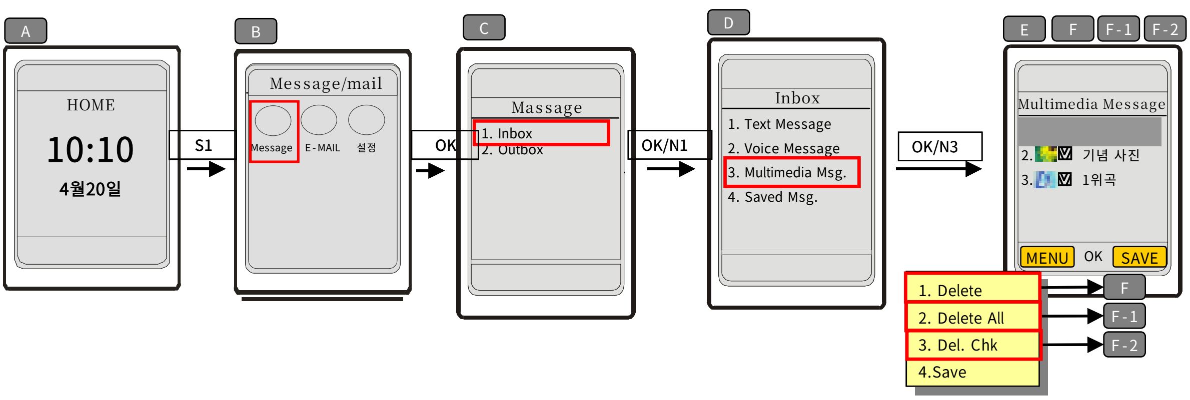
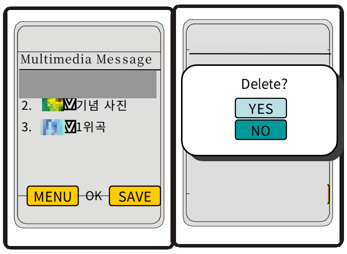


Fig.10 Storyboard of system design elicited from task procedure model

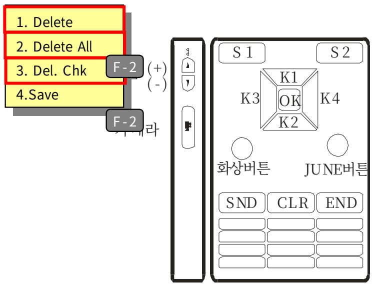
When first design process was ended, whole process were performed again to refine the three models of LTM to find out and revise UI problems in detailed level. The information on system could be founded only after a visually concrete prototype, like storyboard, is designed because it enables designers to simulate their system as like users do. All of interaction models, SIR, interface unit, and task procedures could be changed or updated at this stage, too.

System Information Requirement



Path : MESSAGE/Message /Inbox/Multimedia Message

Sub Task	Interface Components
1. Delete	Soft Button Menu(S1)
2. Delete All	Soft Button Menu(S1)
3. Delete Checked	Soft Button Menu(S1)
4. Save Message	Soft Button Menu(S1)
5. Save	Soft Button Menu(S1)
6. Select	Soft Button Menu(OK)



Sub Task	Interface Components
Call Related Task	S1
Call Related Task	S2
Move Focus(Up)	K1
Move Focus(Down)	K2
Select	OK
Home, End	End
Cancel	CLR
Select in List	N
Check Message	TOUCH

Fig.11 SIR and detailed design information requirements for system UI design

5. Conclusion

Linked Task Model supports iterative design process and prediction and solution of UI problems in a system hardly detected in the early stage of design by a single model. LTM also supports the user-centered design by adopting task procedure model as a basic interaction model and supporting it with information hierarchy model and interface model. In the case study of application of LTM onto the cellular phone interface design process, designers and UI engineers could find out various kinds of UI problems and suggested creative ideas evaluated by various team members with SIR and Interface unit. The result of UI design in the accomplished UI documents was easily shared between teams. The future studies should be focused on the development of computer-supported environment for LTM.

6. Reference

1. Biere,M, Bormsdoft,B., Szwillus,G. The Visual Task Model Builder, Computer Aided Design of User Interfaces II, 245-256(1999)

2. Burns,C.M., Vicente,K.J. Model-based Approaches for Analyzing Cognitive Work: A Comparison of Abstraction Hierarchy, Multilevel Flow Modeling, and Decision Ladder Modeling, International journal of Cognitive Ergonomics 5(3), 357-366(2001)

3. Rasmussen,J., Vicente,K.J. Coping with Human Errors Through System Design: Implications for Ecological Interface Design, Int. J. of MMS,31,517-534(1989)

4. Limbourg,Q. Towards Uniformed Task Models in a Model-Based Approach, DSV-IS, 164-182(2001)
5. Silva,P.P. User Interface Declarative Models and Development Environments: A Survey, DSV-IS 2000,207-226(2001)
6. Dittmar,A., Forbrig,P. Methodological and Tool Support for a Task Oriented Development of Interactive Systems, Computer Aided Design of Uer Interfaces II, 271-274(1999)
7. Hix,D., Hartson,H.R. Developing User Interface Ensuring Usability Though Product & Process, John Wiley & Sons(1993)
8. Pong,F., Dubois,M. Formal Specification of Complex Coherence Protocols Using Symbolic State Models, Journal of ACM(1998)
9. Farenc,C., Palanque,P. A Generic Framework Based on Ergonomics Rules for Computer Aided Design of User Interface, Computer Aided Design of User Interfaces II, 281-292(1999)
10. Paterno,.F., Mancini,C, Meniconi,S. Concur Task Trees: A Diagrammatic Notation for Specifying Task Models, Proceedings Interact'97, July(1997)
11. Scogings,C.J., Phillips, C.H.E. Linking Task, Dialogue and GUI Design: A Method Involving UML and Lean Cuisine+, Interacting with Computers, 14, 69-86(2001)
12. Lim,Y.K., Sato,K. Development of Design Information Framework for Interactive Systems Design, Asian design conference, BD.13(2001)
13. Yoon,W.C., Park,J., Lee S.H. A Diagrammatic Model for Representing User’s Interface Knowledge of Task Procedure Knowledge of Task Procedures, In Proceedings of Cognitive Systems Engineering in Process Control, Kyoto, 276-285(1996)