International Journal of Hybrid Information Technology
Vol.9, No.3 (2016), pp. 375-388
http://dx.doi.org/10.14257/ijhit.2016.9.3.35

A Mobile Interactive System Generation Framework from Conceptual Model to Development Paradigm

Juanni Li[1,2], Qingyi Hua[1*] and Xiang Ji[1]

[1] *Department of Computer Science, School of Information Science and Technology,*
Northwest University, Xi'an 710075, China
[2] *School of Science, Xi'an Shiyou University, Xi'an 710065, China*
**Corresponding author, e-mail: huaqy@nwu.edu.cn*

Abstract

The design of interactive systems is a process of transformation from problem domain to implementation domain. A mobile interactive system design differs from traditional software design in the perspective providing to conceptual modeling. With a diversity of user requirements and devices, it more concentrates on knowledge about the context of use rather than static information of problem domain. This paper proposes a multi-dimension description method which can describe those concepts related to special user requirements in the special context from four aspects: the set of static conceptual elements, process set, constraint set and interactive set. Then based on the conceptual model, a mobile interactive system generation framework is proposed to accurately map those concepts into requirement analysis and the implementation process; under the guidance of design pattern, a usable and useful interactive system can be achieved. Finally, we give a prototype of campus navigation system in which we successful used the generation framework.

Keywords*: conceptual model, dynamic environments, design pattern, development paradigm, the context of use*

1. Introduction

Currently with the integration of the ever-growing information technology into our daily life, more and more people are enjoying the convenience brought about by mobile services. Meanwhile the variety of mobile devices, development platforms, interactive mode and the context of use also bring about challenges to the designers and developers of the interactive systems. How can user interfaces be generated for fulfilling the same needs although they have a different concrete appearance. Conceptual model-based approaches could represent a feasible solution for addressing such issues.

The term of conceptual model originally emerged in the database field and it was used to represent the data and their relationships. But now, the scope of conceptual model has been gradually broadened to these fields: Software Engineering, Knowledge Engineering, Human-Computer Interaction and Simulation, *etc.* According to the definition work of ISO/TC97/SC5/WG3 [1] on Conceptual Sachems and those concepts in computerized information systems, we can conclude the conceptual model (CM) is a set of concepts and their relationships, which embodies user's shared understanding for some perspective with respect to some domain of interest. Natalia Juristo [2] specified the properties of conceptual model: abstraction, non-ambiguity, ease of understanding and verification and implementation independence. The aim of modeling concepts is to formally define aspects of the physical and social world around people for the purposes of understanding

ISSN: 1738-9968 IJHIT

and communication among the various "stakeholders" (customers, users, developers, testers ,etc.) and enhance traceability, reliability, availability of the interactive system .

In mobile environments, the heterogeneity of technologies and the diversity of context add some properties to interactive systems, such as randomness, context-sensitive, individuation. And thus, conceptual modeling becomes more important as the problem domain moves further away from basic office needs familiar to developers. In order to faithfully represent the problem to be solved in user domain and map these data to the development process, many universities and research institutions focus on the study of conceptual model [3-5], which can be mainly divided into the following groups: classification and integration of concepts; the formal description methods; the mapping from concept to development paradigm; verification and assessment etc.

While the proximity of computing process and the physical process becomes more close, a mobile interactive system design differs from traditional software design in the perspective providing to conceptual modeling:

- In mobile environments, CMs would explicitly represent, not only these information within the interactive system, but also those uncontrollable variables in the external environments, which can affect the effectiveness of the courses of action or the values of the results;
- CMs must represent the elements within the problem domain without bringing in computational connotation;
- CMs are generated from reality for the purpose of gaining an understanding of user needs in the dynamic situation.

The remainder of this paper is structured as follows. In Section 2 shows existing description methods of CMs and some technologies from the conceptual model mapping to development paradigm. In Section 3, we propose a method to describe interactive systems in mobile environments from multiple dimensions: static set of elements, dynamic process description, domain knowledge constrains, interactive process. In Section 4, a mobile interactive system generation framework is proposed to accurately map those concepts in problem domain into the development process. Afterwards a case study of campus navigation system will be shown in Section 5 to illustrate the practical use of our framework, and then we conclude and shed light on future directions in Section 6.

2. Related Works

2.1. The Description Methods of Conceptual Model

The objective of CMs used in developing process is twofold. On one hand, a standard vocabulary can make the communication more smooth between users and designers, and reduce the dangerous risks of misunderstanding caused by semantic vagueness. On the other hand, a computing-oriented formal description method can be easily integrated into the usual development process to ensure the functionality and usability requirements of interactive systems. The entity-relationship model [6], proposed by Peter Chener, perhaps is the first conceptual model to be used widely world-wide. This model assumes that applications consist of entities and relationships and incorporates some of the important semantic information from three aspects: information structure, operation and constraints. The entity-relationship model describes the information in an easy way, but there is the lack of capacity of systemic organization .

The OORAM method [7] is a reference frame for a family of object-oriented methodologies. It offers the role model as the basic unit to describe the whole system and each one provides a more fine-grained control of message passing than the more common class or type. The role model specifies not only the messages that must be understood by an object, but also who are allowed to send the different messages. The OORAM method

proposes some different views, each of them can describe the problem space from different aspect, however in complex dynamic environments, it is difficult to ensure the generality and consistency of the concepts.

James Rumbaugh, *etc.*, [8] illustrate an OMT (Object Modeling Technique) approach based on the model-driven, which describes the problem space from three aspects: object model provides uniform static objects and structure of systems; functional model organizes objects to achieve a specific requirement; dynamic model are used to capture object interactions between computational objects. Graphical representation of the analysis process cannot cover all the information in OMT method required to build the system.

KAOS (Knowledgeable Agent-oriented System) [9] provides a language and method for goal-driven requirements elaboration. Major components of the KAOS architecture include agent structure, dynamics, and properties; the relationship between agents and objects; and the elements of agent-to-agent communication. Each agent has a generic agent instance, which implements as a minimum the basic infrastructure for agent communication. In order to help users and system designers analyze and describe the higher-level agent centered requirement of software systems, some scholars extend KAOS by modeling the uncertain property and integrates a fuzzy version of Z with the KAOS language [10].

The Unified Modeling Language (UML) emerged in the mid-1990s through the combination of previously competing object-oriented (OO) software engineering methods developed by Booch, Jacobson, *et al.*, Now UML has been adopted as the standard modeling language for modeling software systems [11]. It could be used to visualize, specify, construct and document the artifacts of a software system. However, for the UML notation to be effectively applied, it needs to be used with an object-oriented analysis and design method.

The Table 1 summarizes the evaluation of the aforesaid five conceptual models from four aspects: description ability, reasoning ability, scalability and the ability to be learned.

Table 1. Main Conceptual Model Features

	Description ability	Reasoning ability	Scalability	Learnability
ER	OK	so-so	so-so	great
OMT	so-so	OK	so-so	so-so
OORAM	OK	OK	OK	great
KAOS	OK	great	so-so	OK
UML	great	great	great	OK

2.2. Mapping the Conceptual Model to the Development Paradigm

Each CM acts like a pair of glasses and designers use them to observe the domain and user reality. Different glasses highlight different features, tone down others and hide other. It is difficult to retrieve anything that has been filtered through the CM, even if this information is required by the development process. If we want to recover the information lost in the CM filter, the only way is to reanalyzes the problem domain using a different CM [2]. However CMs used nowadays are oriented to specific development paradigm, such as OMT and OORAM, which are more or less directly related to specific implementation characteristics and these is a mapping relationship between CMs and paradigms. In mobile context, the rapidly growing amount and complexity of information available have compounded the problems of technical information delivery. So it is needed that a CM can be addressed by means of several different development approaches.

Oscar Dieste [12] proposes the generic conceptual models which are more expressive and less dependent on development approaches. The generic conceptual models build the concepts of interactive system through element maps, dictionaries and narrative description, but there is no detailed explanation how the model is applied to the process of system implementation. Coleman [13] and Champeaus [14] address the incompatibility between the CMs used in different development approaches, which represent the different problem issues. Once CM is performed using models bound a development approach, the development process is most likely to continue with the same approach.

Nowadays Ambient Intelligence implies the need for context-aware adaptation of user interfaces, which brings the new challenge to CMs. The current evolution of mobile technologies means that traditional CMs are not always suitable in dynamic environments and, consequently, need to be extended from internal system information to external service information, environmental information and user information. On the other side, the design of mobile applications is a tedious task owing to its intrinsic variability, such as: multiple computing platforms, multiple channels, multiple environments, and multiple users. To accommodate a diverse range of technical and business requirements, Generic CMs independent of any development consideration are needed. These models include a series of abstractions that are rich enough to adapt more easily to a wider range of problems.

3. The Description Method of CMs from Multi-Dimension

With information technology weaving themselves into the fabric of everyday life, user requirements of mobile applications are not only office needs for office staff, but also life and entertainment needs for everyone. Compared with traditional static environments, the task objective in cyber-physical space become more changeable and susceptible to interference. The same service, different users have their own individual requirements, and even the same user, he also has different requirements in the different context of use. So user objective is role-sensitive and context-sensitive in dynamic environments. According to [15] we can get the formal definition of user objective.

UserGoal *Achieve/Cease/Maintain/Avoid/Optimize[name]*
InstanceOf *…*
Role *userrole1, userrole2…*
Context *specific context description*

These sets of concepts ,which related special users to achieve specified goals in the specified context of use, includes knowledge elements, tasks, functions, behaviors and strategies. If there is only one view perspective to describe all information, it must be too complicated to be understood. So in this paper, we describe those concepts related to special user requirements from the following aspects: the set of static conceptual elements, the set of process, constraint set and interactive set (as shown in fig1), and each aspect independence makes problem comprehension easier. We can describe the concept set as Sconcept=(E,S,C,I), Where E = $\{e_1, e_2, ..., e_n\}$ is the set of elements; S=$\{s_1,s_2,\ldots,s_n\}$ is the set of states, and each state also can be expressed as Si = (S_{i0}, Evn, A, S_{in}), where S_{i0} is initial state, Evn is trigger event, A is the active set, S_{in} is target state after being transformed; C=$\{c_1,c_2,\ldots,c_n\}$ is the set of constraint, which includes element constraints and process constraints; I$\{i_1,i_2,\ldots,i_n\}$ is the set of interaction, and each interactive process can be expressed as SI=(E,M,Evn), where E is the set of elements used in interactive process, M is the set of messages between elements, Evn is the set of events corresponding to the message.

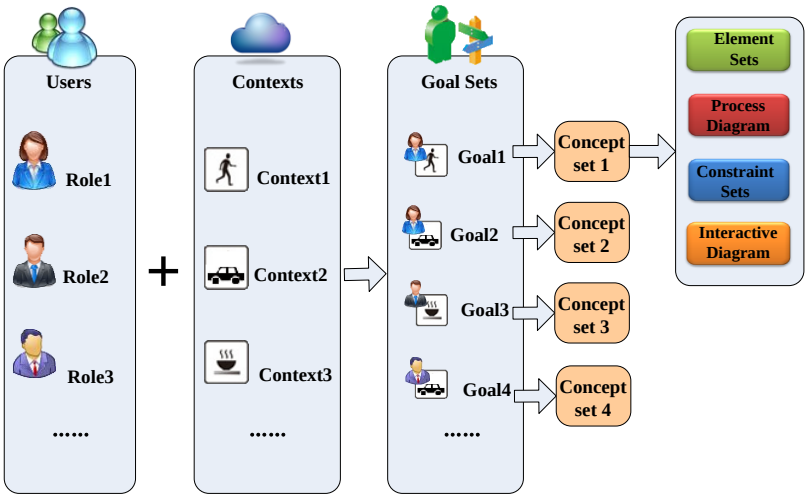


Figure 1. The Description Method of Conceptual Model from Multi-Dimension

3.1. The Set of Elements

CMs can explicitly represent part of the real world that will be implemented in the software system. The set of elements is defined as those critical information in problem domain which are represented in an uniform, abstract, verifiable way and endowed with standardized semantics. A sharable and reusable set of elements within the same domain can support better communication and understanding among the various "stakeholders". When the important elements in problem domain have been identified, we should describe the above elements exhaustively and determine the associations between them clearly. In general, those important elements include all the name, academic vocabulary, glossary, as they are shown in the Requirements Document. The identification of the well-suited development approach is based on a joint evaluation of key elements. Berry D, *et al.*, [16]. introduce a method to construct the set of elements through extracting noun list in "Use Case".

As a specific element, we give it a precise definition of semantics, in addition, we describe the element from multi views perspectives: synonyms/abbreviations, type, attribute, importance, the use of context. Among them, the element types can be classified by Abstract Type, Environment Type, User Type, System Type, Interactive Type; and the use of context explicitly represents geographic information, time information, social environmental information related to a given element. For example, in the "Mobile Campus" system, the element of "Course Information" can be described in Table 2.

Abstraction mechanisms of organizing elements inherent the UML basic relationship: dependency, generalization, association, aggregation; and add the contextualization relationship to more accurately describe the dynamic information. The contextualization relationship indicates there is different property or parameter in the different context when we describe the same element. A well-known example of such relationship is when a system changes its behavior depending on the location. The "Mobile Campus" system, for instance, user objective is to find a path to the destination. As far as the element of "path" concerned, if the user current location information cannot be captured, the path would be displayed from "start location" to "end location"; while the user current location information be captured, the path would be displayed from "current location" to "end location".

Table 2. The Element of "Course Information"

name	Synonyms/ abbreviations	Type	Attribute	Importance	The use of context
Course Information	CInfo	System	Class, Teacher, School hour, Teaching Places, Teaching Time		➢ User queries Timetable ➢ User queries Classroom ➢ User queries Teacher

Element *Path*
Type *Interactive*
Attribute *from: Location, to: Location*
Variant *(s: input source): {position provider, GPS, Wi-Fi}*
Variant *(u: user input): { type in, list select, map select }*
Context *get location information*
from← s
to← u
Context *not get location information*
from← u
to ←u

3.2. The Process Set of State Transition

While the set of elements describe the structure or static aspect of the system, the process set of state transition should contain or explain the behavioral dynamic aspect, in other words, the process set of state transition focus on the "how" and it intends to support the execution of the process. In the context of traditional interactive systems, user can predict task performance well enough so a plan set of activities will generate a reliable result. However in the mobile environments, advancing development of context-aware interactive systems gives incentives to increase the amount interactive channels and the diversity of devices, there are more unpredictable external influences. So instead of description user behavior in process- or action- oriented way as usually done in current methods, our approach will organize activities according to the system state transition in a state-based, declarative way . The following steps are included in the method.

1. Analyzing user objective in the special context of use;
2. Represent the desired state when the objective be achieved.
3. Keeping the interactive system in the desired state.
4. For some reason, there is some difference between the current state of system and desired state.
5. Some activities can be arranged to compensate for the effects of these influences, and then make the system back to the desired state.

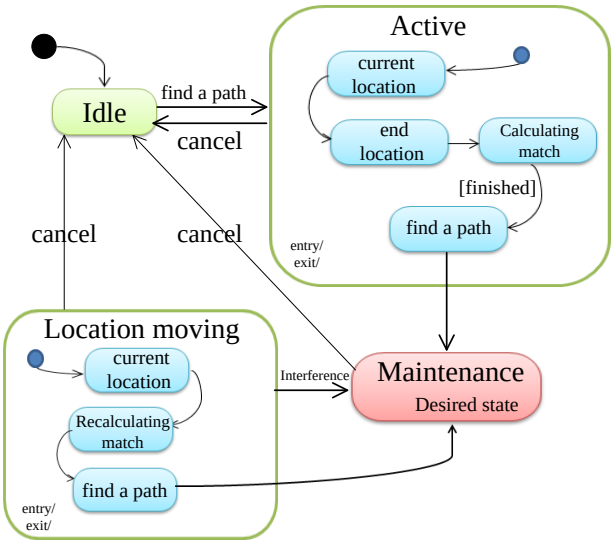


Figure 2. State Diagram of Path Query Process in Mobile Environments

Taking the process of "find a path" in mobile environments for example, as shown in Figure 2. Initially the system is in "Idle" state, and when users want to get a path from "start location" to "end location", the "find a path" event will trigger the state transition from "Idle state" to "Active state". The compound states "Active" which contains one or more regions, such as: getting current location, getting end location, calculating the match path and presenting it. And then the system reaches the desired state. But in dynamic environments, with the change of the current location ,"location moving" state is triggered, the information of current location needs to be re-captured, the match is re-calculated and then the new path is presented, the system reaches the desired state again.

3.3. Constraint Set

Constraint Set is specified as the rules and limitation description of domain knowledge which can be applied in the development process of interactive systems [17]. Domain-imposed requirements are facts of nature and reflect domain laws. This means that Constraint Set can be used to contract the space of solution and improve the efficiency of problem-solving. Domain constraint may be classified into element constraint and process constraint. Element constraint describes the rules and limitation of elements in special domain. For example, as a borrower, the time during he have borrowing a book is limited. The process of constraint condition operationalizing the element of "borrower" can be shown:

Element *Borrower*
Type *User*
Attribute *name, ID, role*
Constraint *(∀user: Borrower, b:book)*
Borrowing(user,b).time<LimitedBorrowingPeriod

Process constraint is any factor that limits the order of execution among interactive activities. For example there are a series of activities that should be performed by specified users to achieve "login", such as: input user name, input password, if and only if the first two activities terminate, and then user can enter the system. The process is shown as following (the formal definition about the temporal relationship between activities you can see [18])

Goal: *Achieve[Entering]*
InstanceOf *: SatisfactionGoal*
Concerns *: user, system,*

Context：*mobile environment, PC environment …*
Process：Ps_1*=(username/ / password >> enter in)*

3.4. Interaction Set

Interaction set focuses on the dynamic dialog process between users and systems. A interactive process is a collection of ordered steps intended to achieve a particular goal. Interaction set can be composed by a pair of elements and some messages passed by them. We can construct a interaction set according to sequential relationship of achieving a task, or structural relationship among those interactive objects. In the mobile environment, user input information is no longer the only driving force to the system, the context information implicitly existing in the outside environment is treated as the basis of selection strategy. The interactive process of "Campus navigation system" can be shown as Figure 3.

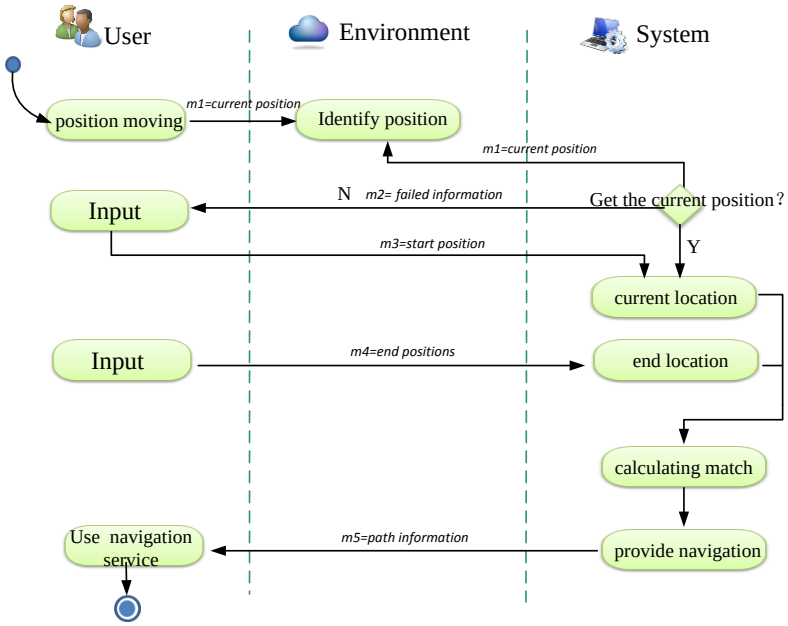


Figure 3. Interactive Diagram of Path Query Process in Mobile Environment

We can describe the above interaction set as SI=(E,M,Evn), Where E includes all objects related the whole interactive process, E={user, environment, system}; M is the message set in the process, (M⊆E×E), M={m_1=current location, m_2= failed information, m_3=start location, m_4=end location, m_5=path information}; Evn is event set, and each event contains message source, message, destination. If the system can get the information of user's current location, Evn would be shown as Evn_i={system, m_1, environment}, while the system cannot get the information, Evn would be shown as Evn_j={system, m_2, user}.

4. Generation Framework from Conceptual Model to Development Paradigm

The generation framework of an interactive system is a description of what the basic components of its implementation are, and how they are connected to support the required functionality and interaction with the user. The Generation Framework from Conceptual Model to Development Paradigm includes four processes: Need-oriented analysis,

Construction of a conceptual model, Construction of a design model and Computing-oriented analysis. As shown in Figure 4.

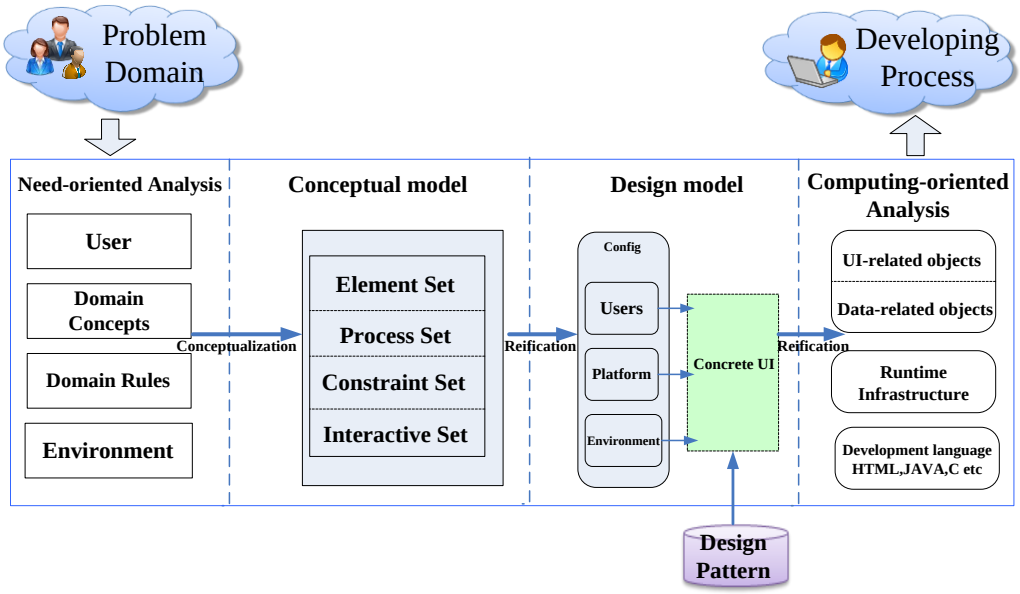


Figure 4. Generation Framework from Conceptual Model to Development Paradigm

4.1. The Process of Need-Oriented Analysis

The need-oriented analysis is a process of extraction user's requirement from current task situation through interviews, observation, tests of prior versions of the system, and documenting these data. The need-oriented analysis process is directly linked to the user need and is independent of the development approach. The goal of need-oriented analysis is to give the software engineer an understanding of the need in question. In mobile environment, interactive systems not only can help users achieve their goals effectively and efficiently, but also pay more attention to user experiences, so in the process of need-oriented analysis, apart from domain concepts and rules related functional requirements, it is needed to analyze user's characteristic, interests, experience of use computer and all environment information related the context of use. We try to change the unified requirements in static situation into the specified requirements of specified users to achieve specified goals in a specified context of use, and then make the interactive system more understandable and useful to users.

4.2. The Process of Modeling Concepts

How to switch from the need-oriented analysis step to the synthesizing step of user interface design, CMs can help to bridge the gap between them. Conceptualization is a semi-formal specification technique, combining formal elements for the specification of task structures and internal rules with informal domain concepts descriptions expressed in colloquial language. In this paper, the process of modeling concepts describes a specific problem according to four dimensions: Element Set defines those critical information in an uniform, abstract, verifiable way, thereby the abstract description of a specific system is semantically closed; State Set describes exactly state transitions caused by sequence of action; Constraint Set specifies the rules and limitation description of domain knowledge which can be applied in the development process; Interaction Set focuses on the dynamic dialog process between user and system. Each set includes a series of abstractions that are rich enough to adapt more easily to a wider range of problems.

4.3. Construction of a Design Model

Constructing a designer's model is a process to convert those data in concept models to specific platform, specific devices, and then generate a user interface prototype. In mobile context, a wide variety of computational devices and dynamic environment lead to the same service have different usability requirements in the different context of use. In order to accurately map those concepts of need-oriented analysis to design process and achieve a higher level of usable and useful interactive system, designers need to acquire not only the knowledge of design experience, but also the knowledge of how to use them. It has been summed up many common design patterns in pattern libraries, such as Jenifer Tidwell [19]、Martijn van Welie [20]、Yahoo [21] and so on. Those pattern libraries provides a good platform for developers to learn and disseminate knowledge and design experience, but these is no uniform pattern description specifications, such as the pattern library built by Jenifer Tidwell describes a specific design pattern from “Name” “Figure” “What” “Informa” “Why Use when” “How” “Examples” “In other libraries”, whereas the pattern library imposed by Welie from “Name” “Problem” “Solution” “Use when” “How” “Why” “More examples” “Literature” “Comments”. So we give a unified description method, as shown in Figure 5.

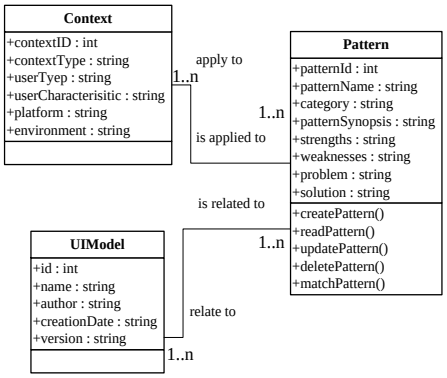


Figure 5. The Static Structure Diagram of Design Pattern

4.4. The Process of Computing-Oriented Analysis

The computing-oriented analysis process is directly linked to objects and their relationship used in the system from developer perspective, and during the software system-oriented analysis, the software engineer has to select the best development approach for building the software that is to meet the above needs. Accordingly, the computing-oriented analysis is dependent on the development approach, development environments, operating systems, development tools, programming languages *etc*. The elements being depicted in CMs are only mentioned as nouns in these short descriptions, while in computing-oriented analysis process, typically elements are introduced as instances of classes, which define object attributes, as known form object-oriented programming languages.

5. A Case Study

Campus navigation system is an application that can provide navigation assistance to users who are not familiar with the geographic information of campus. In traditional way, users need to input the information of their current location and destination location to the systems, and then gradually approach the destination according to other reference on the map. However in this way there is a heavy cognitive burden for users. In the mobile situation, the objective of navigation task will be changed in the different context of use.

For example, when the information of user's current location can be acquired through GPS devices, user want to get real-time visual guide in the map. While the information cannot be acquired, a reference route from origin to destination is needed. Without going into as much detail as all context, there is only a formal description to acquiring the information of current location.

UserGoal *Maintain[navigation]*
Instanceof *satifactionGoal*
Concerns *user, system, environment*
ReducedTo *CurrentLocation, EndLocation, InfromationPresent*
Systemgoal *Achieve[CurrentLocation]*
InstanceOf *LocationGoal*
Concerns *user, system, environment*
FormalDef $(\forall cur{:}currentlocation\ \ s{:}\ system)$
$cur \notin \phi \Rightarrow £[\ getting\ (s, cur)]$
//£P means"P holds in current and all future states"
Usergoal *Achieve[CurrentLocation]*
InstanceOf *LocationGoal*
Concerns *user, system*
FormalDef $(\forall u{:}\ user, end{:}\ endlocation\ \ s{:}\ system)$
$Inputting(u, end) \wedge end \notin \phi \Rightarrow £[\ getting\ (s, end)]$
Systemgoal *Achieve[InfromationPresent]*
InstanceOf *satifactionGoal l*
Concerns *user, system*
FormalDef $\cdots$

The goal of navigation can be decomposed three sub-goals: getting current location, getting end location, displaying navigation information, and the process of state transition and interaction between them can be shown in Figure2 and Figure 3. After modeling all concepts in the system, we can choose an appropriate design pattern from the pattern libraries, such as " Map Navigation" pattern in this case, as shown in fig 6. And then the user interface prototypes are generated as Figure 7.

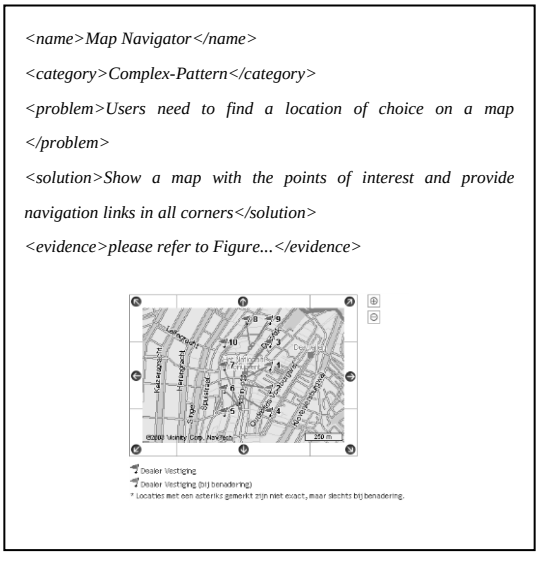


Figure 6. "Map Navigation" Design Pattern

(a) (b)

Figure 7. The UI Prototype of Navigation System

Figure 7(a) is a UI prototype in mobile device that can provide real-time visual navigation information, Figure 7(b) is a UI prototype in PC device that represents path information from origin to destination.

6. Conclusion

Conceptual modeling is a hot research topic in software engineering and human-computer interaction domain. CMs were advocated as a means for describing, documenting, and in fact analyzing problem space, and then with a systematic development method, a usable and useful interactive system can be achieve. The work we propose in this paper will contribute to the aspect of providing a multi-dimension description method which can describe those concepts related to special user requirement in the special context, and then a mobile interactive system generation framework is proposed to accurately map those concepts to the development process, and ensure traceability, reliability, availability for interactive systems.

In future, we intend to continue work to research on formal description and transformation rules of design patterns, development prototype of model transformation tools, which can provide a technical basis for the design and development of multi-target mobile interactive system.

Acknowledgement

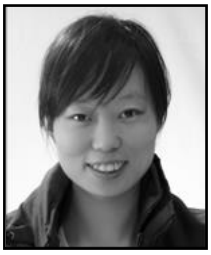
The research is supported by the National Natural Science Foundation of China (Granted No. 61272286) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (Granted No. 20126101110006).

References

[1] "International Organization for Standardization", Technical Committee 97. Sub Committee 5. Working Group 3. Concepts and terminology for the conceptual schema and the information base. American National Standards Institute, **(1982)**.
[2] N. Juristo and A. M. Moreno, "Introductory paper: Reflections on conceptual modeling", Data & Knowledge Engineering, vol. 33, no. 2, **(2000)**, pp. 103-117.
[3] H. Herre, "General Formal Ontology (GFO): A foundational ontology for conceptual modeling", Theory and applications of ontology: computer applications. Springer Netherlands, **(2010)**, pp. 297-345.
[4] J. A. Lara, D. Lizcano and M. A. Martínez, "A UML profile for the conceptual modeling of structurally complex data: Easing human effort in the KDD process", Information and Software Technology, **vol.** 56, no. 3, **(2014)**, pp. 335-351.
[5] Y. Ling, C. Xiao-Hong and L. Jing, "Consistency Analysis of Timing Requirements for Cyber-Physical System", Journal of Software, vol. 25, no. 2, **(2014)**, pp. 400-418.

[6] P. P. S. Chen, "The entity-relationship model-toward a unified view of data", ACM Transactions on Database Systems (TODS), vol. 1, no. 1, **(1976)**, pp. 9-36.
[7] T. Reenskaug, P. Wold and O. A. Lehne, "Working with objects: the OOram software engineering method", Greenwich: Manning, **(1996)**.
[8] J. Rumbaugh, M. Blaha and W. Premerlani, "Object-oriented modeling and design", Englewood Cliffs (NJ): Prentice hall, **(1991)**.
[9] R. Darimont, E. Delor and P. Massonet, "GRAIL/KAOS: an environment for goal-driven requirements engineering", Proceedings of the 19th international conference on Software engineering. ACM, **(1997)**, pp. 612-613.
[10] L. Zongtian, S. Kun and S. Zhiyong, "FKAOS: A Method for Agent Oriented Requirement Analysis", ACTA ELECTRONICA SINICA, **(2003)**, pp. 2171-2174.
[11] G. Booch, J. Rumbaugh and I. Jacobson, "The unified modeling language user guide", Addison-Welsley Longman Inc, **(2010)**.
[12] O. Dieste, "Development-Paradigm Independent Conceptual Models", available at http://grise.upm.es/miembros/oscar/documents/seke01.pdf.
[13] D. Coleman, P. Arnold and S. Bodoff, "Object-oriented development: the fusion method", Prentice-Hall, Inc., **(1994)**.
[14] O. Pastor, J. Gómez and E. Insfrán, "The OO-Method approach for information systems modeling: from object-oriented conceptual modeling to automated programming", Information Systems, vol. 26, no. 7, **(2001)**, pp. 507-534.
[15] A. Dardenne, A. Van Lamsweerde and S. Fickas, "Goal-directed requirements acquisition", Science of computer programming, vol. 20, no. 1, **(1993)**, pp: 3-50.
[16] D. Berry and J. Mullaly, "Designing for the user with OVID: bridging user interface design and software engineering", Indianapolis, IN: Macmillan Technical Pub, **(1998)**.
[17] W. Kang-Heng and J. Yun-Fei, "Planning with Domain Constraints Based on Model-Checking", Journal of Software, vol. 15, no. 11, **(2004)**, pp. 1629-1640.
[18] F. Paternò and G. Faconti, "On the use of LOTOS to describe graphical interaction", People and computers, **(1992)**, pp. 155-155.
[19] M. van Welie, "Patterns in Interaction Design", available at http://www.welie.com, Retrieved on, **(2012)** March 13.
[20] J. Tidwell, "Designing Interfaces, Patterns for Effective Interaction Design", available at http://www.designinginterfaces.com , Retrieved on, **(2012)** March 13.
[21] "Yahoo! Inc. Yahoo Design Pattern Library", **(2012)**, available at http://developer.yahoo.com/ypatterns/.

Authors



Juanni Li, She is a Ph.D. Candidate of Computer Science in School of Information Science and Technology, Northwest University Her research interests include human-computer interaction and model-oriented development.

Qingyi Hua, He is a Supervisor of PhD Candidates, professor of Computer Science in Northwest University, a member of professional committee of human-computer interaction in China Computer Federation. His research interests include human-computer interaction and UI engineering.

Xiang Ji, She is a Ph.D. Candidate of Computer Science in School of Information Science and Technology, Northwest University. Her research interests include human-computer interaction and interface design pattern.

