Describing multiple aspects of use situation: applications of Design Information Framework (DIF) to scenario development

Y.-K. Lim, School of Informatics, Indiana University, 901 E. 10th St., Bloomington, IN 47408, USA
Keiichi Sato, Institute of Design, Illinois Institute of Technology, 350 N. LaSalle, Chicago, IL 60610, USA

Scenarios have been broadly used to describe the context of a user's experience with products. Even though multiple aspects of user experience may be embedded in scenarios, they are not generally incorporated in a structured way. This research intends to develop methods of constructing scenarios by integrating multiple aspects of use situations with a coherent and accurate structure for evaluation. We introduce Design Information Framework (DIF) as a mechanism to accommodate multiple viewpoints and to represent them for effective management of design information. This research addresses the incorporation of this framework in a structured form of scenario generation to facilitate design problem identification and analysis from holistic viewpoints.

Keywords: design rationale, design tools, modelling, user behaviour, scenarios

We start with a story that shows a common problem caused by focusing on a single aspect of a use context when designers generate design solutions.

A product designer was trying to understand the processes used in blood testing facilities in hospitals. While observing the phlebotomists—personnel trained to take blood samples—the designer identified a problem. When holding the needle in a patient's arm with one hand, the phlebotomist needed to exchange tubes with the other hand. In this situation, there was a possibility that the phlebotomist could lose control of the needle, or mix up the order of the tubes or cause pain to the patient. Mixing up the tube order causes chemical contamination of blood samples which results in a faulty blood test. As a solution to this

Corresponding author:
Y.-K. Lim
younlim@indiana.edu

www.elsevier.com/locate/destud
0142-694X $ - see front matter *Design Studies* **27** (2006) 57–76
doi:10.1016/j.destud.2005.04.004

problem, a tube container that assembles the tubes in the preset order was introduced. This solution was generated by looking into the work procedure of phlebotomists focusing only on the specific operational aspect. It reduced the possible mistakes involved in placing the tubes in order, but the designer realised that this solution causes other problems such as further crowding an already tight workspace, and also creating more documentation tasks that modify the original flow of information and phlebotomists' activities in the space.

This story documents an instance of concept generation in a design process (Galvao, 2001). The solution in this story was generated by focusing on a single aspect of the activity of drawing blood. However, by combining several different aspects of the observed environment, designers can identify the problems that might have been overlooked by only focusing on the single aspect.

This story illustrates the following needs: (1) to recognise relevant aspects of a situation of use, (2) to manage examination of information across multiple aspects, and (3) to identify requirements for generating solutions that address problems discovered in multiple aspects. To address these needs, our research developed a mechanism for generating a holistic view by identifying problems and requirements from multiple aspects, and incorporating them into a scenario description.

A scenario is an imagined story of an event. A story illustrates an event that can be understood by most people. Like the example of the story of a designer's experience above, stories are an effective means of communicating experience and activities in context. Because of their narrative format, people with different backgrounds can easily understand them. Scenarios have been used for various purposes in the design process (Jonas, 2001), and can be categorised into the following five classes: problem description, future prediction, concept generation, requirements analysis, and detailed system design (Campbell, 1990; Carroll, 1995; Kaindl, 1995; Schoemaker, 1995; Simon, 1996; Alexander, 2000).

However, many scenarios used in design processes are not created by adapting a specific structure to build them. Although scenarios still have the advantage of boosting communication among development team members, an analytical use of scenarios in the process cannot satisfactorily be performed without having a well-specified mechanism for using them in the process. There have been good attempts to use scenarios in the software development process by re-structuring the scenarios into object-oriented forms, design rationale representation, and some informal card-sorting method (Carroll, 1995). These examples prove the potential for using scenarios for analytical purposes in design. Nevertheless, existing scenario development approaches do not use consistent methods for accommodating and manipulating multiple aspects of a use situation. Although designers sometimes do represent multiple viewpoints in a scenario, they often fail to incorporate certain important aspects of a situation. In this regard, we introduce a structured scenario generation mechanism that accommodates multiple aspects of the situation of use by decomposing complex situations into chunks, structuring them, and representing their interconnection. In order to develop this mechanism, we introduce Design Information Framework (DIF) that provides a fundamental structure for generating and integrating models that represent various aspects of the use situation which we call 'aspect models.'

1 Design Information Framework (DIF) and aspect models

Before introducing the scenario generation mechanism, we should explain what Design Information Framework (DIF) is and how we can generate aspect models using the DIF, which will provide the foundation for developing the scenario generation mechanism.

Design Information Framework (DIF) was developed to enable designers to organise and manipulate information throughout a design process (Lim and Sato, 2001). It is a unified information platform that utilises a modular information structure as a driving mechanism for developing various tools for design activities. One of the key issues addressed by the development of DIF is an incorporation of various representations of information created by the members of different teams having different backgrounds in the design process, and this fact supports the communication among them by translating one's representation to others through DIF.

In a design process, templates for archiving information into a DIF-structured database can be generated, and all types of design information, such as user studies data, design concepts, models, scenarios, and prototype models, are then structured by those templates (Lim and Sato, 2001) (Figure 1). DIF represents information on two levels: the lower level of representation consists of primitives that cannot be further decomposed into smaller conceptual units; the higher level of representation consists of elements which are composed of the primitives (Lim and Sato, 2001).

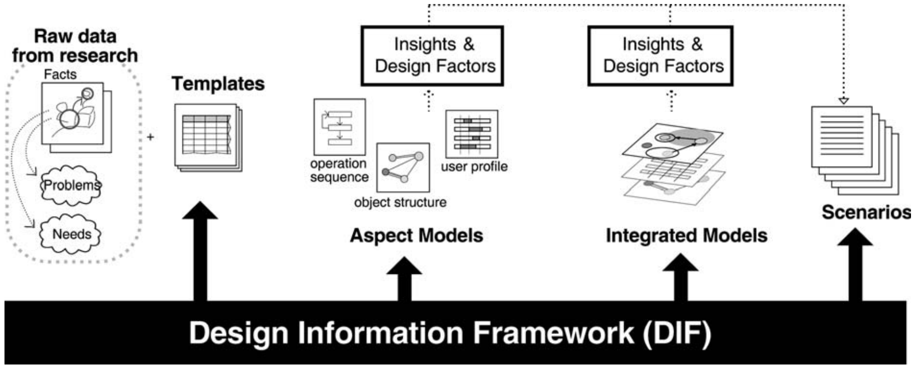


Figure 1 DIF in a design process as a platform

DIF has an open structure that does not limit types of information to be accommodated to a particular format. Each design project requires a different set of elements to be represented in DIF, and this set is called Project-based DIF (P-DIF) (Figure 2). The P-DIF can be determined by the nature of the project and the methodology used for the project. For example, if a scenario-based design method is used for an interactive system design project, the P-DIF for the project can include elements such as *goals*, *activities*, *settings*, and *conditions*, which correspond to theatrical play scripts that are typically structured with events, scenes, and props.

Once the information is organised with P-DIF, it can then be translated into various formats such as analytical aspect model descriptions, narrative scenarios, and specifications by using a set of elements and primitives as an intermediate mechanism between different representation formats. Each aspect model is defined by a set of parameters. For example, 'Mary writes down the information on the form, and administers the blood test to Nancy' describes an operation sequence; it can be represented with an operation sequence aspect model. It corresponds to the element, *action*, which is the parameter used to define this aspect model.

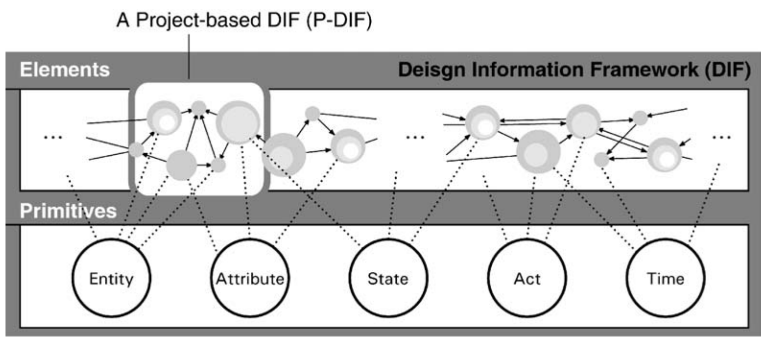


Figure 2 The structure of Design Information Framework (DIF) and P-DIF

'A phlebotomist, Mary, performs Specimen Collection on her patient, Nancy, who is 70 years old, and needs a blood test for a diagnosis of her heart condition' is from another aspect that focuses on the characteristics of the user as an actor in the scenario. It can be represented with the user profile model. This information element can be decomposed into primitives as follows: 'Mary' and 'Nancy' are *entities*, and 'a phlebotomist', 'a patient', and '70 years old' are *attributes*. Other primitives include *entity*, *attribute*, *state*, *act*, and *time*.

A good design solution corresponding to a problem identified on one aspect is not guaranteed to be a good solution for other aspects of the situation of use. One benefit of using aspect models lies in associating them into an integrated view which is a way to discover hidden problems. For instance, a problem found in the information flow aspect model can be viewed in other aspects such as an operation sequence or a physical layout of the space. Through this aspect integration and cross-referencing process, we can systematically understand the nature of a problem from multiple viewpoints. Integration of different aspects such as the spatial aspect of an environment with the operational aspect in the same environment is possible by associating different aspect composition rules that define the method of constructing aspect models (Lim, 2004). The aspect composition rule for defining each aspect can be constructed with the use of DIF. DIF enables specification of a set of components that define each aspect model and relations between different aspects. As Figure 3 shows, for example, a description of the spatial aspect, 'The computer is located on the table at the corner of the room,' can be represented with DIF structure: an *entity*, 'the computer,' and a location *attribute* of this entity, 'on the table at the corner of the room,' and this set, {*entity*, location *attribute*} forms the necessary components to define the spatial aspect. A description of the operational aspect, 'Nancy is operating the computer,' can be represented with DIF: *entities*, 'Nancy' and 'the computer,' and an act, 'is operating.' Mapping one aspect to another aspect like these two aspects is possible by matching commonly shared elements in the sets such as *entities* (one of primitives) for this example (Figure 3).

2 *Scenario generation for representing multiple aspects of use situation*

The proposed scenario generation mechanism utilises the structure of DIF and aspect models. It consists of the following three steps: (1) organising and structuring user research data with the P-DIF to prepare for creating aspect models that represent user experience that are

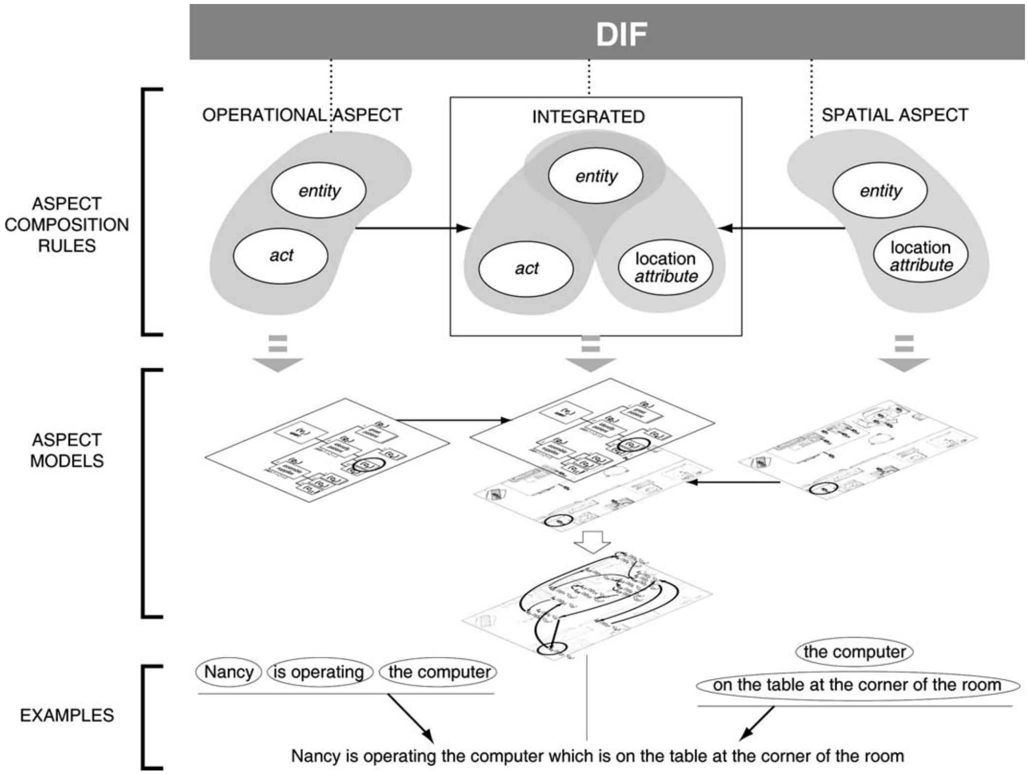


Figure 3 The mechanism of integrating different aspects of a situation of use through DIF

researched, (2) producing multiple aspect models of the user experience for identifying problems and patterns, and (3) generating a scenario by translating the aspect models into a narrative format based on the structure of a scenario building block, which will be explained later in detail.

2.1 Organising design information with DIF

In order to organise design information with DIF, we defined the meaning of each primitive as shown in Table 1. Precise definition of each primitive is critical in terms of defining an aspect model that consists of elements that are defined by a set of primitives, as explained in the previous section.

Figure 4 shows how the same information from the introductory story can be organised using the P-DIF format. Each element is described by a set of primitives. The organised information is then readily translated into the sentence format as shown in Figure 4. *Action* c_{22} "a

Table 1 Definition of primitives in DIF

Primitives	Definition
Entities (E)	Basic indicators of things in the world. Each entity can be described by a set of attributes (Yoshikawa, 1987).
• Agents (E^a)	Entities which are the performers of acts.
• Human (E^h)	Human entities
• Users (E^u)	Human agents ($E^u \equiv E^a \cap E^h$)
• Objects (E^o)	Entities which are subjects of acts. They could be outputs of activities or actions.
• Tools (E^{tl})	Entities which are used as tools for acts. These could indicate inputs or intermediate objects to produce the outputs.
• Spaces (E^c)	Entities which define spaces.
Attributes (R)	Properties that describe entities.
Time (T)	Indicators of time • time duration • time point • temporal relations (Sato, 1991) - if X and Y indicate certain events, the relations are as follows: X before Y, X equals Y, X meets Y, X overlaps Y, X during Y, X starts Y, X finishes Y
States (S)	Changeable attributes
e.g. Locations (L)	One example of states could be 'Locations' which indicate placement within a space.
State changes (V)	It describes the change in an entity's state.
Acts (A)	Acts or behaviours performed by agents (E^a) to create an output (objects (E^o)) through the state changes of agents (E^a) or tools (E^{tl}).

phlebotomist performs a venipuncture with a syringe'' as shown in Figure 4 can be represented with the DIPs, an *agent* ''phlebotomist'', an *act* ''performing'', an *object* ''venipuncture'', and a *tool* ''syringe'' as follows:

$$\begin{aligned} \mathtt{c}_{22} &= \text{``a phlebotomist performs a venipuncture with a syringe after } \mathtt{c}_{21}\text{''} \\ \mathtt{e}^{u}_{1} &= \text{``phlebotomist''} \\ \mathtt{a}_{22} &= \text{``performing''} \\ \mathtt{e}^{o}_{12} &= \text{``venipuncture''} \\ \mathtt{e}^{tl}_{14} &= \text{``syringe''} \\ \mathtt{t}_{20} &= \text{``after } \mathtt{c}_{21}\text{''} \\ \mathtt{c}_{22}\, &(e^{u}_{1}, a_{22}, e^{o}_{12}, e^{tl}_{14}, t_{20}) \end{aligned}$$

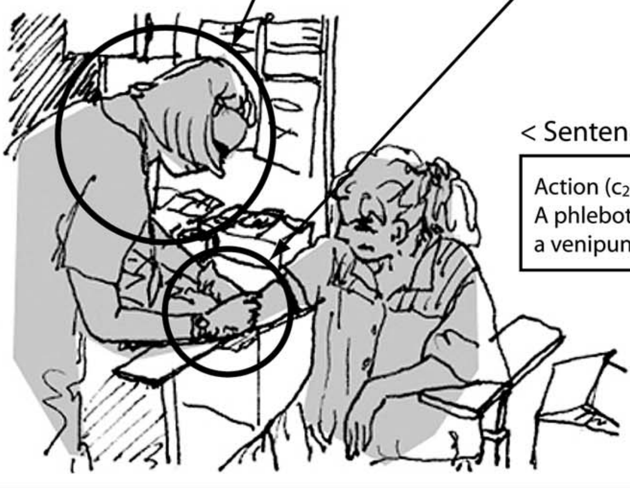
where $\mathtt{c}_{21}$ is required to be performed prior to the initiation of $\mathtt{c}_{22}$.

Maintaining a unique identifier for each discrete element and primitive from DIF-based data is useful when we have a computer-supported tool with DIF-based database. It allows re-visiting the original user study data that are used to create aspect models in order to recall the real situation. Besides this, it also enables automated generation of aspect models and integration of them, which will be further studied as an

<P-DIF Formated Data>

DIL	DIPs				
ACTION (C)	**TIME (T)**	**USER (E^u)**	**ACT (A)**	**OBJECT (E^o)**	**TOOL (E^{tl})**
explaining a procedure (c_{15})	t_{15}	phlebotomist ($e^u{}_1$)	explains the procedure (a_{15})	patient ($e^o{}_2$)	
tying patient's arm with a tourniquet (c_{16})		phlebotomist ($e^u{}_1$)	binds (a_{16})	patient's arm ($e^o{}_3$)	a tourniquet ($e^{tl}{}_7$)
finding a vein (c_{17})	t_{16}	phlebotomist ($e^u{}_1$)	finds (a_{17})	a vein ($e^o{}_8$)	phlebotomist's eyes ($e^{tl}{}_{16}$) and hands ($e^{tl}{}_{17}$)
telling possible problems (c_{18})		phlebotomist ($e^u{}_1$)	tells possible problems (a_{18})	patient ($e^o{}_2$)	
palpating a vein (c_{19})	t_{17}	phlebotomist ($e^u{}_1$)	palpates (a_{19})	a vein ($e^o{}_8$)	phlebotomist's hands ($e^{tl}{}_{17}$)
tracing a vein (c_{20})	t_{18}	phlebotomist ($e^u{}_1$)	traces (a_{20})	a vein ($e^o{}_8$)	phlebotomist's hands ($e^{tl}{}_{17}$)
cleaning a site (c_{21})	t_{19}	phlebotomist ($e^u{}_1$)	cleans (a_{21})	patient's arm ($e^o{}_3$)	gauze ($e^{tl}{}_{15}$)
performing a venipuncture (c_{22})	t_{20}	phlebotomist ($e^u{}_1$)	performs (a_{22})	a venipuncture ($e^o{}_{12}$)	a syringe ($e^{tl}{}_{14}$)

<Video Data>



< Sentence Format >

Action (c_{22}):
A phlebotomist ($e^u{}_1$) performs (a_{22})
a venipuncture ($e^o{}_{12}$) with a syringe ($e^{tl}{}_{14}$).

Figure 4 An example of organising user data with DIF through analysis of video data extension to this research in the future. Some of other examples of the definitions of elements are shown in Table 2. Each element and primitive are represented with symbols to denote the definition of each element in a consistent manner.

2.2 Producing aspect models of a use context

Each aspect of a use situation can be represented as a model, and those models can be created from the user study data organised with DIF. The variables used for producing aspect models are distinct sets of elements and primitives. An integrated model, comprised of multiple aspect models representing multiple viewpoints, enriches understanding of use situations, and reveals hidden problems which otherwise cannot be captured with a single aspect model (Sato and Lim, 2000).

Figure 5 shows examples of aspect models. Figure 5(a) is a Hierarchical Task Analysis (HTA) model and Figure 5(b) is a spatial layout model. The HTA model is built with a set of elements, *activities* (v_i), *goals* (g_i), *plans* (p_i), and *actions* (c_i). The spatial layout model is composed of the collection of *entities* (e_i) and *location-attributes* (l_i). Numbers following each symbol denote instance numbers. The instance numbers for each element and primitive are kept as a unique identifier for each different instance of primitives. If the same kind of primitives such as agent entities (e_i^a), object entities (e_i^o), and tool entities (e_i^{tl}) have the same

Table 2 Definition of several elements in DIF

Elements	Definition	Relations with primitives and other elements
Actions (C) A set of Actions is indicated as C.	Actions are carried out by Agents, and executed by doing certain Acts with Tools as media to achieve the Objects of the Actions. A process of several steps of Actions defines an activity, a long-term formation that takes place over time to transform the objects into outcomes (Nardi, 1996).	C (E^a, A, E^o, E^{tl}, T) E^a: a set of agent entities, A: a set of acts, E^o: a set of object entities, E^{tl}: a set of tool entities, T: a set of time indicators
Goals (G) A set of Goals is indicated as G.	A Goal leads Actions by shaping Plans. It is defined with the agent of the Goal, the Goal act, and the Goal object. Goals make predictions about the actions and the preconditions for those actions (Schank and Abelson, 1977).	G (E^a, A, E^o)
Plans (P) A set of Plans is indicated as P.	A Plan decides Actions an Agent needs to perform and a method of performing Actions. It is defined with who is the plan-agent and what is the planning act. By finding the plans, the user intentions of an action could be unfolded (Suchman, 1987).	P (E^a, A)

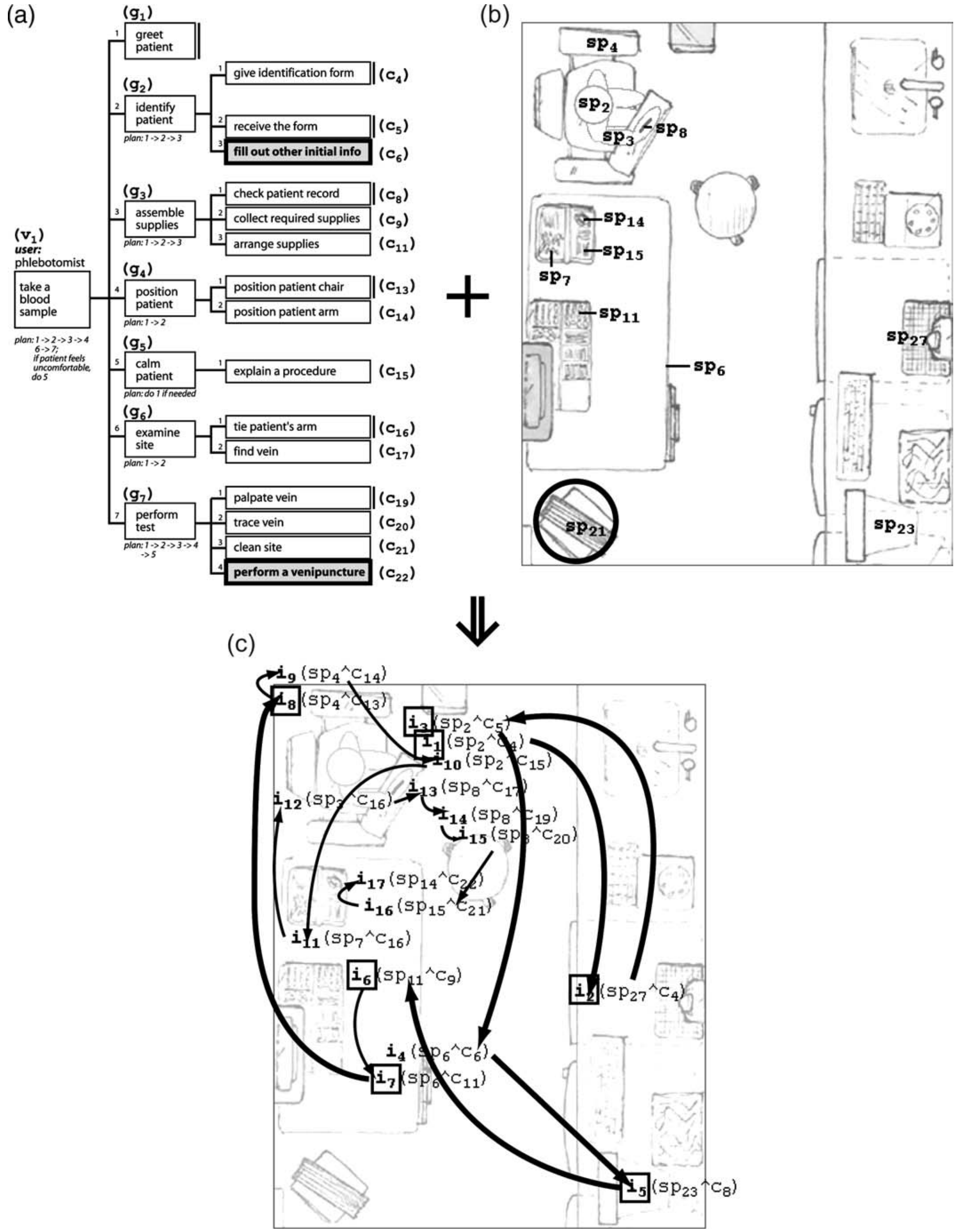


Figure 5 Examples of several Aspect Models. (a) HTA model – The elements, an activity *(AV), goals (G), and* actions *(C) are used. (Note: number sequence is from original user data.) (b) Spatial layout model – A space setting (SP) is represented by* entities *(E) and their* location-attributes *(L). (c) Integrated model – The shared primitives,* entities*, for both models are used for integration*

instance number, they indicate the same entity in the real situation. For constructing the HTA model, the goal structure and the task execution methods for achieving those goals of the specific user (the phlebotomist) were identified. Bold lines in the model representations such as c_{22} shown in Figure 5(a) indicate problems found during the observation. This enables designers to review the original observation data related to the problems in order to systematically incorporate those problems into the scenario construction.

Each aspect of a use situation can be represented as a model, and those models can be created from the user study data organised with DIF. The variables used for producing aspect models are distinct sets of elements and primitives. An integrated model, comprised of multiple aspect models representing multiple viewpoints, enriches understanding of use situations, and reveals hidden problems which otherwise cannot be captured with a single aspect model (Sato and Lim, 2000).

Figure 5 shows examples of aspect models. Figure 5(a) is a Hierarchical Task Analysis (HTA) model and Figure 5(b) is a spatial layout model. The HTA model is built with a set of elements, *activities* (v_i), *goals* (g_i), *plans* (p_i), and *actions* (c_i). The spatial layout model is composed of the collection of *entities* (e_i) and *location-attributes* (l_i). Numbers following each symbol denote instance numbers. The instance numbers for each element and primitive are kept as a unique identifier for each different instance of primitives. If the same kind of primitives such as agent entities (e_i^a), object entities (e_i^o), and tool entities (e_i^{tl}) have the same instance number, they indicate the same entity in the real situation. For constructing the HTA model, the goal structure and the task execution methods for achieving those goals of the specific user (the phlebotomist) were identified. Bold lines in the model representations such as c_{22} shown in Figure 5(a) indicate problems found during the observation. This enables designers to review the original observation data related to the problems in order to systematically incorporate those problems into the scenario construction.

Figure 5(c) shows an integrated model between the HTA model (Figure 5(a)) and the spatial layout model (Figure 5(b)). This model reveals problems by allowing designers to understand the task procedure in the context of the object arrangement. To integrate different models, the common primitives between different aspect models are identified to define the composition rule of each aspect model and the relationship between those models. In the following two descriptions, both the *tool*,

e^{tl}_{14} and the *entity*, e_{14} indicate the syringe. Therefore, an *action*, c_{22}, and a *spatial setting*, sp_{14}, are described as follows:

$$c_{22}\ (e^{u}_{1}, a_{22}, e^{o}_{12}, \underline{\mathbf{e^{tl}_{14}}}, t_{20})$$
$$sp_{14}\ (\underline{\mathbf{e_{14}}}, l_{14})$$

Several examples of the aspect models are described in Table 3. The more aspect models are integrated, the more holistic a situation can be simulated. However, it will easily become too complex for effective comprehension and analysis. In this scenario development mechanism, the association and integration of multiple aspect models are achieved by a step-by-step approach to deal with the complexity. It will be explained in the next section.

2.3 Scenarios generation using aspect models

Since aspect models represent a use situation from different viewpoints, together they can be used as a resource for constructing scenarios. The P-DIF provides the parameters to construct aspect models required in the design process, and the aspect models then generate components of scenario content as building blocks of the scenario (Figure 6) (Lim and Sato, 2003). For example, a layout model describes the spatial arrangement of the entities in the scenario. User profiles and a role-activity model provide the information about actors such as their jobs, roles, and basic relations with other actors. This information sets the scene for the scenario and provides background information on actors. When the scene contents influence the outcome of events, they need to be embedded in the story.

Table 3 Examples of aspect models with DIF (Diaper, 1989; Johnson, 1992; Kirwan and Ainsworth, 1992; Cross, 1994)

Function Structure	This model shows the hierarchical structure of Functions.
Hierarchical Task Analysis (HTA)	This is a task structure model that shows sequences and choices. The integration of the elements in the set Goals, Plans, and Actions produces this model.
Operation Sequence	This shows a sequential pattern of Actions.
Timeline Analysis	A graph of the set of Actions considering only two primitive sets; one is the set Acts (A), and the other is the set Time (T).
Spatial Layout	This model can be defined by Entities (E) and Locations (L) of those Entities.
Link Analysis	This model is defined by the relationships among the set A, T, and E^{o}, which are the parts of the primitive sets that compose the set of Actions.
Influence Model	This model is described by the integration of the elements in the set Roles, Conditions, and Requirements.
Information Flow	This model defines the flow of information entities in work environment. It is defined by the sequence of Actions considering three primitive sets; the set E^{u}, the set E^{o}, and the set E^{tl}.

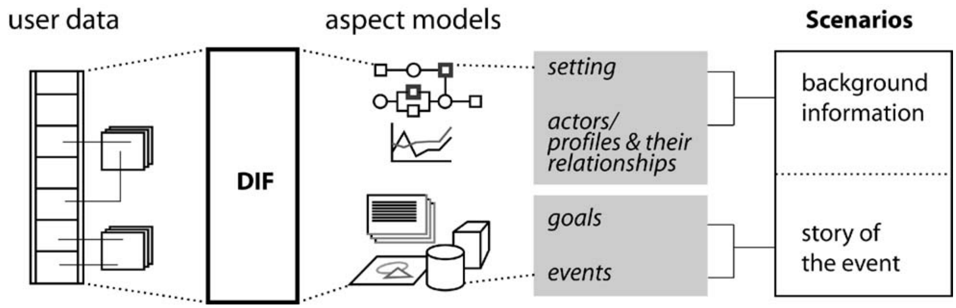


Figure 6 The process of generating scenarios with DIF

A typical scenario building block is defined by one of the following four information categories (Breitman and Leite, 1998; Carroll, 1998):

- **Actor Profile**: description about agents or actors who are the main subjects of the scenarios
- **Scene Setting**: description of the relationships among physical elements in the situation or among the actors involved in the scenario
- **Goals**: what the actors want to achieve
- **Events**: the main contents of the scenarios, describing actors' actions

The body of the scenario is constructed by events and goals as performed by actors. Goals drive actors' basic actions in a given event, and every scenario involves at least one actor and one goal. A Hierarchical Task Analysis (HTA) model provides the structure of the event in the scenario along with its goal and task structure. The explicit accommodation of the goal and task structure in scenario development helps using those scenarios for the various types of task analysis. Johnson et al. (1995) used Task Knowledge Structure (TKS) as the basic foundation for interpreting scenarios. The TKS describes the mechanism of carrying out activities by users with their task knowledge.

The scenario in Figure 7 was created by collecting relevant elements from several aspect models. The elements in the spatial layout model (Figure 5(a)) were used to form the content of the scene description in the scenario. For example, an element, $\mathtt{sp}_{21}$, in the spatial layout model was referred to when we created the scenario. It can be described with the primitives, *entity* $\mathtt{e}_{21}$ and *location attribute* $\mathtt{l}_{21}$, as follows:

$\mathtt{sp}_{21}$ (e_{21}, l_{21}), e_{21} = file container, $\mathtt{l}_{21}$ = on the biohazardous container.

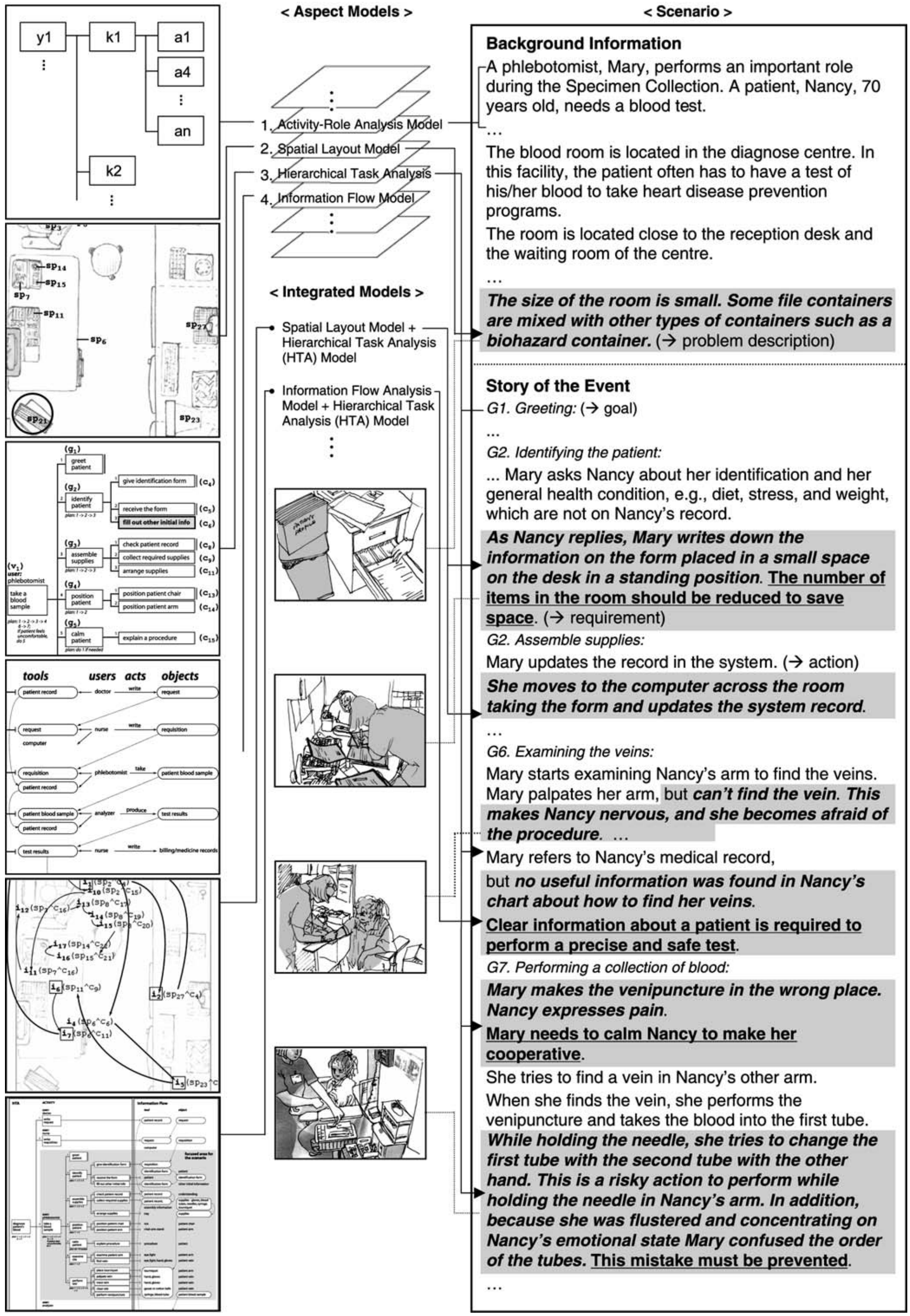


Figure 7 An example of scenario construction

And then it can be translated to a sentence, ‘a file container is on the biohazard container,’ which we can directly use to build the scenario content.

In the scenario in Figure 7, problem descriptions like $\mathtt{sp}_{21}$ are highlighted with bold italic letters. Requirements are highlighted with bold underlined letters. Information like the location of the blood room and the arrangement of the supplies within the room, in this case, is the kind that can be described by referring to the spatial layout model.

Actors within scenarios have *goals* to achieve through *actions* according to *plans*; it is helpful then to tell the story of *events* by articulating the goals that drive the actors. The HTA model was used to compose the story around the event of a phlebotomist drawing a patient’s blood that supports the goal-based structure of the scenario. The goals were distinguished in the scenario description as shown in Figure 7 (G_1 to G_7), making it easy to identify and codify the actions used to achieve.

Not only for the description of the scene in the story above, we can also take a great advantage of using aspect models as the source for creating the story of the event part in the scenario. The action ($\mathtt{c}_8$) of checking the patient’s record on the computer in the HTA model (Figure 5(a)) can be described with the DIF element and primitive notions as follows:

an action	$\mathtt{c}_8$ “checking the patient’s record” is comprised of
a user	$\mathtt{e}_1^{\mathrm{u}}$ “phlebotomist”
performing an act	$\mathtt{a}_8$ “checking”
on an object entity	$\mathtt{e}_{35}^{\mathrm{o}}$ “patient’s record”
using a tool entity	$\mathtt{e}_{23}^{\mathrm{tl}}$ “computer”
at a specific time	$\mathtt{t}_8$

$\mathtt{c}_8$ $(\mathrm{e}_1^{\mathrm{u}}, \mathrm{a}_8, \mathrm{e}_{35}^{\mathrm{o}}, \mathrm{e}_{23}^{\mathrm{tl}}, \mathrm{t}_8)$

Using this notation, we can track different parts through other aspect models to identify potential problems. For example, when action $\mathtt{c}_8$ in the HTA model is positioned in the spatial layout model, it is clear that the action requires users to move across the room to accomplish the goal.

Because the diagrammatic representation of aspect models effectively visualises the patterns of users’ experiences and the problems in use situations, the scenario development with this mechanism can easily represent the important user information like those problems and requirements extracted from user studies, which can be directly utilised for generating new product concepts.

3 Using scenarios to generate design solutions

The primary objectives of this scenario development mechanism are to view a situation through multiple aspects and to generate design requirements for system development. The design requirements elicited from the created scenario represent the criteria to be satisfied by a design solution for the identified problems. For example, as shown in Figure 7, cross-examination of the multiple aspect models revealed a problem, 'She moves to the computer across the room with the form and updates the system record.' The layout of the room regarding the spatial aspect becomes a problem for upgrading the patient information regarding the operational aspect. The corresponding requirement on this problem was specified as 'The room layout should reflect the pattern of phlebotomist's task execution.' As this example shows, the scenario indicates the requirements that correspond to the problems captured in user research from multiple aspect models, and particularly specifies the user needs to be satisfied in the solution. They are different from the performance requirements that specify how a system should be implemented. The performance requirements set limits to what has to be achieved by design to narrow the range of acceptable solutions (Cross, 1994). The requirements shown in the scenario guide the direction of generating new solution concepts.

Here are some examples of the requirements specified for this case study: (1) avoid conflicts among phlebotomist's tasks caused by the room layout (from the HTA model and the spatial layout model), (2) reduce the number of devices and containers to save space (from the spatial layout model), and (3) improve recording and transferring information accurately to other sections of the hospital (from the information flow model).

In the story given at the beginning of this paper, the solution that was generated from a single aspect for preventing phlebotomists from mixing up the order of tubes created other problems such as crowding a small space by its volume and creating more documentation tasks, which conflicts with the identified requirements. The requirements from multiple aspects, which were easy to miss by looking at just the operational aspect, were clearly identified by creating the scenario that accommodates the multiple aspects of the situation. All requirements including these example requirements then guided the generation of new concepts, and the final concept shown in Figure 8 (Galvao, 2001),

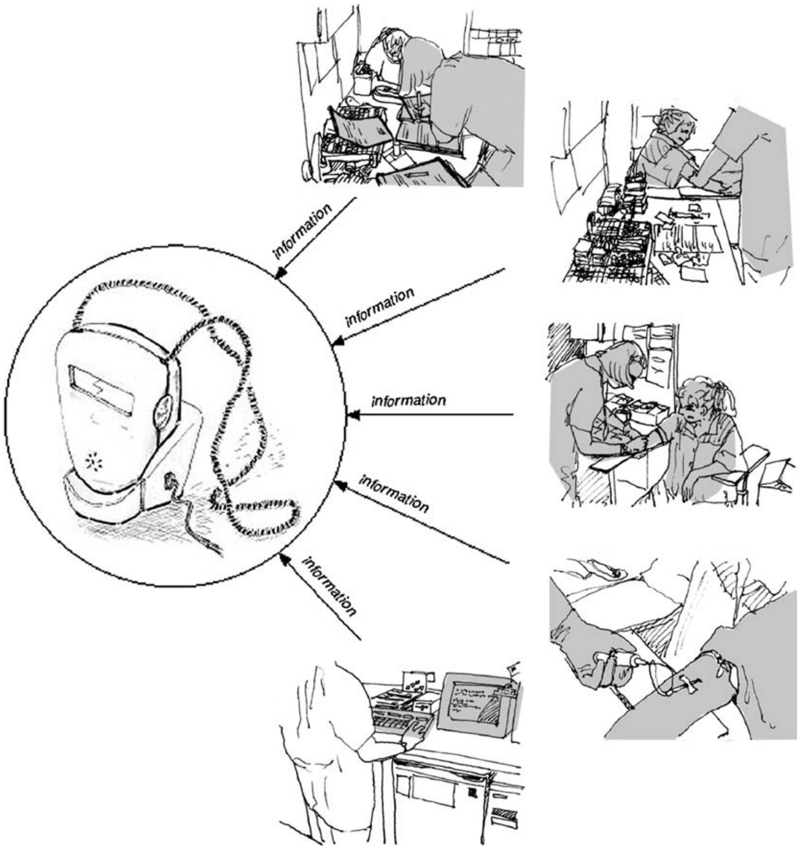


Figure 8 An example of a Solution Idea

a small device that supports the recording of patient information, was generated with a new direction from the initial concept presented at the beginning. By recording into the device what a phlebotomist says and what a patient says during the procedure, detailed information about what happened in the task procedure could be effectively captured. Through complementing the recorded data with other kinds of patient information, phlebotomists can easily understand what should be considered for each patient during the procedure.

This type of requirements helped the designer to generate the concepts of primary functions and usages of the product rather than to identify the solution details such as manufacturing materials or forms. Instead of only focusing on the operational aspect like the initial concept, the new concept also addresses the information flow aspect by supporting continuous patient history information on examination, and the spatial aspect by reducing paper-based documents and footprint requirement of the computer that affect both the size of the room and the layout of the room which correspond to the example requirements listed above.

This design example demonstrates that the scenario generation mechanism effectively supports identification of requirements that guide the generation of appropriate solution concepts at the early stage of the design process. The mechanism provided means to develop better understanding of users' situation by encouraging the generation and integration of multiple aspect models with DIF. Extending the problems identified from those multiple aspect models to the requirements enabled the effective application of the understanding of users' situation into the actual design solution generation.

4 *Discussions and future study*

This research developed a new mechanism for constructing scenarios in order to make the scenarios an effective means of supporting problem solving, communication, and more useful description of users' activities in context. Through this research, the following are achieved:

- developing a general structure of scenarios for providing a framework of scenario construction;
- viewing problems holistically through multiple aspect models; DIF provided the basic infrastructure to integrate different aspects of use situations. The formalised description provided an effective mechanism of integration;
- creating a method for generating scenarios through use of aspect models based on DIF structure;
- providing the environment for the evaluation of solution ideas in the early stage of the development process with scenarios, which can work as conceptual prototypes for simulating a new use context.

The major contribution of this scenario generation technique to design is that designers can effectively analyse complex use situations through multiple aspects, and identify problems and requirements that lead to further design problem solving. The scenarios that clearly embed rationales for solutions become valuable reference sources of use context information throughout the design process.

In this research, we did not address the issue of how to prioritise the aspects of user experience to model them. However, this issue is one of the critical issues in design especially when we deal with multiple aspects of concern. Applying a better mechanism to the prioritisation will further improve the usefulness of the scenario descriptions in design. For future studies, we will focus on the mechanism of selecting and prioritising the multiple aspects of user experience, especially in terms of utilising them into the scenario construction.

Acknowledgement

We wish to thank Takashi Kiriyama and Eli Blevis for the valuable comments, and Adriano Galvao for providing the design case to validate our work. We also thank Tangible Knowledge Research Consortium that sponsored this research at the Institute of Design at the Illinois Institute of Technology.

References

Alexander, D (2000) Scenario methodology for teaching principles of emergency management *Disaster Prevention and Management* Vol 9 No 2 pp 89–97

Breitman, K and Leite, J (1998) A framework for scenario evolution, in *Proceedings of the 1998 International Conference on Requirements Engineering (ICRE'98)*, The IEEE Computer Society pp 214–221

Campbell, R (1990) Developmental scenario analysis of Smalltalk programming, in *Proceedings of CHI '90*, ACM Press, Seattle, WA pp 269–276

Carroll, J M (1995) *Scenario-based design: envisioning work and technology in system development* John Wiley, New York, NY

Carroll, J M (1998) *Making use: scenario-based design of human–computer interactions* The MIT Press, Cambridge, MA

Cross, N (1994) *Engineering design methods: strategies for product design* John Wiley & Sons, Chichester, UK

Diaper, D (1989) *Task analysis for human–computer interaction* Ellis Horwood, New York, NY

Galvao, A B (2001) *Design for phlebotomy: user-studies and conceptual development Research and demonstration project report* Institute of Design, Illinois Institute of Technology, Chicago, IL

Johnson, P (1992) *Human–computer interaction: psychology, task analysis, and software engineering* McGraw-Hill, London

Johnson, P, Johnson, H and Wilson, S (1995) *Rapid prototyping of user interfaces driven by task models Scenario-based design: envisioning work and technology in system development* John Wiley, New York, NY pp 209–246

Jonas, W (2001) A scenario for design *Design Issues* Vol 17 No 2 pp 64–80

Kaindl, H (1995) An integration of scenarios with their purposes in task modelling, in *Proceedings of DIS '95*, ACM Press, Ann Arbor, MI pp 227–235

Kirwan, B and Ainsworth, L K (1992) *A guide to task analysis* Taylor & Francis, London

Lim, Y (2004) Multiple aspect based task analysis (MABTA) for user requirements gathering in highly-contextualized interactive system design, in *Proceedings of Tamodia 2004*, November 15–16, ACM Press, Prague, Czech Republic pp 7–15

Lim, Y and Sato, K (2001) Development of design information framework for interactive systems design, in *Proceedings of the 5th Asian International Symposium on Design Research*, Seoul, Korea

Lim, Y and Sato, K (2003) Encoding scenarios with Design Information Framework for the generation of multiple aspect models for system implementation, in *Proceedings of the 7th World Multi Conference on Systemics, Cybernetics and Informatics (SCI 03)*, July, Orlando FL

Nardi, B A (1996) *Context and consciousness: activity theory and human–computer interaction* The MIT Press, Cambridge, MA

Sato, K (1991) Temporal aspects of user interface design, in *Proceedings of the '91 International Symposium on Next Generation Human Interface*, Institute for Personalized Information Environment, Tokyo

Sato, K and Lim, Y (2000) Physical interaction and multi-aspect representation for information intensive environments, in *Proceedings of the 2000 IEEE International Workshop on Robot and Human Interactive Communication*, Osaka, Japan

Schank, R and Abelson, R (1977) *Scripts, plans, goals, and understanding: an inquiry into human knowledge structures* Lawrence Erlbaum Associates, Hillsdale, NJ

Schoemaker, P J H (1995) Scenario planning: a tool for strategic thinking *Sloan Management Review* Vol 36 No 2 pp 25–40

Simon, H A (1996) *The sciences of the artificial* The MIT Press, London, England 139–167

Suchman, L (1987) *Plans and situated actions: the problems of human–machine communication* Cambridge University Press, New York, NY

Yoshikawa, H (1987) General design theory and artificial intelligence, in **T Bernold** (ed.) *Artificial intelligence in manufacturing*, Elsevier Science pp 35–61