Applying Task Analysis to Facilitate the Design of Context-Aware Technologies

Yun-Maw Cheng
Institute of Information Science, Academia Sinica, Taiwan
kevinc@iis.sinica.edu.tw, http://www.iis.sinica.edu.tw/~kevinc

Chris Johnson
Department of Computing Science, University of Glasgow, UK
johnson@dcs.gla.ac.uk, http://www.dcs.gla.ac.uk/~johnson

Abstract: Current research in design of Context-Aware applications appears very technology focused, in particular software and sensor development and deployment rather than utilizing Human-Computer Interaction (HCI) principles such as task analysis to assist design of the applications. Developers specify what context-aware behavior to implement and determine what context information is needed based on intuition and introspection when they design applications of this kind. As a result, users of context-aware applications may have to repair the inappropriate predictions which the applications make about based upon likely user tasks, and immediate environments. This paper describes the approach of utilizing Hierarchical Task Analysis (HTA) to analyze the interaction between users and context-aware applications. The purpose of this research was to address the issue that there is not enough knowledge about where to discover and how to exploit context information. This is arguably the biggest obstacle when designing this type of applications. Case studies on analyzing existing context-aware applications are presented. The intention is to demonstrate the effectiveness and indicate the limitations of using HTA to better understand context-aware interaction. It is important to stress that we intend to exploit existing task analysis techniques instead of creating a new approach to validate existing context-aware applications. HTA provides an easy entry level for developers who have little knowledge about task analysis to inspect the existing context-aware applications and understand what previous developers of the applications consider context information and how they transform the information as input to their applications. This paper also describes a hybrid task analysis approach for modeling context and facilitating design of applications of this type. A case study is presented to validate this approach.

Keywords: Context-Aware Application, Hierarchical Task Analysis (HTA), Scenario-Based Design, Entity-Relationship Modeling

Introduction

Context-Aware Computing is currently a hot topic in multidisciplinary research fields. The trend is that computing devices and applications serve their users beyond the traditional desktop into diverse environments. A number of researchers have made claims about the benefits of context-aware applications. The applications can exploit not only explicit input from their users but also implicit input both from the users and their immediate surroundings to provide information tailored to the users' tasks (Chen and Kotz, 2001, Dey, 2001, Schmidt, 2000a). According to Schmidt, an implicit input is a user action that the user does not intend to perform to interact with an application. However, the application recognizes its meaning and considers as an input (Schmidt, 2000a). Implicit inputs may come from sensors which sense factors about user's activities, surroundings, location, etc (Masui and Siio, 2001, Selker and Burleson, 2000). The effect is to reduce the explicit input efforts as well as attention performed by the users when using applications of this kind. However, this increases the complexity associated with the design of context-aware applications. Developers must know what changes from users or environments are related to the tasks that the users perform to achieve goals with help from the applications. The changes may be considered as context information if the connection between user and application tasks and the changes can be found. However, the current state of this field is that there is no systematic approach to identify context information of applications uses in different domains. This results in that context information is defined by intuition and handled in improvised manner. In order to validate the claimed benefits, first of all, we must understand what context is considered and how it is exploited in the existing applications. However, it is a non-trivial task due to the considerable disagreement over the definition of "context-awareness" in human-computer interaction.

When designing an application, the first step is to understand what the application should be doing. Task analysis can help developers understand what needs to be accomplished by the application and break down the major task, the purpose of the application, into the simplest component part. Each component consists of clear goals and tasks. In order to carry out the application, developers need to know what information is needed for each task. The information may come from the user input or the state of current runtime environment. Traditional human-computer interaction requires explicit user input whereas context-aware applications can adapt both explicit and implicit input that is regarded as context information. In the case of designing a context-aware application, developers need to know what user tasks are necessary to operate the application and also need to figure out which part of user input can be transferred to the application task in order to increase the level of context-awareness of the application.

According to Lieberman et al, it is important to identify user, user and application task, and application models of a context-aware application in order to validate how it simplifies the interaction scenario. User model holds information about the user's current and past state and preferences that related to the current tasks. The user and application task model captures actions that are performed by a user to complete a task with help from an application. Application model describes the capabilities of the application itself (Lieberman and Selker, 2000). This paper reviews the existing context-aware applications that have emerged to help users in different domains. The hypothesis is that by utilizing HTA and our proposed hybrid approach in the light of Scenario-Based design and Entity-Relationship Modeling techniques for analyzing the interaction between users and context-aware applications, we can see what and how user's tasks could be reduced and better understand what context-awareness entails.

Describing Context-Aware Applications Subject to HTA

Hierarchical Task Analysis (HTA) focuses on the way a task is decomposed into subtasks and the order and conditions where these are executed. They are represented as a hierarchy of tasks, subtasks and plans. The output of HTA can be represented textually and diagrammatically (Dix et al., 1998, Shepherd, 1989). This technique has the potential to represent goal structures and cognitive decision-making activities. It is similar to GOMS in terms of its nature of decomposition, sequence representation, and task operation selection. However, GOMS is designed to express the internal cognitive processes when a person performs tasks. Thus, application actions as well as feedback are not stated in a GOMS task model. This raises difficulties for developers to capture application tasks in application-supported scenarios. In this research, we mainly focus on observable users' actions rather than their internal cognitive state. Therefore, this technique is not suitable for our approach.

Some researchers argue for the advantage of HTA is at an early stage in the design process (Carey et al., 1989). It can provide brief picture of user tasks and basic functional specification of the proposed application. It can also be exploited as a rough-and-ready tool at this stage. The nature of decomposition enables developers to concentrate on parts of the overall task without losing the picture of overall task activities. In addition, the top down structure ensures completeness and is easy to comprehend (Carey et al., 1989, Shepherd, 1989). Regarding the task-design mapping, HTA provides a clear description of all task functions for mapping on to the new interactive application. It is also ideal for the identification and mapping of information input and output requirements in design of applications (Carey et al., 1989). We can take advantage of this easy yet hands-on task analysis technique to identify the input and output of a context-aware application in order to understand how the developers of previous projects consider context information and how they transform the information as input to their s. The textual representation of HTA in this paper follows the format listed in (Dix et al., 1998). The following sections focus on the existing context-aware applications in different domains. In particular, we are interested in the analysis of user, user-application, and application tasks in each application scenario. These are highlighted after each task and sub-task. We look at the comparison between context-aware and non-context-aware approach based on the same scenario to illustrate differences in the resulting effect. This emphasizes the benefits of context-aware applications.

Office Utilities: The early demonstration of context-aware applications focused on tailoring, disseminating, and presenting information to users based on the current locations in the office domain. The intention is to improve not only the interaction between office workers but also the effectiveness of using computers, printers, and electronic equipment that helps with tedious repetitive tasks. The following describes call-forwarding service in this domain.

The call-forwarding application scenario is that the users can be tracked within a building and phone calls are forwarded to the nearest phone to them. The first demonstrations were based on the Active Badge application (Want et al., 1992). Users are required to wear badges and move around the building. Their location information can be obtained and updated to the database by the application. The database contains information about the users' current or most recent location, whether or not they are in their working places or offices. It also contains status message, and the nearest phone extension. When the receptionist receives a phone call for a particular user, she can use the database to look up the recipient's location information and forward the phone call to his/her last known location. The following scenario is based on the call-forwarding application at AT&T Laboratories Cambridge and assumes that the intended recipient exists (Cambridge, 1992a) (Cambridge, 1992b). This is not the only HTA we can produce. It is simply an example. The HTA of call-forwarding is as follows:

<u>Non-Application Support</u>:

```
Receptionist:

0. Receptionist forwards calls to intended recipient

       1. pick up the incoming call (User task)
       2. converse with the caller (User task)
       3. identify the intended recipient (User task)
       4. check the recipient's status (User task)
              4.1 check the recipient's in/out status from in-out board (User task)
              4.2 check the recipient's schedule (User task)
       5. appoint the next call with the caller for the recipient (User task)
       6. forward the message to the recipient (User task)
       7. check the recipient's extension number (User task)
              7.1 Get the phone list (User task)
              7.2 Look up the phone list to obtain the recipient's extension number (User
              task)
       8. forward the call to the phone in the office or somewhere close to (User task)

Plan 0: 1 - 4 in that order
        if the recipient is not available
        then 5 - 6
        else 7 - 8

Plan 1: do 4.1 - 4.2 in that order

Plan 2: do 7.1 - 7.2 in that order

Recipient:

0. update the current status (in/out, in a meeting, and etc.)
       1. go to reception desk/office (User task)
       2. inform the receptionist about in/out status and activity status (User task)

Plan 0: do 1 - 2 in that order


0. receive calls from others

       1. answer the nearest phone (User task)

Plan 0: do 1
```

We refined the HTA and include Active Badge in the analysis to see what happen when users use this technology.

<u>Application Support</u>:

```
Receptionist:

0. receptionist forwards calls to intended recipient
       1. pick up the incoming call (User task)
       2. converse with the caller (User task)
       3. identify the intended recipient (User task)
       4. check the recipient's status from application database (User-application task)
```

```
        5. appoint the next call with the caller for the recipient
        (User task)
        6. update the message to the recipient correspondent database entry (User-
        application task)
        7. forward the call to the phone close to the recipient
        (Use the result from task 4) (User-application task)

Plan 0: do 1 – 4 in that order
        if the recipient is not available
        then 5 – 6
        else 7

Recipient:

0. update current status
        1. operate the Active Badge (User-application task)

Plan 0: do 1

0. Receive calls from others
        1. answer the nearest phone (User task)

Plan 0: do 1
```

The HTA reveals that the application reduces the number of user tasks required for the receptionist and the recipient to perform call-forwarding. The application considers its users' location and their current status as context information. The application can obtain the context information about the user's identity, location, and timestamp of last seen and update the database automatically. That means the application considers the recipients' movement as an implicit input. This makes the interaction flows of completing the task smoother than doing the task without application support. It reduces the physical activities required to explicitly update information about the recipients' current situation. Updating one's current location requires that the telephone recipients go to inform the receptionist. Also, receptionist need not iterate to check recipients' current status and extension number using paper based list. Instead, the application can help integrate the information about the last known location, status, and closest phone extension to the recipients.

It is argued that the users want to have more control over the subsequent interaction tasks, depending on their current situation such as they do not want to take unexpected calls or receive instant messages when they are in a meeting (Adams, 2002, DeVaul and Dunn, 2001, Schigeoka, 2002). User's preference should be taken into account to make the application meet their users' social need. This illustrates key point about HTA gives no user perception of preference in each step.

Tour Guides: When a person visits a city or an exhibition, she can go to an information centre or counter to get a paper-based map and use it to guide herself. However, visitors might get lost if they cannot find the link between the physical place and the map. Other situation such as they might want to have personalized visiting routes. Context-aware applications in this domain tend to provide their users with information about their current location and suggest routes based on user's preferences (Chan, 2001b, Davies et al., 2001, Long et al., 1996, MacColl et al., 2002, Oppermann and Specht, 1999, Spasojevic and Kindberg, 2001, Youll et al., 2000). The user's preference may be obtained from a history of where previous users have been or the user's interests (Galani and Chalmers, 2002).

Many indoor exhibitions, for instance, museums provide their visitor not only with paper-based guides but also tape recoded guides. Both mediums provide predefined visiting routes and lack flexibility to adjust itself to suit their users' needs based on their current situation. For example, a visitor may feel bored with her current route or attracted by a particular exhibit. She may want to have another choice of visiting path. The paper-based and audio guide cannot support the dynamic nature of visitors' interests. The Hippie application was developed to avoid this limitation. It is an context-aware guide application in an indoor environment (Broadbent and Marti, 1997, Oppermann and Specht, 2000). The visitor carries a PDA and wears earphones while walking within the museum. Each exhibit is equipped with an infrared transmitter, which is used as a link to the corresponding digital information stored in the application. In the original prototype, information about the exhibits in the museum was cached on a PDA. The current development of Hippie has incorporated wireless LAN to provide dynamic information to the users.

Non-Application Support:

```
0. visit a museum using paper-based guide
       1. obtain a paper-based guide from the counter (User task)
       2. choose a categorized visiting path (User task)
       3. follow the categorized visiting path (User task)
       4. walk abound the museum (User task)
       5. stop at the interested exhibit (User task)
       6. look up information about the exhibit on the guide (User task)
       7. read the description of the exhibit displayed around the exhibit (User task)
Plan 0: do 1
        if categorized visiting route provided on the guide
        then do 2 – 7
        else do 4 – 7

0. visit a museum using audio guide
       1. obtain a audio guide from the counter (User task)
       2. choose a categorized visiting path (User-application task)
       3. follow the categorized visiting path (User task)
       4. walk abound the museum (User task)
       5. stop at the interested exhibit (User task)
       6. press the number displayed on the exhibit on the audio guide keypad (User-
       application task)
       7. hear the description of the exhibit displayed around the exhibit (User task)

Plan 0: do 1
        if categorized visiting route provided on the audio guide
        then do 2 – 7
        else do 4 – 7
```

Application Support:

```
0. visit a museum using context-sensitive mobile computing system (Hippie)
       1. obtain the device from the counter (User task)
       2. choose a preferred visiting path organized by the system
       (User-application task)
       3. follow the visiting path (User task)
       4. stop at the exhibit interest you (User task)
       5. information about the exhibit is presented through the earphone (Application
       task)
       6. want to discover different topic (User task)
       7. change current visiting path to another (Application task)
       8. follow the visiting suggested by the system (User-application task)

Plan 0: do 1 – 4 in that order
        if the visitor is attracted by something else
        then do 5 – 8 – 3 – 4
```

The Hippie development team claimed that the application utilizes the user's location/presence and preference as context information to simplify the user's visiting task. From the HTA, the task 2 and 3 in the non-application support are conditional while they are unconditional tasks in the application support. This emphasizes that a personalized visiting path is an essential function for a context-aware guide application. To personalize a visiting route for the visitor, the application asks the visitor what kind of tour they would prefer and then guides the visitor based on her preference. At this stage, the application needs to gather context about its user's preference explicitly from the visitor. During the visit, the application shows the visitor her current location and the path to the next planed exhibit on the PDA. This reduces the effort that the visitor needs to check the paper-based guide. An audio guide is arguably better than a paper-based guide because the visitor can visually focus on the physical environment and audibly receive the direction guide to the next exhibit. The HTA revels that the user's task of finding a description of an exhibit in the non-application support section can be transformed to an application task by adapting to user's location information. The task 5 to 7 in the non-application support section requires explicit interaction between the visitor and a paper-based guide or an audio guide. The visitor has to match the label on the exhibit with either the label on the paper-based guide or press the corresponding label (i.e. number) on the audio guide keypad to read or hear the description. However, the application support can detect the visitor's location and provide the information about exhibits automatically. As for task 6 to 8 in the application support, it is an exclusive for the context-aware guide application. For example, the visitor may be interested in a specific exhibit and want to know more about it by selecting the detail information option on the PDA's screen. The application can sense the implicit changes about the visitor's interest from the interaction between the visitor and the PDA. It may then suggest a new route to visit the rest of the exhibits in the museum.

Social Enhancement: In this application domain, we focus on the application that can recognize and adapt themselves to their user's current social situation while providing services. Current development of mobile phone is not designed for context-aware. Users must set an appropriate operation mode for their social setting. However, users often forget to setup their mobile phone to meet the current situation. Research on context-aware mobile phone focuses on the user's current situation, for example, location, activity, and co-location of the user and her mobile phone (i.e. in the pocket, in the user's hand, on the desk, etc.) and utilizes the information to enhance the quality of usage in terms of social aspects (DeVaul and Dunn, 2001, Lijungstrand, 2001, Schmidt et al., 2000, Tuulari, 2000). For example, a mobile phone detects that its user is in a meeting and does not want to receive any call except emergency ones. The mobile phone can then adjust itself to the "meeting" mode and apply the appropriate call filter during the meeting. Inspired by instant messaging (IM) services, Schmidt et al implemented their concept of "context-call" over the Wireless Application Protocol (WAP) (Schmidt, 2000b). In this case, the user or the mobile phone itself can publish the current situation and contact method to the central server. Callers contact the user by making a context-call in the same way as using IM services to see the status of a recipient and decide to make a call, leave a message, or call the user later.

The following scenario is based on the context-call development in TecO (Schmidt, 2000b). The scenario shows that a person is in the middle of a meeting at the customer site. One of her colleagues is calling her about going for a drink later.

<u>Non-Application Support</u>:

```
The person in a meeting:

0. change the mobile phone status to "meeting" mode
       1. press appropriate key set on the mobile phone
       (User-application task)
       2. check/answer the phone (User-application task)

Plan 0: do 1
        if the incoming call goes through the filter
        do 2
```

<u>Application Support</u>:

```
The person in a meeting:

0. change the mobile phone status to "meeting" mode
       1. check/answer the phone (User-application task)

Plan 0: if the incoming call goes through the filter
        do 1
```

The HTA shows the task 1 in non-application support is transferred from explicit user-application task to application task. The user's activity of walking into the meeting room is considered as implicit input for the system. The application support approach allows the user focus on his current tasks in the meeting with her customer and do not have to explicitly adjust her mobile phone to "meeting" mode. The user need not worry about whether the mobile phone has been set to an appropriate mode.

Games: A number of recent research implementations have built context-aware games to expand the arena from virtual space to mixtures of virtual and physical space (Bjork et al., 2001, Falk, 2001, and Headon, 2001). The aim is to evaluate how traditional game design can benefit from mobile computing, wireless communication, and sensor technologies. They want to investigate how to maintain and encourage social interaction in play. We look at "Pirates!", a context-aware multi-player game, and apply HTA to illustrate the differences between context-aware support and traditional game playing. This game exploits context information about its player's location, other players' location, and the location of game objects, such as treasures. The game scenario is that each player represents the captain of their ship. They have to walk around the physical game arena to obtain treasure and earn points. They may, however, be engaged in a battle with other ships nearby. Playing this game, the player carries a handheld device equipped with a wireless connection and a sensor receiver while they are moving around the physical game environment.

Non-Application Support:

```
0. play the Pirates! game
       1. move the game character around using game pad or keyboard (User-application
       task)
       2. search for treasures (User-application task)
       3. attack other ships (User-application task)
Plan 0: do 1 - 2 in that order
        if encounter other ships
        then do 3
```

Application Support:

```
0. play the Pirates! game using context-sensitive mobile computing device
       1. move around the game character by physically walking around the physical game
       arena (User task)
       2. search for treasures (User-application task)
       3. attack other ships (User-application task)

Plan 0: do 1 - 2 in that order
        if encounter other ships
        then do 3
```

From the HTA, we see the task 1 in the non-application support can be transferred from explicit user task to implicit user task. Namely, the application regards the player's movement is an implicit input. The game character, the ship, moves while the players walk around instead of pressing the buttons on a game pad. The benefit of application support is that the player can immerse into the game. The immersive experience in the game play would increase the level of excitement when the player playing the game (Headon, 2001, Schneider, 2001). Augmented Reality (AR), which tackles the research issue of interaction between human, physical, and virtual entities, is rather suitable to describe the interaction between the player, game application, and physical and digital game arena. Many researchers in this field tend to exploit the context-aware game applications as social interaction test-bed to discover more about how the players react to each other on particular game tasks (Dennis, 2001, Pering, 2001, Schneider, 2001).

We learned a lot about HTA. It does not capture human factor and social issues very well. Call-forwarding applications with active tracking sensor mechanisms allow users concentrate on performing relevant tasks to deal with their current situations without the disturbance from the application tasks. However, users may lose control of their privacy. In the case of smart mobile phone interaction, the real issue is not work saved for users but annoyance to their colleagues. In the context-aware game scenario, the HTA cannot address the issue of enjoyment.

A Hybrid Approach for Modeling Context Information

This section introduces a systematic approach for finding innovative uses for future technologies. It is to extract user tasks from situations that are elicited from a scenario. As noted by Carrol, scenarios are stories about people and their activities (Carroll, 2000a and Carroll, 2000b). Each scenario has a setting that explicitly describes the starting state of the current situation and implicitly depicts the characters that take part in the situation in the scenario. Each scenario has actors who perform tasks to achieve goals in different situations in a scenario. Each task can be regarded as what needs to be done in the situation. We analyze the user tasks in terms of the answers to the questions, "Who should be responsible for the situation?", and "What should be known to act on the situation?". HTA is utilized to picture and describe what happens in a scenario and presented in user, user-application, and application tasks performed in a scenario. In order to figure out the transformation between user and computer application tasks, we also adapt the Entity-Relationship Modeling to identify the relationships between entities, actors, and actions described in the HTA. In addition, the user would feel easier to stay in role and resolve any potential hesitation if the adapted scenario can reflect situation based on the user's previous experiences with realistic reasons for performing the tasks. The closer that the scenario represents reality, the more chance the useful context is discovered.

A Case Study

To illustrate how the approach can be applied, we introduce a case study, Virtual Notelet (Cheng and Johnson, 2001). Using this application, the user carries a wireless-enabled mobile device and walks around in the office environment. When the user stands in front of an office door, the device displays the occupant's status in the office and virtual notes virtually attached on the office door. The user can adjust her status in her office and leave virtual notes to others.

Interaction at office doors happens frequently in office environments. Office doors do not simply act as physical barriers to particular rooms. They also play a significant role in communicating information about the location and availability of the occupant. In a wider sense, they can also be thought of as a medium of communication for information from the occupant to her colleagues and vice versa. The doors to communal and shared locations play a similar role. It is often possible to tell if a room has been booked by looking for notes attached to the door. Similarly, if a meeting is being conducted then the same approach can be utilized to indicate whether it is socially acceptable to interrupt that meeting or not. People in the office domain may stop by an office door in order to find out if a colleague is in her office. In this situation, the visitor encounters a problem about whether to enter or not to do so whereas the occupant faces intrusion if she is engaged in something. How does the visitor put an anchor to the uncompleted activity in order to resume the interaction with the occupant if the occupant is not in the office or unavailable for the visitor? From our preliminary observation, some staff in our department use annotations to indicate their current status in the offices. Some of them provide their visitors with Post-It notes, which are attached on the door, so they can leave messages on the door. Some authors argued that people's actions at a door are determined by the status of the occupant in the office (Selker, 2000). Their observation shows most visitors perform the action "knock and wait", "check status", or leave notes". The actions such as "walk in" and "knock and walk in" are rarely happen. This reflects the importance of the annotation on an office door.

There are two situations we are interested. Firstly, a person stands in front of an office door and checks the status of the occupant. Secondly, an office occupant stands in front of her office door, and manipulates the annotations to indicate her in/out status, check the messages left by the visitors, and opens the door in order to enter her office or close the door to leave her office. The following describes three different situations when using the Virtual Notelet. The user task at office door is described using HTA to show a high level view of the interaction. The label, Role, is a light weight user model to indicate the type of user in the application scenario. We also utilize the entity-relationship modeling to figure out the entities, actions, and actors involve in the interaction n the scenario. The relationship between the entities, actions, and actors is important when we try to transform user tasks to computer tasks.

Situation 1: Approaching a colleague's office and want to know her status in the office when standing in front of the door.

Role: visitor

```
0. in order to meet the colleague in her office
       1. walk by the office door
       2. check the context board attached on the door
       3. leave a note (using  Post-It notes)
              3.1 write message on the note
              3.2 detach the note from the pile of Post-It notes
              3.3 attach the note on the door
       4. knock the door
       5. wait few seconds
       6. open the door
       7. walk in

Plan 0: do 1-2-4-5-6-7 in that order
        When the occupant is away from the office or busy in the office do 3

Plan 3: do 3.1-3.2-3.3 in that order
```

Object Visitor **human actor**

Actions:

```
V1-1: walk to the office
V1-2: check occupant's status showing on the door
V1-3: leave notes
V1-4: knock, open the door, and walk in the office
```

Object Post-It note **simple**

Attributes:

Affordances: `hold/fold/attach/detach/draw or write`

Events:

```
E1-1: occupant is free in the office
E1-2: occupant is busy in the office
E1-3: occupant is not in the office
```

Relations: object-object

```
Location (Post-It notes, office door)
Location (context board, office door)
```

Relations: action-object

```
patient (V1-2, context board and notes)
        - Visitor "sees" the context board and notes attached on the door
instrument (V1-3, Post-It notes)
        - Visitor writes down messages on the note and attaches it on the door
        using its self-attaching area on the back
patient (V1-4, door)
        - Visitor knocks and opens the door
```

Relations: action-event

```
before (V1-2, V1-3)
        - Visitor must check the office occupant's status before deciding whether
        to leave a note
triggers (E1-1, V1-4)
        - "Occupant's status is free" triggers the visitor to knock, open the door,
        and walk in the office
triggers (E1-2, V1-3)
        - "Occupant's status is busy" triggers the visitor to decide to leave a
        note
triggers (E1-3, V1-3)
        - "Occupant's status is away" triggers the visitor to decide to leave a
        note
```

As mentioned previously, the task analysis is not an end in itself. For instance, the events listed in the previous paragraph are very unlikely to provide a complete description of the changes that must be considered by the system. In contrast, the HTA represents an initial stepping stone between the informal scenario and the more detailed information required to move towards a prototype implementation. Both the scenario and the task analysis are refined by the insights that are provided once users can access the system. Considering building a context-aware application to help the user perform these actions we should determine what actions the application needs to perform and what input it expects. From the task analysis listed above, there are two human actor actions, V1-2 and V1-3, we are interested in. In more detail, the application should display the occupant's status in the office and provide a Post-It note like function so that a visitor can write messages and post it on the office door. From the application point of view, when its user stands at an office door it must first identify his/her role in the ongoing interaction. This can be done by requiring the user to "login" so the application knows whether she is a visitor or an occupant in the office. The login process can be implicitly adapting sensing technology or explicitly asking the user to type in her ID and password. Once the user's role is obtained, the application can perform subsequent actions. For instance, the application displays the occupant's status in the office. The acquisition of the information is described in situation 2. As shown in the action-event relations section, the occupant's status in the office determines the visitor's subsequent action, "leave a note or knock the door". The occupant's status can be regarded as an input to the application to activate its Post-It like function to the user. The user can write messages on the virtual note and virtually attach it on the office door.

Object Virtual Notelet **non-human actor**

Actions:

```
VN1-1: identify the user's role
VN1-2: display the occupant's status in the office
VN1-3: activate note editor
VN1-4: associate the virtual note with the physical office door
```

Object Virtual Post-It note **simple**

Attributes:

Affordances: `virtually attach/detach/draw or write`

Relations: object-object

```
        Location (virtual note, computing device (i.e. PDA))
```

The following indicates the input to each action performed by the Virtual Notelet. Reversely, the input is interpreted and presented as context information related to the application actions. It is important to note that these are not the label listed in the entity-relationship model.

Input to Virtual Notelet

```
        user ID and password • VN1-1
        occupant's status in the office (i.e. in/out) • VN1-2
        occupant's "bust" or "away" status • VN1-3
        user fires "attach" command • VN1-4
```

Context in Virtual Notelet

```
        [user] • VN1-1
        [office occupant's status] • VN1-2, VN1-3
        [virtual note manipulation command] • VN1-4
```

The following describes the interaction at occupant's office door when coming back to the office.

Situation 2: Office occupant is approaching to her office door from outside of the office and she wants to adjust her in/out status and check the notes left by others when she stands in front of the door.

Role: occupant

```
0. enter her office
        1. walk by her office door
        2. adjust the in/out status on the context board on the door
        3. check the notes attached by visitors on the door
        4. remove notes
        5. open the door
        6. walk in

Plan 0: do 1-2-3-5-6 in that order
              if any note attached on the door then do 4
```

Object Office occupant **human actor**

Actions:

```
                V2-1: walk to the office
                V2-2: adjust in/out status showing on the door
                V2-3: check and remove notes from the door
                V2-4: open the door and walk in the office
```

Object Post-It note **simple**

Attributes:

Affordances: hold/fold/attach/detach/draw or write

Object context board **simple**

Attributes:

Affordances: adjustable indicator

Events:

```
        E2-1: notes left by the occupant are attached on the door
        E2-2: notes left by other visitors are attached on the door
```

Relations: object-object

```
        Location (Post-It notes, office door)
        Location (context board, office door)
```

Relations: action-object

```
        patient (V2-2, context board)
                - occupant adjust the in/out status displayed on a context board
        patient (V2-3, Post-It notes)
                - occupant "see" and remove the notes attached on the door
        patient (V2-4, door)
                -   occupant open the door and walk in
```

Relations: action-event

```
        before (V2-2 or V2-3, V2-4)
                - occupant adjust her in/out/busy/free status and check notes attached on
                the door before she enter the office
```

```
triggers (E2-1 or E2-2, V2-3)
        - notes left and attached on the door by the occupant or others trigger the
        occupant perform the action "remove the notes"
```

When the user arrives at her office door the application must identify the relationship between the user and the office as described in situation 1. If the user's role is identified as the occupant of the office she can adjust her in/out status manually on the computing device or implicitly updated by the application if it embodies a more sophisticated user model (i.e. meeting schedule, location, and etc.). If a virtual note has been left by others or the occupant herself, the application displays the notes on the user's computing device.

Object Virtual Notelet **non-human actor**

Action:

```
VN2-1: identify the user's role
VN2-2: activate the virtual context board
VN2-3: display virtual notes and provide the user with note manipulation
function
VN2-4: modify the relations between the virtual note and the physical door
```

Object virtual context board **simple**

Attributes:

Affordances: `virtually adjustable indicator`

Object Virtual Post-It note **simple**

Attributes:

Affordances: virtually `attach/detach/draw or write`

Relations: object-object

```
Location (virtual note, computing device (i.e. PDA))
Location (context board, computing device (i.e. PDA))
```

Input to Virtual Notelet

```
user ID and password • VN2-1
occupant stands at the door • VN2-2
notes attached on the door • VN2-3
user fires "remove" command • VN4
```

Context in Virtual Notelet

```
[user] • VN2-1
[occupant's location] • VN2-2
[virtual note attached on the door] • VN2-3, VN2-4
```

As mentioned, "Input" indicates the input to each action performed by the Virtual Notelet. Reversely, the input is interpreted and presented as context information related to the application actions. Situation 3 describes the occupant's interaction at the door when leaving the office.

Situation 3: Office occupant is approaching to her office door, opening, walking out, and locking the door. She adjusts the in/out status and may leave notes to state further information when she stands at the door.

Role: occupant

```
0. leaving her office
        1. open the door
        2. close the door
        3. adjust the in/out status on the context board attached on the door
        4. check and remove notes
        5. leave a note for extra message
                5.1 write message on the note
                5.2 detach the note from the pile of Post-It notes
                5.3 attach the note on the door
        6. lock the door and leave

Plan 0: do 1-2-3-6 in that order
        if any note attached on the door then do 4.
        if further message is needed then do 5
```

```
Plan 5: do 5.1-5.2-5.3 in that order
```

Object Office occupant **human actor**

Actions:

```
V3-1: walk to the office door
V3-2: open the door, walk out, and close the door
V3-3: check and remove notes from the door
V3-4: adjust in/out status showing on the door
V3-5: lock the door
```

Object Post-It note **simple**

Attributes:

Affordances: `hold/fold/attach/detach/draw or write`

Object context board **simple**

Attributes:

Affordances: `adjustable indicator`

Events:

```
E3-1: occupant is free in the office
E3-2: occupant is busy in the office
```

Relations: object-object

```
Location (Post-It notes, office door)
Location (context board, office door)
```

Relations: action-object

```
patient (V3-2 and V3-5, door)
        - occupant open, close, and lock the door
patient (V3-3, Post-It notes)
        - occupant check and remove the notes
patient (V3-4, context board)
        - occupant adjust her in/out status displayed on the context board
```

Relations: action-event

```
trigger (E3-2, V3-3)
```

When the user stands at the office door, the application must identify the relationship between the user and the office as described in situation 1 and 2. If the user's role is identified as the occupant of the office she can adjust her in/out status manually on the handheld or it can be implicitly updated by the application if it embodies a more sophisticated user model (i.e. meeting schedule, current location, and etc.). If a virtual note has been left by others or the occupant herself, the application displays the notes on the user's handheld device. From the user task analysis listed above, we see that the office occupant's status and notes that attached on the office door interest both a visitor and an office occupant when they stand at the door and influence their subsequent tasks.

Object Virtual Notelet **non-human actor**

Action:

```
VN3-1: identify the user's role
VN3-2: activate the virtual context board
VN3-3: display virtual notes
VN3-4: modify the relations (remove) between the virtual note and the
physical door and provide the user with note manipulation function
```

Object virtual context board **simple**

Attributes:

Affordances: `virtually adjustable indicator`

Object Virtual Post-It note **simple**

Attributes:

Affordances: `virtually attach/detach/draw or write`

Relations: object-object

```
Location (virtual note, computing device (i.e. PDA))
Location (context board, computing device (i.e. PDA))
```

Input to Virtual Notelet

```
user ID and password • VN3-1
occupant stands at the door • VN3-2
```

```
notes attached on the door • VN3-3
user fires "remove" and "create" virtual note command • VN3-4
```

Context in Virtual Notelet

```
[user] • VN3-1
[occupant's location] • VN3-2
[virtual note attached on the door] • VN3-3
[occupant's status in the office] • VN3-4
```

To sum up, the context information supported by the initial prototype of the Virtual Notelet application will include "the role of the user", "the user's location", "office occupant's status", and "virtual note on the office door". The user's location triggers the information presentation about the occupant's status in the office and notes left by the occupant or others virtually attached on the door. The context, occupant's status in the office, determines the visitor's tasks at the office door such as leaves a note or "knocks and walks in".

Conclusions

Most existing context-aware applications were designed in an improvised way. The aim of this paper is to utilize task analysis techniques as tools for thought for designers when they design applications of this kind. It describes the approach of exploiting HTA to identify tasks in existing context-aware applications in different domain in attempt to better understanding of context-awareness. We are interested in user tasks in a scenario. In particular, we focus on the actors, goals, and settings of a scenario. We concentrate on the way users perform tasks to accomplish goals. The point is that task analysis can help us to move from a scenario to a more concrete design. User testing can then be used to observe limitations with the task analysis that will only be apparent when real people actually start to use the application. We also utilize Entity-Relationship Modeling to identify the relationship among actors, objects, and actions when users perform their tasks. The aim is to figure out which part of user tasks can be transformed to applications if the level of context-awareness is increased on the application side. The hybrid task analysis technique is applied to the design scenario of Virtual Notelet in order to identify actors, goals, and actions that include both with and without application support when users perform tasks in the application scenario. Our finding is that HTA and the hybrid approach do not reduce the complexity nature of context-aware applications but provide a blueprint of modeling context information for application uses in diverse domains. It is important to stress that this approach should encourage developers with more knowledge about underlying technologies of context-aware applications to concentrate more on observation of user activities that are performed to achieve goals in specific application domains. The hope is that more novel and meaningful context-aware applications will be discovered and developed.

References

Adams, D. (2002). Programming Jabber, 1st edition. O'Reilly, Sebastopol, CA, USA.

Bjork, S., Falk, J., Hansson, R. and Ljungstrand, P. (2001). Pirates! - Using the Physical World as a Game Board., Interact 2001, IFIP TC.13 Conference on Human-Computer Interaction, Tokyo, Japan, 2001.

Broadbent, J. and Marti, P. (1997). Location Aware Mobile Interactive Guides: usability issues, Fourth International Conference on Hypermedia and Interactivity in Museums (ICHIM97), pp.88-98. Paris, 1997.

Cambridge, AT&T Laboratory (1992). Active Badge on ``Beyond 2000'',
http://www.uk.research.att.com/pub/videos/qsif-200/beyond-qsif-200.mpg, 1992

Cambridge, AT&T Laboratory (1992). The Active Badge System,
http://www.uk.research.att.com/pub/videos/qsif-200/badge-qsif-200.mpg, 1992

Carey, M. S., Stammers, R. B. and Astley, J. A. (1989). In Task Analysis for Human-Computer Interaction (Ed, Diaper, D.) Ellis Horwood, pp. 56-74.

Carroll, J. M. (2000a). Five reasons for scenario-based design, Interacting with Computers, **13,**
pp. 43-60.

Carroll, J. M. (2000b). Making Use Scenario-Based Design of Human-Computer Interactions, The MIT Press, Cambridge, Massachusetts.

Chan, W. (2001). Project Voyager: Building an Internet Presence for People, Places, and Things, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

Chen, G. and Kotz, D. (2001). A Survey of Context-Aware Mobile Computing Research, Dept. of Computer Science, Dartmouth College

Cheng, Y-M. and Johnson, C. (2001). The Reality Gap: Pragmatic Boundaries of Context Awareness, Blandford, A., Vanderdonckt, J. and Gray, P., *IHM-HCI 2001*, pp.412-427. Springer, Lille, 2001.

Davies, N., Cheverst, K., Mitchell, K. and Efrat, A. (2001). Using and Determining Location in a Context-Sensitive Tour Guide, IEEE Computer, **34,** 35-41.

Dennis, B. (2001). On Integrating First Person Shooter Games and Ubiquitous Environments, UbiComp2001, Atlanta, U.S., 2001.

DeVaul, R. W. and Dunn, S. (2001). The Context Aware Cell Phone Project, http://www.media.mit.edu/wearables/mithril/phone.html

Dey, A. K. (2001). Understanding and Using Context, Personal and Ubiquitous Computing, **5,** pp. 4-11

Dix, A., Finlay, J., Abowd, G. and Beale, R. (1998). Human-Computer Interaction, Prentice Hall Europe, 2nd edition.

Falk, J. (2001). The Nexus: Computer Game Interfaces for the Real World, UbiComp2001, Atlanta, U.S., 2001.

Galani, A. and Chalmers, M. (2002). Can You See Me? Exploring Co-Visiting Between Physical and Virtual Visitors, Museums and the Web 2002, Archives & Museum Informatics, Boston, U.S., 2002.

Headon, R. and Curwen, R. (2001). Ubiquitous Game Control, UBICOMP Workshop on Designing Ubiquitous Computing Games, Atlanta, GA, USA, 2001.

Kindberg, T. and Barton, J. (2001). A Web-based nomadic computing system, Computer Networks (Amsterdam, Netherlands), **35,** pp. 443-456.

Lieberman, H. and Selker, T. (2000). Out of Context: Computer systems that adapt to, and learn from, context, IBM SYSTEM JOURNAL, **39,** pp.617-632.

Lijungstrand, P. (2001). Context Awareness and Mobile Phones, Personal and Ubiquitous Computing, **5,** 58-61.

Long, S., Kooper, R., Abowd, G. D. and Atkeson, C. G. (1996). Rapid prototyping of mobile contextaware applications: The cyberguide case study, conference on Human Factors in Computing Systems ,CHI'96, pp.97-107. 1996.

MacColl, I., Brown, B., Benford, S. and Chalmers, M. (2002). Shared Visiting in EQUATOR City, Collaborative Virtual Environments 2002, Bunn, 2002.

Masui, T. and Siio, I. (2001). Real-World Graphical User Interfaces, Thomas, P. and Gellersen, H. W., *HUC 2000*, pp.72-84. Springer-Verlag, Bristol, U.K., 2001

Oppermann, R. and Specht, M. (1999). A nomadic Information System for Adaptive Exhibition Guidance, ICHIM99, International Cultural Heritage Meeting, pp.103-109. Washington, D.C., U.S., 1999.

Oppermann, R. and Specht, M. (2000). A Context-Sensitive Nomadic Exhibition Guide, Thomas, P. and Gellersen, H. W., *HUC2K*, pp.127-142. Spring-Verlag, Bristol, U.K., 2000.

Pering, T. P., C. (2001). Mercantile: Social Interaction Using a Mobile Computing Platform, UbiComp2001, Atlanta, U.S., 2001.

Schigeoka, I. (2002). Instant Messaging in Java, 1st edition, Manning Publications Co.

Schmidt, A. (2000a). Implicit Human Computer Interaction Through Context, Personal Technologies*,* **4,** pp. 191-199.

Schmidt, A., Takaluoma, A. and Mntyjrvi, J. (2000b). Context-Aware Telephony over WAP, pp.225-229. Springer-Verlag, London, Ltd., 2000.

Schneider, J. and Kortuem, G. (2001). How to Host a Pervasive Game - Supporting Face-to-Face Interactions in Live-Action Roleplaying, UbiComp 2001, Atlanta, GA, USA, 2001.

Selker, T. and Burleson, W. (2000). Context-aware design and interaction in computer systems, IBM SYSTEM JOURNAL*,* **39,** pp. 880-891.

Shepherd, A. (1989). In Task Analysis for Human-Computer Interaction(Ed, Daiper, D.) Ellis Horwood, pp. 15-55.

Spasojevic, M. and Kindberg, T. (2001). A Study of an Augmented Museum Experience, HP Laboratories, 19.07.2001

Tuulari, E. (2000). Context aware hand-held devices, MSc Thesis, VTT ELECTRONICS, NETWORKING RESEARCH, TECHNICAL RESEARCH CENTRE OF FINLAND, OULU

Want, R., Hopper, A., Falcao, V. and Gibbons, J. (1992). The Active Badge Location System, ACM Transactions on Information Systems, **10,** pp. 91-102.

Youll, J., Morris, J., Krikorian, R. and Maes, P. (2000). Impulse: Location-based Agent Assistance, Fourth International Conference on Autonomous Agents, Barcelona, Spain, 2000.