A Multimodal Interaction Framework for Pervasive Game Applications

Carsten Magerkurth, Richard Stenzel, Norbert Streitz, Erich Neuhold
Fraunhofer IPSI
“AMBIENTE - Workspaces of the Future”
Dolivostrasse 15
D-64293 Darmstadt, Germany
+49 (0) 6151/869-997

{magerkurth; stenzel; streitz; neuhold}@ipsi.fraunhofer.de

ABSTRACT

In this paper we present STARS, a platform for developing computer augmented board games that integrate mobile devices with an interactive table. The aim of STARS is to augment traditional board games with computing functionality, but without sacrificing the human-centered interaction dynamics of traditional tabletop games. STARS consists of a specialized hardware setup and an interaction framework that dynamically couples mobile and stationary input and output devices with the game table. Depending on a current device configuration, different input and output modalities are available which can either be explicitly utilized by the human players or determined by the STARS platform with regard to a set of mode-specific attributes, available interface services, and demands from a specific game logic.

1. INTRODUCTION

Gaming has been a hot topic in many cultures for a long time with board games such as Chess or Go having a thousand years of successful history. In recent decades, with the advent of personal computer technology, many exiting new possibilities have emerged to create fascinating games with computers. Computer games can apply complex simulations to create a believable, immersive game world. 3D graphics and sound add to an ultra-realistic perceptional experience. Network technology connects dozens of players around the globe to share a single game instance — At the same time, the old-fashioned board game is still going strong with current titles such as *The Settlers* or *Dungeons & Dragons* selling millions of copies in Germany alone [5].

Even though computer technology opens up an endless number of exiting ways to surpass traditional board games, playing computer games is still being perceived as a mostly isolated activity [24]. It stresses the rigidly defined interaction between the human and the computer instead of the unbound human-to-human interaction that makes board gaming a pleasant social activity [11].

Our aim is to support gaming as a social activity between humans and to exploit the beneficial features of computer technology at the same time. To address both of these goals, we have developed a generic platform for realizing computer-augmented board games called STARS (SpielTisch-AnReicherungsSytem, which is a German acronym for Game Table Augmentation System). A key objective for the development of STARS was to ensure that interaction experiences always remain human-centered. We try to achieve this by applying a flexible multimodal interaction framework that integrates different input and output modes where appropriate to allow for a mostly natural interaction. The range of supported game types in STARS includes simple mainstream board games such as Monopoly, but the system is best experienced with more complex tabletop strategy or roleplaying games. An introduction to these can be found here [7].

Playing pervasive board game applications is quite similar to playing traditional board games in terms of the enjoyable experience with friends. However, in augmented games there is an additional instance present at the game table: The computer controlled game logic. The game logic can enforce game rules, compute complex simulations in the game world, keep track of players inventories, and perform other mundane tasks. Depending on the nature of the game, it might also take a more active role and communicate with the players over different modalities. For instance, it might tell one player over his earphone about an important game event and leave it to the player, if he tells the others about it or not.

Augmented games can also change the ways players communicate with each other. In addition to the traditional face-to-face interactions, new modes of e.g. clandestine communication are provided. For games with both cooperative and competitive elements, exiting new situations can emerge, when alliances and conspiracies are forged in secrecy, while the open diplomacy at the game table speaks a different language.

Through the use of mobile devices, game-related activities not requiring the presence of other players are also supported. For instance, parts of the administration of a player’s character in a roleplaying game can be performed asynchronously with his PDA (see figure 2) even when he is not present.

2. THE STARS SETUP

2.1 Context of Development

The STARS platform was developed in the tradition of our earlier work on i-LAND, the ubiquitous computing environment for creativity and innovation [18], consisting of several Roomware® components, e.g., interactive walls and tables and mobile devices such as PDAs. In addition, a special software infrastructure (BEACH) was developed that supports cooperative work and synchronous sharing of hypermedia information [20]. STARS utilizes now two of the second generation of our Roomware® components [19], in particular, the InteracTable® and the DynaWall®, but provides a new software infrastructure. Our work follows the Ambient Intelligence (AmI) paradigm in that we provide a user friendly, adaptive computing environment with unobtrusive access to its computing functionality.

2.2 Integrated Device Components

Players interact with each other and jointly modify the state of the game board on the table surface. In addition to the table, other mobile and stationary device components can be dynamically integrated into a STARS session, i.e. they are utilized if available and substituted if not. In the following sections we briefly describe each supported device and its purpose within STARS. Figure 1 illustrates the hardware setup.

2.2.1 Table

The game table provides the game board which is displayed on its surface. It provides a tangible interface with ordinary playing pieces that are tracked by an overhead camera. Currently, we use

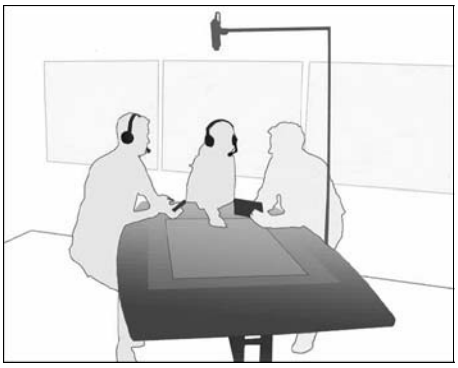

Figure 1. The STARS setup.

the InteracTable, which is a Roomware component that features a touch sensitive surface used for gestures and menus.

In contrast to other augmented tabletop applications [e.g. 17, 18, 11] the STARS setup does not use top projection from a projector mounted above the table in conjunction with sensing technology embedded in or under the table surface. Instead, we use the table's embedded plasma display and the camera mounted above the table for additional sensing. This approach has the advantage of robustly providing a high-quality image on the table surface, no matter how the lightning conditions are. Furthermore, unlike top-projection, the game board is not displayed on top of the playing pieces, when it should be beneath them. By providing both a camera and a touch display, multiple types of interaction can be realized in contrast to mere object placement detection.

2.2.2 Wall Display

The wall display is used for showing game relevant public information that each player can view at any time. It consists of the DynaWall®, which is a Roomware component with a rear-projected, touch sensitive interaction space for computer-supported cooperative work [19]. On the wall display, arbitrary media such as videos or animations can be played which are triggered either by a player or by the logic of a STARS software application. Current STARS games use the wall for showing players' scores (Monopoly) or to display a map of a dungeon that the players explore (KnightMage).

2.2.3 Personal Digital Assistants (PDAs)

Each player can bring a PDA to connect with the other STARS components. Currently we use Compaq IPAQ 3950 and Toshiba e740 Pocket-PCs with integrated 802.11b cards. The usage of PDAs is twofold. First, they are used for taking private notes and for administering private data. For example, one of our sample

Figure 2. PDAs administrate private data.

applications is a fantasy role-playing game where the PDA provides access to a player's inventory of carried objects and his character attributes (health, strength, etc). This is shown in figure 2. The second use for the PDAs are private communication channels that allow both a STARS game logic and the players to send private messages to other players (see figure 3).

Figure 3. Hidden Communication through PDAs

2.2.4 Audio Devices

STARS provides both a speech generation and a speech recognition module with multiple input and output channels based on the Microsoft Speech API [12]. Depending on the type of game, players wear earphones to receive private messages either generated by the STARS game logic or by other players. A public loudspeaker is used to emit ambient audio samples or music to atmospherically underline the action at the game table. Headsets are available for verbal commands to the game logic. Speech recognition is currently based on an XML command list that allows natural language variations .

3. INTERACTING WITH STARS

The key idea behind the development of the system was to enrich traditional board games with pervasive computer technology and at the same time take care that the social dynamics of a board game session remain intact. By integrating too much functionality into the digital parts of the system, the players might easily experience interacting more with a computer game than with each other. For instance, in many board games, dice are used to generate random numbers that add variability to the game progression. Although random number generation is a trivial task for a computer program, STARS never simulates rolling any dice, because for most players the act of rolling dice is a highly social activity involving skillful rolling techniques which are permanently supervised by the other players. Sometimes, very interesting group dynamics can emerge from different perspectives about what "rolling correctly" means. To preserve such unbound interaction situations, care must be taken not to rigidly enforce certain interaction patterns, but to provide a flexible approach that adapts to the dynamics of a given game situation.

3.1 The Pool of Modes

During a game session, interaction between players and/ or the computer is manifold. Sometimes, short bursts of information must be communicated clandestinely from one player to another using e.g. PDAs. Sometimes, complex spatial formations of playing pieces must be collaboratively developed on the game board. And sometimes, there is human discussion that does not involve the game media at all.

For different interaction needs, different media and modalities are more suitable than others. In STARS, we attempt to let the computer assist in finding optimal compositions of modes. We are aware of the dangers associated with applying a "mechanistic plug-and-play manner to create the ultimate multimodal translation device" that Oviatt challenges in the myths of multimodal interaction [13]. However, by specifically addressing the uniqueness of different modes and modalities, we believe that a dynamic, reconfigurable, and context-aware composition of modes is beneficial and feasible for the relatively narrow task domain of pervasive game applications.

To interact with the STARS platform, the device components described in section 2 provide a set of input and output modes shown in table 1 and table 2. Each device can provide 0..n input and 0..n output modes. Currently, we do not use devices with multiple output modes, so that table 2 effectively lists the devices used for output. Additional devices such as mobile phones might be used in the future and provide multiple output modes, e.g. vibration and showing information on the display.

Each mode is further characterized by a set of capability attributes relevant for the *Interaction Manager*, the STARS software component that is responsible for the composition of modes. These capability attributes describe the suitabilities of a mode for different interaction requirements. Input and Output modes, naturally, do not share an identical set of relevant attributes.

Some of the modes are tightly connected to a single interface task and are unsuitable for others. For instance, the positions of the playing pieces are altered by moving them on the board in an as natural way as in a traditional board game. However, moving the pawns can still be part of an interface composition, e.g. the movement can be combined with a verbally uttered command such as "take", when the pawn is placed on another game object like e.g. a bag of gold. This is in the tradition of Bolt's Put-That-There [4] paradigm, where the "that" is already implicitly included in picking up the pawn.

The attributes that characterize the modes in table 1 and table 2 are defined as follows: "Private" refers to the capability of obtaining input from one person without others perceiving the input made, or if output can be directed towards single individuals without others perceiving this output. "Spatial" refers to the positions of game objects on the board, e.g. if an interface component can be placed "near" a pawn. "Generic" refers to being generically applicable in contrast to being dedicated to certain tasks. Being not generic implies humans to expect a certain affordance, so that they would be disturbed if the mode was used in an unexpected way. "Complex" refers to different degrees of input complexity, e.g. choosing from two alternatives (simple) to multiple parameters (more complex). "Simultaneous" means the ability to use one mode without preventing the usage of another. "Audible" and "Graphical" refer to the capability of rendering images or playing sounds, respectively. "Textual" finally refers to the capability of communicating textual information.

Table 1. Capability Attributes of Input Modes.

Mode \ Attr.	Private	Spatial	Generic	Complex	Simultaneous
1. Table Pawns	-	+	-	-	+
2. Speech	-	-	+	o	+
3. Table Gestures	-	+	+	-	+, except **4.**
4. Table WIMP	-	+	+	+	+, except **3.**
5. PDA	+	-	+	+	+

Table 2. Capability Attributes of Output Modes.

'Mode' \ Attr.	Private	Spatial	Audible	Graphical	Textual
a. Table Display	o	+	-	+	+
b. Wall Display	-	-	-	+	+
c. PDA Display	+	-	-	o	+
d. Loudspeaker Audio	-	-	+	-	o
e. Earphone Audio	+	-	+	-	o

Each mode from table 1 and table 2 is characterized as being well capable (+), moderately capable (o), or incapable (-) in terms of the given attributes. Whenever applicable, these rankings were derived from Modality Theory [1, 2]. Bernsen's meta-analysis showed that several modality properties were uniformly found throughout the scientific body of literature regarding multimodal interaction and can thus be claimed as being universal modality properties. For instance, modality property 1 states that linguistic input and output is "unsuited for specifying detailed information on spatial manipulation" [2] which is reflected in the tables. Also, static graphic modes are suitable for representing large amounts of (complex) information, whereas, e.g., synthetic speech output increases cognitive processing load, as found for two other modality properties. Some of the presented attributes are also based on personal experience, common sense, or relate to obvious properties of a mode, e.g. the ability to display graphics or create audio.

The "o" (moderate capability) for private table output refers to the ability of STARS to rotate table interface components to the position of the addressee, so that she can perceive this visual information better than the other players, even though they are not completely hindered to perceive the information as well. They are, however, aware that the information is probably irrelevant for them. The interface is described in more detail in [10].

The "o" for loudspeaker and earphones' textual output refers not only to the cognitive overhead, but also to the time it takes to synthesize speech. Longer textual information is therefore not practical for audio output.

3.2 Composition of Modes

The characterization of modes in the preceding section helps deciding which modes are candidates for user interaction, when certain requirements apply for the desired interaction. The modes' capability attributes are not meant to directly generate interfaces, but to help evaluating the interface services available for each mode in the context of a concrete desired interaction. The STARS software component that composes the actual interfaces is the Interaction Manager.

4. THE INTERACTION MANAGER

Regarding user interaction, the Interaction Manager is the predominant software component in STARS. It maps *Interaction Requests* from the higher level of the game logic to the lower level of the interface services available for a mode. By doing so, it takes into account hints within the Interaction Requests about certain characteristics a mode should have.

It also provides a flexible, high-level programming interface that allows STARS game applications to formulate Interaction Requests that are easy and mode-independent to use and at the same time very flexible by allowing to specify different requirements for each mode involved.

4.1 Input and Output

The Interaction Manager always differentiates between input and output modes, which is advantageous when dealing with devices that are not capable of jointly providing input and output modes. Such a distinction could be challenged, when e.g. in a traditional graphical user interface (GUI) input and output are very tightly coupled. In our case, however, it provides the flexibility needed for cross-device interaction. For instance, a simple pop-up menu could currently be placed on the table surface where the user would select one of the menu alternatives by just tapping on it. However, the touch functionality of the table might be unavailable for input, e.g. because it was locked by a different Interaction Request or simply because a specific table hardware might not provide touch sensitivity. The Interaction Manager could then decide to

(a) display the menu on the table anyway, but await the corresponding input via speech. This would cause the menu to be drawn differently to indicate that input is awaited via speech.

The Interaction Manager might also decide to

(b) show the menu on the PDA and also await input there.

If the number of alternatives to choose from was small enough, the Interaction Manager might even provide both output and input via auditory channels.

In the current case, the Interaction Manager would probably prefer (b), if a demand for spatial placement was emphasized in the Interaction Request, whereas a need for private interaction would favor (a). Also, (b) profits by both input and output modalities being closely coupled on the same device, which is also taken into account by the Interaction Manager. A detailed description of weighing modes against each other is given further below.

4.2 Complex Interactions

Only very basic interaction situations in STARS involve exactly one input and one output mode. Frequently, interactions are more complex in that they allow input on alternative modalities, jointly utilize multiple input modes, or consist of multiple, parallel interactions.

4.2.1 Input on Alternative Modes

Interactions can involve multiple Input modes that are suited for receiving the desired input and eventually terminate an interaction. If available, all of these alternative input modes are communicated to the user in the order of their suitability as determined by the Interaction Manager. For instance, the popup menu described above can render information for the user that alternatives can either be chosen by tapping or uttering the corresponding command. Please see the left and center parts of figure 4, where a

Figure 4. Different realizations of one Interaction Request.

the user is prompted to chose a weapon either by tapping on it (left) or additionally via the second alternative mode of speaking (center). If the table surface did not include touch sensitivity, PDA input might have become an option, although the menu output probably would have been rendered directly on the PDA display, and not on the table surface (figure 4, right).

4.2.2 Multimodal Input

Multimodal interaction, in which multiple input modes are required, is supported through grouped Interaction Requests that work similar to single requests with the exception that mode suitabilities take interferences between modalities into account and are thus computed a bit differently. For instance, input involving different human effectors is generally preferred over single effectors. Eventually, the application can provide additional hints regarding timely successions and terminating conditions.

A typical example of an interaction with multiple input modes is the combination of pawn placements and spoken commands: In STARS, game objects on the game board can either be physical such as pawns, or virtual, i.e. being displayed on the table surface. Frequently, players move their pawns over a virtual game object to perform an operation on the object with the pawn. When more than one operation is feasible, the operation must be conveyed to STARS, e.g. by speaking the appropriate command. For instance, a player might place her pawn on a chest and utter "open" or "find trap" (which can make a considerable difference!).

A example for a multimodal interaction involving spatial information and a high demand for privacy is shown in figure 5. A player places his pawn in front of a locked door (figure 5, left) on the game board and opens it with a silver key he takes from his inventory of carried objects (figure 5, middle, right). While the changne in the pawn's location is obvious to the other players, the exact action in front of the door is concealed behind the incasement of the PDA..

A high privacy level would not be required, if all the players were

Figure 5. Interaction with Pawn and PDA.

standing at the door. In this case, the Interaction Manager would probably abort the PDA in favor of displaying a pop-up menu next to the door on the table surface for everyone to see.

4.2.3 Parallel Interactions

Human-centered interaction design must provide a natural way to interact with a computing system. This implies that users should be able to express themselves freely without having to follow application-defined interaction paths. For instance, a player in an augmented board game should freely decide to either move a playing piece, tap on other objects that are on the game board, or jot something down on her private PDA workspace. Such parallel interactions are realized in STARS through multiple Interaction Requests being scheduled in parallel. In fact, the system's standard behavior is to process as many parallel Interaction Requests as the involved modes permit, while exclusive and blocking interaction (such as the system modal shutdown dialog) have to be hinted separately in an Interaction Request.

Please note that limitations at the device hardware level do not necessarily have to be reflected within the Interaction Manager. The touch display of our game table is currently only able to track one finger at a time. Nevertheless, multiple Interaction Requests working with the table surface can be active simultaneously, because resource conflicts are resolved already at the device layer. This way, e.g., multiple menus can be used on the game board at the same time.

4.2.4 Application-provided Interactions

Even though it is highly desirable to let an application formulate Interaction Requests at a very abstract level and have the framework decide how to generate an appropriate interface given device and modality constraints, sometimes this just does not work. As long as interfaces remain relatively primitive and the related task domains are relatively narrow, an algorithmic approach is feasible. However, when highly specialized interfaces are required that are not intended for adaptation across modes, STARS applications can implement interface services themselves and register them at the Interaction Manager.

4.3 Finding Modes

The Interaction Manager uses a rule-based system to determine how to map an Interaction Request to one or more input or output modes. It regards the hints an application provides for the generation of the interface and compares it with every single input and output mode available as well as for combinations of modes and certain constraints for more complex Interaction Requests. A suitability index is calculated for every combination of modes with the highest index being the most suitable. Indices are calculated by weighing every hint about the characteristics of the desired interaction with the attributes of the regarded mode. The sums of these weighings make up the isolated suitabilities of single modes to serve for input or output. Isolated suitabilities can thus be expressed as:

$$I_m = \text{argmax}_m\{ \Sigma_{(i = 0..\text{max Attr})} \text{Hint}_{m_i} * \text{Attr}_{m_i} \}$$

These single suitabilities are further moderated by rules about their combination such as the aforementioned preference of different effectors. The result of each of these calculations is a suitability index from which the highest index is finally chosen.

For simplicity, table 1 and table 2 only include rough measures of the included attributes. For the calculations within the Interaction Manager, every attribute is represented as a signed byte value. The exact values used are a matter of experimentation and can, in fact, be altered while the system is running, so that dynamic reconfiguration does not only relate to the devices within a STARS session, but also to modality composition. The values of the hints within the Interaction Requests are within the same range as the modality characteristics.

4.4 Controlling the Interaction Manager

The programming interface to the Interaction Manager is designed to allow a high level formulation of Interaction Requests that keep the game application independent from the current device configuration. Even though it is possible to hint for specific modality compositions and interface services, the high abstraction level allows for a more generic and content-centered application development. A second key feature of the Interaction Manager is the ability to dynamically alter Interaction Requests during program execution, i.e. Interaction Requests are not created using hard-coded API-calls, but reside in XML-based resources. This

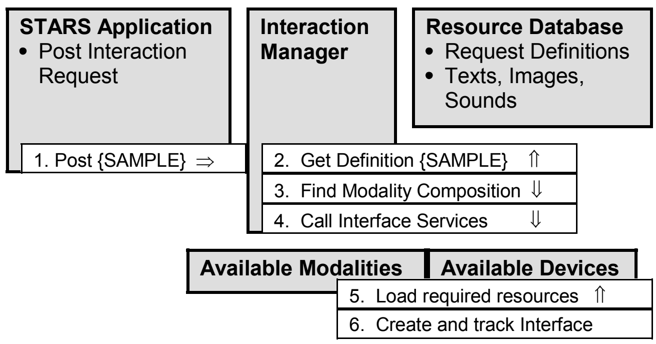

Figure 6. Creating an interface.

allows for tweaking Interaction Requests while a game is in progress.

4.4.1 Flow of the Interface Creation

As shown in figure 6, a STARS application simply provides an identifier for the desired interaction. The Interaction Manager then retrieves a more detailed description from the resource database (interface type, mode hints, parameters etc), finds an appropriate modality composition and - after internal scheduling and synchronization with other Interaction Requests - calls related Interface Services implemented at a device layer. These services then retrieve the required data to build an interface and eventually create the interface. After a terminating condition has been met, the interaction result is transmitted back to the application as defined in the resource database.

4.4.2 Interface Services

The workhorses regarding the creation, presentation, and tracking of interfaces are the Interface Services that are written for every device integrated in the STARS setup. The functionality of Interface Services available for each device varies with the characteristics of the device's modes. Due to the somewhat limited domain of board gaming in terms of interaction variations, the number of services currently implemented is yet manageable. Following the Interaction Manager's distinction there are Interface Services for output and for input/ tracking of user actions. Output services rank from simple DISPLAY(resourceID) to several more complex MENU() variations. What a service such as DISPLAY() actually does, depends on the parameters in the resource database and mostly on the specific device characteristics, e.g. an earphone's only way to display a resource is to read any text/ audio sample included or otherwise fail. Input services include the tracking of pawns, spoken commands and various pen/ finger related services. In addition to the input and output services, several higher level services exist that wrap single basic services for handling dependencies, e.g. when the same device is used for joint input and output.

5. STARS SOFTWARE ARCHITECTURE

While the Interaction Manager is the single most important instance related to turning Interaction Requests into actual interfaces, other components are also involved in adding to the experience of playing pervasive game applications. In this section, the general architecture of STARS is presented. The system's logical components are distributed among four layers of decreasing abstraction. While a discussion of each component is beyond the scope of this paper, we will only briefly describe the architecture and get into more detail on those components that are relevant for user interaction or otherwise closely related to the Interaction Manager.

The first and most abstract layer in STARS is the Application Layer. It implements functionality specific to every game application running on the STARS platform. Since STARS is not a game itself, but a platform to realize games on, the entire content of each game application must be defined here. This includes first and foremost the game logic, i.e. the set of rules necessary to define a game, as well as the artwork and interaction resources in the Resource Database. Also, any additional functionality not covered by a deeper STARS layer must be implemented here, e.g. sophisticated dialogue definitions that go beyond the capabilities of the Interaction Manager.

On the Game Engine Layer, certain sets of tools common for most board games are located to allow for a higher abstraction on the Application Layer. For instance, the game board is represented here along with basic scripting support and object definition templates. Also, building upon the definition of users present in the succeeding Core Engine layer, the participating players are administrated here with common properties found in most tabletop games such as score counters or an inventory of carried game objects.

The Core Engine Layer includes core functionality regarding the STARS hardware setup that is not restricted to game applications,

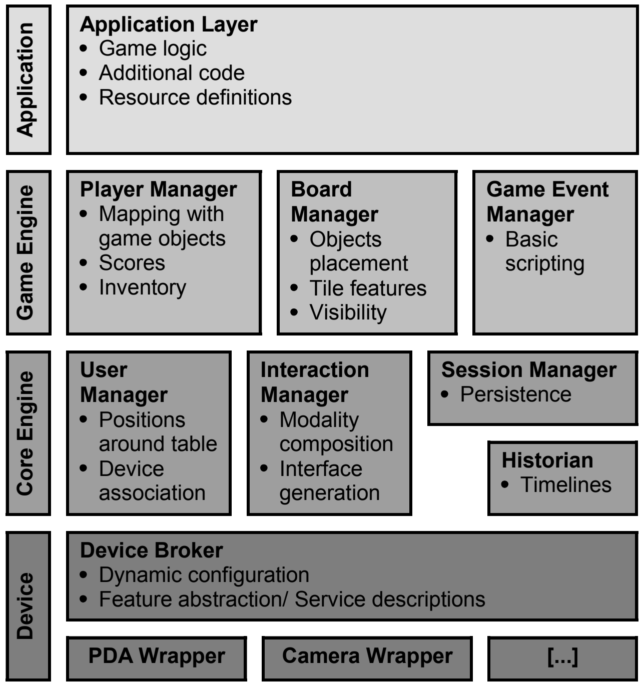

Figure 7. The STARS software architecture.

but has a broader scope. Apart from session management and logging support, the Interaction Manager is the predominant component of this layer. It is indirectly related to the User Manager that provides the associations between users and devices. The User Manager is queried when an application hints a private interaction for a specific user and returns the devices associated with the user. Such associations are currently explicitly established, i.e. a user manually logs in to a device. In the future, we plan to explore implicit authentication mechanisms that work e.g. by RF-ID tagging. In our Roomware environment, we already provide an infrastructure for implicit authentication, please see [14].

On the final Device Layer, the available STARS hardware components are administered. This includes dynamic integration, service enumeration and feature abstraction. For each device, a wrapper exists that implements the available Interface Services that are called by the Interaction Manager. Devices register at the Device Broker which is responsible for administering communication between a device and STARS, or other applications in our ubiquitous computing environment, respectively.

Figure 7 gives an overview of the software architecture.

5.1 System Implementation

We have built the STARS platform almost entirely[1] on the .NET architecture to allow a flexible distribution of components among different computers.

Although there are real-time multimodal interaction frameworks that require only moderate computing resources [e.g. 9], computational power is still regarded a critical issue with multimodal interaction, especially for modalities requiring

[1]PDA implementations are currently based on MFC/ GapiDraw. This will change with the advent of the PocketPC 2003 release.

complex calculations such as speech recognition or visual detection.

Currently, the Camera Device Wrapper is being executed separately on a Pentium 1.6 Ghz machine that runs several image manipulation filters on the camera vision to detect both the playing pieces and the human arms reaching over the table surface. By running the Camera Device Wrapper alone we can thus optimize the frame rate of the image detection (currently about 10 frames/sec). The Device Broker process also runs separately on a backend server, where it fulfills additional device coupling tasks not directly related to STARS, but to other applications in our ubiquitous computing environment. The remaining components run jointly on the Pentium 800 Mhz machine integrated in the game table. Due to the common turn-taking interaction styles both for tabletop pawn movement and speech recognition, the number of session participants does not influence computing demands significantly, and therefore the system remains quite stable.

6. EXPERIENCES WITH THE PLATFORM

6.1 Realized Games

While the STARS platform itself has already reached a quite mature state, we are still in the process of creating games that show off all the additional beneficial features that computer-augmented boards offer over traditional tabletop games. Next to the flexible multimodal interface other important benefits include complex game rules, dynamic game boards, persistency, true private communication, or taking into account the player's viewing angle on the game board. While we have not yet developed a single game to include all of these features, both of our currently realized games demonstrate a set of interesting augmentation aspects. Several other titles are already in the making.

6.1.1 STARS Monopoly

STARS Monopoly is an adapted version of the famous board game. As a twist to the original game, the wall display is used for permanently conveying the financial development of all players as a public information. This fosters private communication between players both to support and to attack struggling players. Because the game money is implemented in the virtual domain, several statistical functions are available for the players to administer their possessions. The numerous textual information on the game board are automatically rotated to the viewing angle of the current player, which makes them significantly easier to read than on a traditional Monopoly game board. As with all games on the STARS platform, a Monopoly session is persistent and can be interrupted and continued at any time.

6.1.2 KnightMage

In the role-playing game KnightMage (see figures 2-5), the players explore and ransack a deep dungeon filled with horrifying monsters. While the players progress further into the dungeon, they solve riddles and find treasures or weapons to help them against the dungeon's inhabitants. KnightMage is a showcase for the computer's capability to provide a dynamically changing game board that is much larger than the table surface. It also uses a so-called fog of war that blanks out unexplored areas of the game board. What makes KnightMage especially interesting is the mixture of cooperative behavior when fighting monsters and the competitive interests when searching for treasures that each involve very different interaction modes. In KnightMage, certain game events are privately communicated from the game logic to a player, e.g. a player's character might hear or find something and decide on her own, if she lets the others know or not.

6.2 User Experiences

The most important question regarding the usage of STARS is if it does work as a board game replacement that adds beneficial computer-augmented features, but does not destroy the atmosphere of playing board games with friends. From our own observations, people play STARS games like they play board games with primarily face-to-face interaction between the players. Also, the extra features, especially the private communication means over audio or PDA modes, are highly appreciated.

Developing games for STARS has proven to be comparatively easy. It is indeed significantly faster to formulate Interaction Requests in a device and mode independent manner than having to query each device and its services directly. But what has turned out to be even more convenient is the ability to store Interaction Requests and assets in resource databases, so that a program's behavior and look can be tuned during runtime without touching any code. This way, even less experienced students can work freely on a game application. In the future, it would be great to store also parts of the game logic and rules in resources to allow for even greater flexibility.

7. DISCUSSION AND OUTLOOK

We have presented STARS, a hardware and software platform to realize pervasive game applications on an interactive table and additional devices. One of its key features is the notion of different, dynamically composable output and input modes that allow for novel and interesting interaction situations. STARS game applications can describe the desired interactions with the players at a very abstract level, so that new devices and modes can be included dynamically at a later time.

For the future, we will explore how the integration of additional devices affects the use of the system. Ultimately, it would be very interesting to augment the scope of STARS to non-gaming applications. Currently, the use of playing pieces, the tailored set of user interface services, and the strong focus on game board interaction narrow the application domains down to board gaming. However, by implementing new software layers in addition to the Game Engine Layer and integrating new devices and device services, the scope of STARS could also be augmented to include other CSCW domains such as engineering tasks, chemistry, or shared architectural design.

8. RELATED WORK

Similar to STARS, [11] are also developing a hybrid board game called False Prophets, in which players jointly explore a landscape on the game board utilizing multiple devices such as PDAs. As with STARS, the aim of the False Prophets project is to unify the strengths of computer games and board games. In contrast to STARS, False prophets uses a custom-made infrared sensor interface with the game board being top-projected on the table surface. Also, False Prophets is not a platform for developing multiple different games, but is currently limited to an exploration setting.

Björk et al. [3] presented a hybrid game system called Pirates! that does not utilize a dedicated game board, but integrates the entire world around us with players moving in the physical domain and experiencing location dependent games on mobile computers. Thereby, Pirates! follows a very interesting approach to integrate virtual and physical components in game applications.

Multimodal table interfaces with a strong emphasis on tangible interaction have been developed by Ishii et al [8, 15]. *Sensetable* is a generic platform for interacting with physical objects on a tabletop surface [15]. Of special merit is Sensetable's capability of altering the properties of the objects on the table surface dynamically via different kinds of switches and knobs. Naturally, objects are required to include an electronic circuit, whereas STARS can utilize arbitrary objects. A similar approach is also found in [16]. Here, digital objects can even pass the boundaries of physical objects seamlessly.

[8] have also developed a hybrid gaming application called PingPongPlus that augments a traditional ping pong game table with several output modalities such as sound and graphical effects projected at the table surface. Even though no new input modalities are introduced, PingPongPlus is highly entertaining to watch and listen to.

[17] present an interesting interface for dealing with photographs on top of a *DiamondTouch* multi-user interactive table. Even though their focus is not on games, they share our vision of supporting recreational face-to-face interaction with unobstrusive and natural interfaces to the digital world.

9. ACKNOWLEDGEMENTS
We would like to thank our colleagues Sascha Nau, Peter Tandler, Daniel Pape, and Alexander R. Krug for their valuable contributions and insightful comments on our work. This paper was supported by a grant from the Ladenburger Kolleg "Living in a Smart Environment" of the Gottlieb Daimler- and Karl Benz-foundation.

10. REFERENCES
[1] Bernsen, N.O.: Defining a Taxonomy of Output Modalities from an HCI Perspective. Computer Standards and Interfaces, Special Double Issue, 18, 6-7, 1997, 537-553.

[2] Bernsen, N.O., Dybkjaer, L.: A Theory of Speech in Multimodal Systems. In: Dalsgaard, P., Lee, C.-H., Heisterkamp, P., and Cole, R. (Eds.): Proceedings of the ESCA Tutorial and Research Workshop on Interactive Dialogue in Multimodal Systems, pp. 105-108, Irsee, Germany, June 1999.

[3] Bjork, S, Falk, J., Hansson, R., Ljungstrand, P.: Pirates! Using the Physical World as a Game Board. In: Proceedings of Interact 2001, Tokyo, Japan.

[4] Bolt, R. A.: "Put-That-There: Voice and Gesture at the Graphics Interface," Computer Graphics 14 No. 3, 262-270.

[5] Costikyan, G.: Don't be a Vidiot. What Computer Game Designers Can Learn from Non-electronic Games. In: Proceedings of the Game Developers Conference 1999. San Francisco: Miller Freeman, 115-139.

[6] Elting, C., Michelitsch, G.: A Multimodal Presentation Planner for a Home Entertainment Environment. Proceedings of PUI'01.

[7] Introduction to Role Playing Games. www.geocities.com/daveshwz/whatis.html.

[8] Ishii, H., Wisneski, C., Orbanes, J., Chun, B., Paradiso, J.: PingPongPlus: Design of an Athletic-Tangible Interface for Computer-Supported Cooperative Play. In: Proceedings of CHI'99, 394-401.

[9] Krahnstoever, N., Kettebekov, S., Yeasin, M., Sharma,R.: A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays. Proceedings of ICMI'02, 161-166.

[10] Magerkurth, C., Stenzel, R.: Computer-Supported Cooperative Play - The Future of The Game Table. Accepted for Mensch&Computer'03.

[11] Mandryk, R.L., Maranan, D.S., Inkpen, K.M.: False Prophets: Exploring Hybrid Board/Video Games. In: Extended Abstracts of CHI'02, 640-641.

[12] Microsoft Speech Homepage: www.speech.microsoft.com

[13] Oviatt, S.: Ten Myths of Multimodal Interaction. Communications of the ACM, November 99, 42, 74-81.

[14] Passage4Windows website. http://www.ipsi.fhg.de/ambiente/p4w_e

[15] Patten, J., Ishii, H, Hines, J., Pangaro, G.: Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. Proceedings of CHI'01, 253-260.

[16] Rekimoto, J., Masanori, S.: Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. Proceedings of CHI'99, 278-285.

[17] Shen, C., Lesh, N.B., Vernier, F., Forlines, C., Frost, J.: Sharing and Building Digital Group Histories. Proceedings of CSCW'02, 324-333.

[18] Streitz, N.A, Geißler, J., Holmer, T., Konomi, S. Müller-Tomfelde, C., Reischl, W., Rexroth, P., Tandler, P., Steinmetz, R.: i-LAND: An interactive Landscape for Creativity and Innovation. Proccedings of CHI'99, 120-127.

[19] Streitz, N.A., Tandler, P., Müller-Tomfelde, C., Konomi, S. Roomware: Towards the Next Generation of Human-Computer Interaction based on an Integrated Design of Real and Virtual Worlds. In: J. A. Carroll (Ed.): Human-Computer Interaction in the New Millennium, Addison Wesley, 553-578, 2001.

[20] Tandler, P. Software Infrastructure for a Ubiquitous-Computing Environment Supporting Collaboration with Multiple Single- and Multi-User Devices. Proceedings of UbiComp'01. Lecture Notes in Computer Science, Springer, Heidelberg, 96-115, 2001.

[21] Ullmer, B., Ishii, H.: The metaDESK: Models and Prototypes for Tangible User Interfaces. Proceedings of UIST'97, 223-232.