University of Crete
Computer Science Department

DEVELOPMENT OF A SMART OFFICE APPLICATION TO SUPPORT MULTI-USER COLLABORATION USING AMBIENT INTELLIGENCE TECHNOLOGIES

By

PANAGIOTIS TOURLAKIS

MASTER'S THESIS

APRIL 2010

To my beloved father Konstantinos,

who passed away during the writing of this thesis.

I made it dad, this is for you.

To my dear mother Anastasia and my brother Dimitris.

Abstract

Ambient Intelligence (AmI) aims to offer people the ability to interact with computer-augmented environments in a natural way, such as by using vocal commands or hand gestures, in order to accomplish various everyday life tasks. Most people spend a big part of their lives in their working environments and, as shown by recent technology trends, they tend to use more naturally operated equipment, such as touch screens and multi-touch surfaces, to simplify their work. However, when it comes to collaboration, most available tabletop interaction solutions are not suitable for large groups of users, since each user needs a personal workspace, while the cost and complexity of commercial interactive surfaces increase along with their size. This thesis proposes a tabletop collaboration system, called SMURF Table, that employs AmI technologies to provide an interactive surface for meetings among multi-user groups in an office environment. The software system consists of two applications, the SMURF Table application and the SMURF Document Manager.

The SMURF Table application creates an interactive surface that is projected on a large desk, providing each user with a personal workspace along with an area for public use. Users are recognized by identification cards that allow the table to automatically load their documents, while each user is provided with a special pen and a personal keyboard for interaction. On SMURF Table documents are automatically rotated to fit each viewer's perspective, while information sharing is achieved by creating document copies that are updated in real time. The application communicates with sensors in the AmI office environment in order to identify users and allow them to interact with the virtual surface.

The SMURF Document Manager is a tool that allows users to connect to the SMURF Table server and manage their documents remotely over an Internet connection. Users have the ability to download documents from the SMURF Table to their personal computers, as well as upload new documents in order to use them in meetings.

SMURF Table has been specifically developed in order to provide users with an efficient solution for group meetings, by supporting collaboration through an easy to use interface that requires no unnecessary initialization or configuration from the users. AmI technologies are used to allow users to associate their documents with personal cards rather than carrying laptops or storage devices, as well as to support multiple simultaneous user interactions. The results of an evaluation session conducted in the context of this thesis confirm the usability of the developed system, as well as its efficiency as a support tool for collaborative meetings.

Περίληψη

Η Διάχυτη Νοημοσύνη στοχεύει στο να προσφέρει στους ανθρώπους τη δυνατότητα να αλληλεπιδρούν με περιβάλλοντα ελεγχόμενα από υπολογιστές με φυσικό τρόπο, όπως με φωνητικές εντολές ή χειρονομίες, ώστε να τους βοηθήσει στο να φέρουν εις πέρας εργασίες της καθημερινής ζωής. Οι περισσότεροι άνθρωποι περνούν ένα μεγάλο κομμάτι της ζωής τους στο περιβάλλον εργασίας τους και, όπως διαφαίνεται από τις τρέχουσες τεχνολογικές τάσεις, τείνουν να χρησιμοποιούν ολοένα και περισσότερο εξοπλισμό που υποστηρίζει φυσικότερους τρόπους χειρισμού, όπως για παράδειγμα οθόνες αφής και επιφάνειες πολλαπλής αφής, ώστε να απλοποιήσουν τη δουλειά τους. Ωστόσο, όσον αφορά τη συνεργασία, οι περισσότερες διαθέσιμες διεπαφές επιφάνειας τραπεζιού δεν είναι κατάλληλες για μεγάλες ομάδες χρηστών, καθώς αφενός κάθε χρήστης χρειάζεται να έχει μια προσωπική περιοχή εργασίας, και αφετέρου το κόστος και η πολυπλοκότητα των εμπορικών αλληλεπιδραστικών επιφανειών αυξάνονται ανάλογα με το μέγεθός τους. Αυτή η διατριβή προτείνει ένα σύστημα συνεργασίας σε επιφάνεια τραπεζιού, που ονομάζεται SMURF Table, το οποίο παρέχει μια αλληλεπιδραστική επιφάνεια για συνεδριάσεις μεταξύ ομάδων χρηστών χρησιμοποιώντας τεχνολογίες Διάχυτης Νοημοσύνης σε περιβάλλον γραφείου.

Το σύστημα αποτελείται από δύο εφαρμογές, την εφαρμογή SMURF Table και το SMURF Document Manager.

Η εφαρμογή SMURF Table δημιουργεί μια αλληλεπιδραστική επιφάνεια η οποία προβάλλεται πάνω σε ένα μεγάλο γραφείο, παρέχοντας σε κάθε χρήστη μία προσωπική περιοχή εργασίας καθώς και ένα «κοινόχρηστο χώρο» για δημόσια χρήση. Οι χρήστες αναγνωρίζονται από κάρτες ταυτοποίησης που επιτρέπουν στο τραπέζι να φορτώνει αυτόματα τα έγγραφά τους, ενώ σε κάθε χρήστη παρέχεται ένας ειδικός μαρκαδόρος και ένα ατομικό πληκτρολόγιο τα οποία χρησιμοποιεί προκειμένου να αλληλεπιδράσει με το τραπέζι. Στο SMURF Table τα έγγραφα περιστρέφονται αυτόματα ώστε να παρουσιάζονται σύμφωνα με την οπτική γωνία κάθε χρήστη, ενώ ο διαμοιρασμός της πληροφορίας επιτυγχάνεται δημιουργώντας αντίγραφα εγγράφων τα οποία ενημερώνονται σε πραγματικό χρόνο. Η εφαρμογή επικοινωνεί μέσω αισθητήρων με το περιβάλλον γραφείου Διάχυτης Νοημοσύνης ώστε να αναγνωρίζει τους χρήστες και να τους επιτρέπει να αλληλεπιδρούν με την εικονική επιφάνεια.

Το SMURF Document Manager είναι ένα εργαλείο που επιτρέπει στους χρήστες να συνδέονται στο server του SMURF Table και να διαχειριστούν τα έγγραφά τους εξ’ αποστάσεως μέσω του Διαδικτύου. Οι χρήστες έχουν τη δυνατότητα να κατεβάσουν έγγραφα από το SMURF Table στους προσωπικούς τους υπολογιστές, καθώς και να ανεβάσουν νέα έγγραφα ώστε να τα χρησιμοποιήσουν σε συνεδριάσεις.

Το SMURF Table αναπτύχθηκε ειδικά ώστε να παρέχει στους χρήστες μία αποτελεσματική λύση για ομαδικές συνεδριάσεις, υποστηρίζοντας τη συνεργασία μέσω μιας ιδιαίτερα εύκολης στη

χρήση διεπαφής που δεν απαιτεί περιττή προεργασία ή παραμετροποίηση από τους χρήστες. Τεχνολογίες Διάχυτης Νοημοσύνης χρησιμοποιούνται ώστε να επιτρέπεται στους χρήστες να συσχετίζουν τα έγγραφά τους με προσωπικές κάρτες αντί να μεταφέρουν φορητούς υπολογιστές ή μέσα αποθήκευσης, καθώς και για να υποστηρίζεται η ταυτόχρονη αλληλεπίδραση πολλαπλών χρηστών. Τα αποτελέσματα της διεξαχθείσας διαδικασίας αξιολόγησης επιβεβαιώνουν τη χρηστικότητα του αναπτυχθέντος συστήματος, καθώς και την αποτελεσματικότητά του ως επιφάνεια συνεργατικών συνεδριάσεων.

Acknowledgements

For their support and encouragement, I would like to thank my supervisor Prof. Constantine Stephanidis.

I would also like to thank my colleagues at the Human-Computer Interaction laboratory of ICS-FORTH, and especially Dr. Dimitris Grammenos, Giannis Georgalis, Dr. Margherita Antona, Sokratis Kartakis, Ilia Adami, Giannis Drosis, Nikos Ftylitakis and Thomas Sarmis.

The work reported in this thesis has been conducted in the context of a scholarship provided by the Institute of Computer Science (ICS) of the Foundation for Research and Technology – Hellas (FORTH), including financial support and the use of the necessary equipment facilities.

Table of Contents

List of Figures

List of Tables

1. Introduction

This thesis addresses the need for a workspace that supports collaboration among multiple users in an Ambient Intelligence environment. Towards achieving this objective, a system was developed, called SMURF Table, which provides a surface allowing users to easily view and share information, supporting at the same time teamwork as well as independent user operations. The collaborative workspace is integrated in a smart office environment that is able to identify users and provide them with interaction capabilities by using various sensors in a way invisible to the users. The provided user interface is rather simple, in order to achieve collaboration among users of varying levels of experience, while remote management capabilities are available to keep users up-to-date when away from the smart office environment. SMURF Table introduces various innovative aspects with respect to the currently available collaborative smart surfaces. In particular, SMURF Table:

- Provides a workspace to support collaboration among multiple co-located users
- Allows easy viewing, duplication and sharing of any type of documents
- Automatically assigns a personal working area to each user, allowing also to dynamically reform areas according to working needs
- Presents users with their personal data automatically, without requiring manual initialization or further setup
- Automatically orients documents towards each user, regardless of their position around the table
- Provides users with intuitive ways to manage their documents
- Provides independent interaction and input methods for each user
- Allows all users to focus on documents of public interest by providing a public area for document display
- Allows users to manage their documents remotely

In order to provide a background for the proposed system, the rest of this Chapter contains a brief introduction to Ambient Intelligence, discussing its main characteristics, as well as to the application of Ambient Intelligence in working environments. The Chapter closes with an overview of collaborative tabletop interfaces, accompanied by various concepts relevant to co-located multiple user interaction.

The rest of the Chapters in this thesis are organized as follows.

Chapter 2 discusses related work in the area of tabletop interfaces. Several popular touch-sensitive hardware surfaces are presented, followed by tabletop interfaces created for various purposes. The Chapter also reports some tabletop interfaces mainly based on Infra-Red and RFID sensing, technologies that are also used by the system proposed in this thesis. At the end of Chapter 2 there is a discussion of existing gaps in the current state of the art that the proposed system is called to address.

Chapter 3 presents the design of the proposed system, describing in detail all the elements that comprise the multi-user workspace, along with the provided functionality.

Chapter 4 discusses the implementation of the proposed system and its integration in the smart environment. A detailed description of all software modules that form the system is provided, followed by an explanation of how they communicate with each other to achieve the desired functionality.

Chapter 5 presents an evaluation process that was conducted in order to determine the developed system's usability, along with the respective results and remarks.

Chapter 6 presents relevant conclusions and discusses future work.

1.1 Ambient Intelligence

The definition of the term 'ambient' is 'existing or present on all sides' according to the Merriam-Webster dictionary [1]. Ambient Intelligence (AmI) is a field of research whose main vision is to augment various environments with intelligence in order to sense people's needs and assist them into carrying out everyday life tasks in a natural way. This form of intelligence is provided by various types of devices and sensors, networked and communicating with each other, that enrich the environment with sensing and responsive capabilities in a way invisible to people. Alcañiz and Rey suggest that 'Ambient Intelligence builds on three recent key technologies: Ubiquitous Computing, Ubiquitous Communication and Intelligent User Interfaces' [2]. The concept of ubiquity is tightly connected to AmI, referring to 'presence everywhere or in many places especially simultaneously' [3].

Ubiquitous computing refers to the presence of computers everywhere, invisibly acting as parts of the environment. The concept of such a computer-augmented environment was firstly presented at the Xerox Palo Alto Research Center in the late 1980's. Weiser proposed that computers could be integrated into the environment, as well as into objects within the environment, allowing people to interact with them while they wirelessly communicate with each other invisibly [4]. Ubiquitous computing encompasses the notions of ubiquity and transparency [5]. User interactions are interpreted through various networked devices that are located in multiple places in the environment, at the same time giving users the impression that they interact with the environment itself rather than with distinct machines. A wide range of devices and technologies could comprise the backbone of such an environment, including handheld devices, pen-based mechanisms, interactive screens and computer vision technology [6].

As shown by the change of trends in computing [7], ubiquitous computing seems to continuously gain ground, especially in recent years. Once, computers were mainframes, very large in size and were operated by many users. This situation changed over the years, when computers became compact in size and were suitable for personal use, establishing the concept of Personal Computers (PC). Lately, the popularity of computers in various forms, such as Handheld PCs, PDAs and Smartphones, has led to a state where one person uses many computers that can communicate with each other to accomplish various tasks, thus establishing a notion of ubiquity in computing. This constitutes the basis of ubiquitous communication, which involves using networking technologies to connect various computers in the environment in order to support communication and information exchange between them, while users do not need to be aware of how devices communicate or cooperate. The evolution of wireless network technologies over the past years proves rather useful in supporting ubiquitous communication, since nowadays almost every computer device is equipped with wireless networking capabilities, whether it is a handheld gaming platform or a desktop PC.

AmI aims to change the way users interact with computers by making a step beyond traditional user interfaces. Traditional interfaces are passive in nature, meaning that their functionality is limited to accepting queries from users and responding by taking the respective actions. Instead of passive request-response interfaces, AmI's purpose is to offer users a more intelligent way of interaction. Intelligent user interfaces are 'able to take the initiative in order to volunteer information, correct user misconceptions, or reject unethical user requests' [8]. Such interfaces are able to sense and collect various types of information regarding the users, the context of use, as well as possible artifacts that the users may use to interact with them. Such information can be extracted by using various technologies, such as visual recognition, location sensing, speech recognition, RFID sensors, or even simple text input from the user. An early example of an intelligent interface is presented in [9]. This interface starts a text dialogue with first-time users, asking them questions about themselves. The user's answers are used to extract keywords that are applied as filters in internally stored information structures representing human stereotypes. The more information the user provides, the more elements of the user's personality can be deducted by the system in order to create a user model. This model can then be used to offer the user a more personalized experience while using the system, by providing personal recommendations and making user-specific remarks. According to [10], these interfaces usually feature multimodal input, multimodal output, interaction management, or even combinations of them. Users are able to provide mixed input, such as speech combined with gestures, receive mixed types of output, such as coordinated text with speech through an animated assistant, as well as receive personalized recommendations and predictive warnings, based on information that the interface collects by monitoring the users' attention, intentions and task progress.

1.2 Ambient Intelligence in the Working Environment

An example scenario is presented below in order to demonstrate some of the numerous possible applications that AmI may have in peoples' working life.

John is working for a multinational company and his office is in a large building in the centre of a big city. Arriving at work, a camera identifies John by his car's license plate and guides him to the parking spot closest to his department's entrance by wirelessly sending voice commands to the

GPS in John's car. After leaving his car, John enters the building and is headed towards his office. He waves his RFID card in front of a panel next to his office door and the door automatically opens granting him access to the interior. At the same time, the intelligent windows in the office sense the exterior light intensity and automatically open up to a point where the appropriate amount of sunlight enters the room. John sits on his chair and, after placing his card on the desk, his computer automatically loads his most recent work documents. An intelligent software assistant checks the time and reminds John of an important meeting that begins in five minutes. John picks up his card, leaves his office and is headed towards the meeting room. A while after the meeting starts, John realizes that he forgot to e-mail the documents he had prepared for the meeting. When his turn arrives, he places his card on the meeting table and his documents are automatically transferred to the computer used for the presentations. When the meeting ends, John picks up his card and his documents are automatically removed from the presentations computer. John returns to his office and places his card on the desk to continue working. An intelligent coffee machine next to the desk knows John's habits and, when the time comes, vocally asks him whether he would like a cup of coffee with the usual amounts of coffee, sugar and milk. John replies, the coffee machine prepares a cup of coffee and informs him that it is ready. Meanwhile, one of John's colleagues is looking for him in order to discuss on some new designs that John had requested. The colleague locates John through a PDA that wirelessly communicates with the environment to look for the place where John's card is found. A camera outside his office recognizes his colleague and informs John, who vocally commands the door to open. The colleague places his card on John's desk and the design documents are automatically transferred to John's computer. When the desk identifies two cards, it assumes that John is in the middle of a discussion and takes care of automatically silencing his office desk phone. After the colleague leaves the office, the desk restores the phone's volume and informs John that while he was busy his wife called twice. The second time his wife called, she left a voice message to remind him of the dinner they had arranged with friends for that afternoon. The computer on which the phone is connected analyzes the message and deducts the calling reason and the dinner hour and place. Immediately, the computer also checks the current time and John's working progress to infer an estimation of the time he needs to finish his work. The computer automatically generates a message, mentioning that John will be late and that he will arrive to the restaurant approximately 35 minutes past the prearranged time. The message is then automatically sent through the internet as an SMS to his wife's mobile phone, so that she knows when to expect him. After sending the message, the computer, having monitored John's actions and typing rate, assumes that John is getting tired and adjusts the lighting conditions in the office to make him feel more relaxed. When John's work is over, the computer automatically retrieves road traffic data through an online system that is updated via a traffic monitoring mechanism installed throughout the city streets. At the same time, the computer also retrieves an online map of the city and uses the traffic data to calculate the fastest path from John's office to the destination restaurant. When the path is calculated, it is uploaded to John's PDA, which is wirelessly connected to the computer. John leaves the office, enters his car and transfers the route data wirelessly to his car's GPS, which guides him to the destination using vocal directions.

The above scenario may seem rather futuristic in general, but it actually encompasses some of the potential elements that can comprise an AmI working environment. Some possible real life scenarios can be found in [11]. Time saving, smart scheduling, flexibility and efficiency are only some of the overall benefits that AmI can offer to people concerning their working life.

Interest towards the conception and realization of smart environments has been rapidly growing for some years now. Although AmI concepts have not yet been applied in real life situations and spaces, there are some efforts coming from research groups and large companies that demonstrate what an intelligent office may look like in the near future.

The BlueSpace project [12, 13] is a prototype workspace created by IBM as an approach towards adaptive working environments. The prototype is a personal workspace equipped with various types of sensors and displays. The sensors are used for measuring environment conditions, such as light, temperature, humidity and noise, while the displays are intended to act as the user's primary working display, colleague information display and environment control panel. There is also a panel on the exterior of the workspace to inform colleagues on the occupant's current activity and availability. BlueSpace uses RFID technology for user identification and workspace adaptation. The goal of the project is to combine multiple such workspaces into a larger working environment that will support collaboration and communication, at the same time providing each user with a personal space that can adapt to their needs.

IDEO's 'Q' [14] is a mobile workstation prototype aspiring to go beyond the traditional meaning of a personal workspace as it is known today. It is a capsule-like piece of furniture that accommodates a single user, who can use a joystick to drive it in order to get close with colleagues in the same environment. The user can connect their PDA to the workstation in order to load personal documents and settings that can be viewed through a set of available displays, while the seat can be adjusted to the user's needs, whether they need to work or relax.

The Animated Work Environment (AWE) [15, 16] is a concept of a future home environment that can accommodate its owner's working needs. AWE aims to combine information technology with robotics and architectural design in order to create an intelligent interior environment that will be able to transform itself according to the user's needs. The environment will consist of various surfaces that the user can configure, control and position through smart interfaces. Depending on the configuration of these surfaces, different environment modes will be available, such as a configuration for document composition, a sleeping state, or a presentation mode. Besides dynamic surface composition, AWE will be augmented with technologies such as sensors for environmental monitoring, RFID for identification and physical object recognition, displays, projectors and input devices.

The integration of AmI into working environments looks rather promising considering the possibilities it can offer to improve peoples' experience at work. As technology rapidly evolves and hardware continuously gets cheaper and more compact, the potential of AmI technologies to be adopted by users and to contribute improving the quality of life increases.

1.3 Tabletop Interfaces for Collaboration

Tabletop interfaces are provided by applications that allow users to accomplish various tasks by interacting with physical or digital information on horizontal or vertical surfaces. Usually, though, the term 'tabletop' is mostly used to refer to horizontal surfaces such as standard tables or large touch-sensitive panels, on which information is either presented through embedded displays or projected by one or more projectors above or below the surface.

Collaborative tabletop interfaces are a category of such interfaces whose main goal is to support information sharing and teamwork among multiple users, by allowing them to interact with personal and shared information. This type of multi-user interaction, also referred to as around-the-table interaction, is characterized by independence and concurrency. Users should be able to work simultaneously on a common surface, while each one should be allowed to perform individual interactions without affecting others' work.

An early effort to characterize co-located collaboration through software applications was the Single Display Groupware (SDG) model [17]. SDG was defined as 'computer programs that enable co-present users to collaborate via a shared computer with a single shared display and simultaneous use of multiple input devices'. This definition distinguishes such applications from traditional groupware, which consists of applications that are run on multiple workstations that communicate with each other over a network. The main difference between SDG applications and traditional single-user or multi-user applications relies on the provided input and output channels. Single-user applications provide a single input channel, through a keyboard or other input device, and a single output channel, usually a monitor or other display. Traditional multi-user applications provide each user with one personal input channel and a personal output channel that is used to display personal and shared information. SDG applications provide each user with a personal input channel, while a single common output channel is used to provide feedback to all users simultaneously.

One of the most important characteristics of a collaborative tabletop interface is how users perceive territoriality on the provided surface. The partitioning of the provided workspace is a key concept towards the coordination of individual user activities [18]. When a working surface is not visually split into distinct areas, people tend to distinguish regions by positioning content in multiple locations depending on the content's use. The position of users themselves around the table is also important, since it affects the partitioning of the common workspace, with the areas closest to each user being considered as personal spaces. The partitioning of a collaborative tabletop surface is crucial to efficient transition between personal and group work [19]. The areas that comprise a common workspace should be organized in order to allow smooth exchange of content, while switching between personal and shared work should be fluid, without interrupting individual user activities. Scott et al. [20] pointed out that a collaborative tabletop surface should contain three types of areas, namely personal, group and storage territories. A personal area acts as a user's dedicated space for manipulating personal content independently from other collaborators. A group area is a common space provided especially for placing content that is shared among all collaborators. A storage area is usually provided as a place where a user can gather and arrange their content. The storage space may provide content indexing capabilities and may also be organized in various forms, such as a pile or table of documents.

In addition to surface partitioning, the content itself should be properly oriented according to users' positions. Orientation during co-located collaboration is crucial to teamwork and independent activity as well, since it enforces comprehension of information, coordination of activities and communication among participants [21]. Users working on a common surface will most probably occupy positions at different sides of the table. Thus, it is important that information is properly displayed to each user individually, as well as to all users simultaneously when referring to shared content. Items positioned within a user's personal space should be oriented towards the respective user, while items in a public area should be oriented in a way that allows information to be legible and easily viewed by all collaborators. Ringel et al. [22] used content orientation as a way to allow sharing of documents. They implemented a mechanism according to which a user orients a document towards the center of the table to render it as public, while reorienting it towards the user makes it personal again.

Multi-user collaboration around a tabletop surface is a social activity in which users participate aiming to accomplish tasks as a team, as well as individually. Similarly to any co-located social activity, possible conflicts and disagreements may emerge mostly due to incompatible concurrent actions [23]. Some common problems that appear on collaborative tabletop interfaces are based on user actions that are taken without knowledge of how they may affect others' work. A few examples of such actions are:

- A user may change the state of a common item while others are still examining it
- A user may gather or reorient common objects towards their personal area while others may need them
- A user may take an item from another user's personal area without permission from the latter
- A user may change the state of the entire workspace while others are still working

A collaborative tabletop interface should be properly designed in order to minimize, if not completely overcome, such issues. Morris et al. [24] have developed two sets of coordination policies concerning interpersonal conflicts during concurrent interaction on tabletop surfaces. They implemented an interface which responds to such social breakdowns in order to help users realize the result of their actions and focus on their tasks. A set of global coordination policies includes interdiction of global state changes if anyone still interacts with items, widgets that allow individual voting on proposed global changes, as well as the use of a special menu for global actions so that attention can be drawn on the potential impact these actions may have on other users. Item specific policies include duplication of an item when two users try to grab it simultaneously, user ranking to allow high-rank users override lower-rank user actions (such as teachers and students), as well as tearing of contested documents in order to encourage negotiation for reassembly.

The system proposed in this thesis provides a collaborative tabletop interface that was designed taking into account users' needs for personal and shared spaces, as well as efficient information sharing through rather simple interactions. A detailed description of the structure of the implemented virtual surface, along with all possible user actions, is provided in Chapter 3.

2. Related Work

Tabletop Interface development has been widely practiced in the past few years for various purposes. Depending on their target user groups such interfaces have been developed to play a very important role as an intermediary to many social aspects of life:

- To support collaboration among team members in an office environment, e.g., [25].
- To aid medical examiners in communicating with their patients and analyzing their behaviors and reactions, e.g., [26].
- To provide educational assistance in the form of interactive lessons or educational games, e.g., [27].
- To help audio professionals, sound engineers and musicians by offering various ways to analyze and process audio streams, e.g., [28].
- To offer users quick access to local and web resources in public environments such as airports or museums, e.g., [29].
- To support remote meetings when participants are not co-located, e.g., [30].

Interfaces intended for tabletop use vary depending on the surface they are designed to be applied on. Tabletop surfaces can be divided in four categories:

- Horizontal table surfaces
- Vertical surfaces
- Multi-touch hardware surfaces
- Combination of horizontal and vertical surfaces.

Horizontal surfaces are basically common tables or desks used to display information that is projected onto them by one or more projectors. Interaction mechanisms on horizontal surfaces are completely implemented and processed through software. Vertical surfaces, often referred to as interactive walls or windows, are similar to horizontal surfaces and differ solely in their vertical positioning towards users. Multi-touch hardware surfaces are surfaces with built-in touch-sensing capabilities. They can recognize multiple input touch points at the same time by using layers of sensors or computer vision techniques. While input sensing is hardware-based, multi-touch surfaces usually are accompanied by appropriate software libraries to provide developers with the means to implement the desired functionality. A surface can also consist of a combination of horizontal and vertical surfaces to provide extended functionality and space. In such combined spaces interaction capabilities may be restricted to one part of the system (e.g. using a horizontal surface for interaction and a vertical surface for extended display) or they may be available on all surfaces.

This chapter describes related work in the area of tabletop interfaces and presents the key features for existing interfaces. Many of these interfaces are based on hardware multi-touch surfaces, thus some commercial and non-commercial surfaces are also presented. The SMURF Table reported in this thesis is a surface with its own tabletop interface that uses RFID recognition and Infra-Red tracking technologies. Thus, some related interfaces that use these technologies are also described. Disadvantages of current approaches and differences between SMURF-Table and related interfaces are discussed at the end of this chapter, along with the approach adopted to fill existing gaps.

2.1 Multi-touch Hardware Surfaces

Multi-touch hardware surfaces are widely used mostly because of their ability to sense at the same time multiple input points created by fingers, hands or other objects. They allow applications to interpret touch points and gestures as pointing and transformation mechanisms.

The key characteristics that make existing multi-touch surfaces differ from each other are:

- The way information is displayed on them
- The ability to distinguish users
- The number of supported simultaneous users
- The types of supported applications and their extensibility
- The provided input capabilities

Perceptive Pixel's Multi-Touch Wall [31] is a 2m x 1.2m vertical multi-touch surface. It is a screen that can support up to four users working simultaneously. The same team also created the Multi-Touch Workstation, a 1.2m x 0.7m horizontal surface in a form similar to an architect's drafting table which is mostly intended for single-user interaction. Both of these surfaces are able to sense an unlimited number of simultaneous touches and are pressure sensitive, so that gloved hands and arbitrary styluses can also be used for interaction. Developers can integrate multi-touch capability into existing applications or create native multi-touch applications from scratch by using the standard software development kit that accompanies these surfaces. A geo-spatial software development kit can also be obtained for use in geographical applications. These software libraries provide special capabilities for viewing, analyzing and manipulating geo-spatially referenced datasets. The applications currently developed for these surfaces include weather broadcasting, medical data analysis, information visualization, digital content creation, storyboarding, financial data analysis and museum exhibition information.

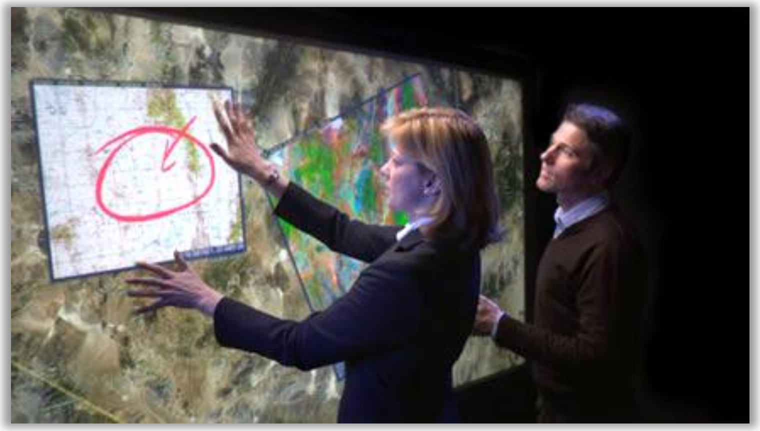


Figure 1. Perceptive Pixel's Multi-Touch Wall

Ideum's MT-50 [32] is an interactive table created especially for use in museums. Its surface, whose dimensions are 1.3m x 0.8m, is made of glass. Information is displayed on the surface by a projector placed inside the table's chassis along with a computer system. Input touch points are recognized by three built-in Infra-Red cameras that can identify up to a total of fifty touch points at the same time. The main purpose of the table is to allow visitors in a museum to interact with virtual exhibits and obtain information in the form of text documents, images or videos. MT-50 comes with the Collection Viewer application which is a multi-touch enabled image and video viewer. GestureWorks and NUITech SnowFlake are software development packages supplied for developers. GestureWorks is a framework that contains multi-touch and gesture libraries that can be used in Adobe Flash, while SnowFlake consists of multi-touch and gesture recognition libraries for use in programming and scripting languages such as C / C++, C#, Java and Python.

Figure 2. Ideum's MT-50 Surface

DiamondTouch [33] is a 1m x 0.8m touch and gesture sensitive top-down projected screen developed at the Mitsubishi Electric Research Laboratories. The surface contains an embedded matrix of antennas that transmit unique signals. Each user is connected to a separate receiver. When a user touches the surface, antennas near the touch point couple a small electrical signal through the user's body and to the receiver. This way multiple touches by a single user are supported and users can be distinguished. DiamoundTouch supports up to four users simultaneously. It comes with a mouse emulator and an on-screen keyboard that allow users to operate some popular Windows Applications using their hands, as well as a multi-user annotation tool that provides users the ability to save mark-ups on documents and images. A software development kit is available that allows developers to create Windows and Linux applications that utilize the surface's touch and gesture capabilities using C / C++, Java or ActiveX Controls.

Figure 3. Mitsubishi's DiamondTouch Surface

Microsoft Surface [34] is a 1.08m x 0.7m multi-user interactive table. Its interior contains a computer system along with a projector that displays information on the surface and a camera-based Infra-Red vision system to handle input. Users are allowed to grab digital information with their hands and interact with virtual content by using their fingers or gestures. The surface can also recognize objects through the cameras by using image recognition in the Infra-Red spectrum to identify tagged items and shapes. Users can place objects on the surface to provide input to some applications or even transfer digital content into their Bluetooth-enabled mobile devices. Microsoft provides developers the Surface software development kit to allow them to create new multi-touch enabled applications for use on the Surface. Developers can use Microsoft's Visual Studio environment, since Surface is fully compatible with it, as well as technologies such as the Windows Presentation Foundation and the XNA Framework. The core applications currently available for Surface include interactive maps with information highlighting, browsing and display of images and videos, music collections creation and manipulation, a newsreader with news stories and videos retrieved from selected web sites and some classic games.

Figure 4. Microsoft Surface

Entertaible [35] is a multi-touch enabled horizontal platform developed by Philips Research Laboratories. It is intended mainly for gaming usage allowing users to play by using their fingers and making gestures or by placing various objects on top of the surface. Its 32-inch surface consists of an LCD display screen and can sense several input touches simultaneously. The input mechanism is Infra-Red based and consists of a series of LEDs and photodiodes placed along the edges of the display. When a finger or object is placed between a LED and a photodiode the latter no longer detects the Infra-Red light from the LED and so input is detected. A software development kit for Entertaible is not yet available and only a few sample applications have been created mostly to demonstrate the platform’s capabilities. Internet connection capabilities will probably be a feature of the system, providing users the ability to play multi-player board games online or download new applications. It is targeted to use in public spaces, such as hotels and gaming centers, and it might be available as a home entertainment platform in the future.

Figure 5. Philips Entertaible

SmartBoard [36] is an interactive whiteboard intended for educational and business usage. Information is displayed on its 1.7m x 1.3m surface by using a projector placed above the whiteboard using a special wall mount. Input sensing is implemented using the DViT technology [37]. Four digital cameras are placed in the four corners of the surface that detect fingers or objects touching the surface and interpret them as input by calculating the distance of the touch point from each camera. Up to two users can work on the whiteboard concurrently because only two touch points can be recognized at the same time. A SmartBoard software development kit is available to allow users create their own applications for the whiteboard using various programming languages such as C++, C#, Visual Basic and Java. Currently, there are many applications available for the whiteboard including text and presentation viewers and editors, support for saving documents in popular formats, painting applications with support for special pens and erasers and interactive lesson and quiz creating tools.

Figure 6. SMART's SmartBoard

2.2 Tabletop Interfaces

UbiTable [38] is a project whose main goal is to provide efficient walk-up setup and fluid user interaction support on horizontal surfaces. The project's design is focused on creating a model of association and interaction between mobile devices and the tabletop and to provide specific information sharing semantics described as private, personal and public. The digital document manipulation functions of UbiTable include document duplication, markup, editing and digital ink for drawing and annotation. The main idea of the interface is to support shared interaction with equal input capability to public documents, but owner controlled document distribution and replication. When users start the application their laptop initializes a handshaking protocol with the tabletop through a trusted channel. A personal space is assigned to that laptop and the user is ready to interact with the table by touching its surface. Private data is displayed on the user's laptop and is not accessible or visible to others. Personal data is displayed on the user's side of the table, visible to all users, but electronically accessible only to the owner. Public data placed in a shared area can be viewed and accessed by anyone, but only the owner is able to distribute and replicate them. Data in the public shared area can be dragged back into the owner's personal or private space at any time. When users leave the table, all copies of their documents are cleared to preserve privacy. The application is entirely written in Java and the interface is built using the DiamondSpin toolkit [39]. The tabletop communication capabilities are implemented using TCP/IP over wireless LAN and the table used to support the UbiTable interface is a DiamondTouch surface.

Figure 7. UbiTable

The SoundTracker project [40] was developed to explore the effects of using individual audio channels during collaboration around a shared display surface. The system uses a DiamondTouch table for the interface and five soundcards, all connected to a standard personal computer. One of the soundcards is connected to a set of speakers, while the rest are connected to separate earbud-style headsets. Users wear a single earbud, so that conversation can be made at normal volume levels. The developed sound libraries allow sound clips and text-to-speech requests to be played, paused or stopped to the desired sound channels. The application provides users with the ability to bind songs to photographic stills from movies by dragging song titles from a song collection onto images from an image collection. A user can listen to a song by touching a musical note icon associated with a song title. The song plays through that user's personal earbud and can be stopped at any time by touching the musical note icon again. While a song plays, a slider appears allowing users to seek forwards and backwards within the song. Image objects are manipulated in a way similar to the song objects, using speech bubbles, instead of musical notes, to listen to brief captions of a specific scene. A user can listen to a song and a scene caption at the same time, and both songs and images can be simultaneously used by more than one user though their personal earbuds. When a user has associated a movie scene with a song, the combined object can be routed to play through the public audio channel using the speakers for all the users to hear. The interface uses the DiamondSpin toolkit to implement object placement and orientation to match users' positions around the table.

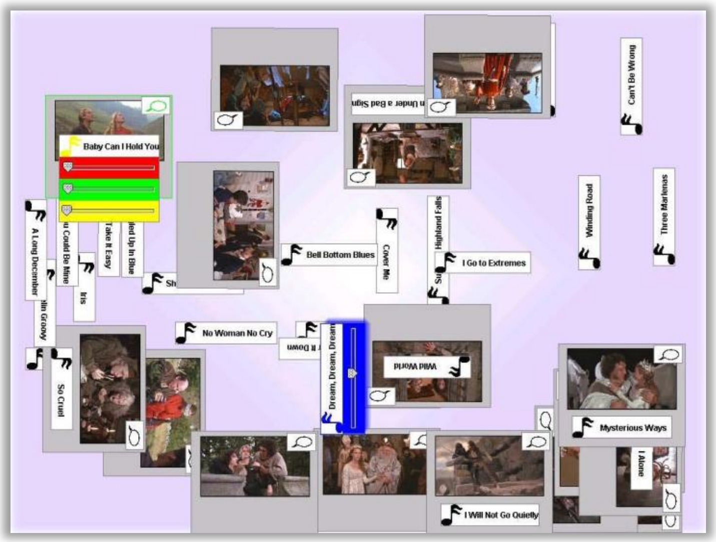


Figure 8. The SoundTracker Interface

TeamSearch [41] is an application that supports co-present collaborative browsing of digital content tagged with metadata on a horizontal surface. Users are able to arrange specific circular query tokens on the surface to form Boolean-style queries. The purpose of the project's development was to examine whether it is more effective to assume that each user's activities are a contribution to a single collective query, or it is best to consider all activities as separate and parallel queries. Users work on a DiamondTouch surface, on which information is displayed by a projector. TeamSearch was written using the DiamondSpin toolkit. The interface consists of a pile of photos in the table's center, some circular widgets representing metadata categories, each of which is subdivided in pieces that contain specific values for that category, some query tokens for each user to move around the table, and an area in front of each user to present query results as thumbnails. A user can form a query by dragging a query token and releasing it over a specific value of a category widget. The photos satisfying the query values appear as thumbnails in the user's area. To refine the query, the user can take another query token and place it over a value of another category widget. This action is similar to a Boolean 'AND' between the two queries. Two implementation alternatives were considered in order to determine when collaboration is more effective. Under the collective queries implementation, when query tokens are placed on category values the system assumes that all tokens form a single query and presents the results in front of all users. Under the parallel queries implementation, when tokens are placed over category values the system interprets each user's tokens as a separate query, displaying the respective results in front of that user.

Figure 9. The TeamSearch Interface

The ClassificationTable, the MatchingTable and the PoetryTable [27] are three applications developed to support cooperative learning of foreign languages using an interactive surface. The applications are designed to run on a DiamondTouch table and were implemented using the DiamondSpin interface toolkit. The ClassificationTable provides users with a set of clues presented as text boxes. There are four categories, one in each corner of the surface, and users can touch and drag clues into the four categories while the system provides feedback regarding the correctness of their classification. The MatchingTable presents users with some images and phrases. The users have to match the phrases with the images by dragging phrase text boxes over photos and releasing them. The correctness of the matching is presented by turning the phrases green when they are correctly matched, or red when the pairing is wrong. The PoetryTable provides users with word tiles that they can move around the surface to form sentences. When a word is double-tapped with a finger, a menu appears that presents some prefixes and suffixes that can be used to conjugate words. Two feedback design alternatives were considered, visual and audio, to determine whether it is best to visually inform users about the correctness of their answers, by appropriately highlighting the respective clues, or by using private and public audio feedback to notify them.

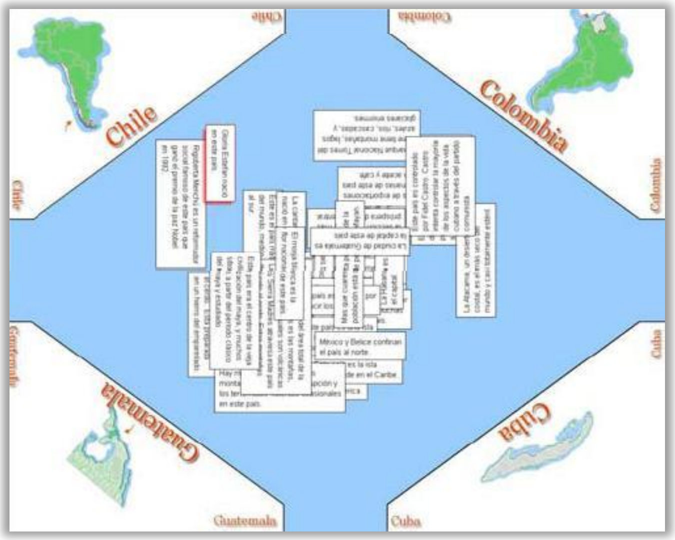


Figure 10. The ClassificationTable Interface

The Tabletop-Mobile Augmented Reality system (TMAR) [42] is a combination of a tabletop interface and a mobile interface that uses augmented reality to allow users to interact with 3D virtual objects. The tabletop interface, named ARTable, consists of a semi-transparent surface, a projector that displays information onto the surface and a camera underneath the surface that can recognize and track objects placed on the table using computer vision software. Objects positioned on the table have special square markers on their top and bottom that can be detected by the camera, so that their exact location and orientation can be obtained at any time. A special frame is applied on the table surface to make it touch sensitive, so that multiple users can simultaneously manipulate virtual contents on the table surface using hand gestures. The physical objects used for interaction are cardboard squares that represent materials, selection tools and other control widgets. The mobile interface consists of handheld personal computers that are equipped with digital cameras that can track the markers on top of the objects. This way, the object viewed can be identified and the pose of the handheld PC is determined relative to the object, so that 3D virtual imagery can be overlaid on the physical objects and viewed through the handheld PC. Users can interact with the virtual 3D models through the handheld PC touch panel. When a virtual object is selected, some action buttons appear on the mobile device allowing users to manipulate the respective object. The use of a tabletop interface and separate mobile devices also inserts a notion of public and private spaces into the system.

Figure 11. The TMAR Interface

The BlueBoard system [29] uses a tabletop application intended for quick walk-up personal use as well as spontaneous small group collaborative use. The system consists of a large vertical touch sensitive screen, equipped with a badge reader for personal identification. A user can log on to the system by placing an HID brand badge close to the screen and, when authentication is completed, the user's personal content is loaded. When one or more users are connected to the BlueBoard, an image appears for each user in a dock on the right of the display. All information displayed on the screen is basically web content. The personal content of a user must be set-up ahead of time by linking all the desired information to that user's home page. The user images, named p-cons, can be touched by a finger to bring up the home page of the respective user. The p-cons are also used for information exchange between users. A user can drag some content and drop it on another user's p-con to provide the latter with a copy of that content. When a person leaves the BlueBoard session by using the personal badge, the content stored in that user's p-con is e-mailed to them and is explicitly removed from the server computer of the BlueBoard the user was working on.

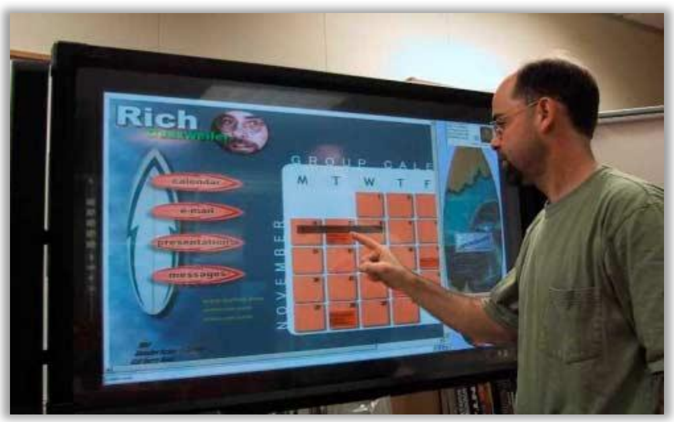


Figure 12. The BlueBoard System

InteracTable and DynaWall [43] are interactive tabletop surfaces created as components of i-Land, a computer-augmented environment that acts as a large workspace intended to support cooperative work. InteracTable is a mobile horizontal surface that allows groups of users to discuss, create, display and annotate digital information objects. Users can work on the table using their fingers, gestures or pens to write or draw on objects. The surface is touch sensitive and information is displayed by a projector placed underneath the surface, within the chassis of the table. DynaWall is large touch sensitive vertical surface that allows users to interact with information objects by using pens and gestures. A take-and-put mechanism was implemented which allows a user to take information objects at one position, walk over without being in contact with the DynaWall, and place them somewhere else using gestures. The developed software, called BEACH, provides an infrastructure for cooperative sharing of information between the devices of the working environment. Users are able to transfer information from the InteracTable to the DynaWall at any time and vice-versa.

Figure 13. The InteracTable Interface

The Shared Space Siggraph [44] is an application that was designed to explore how augmented reality could be combined with tabletop interfaces to enhance co-located teamwork. Users gather around a table and each one wears a head mounted display with a camera attached to it. A set of large cards on the table are used as physical objects for the interface. Each card is labeled with distinct square tracking markers on both sides, so that computer vision techniques can be used to recognize the marks and determine the exact pose of a camera relative to them. The head mounted display and camera are connected to a computer that performs image processing on the video input and draws computer graphics onto the tracked image. The resulting graphics are presented to users through the head mounted display. The application is a card matching game and its goal is to collaboratively match objects that logically belong together. The physical cards are marked with Japanese Kanji characters, each one representing a different object. When users turn over the cards, they see different 3D virtual objects on top of them. When two virtual objects are correctly paired, by placing the respective cards side by side, an animation is displayed involving the objects to notify users that the matching is correct.

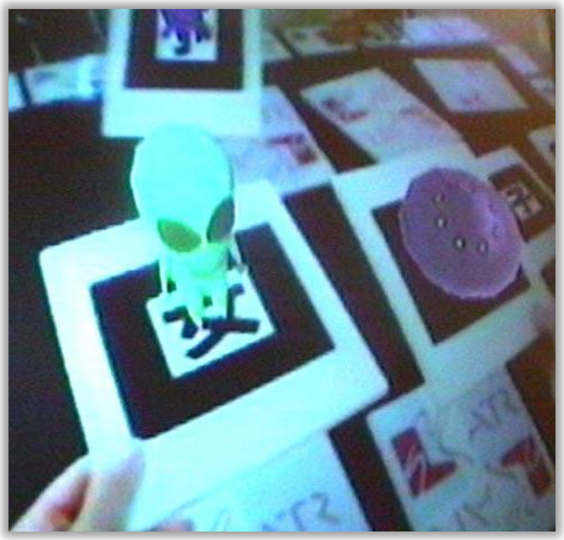


Figure 14. The Shared Space Siggraph Interface

The reacTable project [28] combines a tabletop interface with music technology in order to provide casual and professional users with a platform for dynamic musical synthesis. The application runs on a round table whose semi-transparent surface is used to display information from an underneath projector, as well as to recognize special marks on shaped objects through an underneath Infra-Red camera using computer vision. The objects serve as functions and are categorized into audio generators, audio filters, controllers, control filters, mixers and global objects. Each category corresponds to a specific shape and has its own distinct marker. When an object is placed on the table, a circle is drawn around it to point out its perimeter. Users can configure two parameters for each object depending on its type. The first parameter is configured by rotating the object itself, usually corresponding to frequency or speed, while the second parameter is configured by dragging a finger around the object's perimeter, usually related with amplitude. Musical synthesis is realized by forming topologies using the objects and by moving them around the table. The objects are connected to each other by a set of rules according to their type and proximity with other neighbors, while visual feedback of the resulting sonic topologies is projected on the surface.

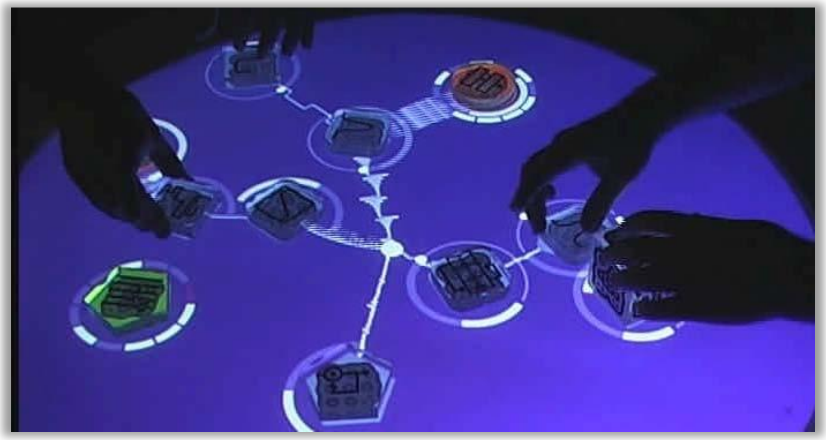


Figure 15. The reacTable Interface

The Shared Speech Interface [26] is an application created for medical assisting purposes in order to examine the challenges of representing speech visually to users around a table. It is designed for use by two users concurrently, to support communication between a deaf patient and a hearing medical doctor. The application runs on a DiamondTouch table and the DiamondSpin toolkit was used to implement the interface elements. Conversation between the two individuals is realized by keyboard input, while the physician is also able to speak through a microphone equipped headset as well as make gestures on the table while discussing. The audio captured from the microphone is used as input to a speech recognition engine and speech APIs are used to convert speech to text and send the respective data to the main application. Text data, coming either from a keyboard or a microphone, appear on the surface in front of the user that contributed that part of the conversation. Each part of the conversation is represented as a speech bubble that can be moved around the table using touches and gestures. The speaking user is provided with three best guesses of their speech and is able to select the phrase that matches their intended speech. All conversation fragments can be deleted at any time to save space by dragging the respective speech bubbles to virtual trashcans.

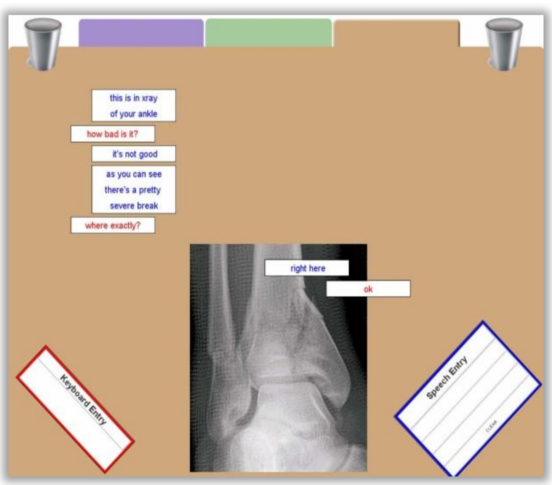


Figure 16. The Shared Speech Interface

Escritoire [30] is a tabletop system created to act as a projected display for a personal computer. Two overlapping projectors are used to display information on a desk and provide a workspace for a single user. A low-resolution projected region occupies the whole surface of the desk, while a high-resolution area is used to project the user's focus of attention. The items displayed on the desk are sheets of virtual paper that the user can view, move around and annotate. Text documents and images from standard applications can be converted to the virtual form used by Escritoire in order to be accessible by the user. The system uses a two-handed input mechanism using two different pens. A large digitizer is used as the user's desk and its pen is assigned to the user's dominant hand for fine tasks such as pointing and annotation. An ultrasonic pen, whose receiver is attached to the edge of the desk, is assigned to the user's non-dominant hand and is intended for coarse tasks such as document selection and sheet positioning on the desk. A client-server system was implemented to support collaboration over a standard internet connection. Several clients can connect to a server that stores all the system states and provides a shared visual workspace which is available to the collaborators as part of their desk. Communication between remote users can be achieved either through audio channels, by using microphones, or by pen gestures that leave visual traces to highlight points of interest. Video conferencing is also allowed through video channels, by using separate display monitors to allow users to see each other.

Figure 17. The Escritoire Interface

Tabletop AgilePlanner [25] is an application created to provide a shared surface for both co-located and distributed agile planning meetings. Its main goal is to emulate collaborations of traditional pen-and-paper based planning meetings. It runs on a SmartBoard horizontal surface and its interface uses Microsoft's Windows Presentation Foundation technologies to provide a shared workspace that displays electronic planning artifacts. The basic elements of the interface are electronic story cards that the user can write on and share with other users, as well as a control palette to allow users to perform various functions on the cards. The electronic cards can be selected, viewed, moved and rotated by finger touches and gestures on the surface, while the functions on the control palette are executed generally by first touching a card and subsequently touching a function button on the palette. A handwriting mechanism is used to allow users to write on the cards, either by using a stylus or their fingertips, while a set of gestures defined by the Windows Presentation Foundation allows the creation of interactions to support creating, deleting and minimizing cards. Distributed collaboration can be achieved by providing a shared workspace when remote AgilePlanners join a meeting over an internet connection. A portion of the surface is used to display a remote system's electronic cards and a set of telepointers is used to act as remote mouse cursors that are updated to display the finger touches on remote systems.

Figure 18. The Tabletop AgilePlanner Interface

Dynamo [45] is a large communal multi-user interactive surface that consists of one or more displays that can be tiled both horizontally and vertically. Its primary goal is to allow users in public spaces to seamlessly manipulate digital media such as documents, video and images, as well as share information with other users. Interaction with the system is allowed by connecting multiple USB mice and keyboards, as well as by connecting remotely through a laptop, PDA, or desktop personal computer. When a user connects to the system, a color-coded telepointer is assigned to that user as a personal mouse cursor, and a personal palette in the same color appears that contains media sources and media sinks. Media sources are basically the files that can be used on the surface. They can be transferred to media sinks, such as USB flash drives, by dragging them on the respective media sink icons. A public palette is also present that allows access to publicly available devices, such as printers. A user can select a personal working region by holding down the mouse and dragging it to indicate the workspace's extent. Digital media can be transferred to the surface from USB flash drives, PDAs, connected personal computers and vice versa. Files can be shared by dragging them from a personal media sink to the public working region of the surface. Dynamo also supports asynchronous sharing of digital media by allowing users to create media parcels that encapsulate multiple media items and can be posted on the surface for recipients to pick up and open.

Figure 19. The Dynamo Interface

RoomPlanner [46] is a room furniture layout tabletop application. It was designed in order to incorporate new interaction techniques that exploit the use of multiple fingers, hand shape and gestural input. The application runs on a DiamondTouch table and is designed for use by two users that sit in front of each other. A public space is shared between users in the center of the table that represents a room from an overhead 2D orthographic view, while furniture pieces are shown as outlines that reflect their relative sizes at the given scale. Each user also has a private space that is positioned immediately in front of them. The interface provides users with editing planes, which are semi-transparent trays on which furniture pieces can be positioned and manipulated. Editing planes can be added to the room or replace existing portions of the room. Four categories of interaction techniques have been developed for use in the RoomPlanner. Single finger techniques include tapping and dragging, to select and move objects respectively, double tapping, to activate context-sensitive menus that provide functions such as object rotation and plane editing, as well as flicking and catching, to send objects directly to another user or receive objects from them. Two finger techniques include rotation and scaling, where one finger is used as the rotation center and the other determines the rotation angle or scaling ratio, as well as parameter configuration, where one finger selects a parameter widget from a menu and the other finger sets the parameter's values by tapping on the respective widget's arrows. Single hand techniques include room rotation by placing a flat hand on the table and rotating it, sweeping of objects by moving a vertically positioned hand on the table, as well as viewing of object properties by displaying a private information box behind a horizontally placed hand on the table. Two hand techniques include object collection by sweeping two vertical hands toward one another, as well as creation of rectangular editing planes by recognizing two corner-shaped hands as the top corners of a rectangle.

Figure 20. The RoomPlanner Interface

WordPlay [47] is a collaborative tabletop interface aiming to aid small groups of users in generating and organizing ideas. The application runs on a table whose semi-transparent surface displays information from a projector located within the chassis of the table. An underneath camera is also used in combination with computer vision techniques to provide multi-touch functionality. Before users start brainstorming, a subject is selected and appears on the surface. Users can contribute ideas to the table either by typing through a software keyboard or by speaking into a microphone. Audio data flows from the microphone to a series of software programs that attempt to parse the audio into text, while natural language processing software divides phrases into their primitive elements. Extensive databases are used in order to retrieve associations relative to the brainstorming subject, based on elements from the parsed phrases. Ideas are presented on the table as text blocks that users can move around, organize, reorient and delete by touching them and making gestures. Users are also allowed to request associations and suggestions from the system by tapping on phrases of interest.

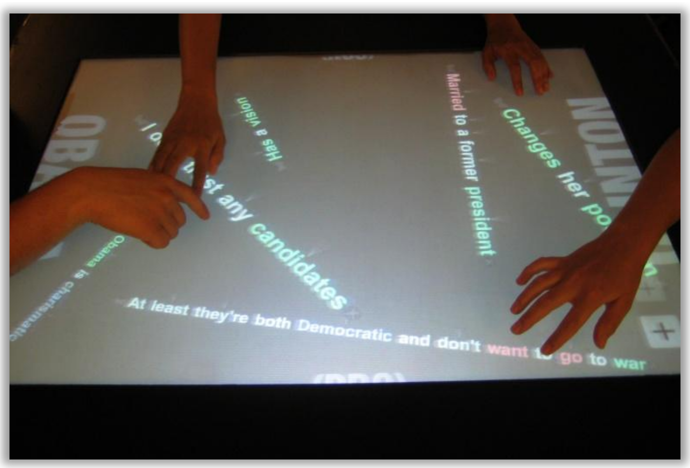


Figure 21. The WordPlay Interface

ConnecTable [48] is a mobile appliance intended for face-to-face meetings. Its hardware platform consists of a small touch-sensitive display tablet that accepts input from a single pen. The interface supports viewing of documents and images as well as simple sketching. The key feature of ConnecTable is that it is equipped with a unique radio-frequency transponder and sensor, allowing two devices to recognize each other. When two ConnecTables are moved next to each other, they establish a connection and automatically form a larger working area for collaboration. When connected, a user can drag documents towards the other user's surface to share information with them. Multiple views of the same document are allowed and documents can be reoriented to fit each user's viewing angle. The common workspace exists while the users are working on it and it can be divided into the two users' private working surfaces by pulling the devices apart from each other. The software functionality of ConnecTable is implemented as a new modified version of the BEACH software that was used in InteracTable and DynaWall.

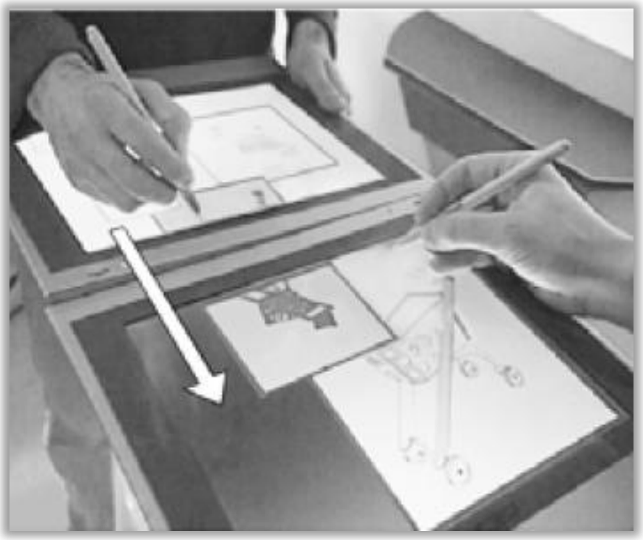


Figure 22. The ConnecTable Interface

2.3 Tabletop Interfaces Using Infra-Red Tracking Techniques

The Elements system [49] is a virtual tabletop environment for the assessment of upper limb function in Traumatic Brain Injury. The primary goal of its design was to extract measurements regarding a patient's performance at a functional and kinematical level. The primary surface of the system is a large horizontal LCD panel, across which simple objects can be placed and moved. Each of the objects has a Nintendo Wii Remote controller placed in its interior. This allows a clinician to measure the patient's ability to manipulate objects because the Wii Remote can sense the rotation, orientation and acceleration of the hand via an embedded three-axis accelerometer. The Wii Remote is also equipped with an Infra-Red sensor that allows efficient position tracking relative to a stable Infra-Red light source. A Nintendo Wii Sensor Bar is used as the Infra-Red source and it is placed above the horizontal panel, along with a stereo camera that is used for object tracking using computer vision. Distinct markers are placed on top of each object and they are used as a point of reference for the camera. This way the camera is able to track each object's position and orientation on the surface. A vertical LCD panel is placed on one side of the horizontal surface, across the user, to display performance information and graphical summaries of a patient's progress over time. The application that runs on the system consists of several tasks. The patients are instructed to move cylindrical-shaped objects to specific locations displayed on the surface. The movements must be completed as quickly as possible and, when objects are placed in the centre of the target areas, the performance results are displayed in graphical form on the vertical information panel.

Figure 23. The Elements System

PlayAnywhere [50] is a computer vision-based interactive tabletop system that was designed to explore interaction capabilities based on new image processing techniques. The system's surface is a standard horizontal desk, on which information is displayed from a projector placed on one side of the desk. An Infra-Red LED illuminant and a camera with Infra-Red pass filter are attached on top of the projector to provide the vision tracking. Touching and hovering above the surface are detected by a technique that exploits the change in appearance of shadows as objects approach the surface. When users point their fingers on the surface, the camera captures images that are binarized to provide images that contain only shadows. Candidate finger positions are generated by finding the highest point on each of the distinct shadows found in the image. PlayAnywhere also contains a real paper tracking system. Physical white pages can be placed on the surface and, when they are tracked by the camera, information is projected onto them. Pages can be moved around the table and rotated up to 180 degrees, while the respective projected information is translated to 'follow' the paper at any position and angle. The software is also able to recognize multiple overlapping pages as well as project different videos on each page without interference on the overlapping areas. Another technique developed for PlayAnywhere is the tracking of one or more parts of a hand when placed over a virtual object. The system continuously analyzes changes in those tracked points to extract information on the movement of the hand, so that simple motions or gestures can be used for interaction on the surface such as to move, rotate and zoom in and out of objects.

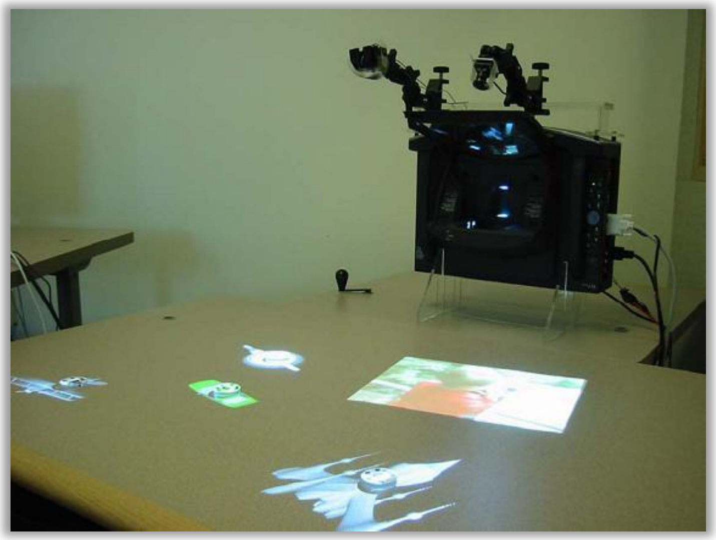


Figure 24. The PlayAnywhere System

Shared Design Space [51] is a collaborative tabletop environment intended for brainstorming and discussion meetings. The system consists of four ceiling projectors displaying information on a table surface and a wall-mounted projector showing data on an adjacent wall. Users can sit around the table and interact with it using digital pens. The key feature of the system is that users can write and draw digital information on virtual pieces of paper as well as physical paper pages. The digital pens used have embedded Infra-Red cameras that track movement by recognizing patterns of small dots printed on the pages. Both the digital papers and the pens have unique IDs, thus making it easy to recognize different documents as well as distinguish users from each other. Physical paper pages are recognized by special markers that are printed on each piece of paper. The coordinates of the physical pages are calculated relative to their markers, so that information can then be projected onto them. When a user taps on a digital object on the table, the system binds that information to the user's pen. When the user places the pen over a real paper, the bound data is translated and projected on the paper. All objects projected on the table can be moved around, scaled and rotated. Users are also able to move notes from the table to the wall by tapping with their pen on special control elements projected on the table. Once documents are projected on the wall, users can interact with them by using their hands. An Infra-Red camera tracks the gestures on the wall surface, allowing for single-handed interaction to scroll through available sketches, as well as two-handed interaction that allows zooming in and out of notes depending on the changing of distance between hands.

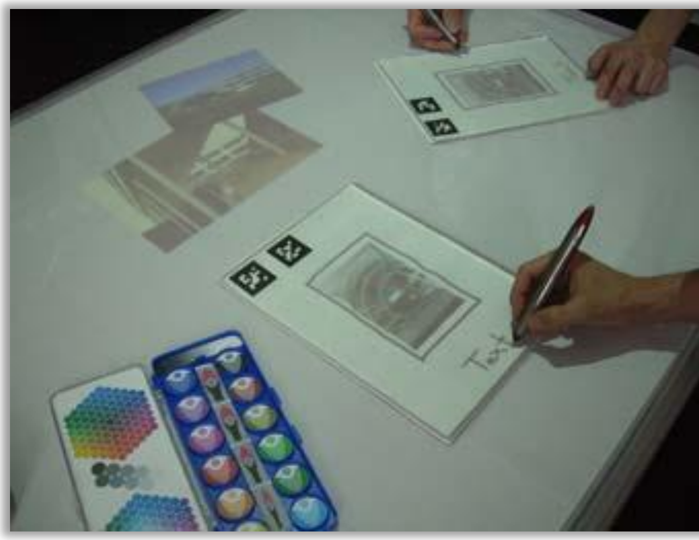


Figure 25. The Shared Design Space Interface

2.4 Tabletop Interfaces Using RFID Technology

The interactive Table with Tangible and Traceable objects [52] is a tabletop platform based on radio frequency identification to provide users with a tangible interface to manipulate objects. The system consists of a top-projected table that is arranged in three layers: the bottom layer is the table's physical surface, on top of which an 8 x 8 grid of RFID antennas is placed to form the middle layer, while the top layer is a surface made of non-conductive material used to project the information on. Users are able to place and move physical objects equipped with RFID tags on the table, making it possible for the system to locate their positions through the antennas, so that various actions can be triggered to display related information on the surface. An example application was built for use with the interactive table, to test and explore its capabilities. The application, called Luminous Zone, allows users to place RFID equipped physical objects over virtual colored zones that are projected on the surface. There is a lighting zone that, when a user places an object over a color zone, makes use of LED lights that are embedded in the table to light up in the color that the object was placed on.

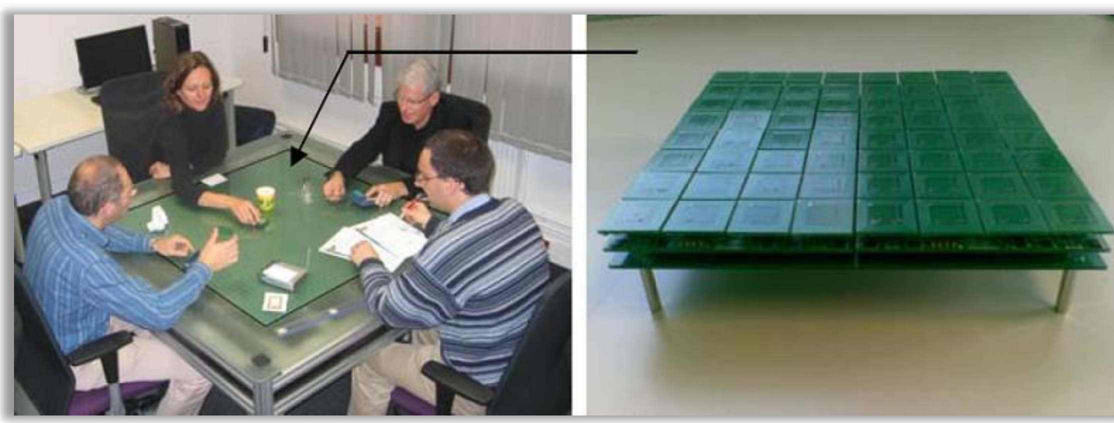


Figure 26. The Interactive Table with Tangible and Traceable Objects

RFIDice [53] are traditional six-sided dice prototypes built to explore the possibilities of using RFID augmented dice in tabletop board games. The approach followed was to add RFID tags to each inner side of a die and embed an RFID antenna in the table surface, so that, when the die rests over the antenna, the side facing down should be recognized to allow the system to infer the side facing up. RFID technology does not allow distance measurement for recognized tags, thus making it complicated to recognize the bottom side of a die solely because, when a die rests over the antenna, it is very possible that multiple RFID tags from adjacent sides will be recognized by the antenna as well. To minimize the chances of multiple tag recognition, a metallic foil was added to the center of a die to shield all tags other than the one at the bottom from the field of the antenna. A non-conductive spacer element was also added to prevent the metallic shielding from directly touching any of the tags, because the tags would then be disabled. Three prototype dice were built, varying in size, in order to explore successful random rolls recognition rate. Many experiments were driven, using metallic foils and spacer materials of varying thickness, to point out which combinations produce more accurate results.

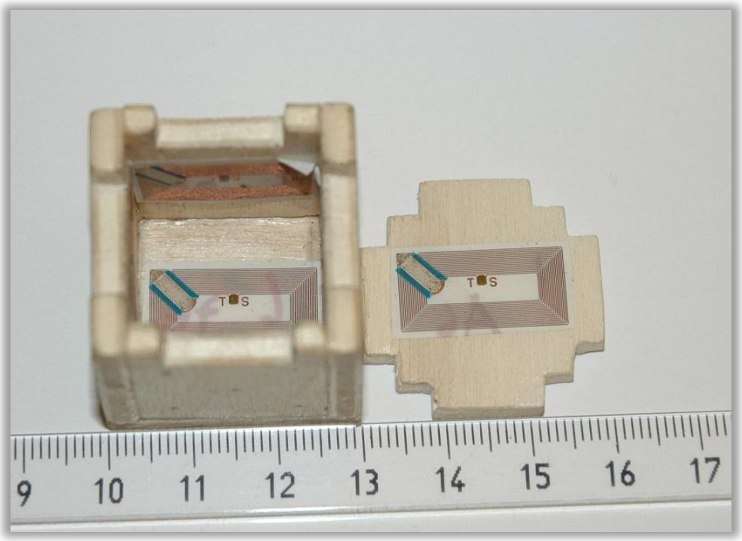


Figure 27. The RFIDice Prototype

Smart Playing Cards [54] is a tabletop card game augmented with RFID technology to provide players with several useful services. The primary goal of the system's design was to relieve players of computational tasks in order to stay focused on their goal. Five RFID antennas were placed underneath a table cover, one immediately in front of each one of four players and one in the center of the cover to be used as the playing area. A deck of cards was also altered by integrating an RFID tag in each one of the playing cards. Initially, the users must place their cards on the surface right in front of them to allow the system to associate each card with a player, thus making it possible to distinguish players while the game runs. The system keeps track of the game score, alerts players of wrong moves, as well as advises novice players of possible moves. All information is presented to the players through their personal mobile phones, as long as they support the Java Mobile Information Device Profile. A game server keeps track of the game state and users can connect to it at any time via Bluetooth to retrieve game statistics and scores.

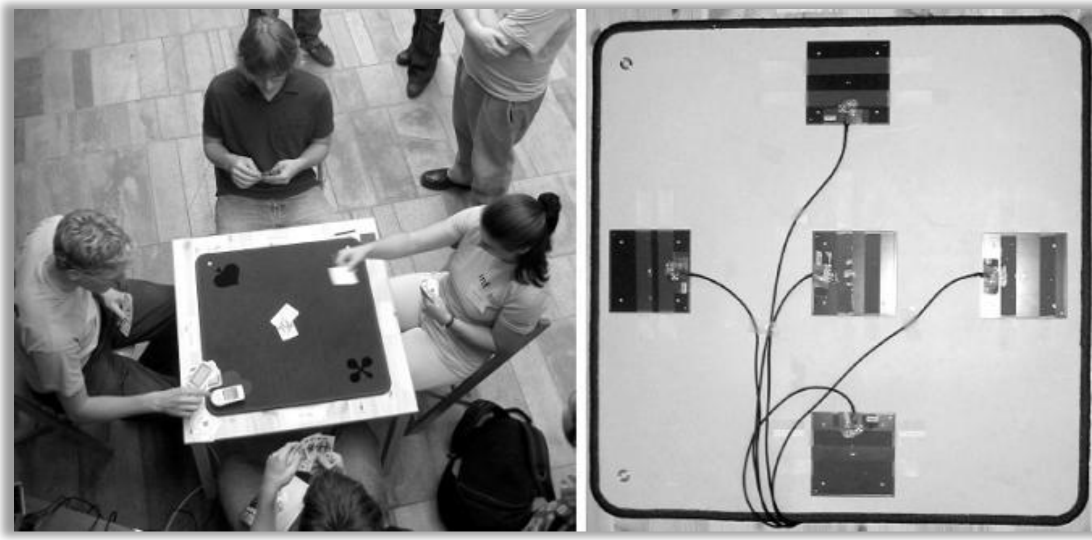


Figure 28. The Smart Playing Cards System

2.5 Discussion

As shown by the miscellaneous interfaces reported throughout the previous sections of this chapter, there is a great need for tabletop applications that strongly support collaboration in many social aspects of life. This need is demonstrated by the wide practice of collaborative interface development by researchers and scientists for many years now, leading to a wide variety of applications ranging from entertainment systems to shared workspaces for professionals.

In such interfaces, information viewing, manipulation and sharing is usually supported through multi-touch and point-sensing surfaces that allow users to interact with visualized data using their hands or other equipment, such as special pens. Many of the existing interfaces are built upon commercial multi-touch surfaces that free developers from the process of creating their own surfaces using custom hardware. Commercial surfaces are also accompanied by software toolkits that allow developers to create applications with integrated touch and gesture sensing capabilities, depending on the surface they are built to run on.

However, when it comes to collaborative group work, current surfaces have some major limitations. Widely used surfaces, such as DiamondTouch and SmartBoard, are intended for very small groups of users, thus presenting a restriction on the number of concurrent users working together. Moreover, none of the existing surfaces, with the exception of DiamondTouch, are able to distinguish users from each other, thus making it impossible to present users with personalized information.

Existing collaborative tabletop interfaces present information to the users in the form of custom data objects that are specifically designed for these applications. Documents, such as text and images, are usually represented by virtual pieces of paper or digital content blocks that users can manipulate in a fixed way defined by the interface that is used. This is a very significant drawback, since new data representation objects must be created for each type of document that is used in a collaborative interface. Additionally, none of the existing interfaces uses commonly used applications to view or edit digital documents. This is not consistent with the way most users are accustomed to handling digital information, and may even require additional training to get used to.

Furthermore, none of the existing collaborative interfaces supports real-time updated shared information. Current interfaces support information sharing by using static copies of information objects that are shared among the users. This means that shared information remains in the same state as long as it is used, without being updated to reflect possible changes.

The SMURF Table presented in this thesis has been designed in order to address the aforementioned gaps in the current state of the art. The system’s design will be presented in Chapter 3 and its implementation will be explained in Chapter 4.

3. Requirements and Design

This chapter describes the design of the **S**MART **M**ULTI-**U**SE**R** O**F**FICE (SMURF) Table system, which is targeted to support flexible collaboration of multiple users around a table in an ambient intelligence office environment. The chapter starts with a brief description of the developed applications, followed by a detailed description of the provided user interfaces and functionality. The chapter closes with a description of the equipment that is used to provide the hardware functionality for the whole system.

3.1 Multiple User Collaboration in a Smart Office Environment

The SMURF Table system has been developed in order to fulfill the following requirements:

- Provide a workspace to support collaboration among multiple co-located users
- Allow easy viewing, duplication and sharing of any type of documents
- Automatically assign a personal working area to each user, allowing them to dynamically reform their areas according to their working needs
- Present users with their personal data automatically, without requiring manual initialization or further setup
- Present information to users correctly, regardless of their position around the table
- Provide users with intuitive ways to manage their documents
- Provide independent interaction and input methods for each user
- Allow all users to focus on documents of public interest by providing a public area for document display
- Allow users to manage their documents remotely.

The SMURF Table application is a virtual workspace, organized in several working areas, that allows users sitting around a table to work simultaneously on personal and shared documents. Each user has their own personal identification card that can be placed on a working area to activate it as a personal workspace and load the user's owned documents. Personal documents are organized in a collection that the user can browse to select the documents that they want to open and manipulate. Users can select, move and manipulate documents by using special pens as pointing devices. Each user is also able to use a personal keyboard to enter and edit text in their documents. Whenever a user needs more space to work on many documents, the personal identification card can be placed over a non-occupied adjacent area to expand the personal workspace. Users can share information by dragging documents to each other's personal areas, resulting in the duplication of the respective documents. Each working area contains a briefcase and a trashcan tool for document management. When a user wants to own a new document, they can drag that document over the briefcase tool to store it in their personal collection. To close an open document, the user can drag it over the trashcan tool to free some space in the working area. When a document is no longer needed, the user can drag the respective element from the personal collection over the trashcan tool to permanently remove it from the collection. Depending on each user's position around the table, documents that are positioned within a working area are automatically re-oriented to fit the respective user's viewing angle. A public area at the center of the entire workspace is used as an area for public documents. When a user needs to show a document to other users around the table, they can move that document to the public area to make it appear in a large connected display. At any time, a user can leave the table by placing the personal identification card over their current working area to deactivate it.

In order to allow users to manage their documents remotely, the SMURF Document Manager has been developed. It is an application that provides users with the ability to use their personal computers or laptops to connect remotely to the SMURF Table system server. It has a simple user interface that allows users to transfer files from the SMURF Table server to their personal computers and vice versa. Users are also able to add new documents to their collection, in order to use them on the SMURF Table, as well as to delete existing documents from their collection. The SMURF Document Manager also adds a dimension of privacy into the system, by allowing users to use their personal computers or laptops as private workspaces.

3.2 SMURF Table

The design process of the SMURF Table was completed in several steps. The main goal of the design was to produce a simple user interface with intuitive elements, so that the target user group would not be limited due to their experience level. Any user can easily interact with the SMURF Table, regardless of prior experience in collaborative tabletop surfaces.

An initial brainstorming session was conducted in order to create a set of design ideas, along with possible alternative approaches, based on ideas gathered by three user interface designers and developers. The best approaches were chosen in a second brainstorming session, resulting in a final set of requirements, a general description of which was presented in the previous section.

In order to determine the desired functionality, interaction methods and the respective user interface elements, a task analysis was conducted that pointed out all the important features of the application's design. A first prototype was built and, after an iterative evaluation and improvement process, the final design of the application was elaborated.

Figure 29 shows the surface that comprises the entire working area of SMURF Table. This is the initial state of the application, when no users actually work on the table. The surface consists of eight smaller rectangular areas that are independent workspaces. In its currently implemented form, the application supports up to seven users, each using one of the areas along the three sides of the table, while the central area is used as a public space. As shown, the SMURF Table surface contains no special interface elements and users can use any of the available working areas as long as they are not occupied.

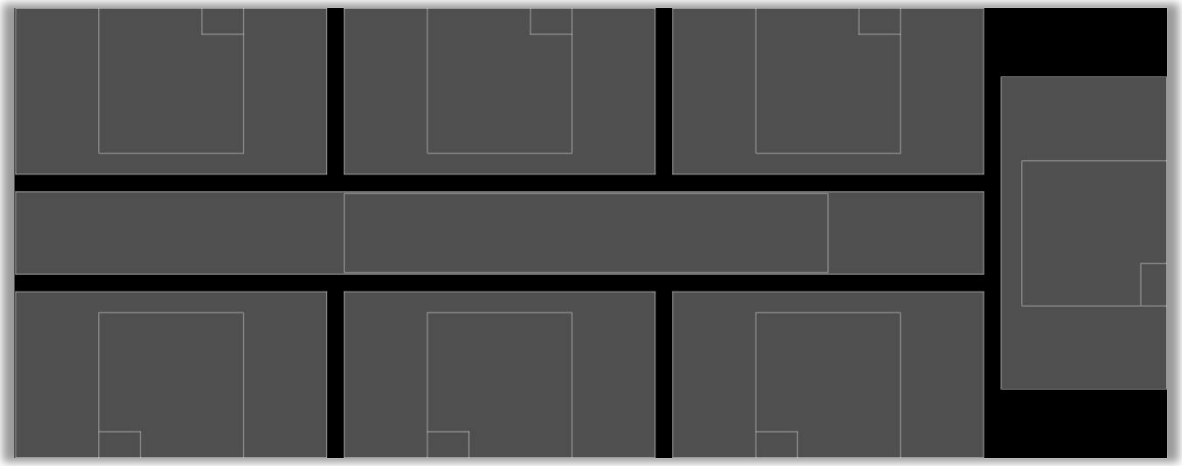


Figure 29. The SMURF Table surface in its initial state

3.2.1 Personal Working Areas

Figure 30 presents an unoccupied working area. Each area contains two rectangular sub-regions: a card slot and a central working region.

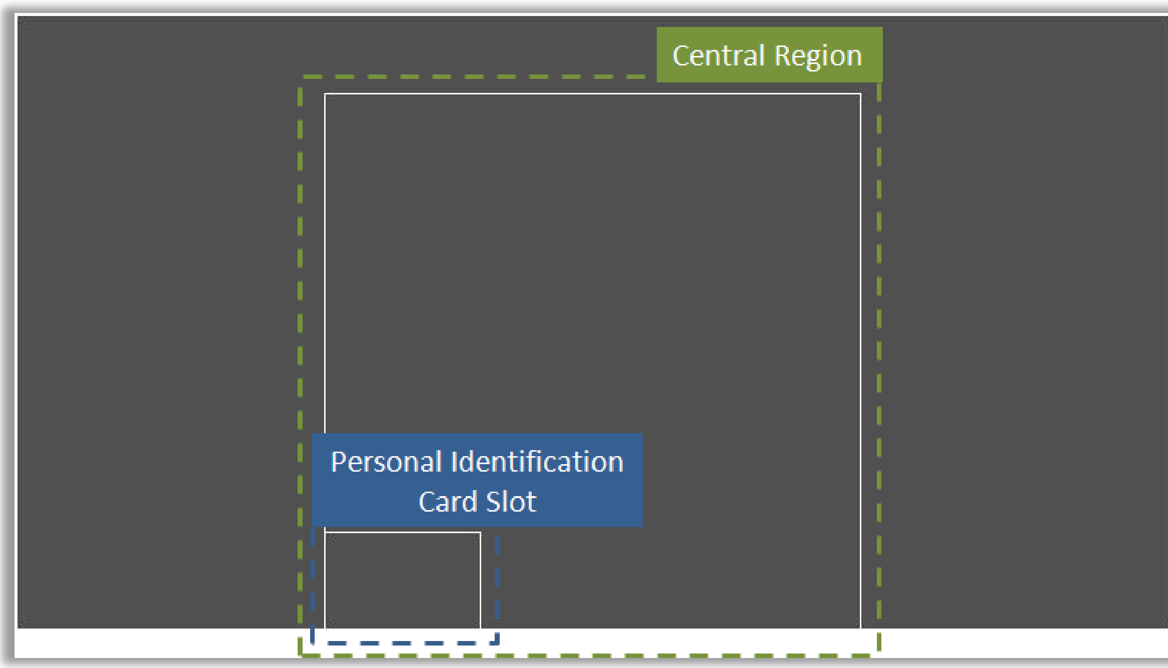


Figure 30. A personal working area

The personal identification card slot (see Figure 30) is a virtual slot where a user can place a personal card on, so that the table can recognize the card holder. The shape and size of a slot are identical to those of the users' cards. Once a card is identified by the system, the respective working area is considered as occupied and is activated. After successful activation, the user can remove the card from the slot. If a working area is activated and the user assigned to that area places the card over the card slot, the area is deactivated and the user can leave the table. The central region is not used when the respective working area is deactivated.

3.2.2 Activated Working Area

Figure 31 depicts the initial state of an activated personal working area. When an area is activated, the user's documents are automatically loaded and some interface elements appear. These elements include a thumbnails region, a briefcase tool, a 'close' tool, a trashcan tool and the activated central working region. The interface elements of the working area are described in detail below.

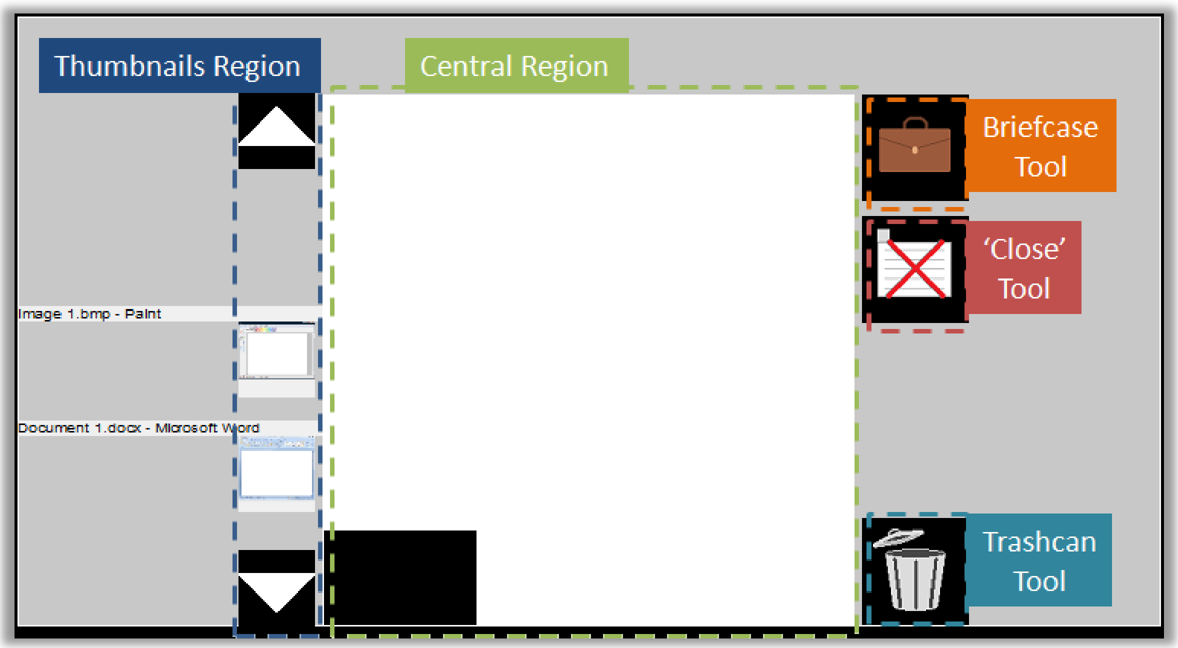


Figure 31. An activated working area

Users can interact with the various elements by using special pens as pointing devices. A pen must be pointed towards the table surface in order to interact with it, similar to the way a common pen is used, but without necessarily touching the table. The special pens are equipped with a button that activates them. When the pen button is pressed down, a red spot appears on the surface to show to the user the exact position being pointed to. The detailed functionality of these pens is described in section 3.4.

Thumbnails Region

The thumbnails region (see Figure 31) is a collection of the user's owned documents. Each document is represented in the collection by a downsized thumbnail. Each thumbnail is accompanied by a textbox that displays the actual title of the document it represents, so that the user can easily distinguish documents from each other. The document thumbnails are organized in pages, each containing up to three thumbnails. There are two arrow tools, one above and one below the thumbnails area, that allow users to browse through thumbnail pages, up and down respectively. The method in which used to organize thumbnails in the collection's pages is described later in this section, along with the addition of new thumbnails to a collection and the removal of existing ones from the collections they belong to.

Central Region

The central region (see Figure 31) acts as the space where new documents are opened. The user can select a thumbnail from their personal thumbnails collection, drag and drop it over the central region to open it.

When the user selects and drags a thumbnail, a semi-transparent copy of the thumbnail appears (see Figure 32) and follows the movement of the pen while it is being dragged over the surface, until the user releases it. These thumbnail copies are called ghost thumbnails, because of their semi-transparent nature, as well as because they disappear as soon as they are released.

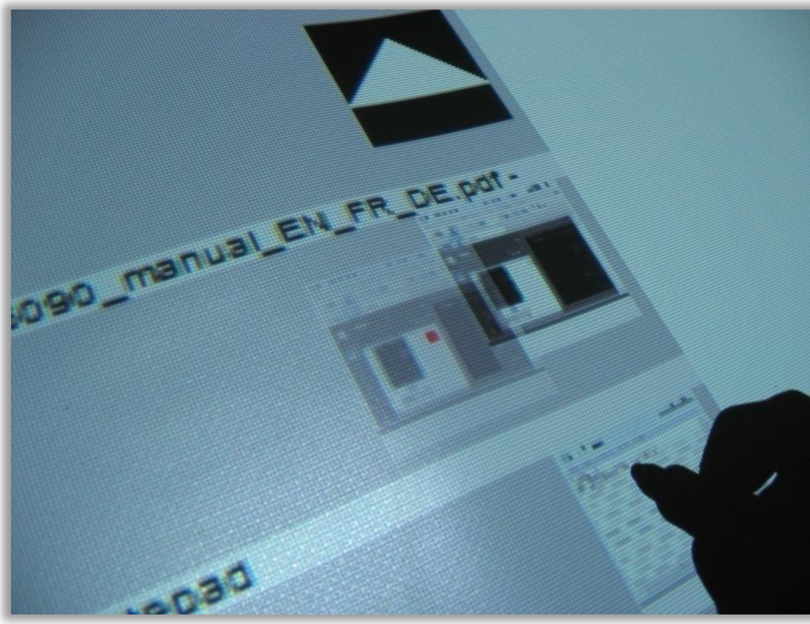


Figure 32. A ghost thumbnail

When a ghost thumbnail is dropped over the central region of a working area, the respective document appears. Figure 33 shows how a document is represented on the table. Basically, each document on SMURF Table consists of two parts, its main interaction area and a special square handle on its top-left corner.

The main interaction area of a document is actually an application window that contains the respective document. Most users are generally accustomed to using specific applications to view specific types of documents, for example viewing web pages in the Mozilla Firefox[1] web browser

[1] http://www.mozilla.com/firefox/

or opening PDF documents with the Adobe Reader[2]. One of the primary goals set when designing SMURF Table was to provide users with the ability to view documents using applications they are familiar with.

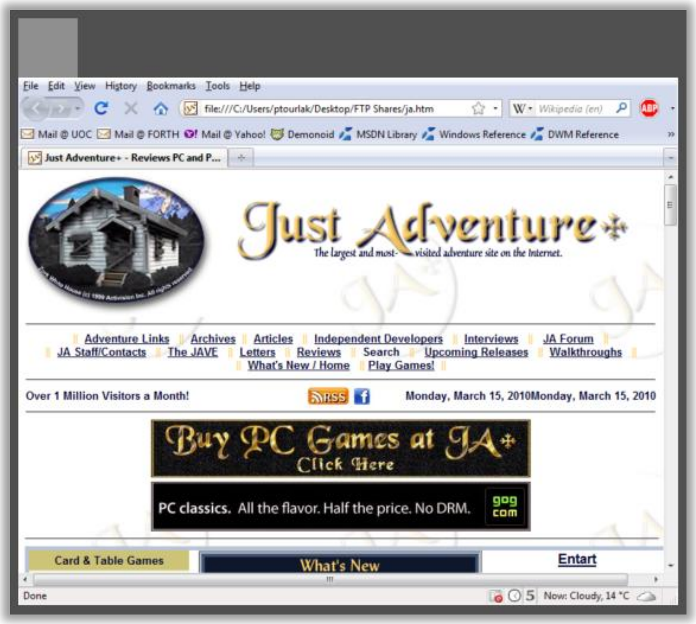


Figure 33. The form of a document on the SMURF Table

The handle on the top-left corner of a document is used for various purposes. When a user wants to move a document around, they can select the document's handle, drag it and drop it at the desired destination point. While a handle is being dragged around, the respective document window is dragged simultaneously in the same movement pattern. The document handles have several other uses that are pointed out throughout the description of interface elements in this section.

Briefcase Tool

The briefcase tool allows the user to store new documents in their personal collection. This happens when a user gets a document from another user. When the user wants to have a copy of a document they do not already own, they can drag the document by its handle and drop it over the briefcase (see Figure 34). When the document handle is released over the briefcase, a copy of that document is added to the user's personal collection. A new thumbnail is created in the user's thumbnails region, along with the new document's title in a textbox. If there is empty space in a page in the thumbnails collection, the new thumbnail will be inserted there. If not, a new thumbnails page will be created and the new thumbnail will occupy the first slot in that page. The order in which thumbnails occupy the available slots in a page is bottom-to-top. The

[2] http://www.adobe.com/products/reader/

first thumbnail in a page will occupy the bottom slot, the second thumbnail will be placed in the middle, while the last one will be positioned in the top slot.

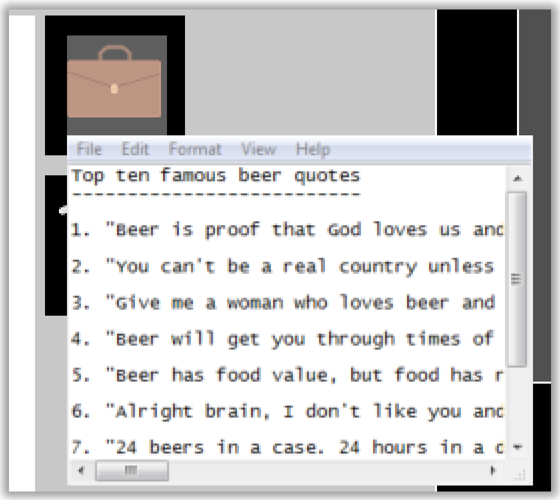


Figure 34. Storing a new document by dropping it on the briefcase tool

'Close' Tool

The 'close' tool allows users to close any of their currently open documents. When a user no longer needs an open document, they can drag that document by its handle and drop it over the 'close' tool. When the document handle is released over the 'close' tool (see Figure 35), the document along with its handle are closed, freeing some space in the working area.

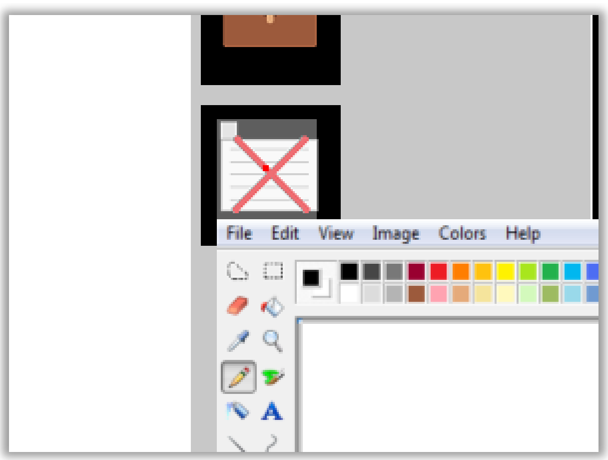


Figure 35. Closing an open document by dropping it on the 'close' tool

Trashcan Tool

To remove a document permanently from their personal collection, the user can drag the respective document thumbnail over the trashcan. The user can select the document thumbnail from the thumbnails region, drag the respective ghost thumbnail and drop it over the trashcan (see Figure 36). This way, the document is permanently removed from the user's personal collection. When a document is deleted from a user's collection, the respective thumbnail is also removed from the user's thumbnails region, leaving an empty slot in the respective thumbnails page. If the empty slot is any other but the top one, the remaining thumbnails are reorganized to occupy the slots towards the bottom of the respective page. If the removed thumbnail was the only one in a thumbnails page, that page is removed from the thumbnails collection.

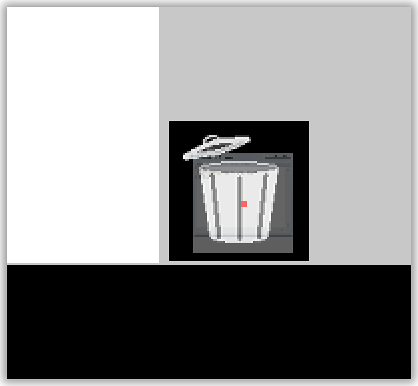


Figure 36. Deleting a document by dropping its thumbnail on the trashcan tool

It is obvious that the briefcase and trashcan tools combined are actually the interface that allows users to manage their personal document collections. Since they provide the ability to store new documents and delete existing ones, they act like a 'Save / Delete Document' feature for document collections.

3.2.3 Document Duplication and Sharing

Users that are gathered around a table for a meeting often need to exchange information and share ideas with each other. SMURF Table supports collaborative information handling by allowing users to duplicate documents and share them with one or more users around the table.

A user can select a document, drag it and drop it at any point within another user's working area. This action results in the duplication of that document, creating a new window containing that document, along with its own handle. Immediately after the duplication, the first user's document automatically returns to the user's working area so that the user won't need to drag it back. Figure 37 shows two users, each having a copy of the same document. The only difference between them, at this point, is that one user actually owns the document, while the other user simply has an exact copy of it in front them.

SMURF Table supports document sharing through multiple ownership of documents. Receiving a copy of a document does not imply ownership. In order to own a document, the user must store it by using the briefcase tool as described previously. In general, the owners of a document are all the users who have that document stored in their collections.

When two or more users have an open copy of the same document, they are able to share ideas and information directly. An important feature of the SMURF Table sharing mechanism is that shared documents are updated in real time, regardless of how many copies of the same document are open on the table. When a user types some text in a document, all users who have a copy of that document actually see the text being typed. At any time, any user can use a shared document in order to share information with the rest of the document's owners.

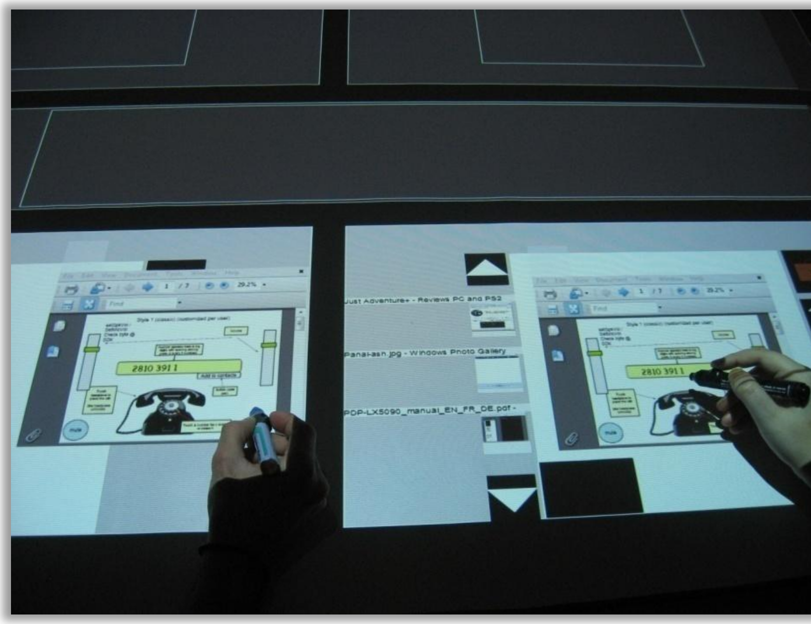


Figure 37. Two users sharing copies of the same document

When a user no longer needs a document they own, they can remove it from their collection by using the trashcan tool as described previously. In that case, if the document has multiple owners, it is not entirely removed from the system, but the user that removed it from their personal collection is no longer considered as an owner of that document. If the removed document has no other owners, then it is completely removed from the system.

3.2.4 Document Orientation

When working around the table, users can use any of the available working areas along the three sides of the table. Depending on the side a user chooses to sit at, their perspective relative to the surface varies. Nevertheless, each user needs the information presented to them to fit their perspective. SMURF Table supports adaptive orientation of all documents and interface elements, according to their position on the surface.

Figure 38 shows the way different working areas are displayed on the table surface. Depending on each working area's position, their respective interface elements are oriented to fit their users' viewing angle. If the viewing angle of the bottom working area shown in Figure 38 is considered as 0°, the interface elements of the right and top areas are presented at 90° and 180° respectively.

Figure 38. Three working areas in different angles

In the same manner, all documents on the table are oriented according to the working area they are positioned in. Figure 39 shows three copies of the same document, each one in a different working area, with different perspective.

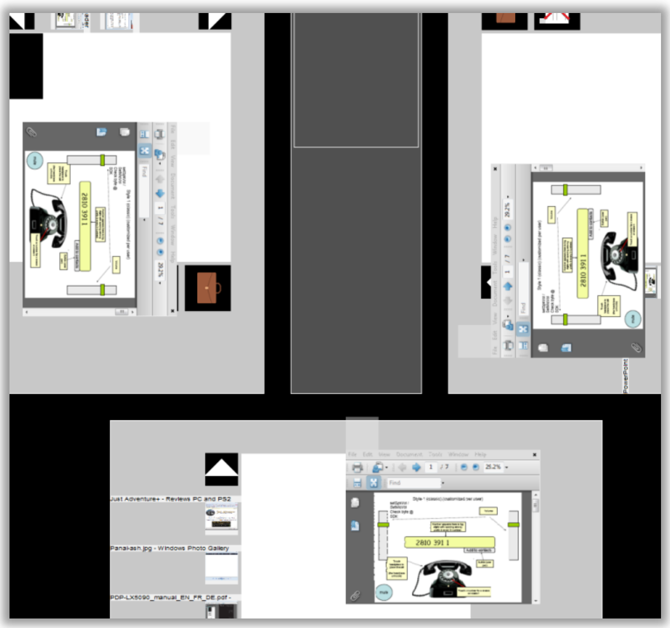


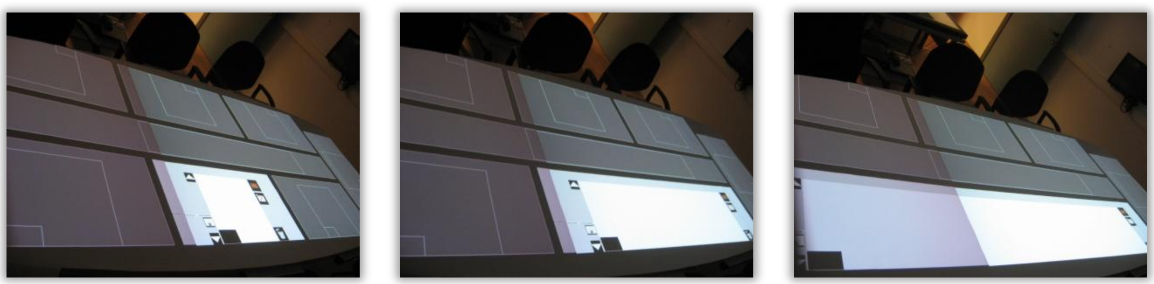
Figure 39. Three copies of the same document in different angles

SMURF Table also supports dynamic reorientation of documents. When a user moves a document around the table by dragging its handle, the angle of the document automatically adapts to the perspective of the working area it is currently over. The rotation of documents is also relative to their respective handles. When a document is automatically rotated, it is repositioned so that the handle remains on its top-left corner regardless of the document's orientation. This feature is very useful, because besides allowing documents to keep their integrity and uniformity anywhere on the surface, also prevents users from losing grip of the documents when moving them around the table.

3.2.5 Dynamic Area Reforming

When a user needs to work on several documents at the same time, it is possible that their personal working area will be entirely covered by open documents. In that case, the user may have to move documents around at times in order to focus on the desired documents, or even close documents that may be reopened later in order to free some working space. SMURF Table overcomes this problem by allowing users to expand their personal space when more space is available.

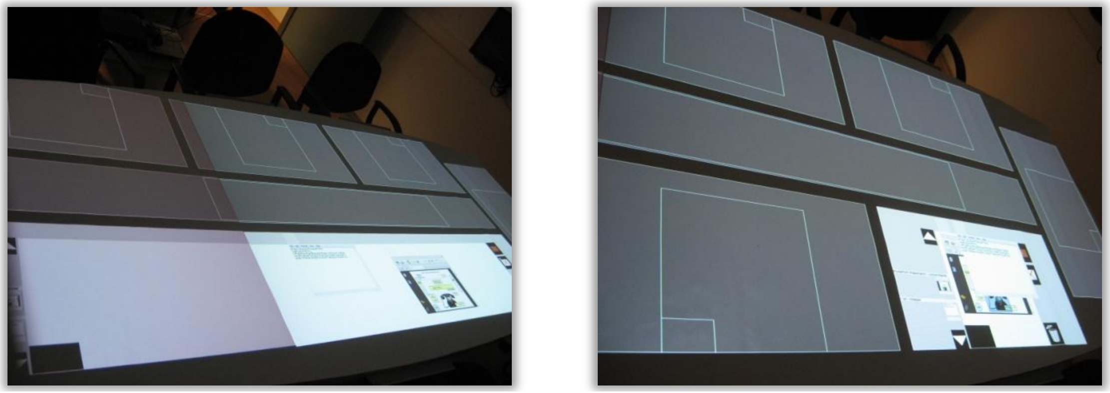
Basically, when a user occupies a working area whose adjacent areas are unoccupied, they have the ability to use their personal identification card in order to occupy two or even three working areas. Figure 40 shows how three distinct working areas can be combined in order to form a larger single working area. In the depicted example, the user initially occupies the middle area on one side of the table. Then the card is placed over the area at the right of the current working area. The user's personal area is now stretched and its surface entirely covers the space that previously consisted of two distinct working areas, including the empty space between their borders. Finally, the user places the card over the working area at the left of the current workspace, resulting in a personal area that occupies the whole side of the table.



(a) (b) (c)

Figure 40. Dynamic area expansion. The user initially occupies the middle area (a), extends it to the rightmost area (b), and finally adds the leftmost area (c)

Users are also able to shrink their personal working areas, in case they don't need the extra space or when another user needs to work on the table and there is no working area available. Figure 41 shows how a user is able to shrink the occupied personal space to a single working area. Initially, the user occupies the space consisting of three distinct working areas. Placing the card over the specific working area to be occupied, the user can shrink the occupied workspace to that area. In the depicted example, the user chooses to shrink the workspace to the working area at the right part of the currently occupied workspace. By placing the card over that area, their entire personal space is automatically set to the selected single working area, while all the user's documents that were spread throughout their previous workspace are automatically repositioned in the reduced personal space, thus the user does not need to drag each one of them back to the smaller area.



(a) (b)

Figure 41. Dynamic area shrinking. The user initially occupies three working areas (a), then shrinks the personal working area to the rightmost area (b)

When a working area is expanded or shrunk, its interface elements are repositioned in order to retain its standard form. As depicted in the previous two examples, the area's central region is reformed according to the new working area size. In each case, the area's thumbnails region is moved so that it always stays at the left of the area's central region, while the briefcase, 'close' and trashcan tools are moved to retain their position at the right of the central region.

There is no particular order in which the user has to combine distinct areas when they need to expand their personal workspace. The only requirements in this case are that the selected areas must be adjacent to the user's current personal area and they must be unoccupied. In the case of area shrinking, the only requirement is that the destination working area must be part of the user's current larger area and not another unoccupied area.

In the current implementation, area reforming is supported for working areas that reside on the two large sides of the table, while the area at the head of the table cannot be reformed due to orientation matters in combination with its adjacent areas. Additionally, area combination is not supported for areas that are positioned across each other, because the resulting area would interfere with the public space at the centre of the table.

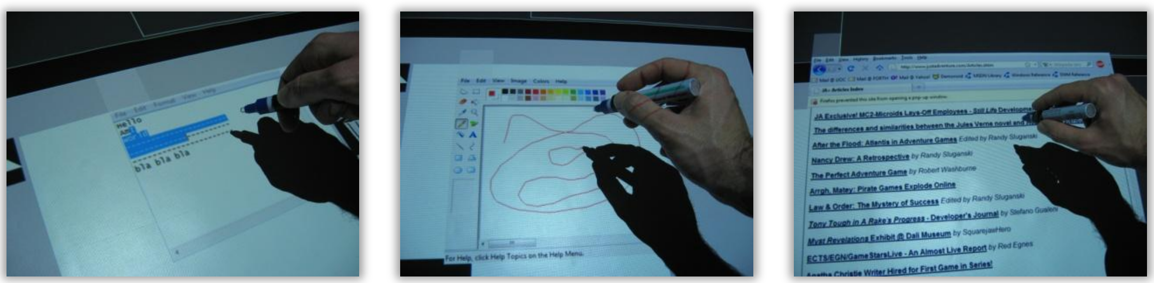
3.2.6 Independent User Input

SMURF Table has been developed in order to provide a meeting space for co-located users that work together and share information. Regardless of the number of users that sit around the table, each one of them needs their own way to interact with their documents, independently from the other users. Independent user input is supported by providing each user with the means to manipulate their documents without interrupting other users' work.

In the current implementation of SMURF Table, two types of user input are supported: a pointing mechanism for virtual object interaction, as well as text input for inserting and editing text in virtual documents.

Each user has a special pen that is used as a pointing device. Users are able to select document thumbnails, focus on specific documents or browse their thumbnails collection just by pointing with their pen. Additionally, they can drag thumbnails or documents over the surface by using their pen and moving it to the desired location.

In addition to thumbnail and document positioning, a pen can also be used to interact with a document itself. Depending on the application that is used to open a document type, a pen can be used to interact with its interface elements in the document window. Figure 42.a shows that a pen can be used to select some text in a text document. In Figure 42.b, the user draws on an empty canvas, by selecting a color from the application's palette and moving the pen over the white space to paint. Figure 42.c shows that a user can also browse the internet by using their pen to follow available links in web pages. In general, depending on the applications that are assigned to open specific document types, a pen can be used to interact with the document content itself, as well as with the available interface elements, such as scrollbars, buttons and textboxes.



(a) (b) (c)

Figure 42. Interaction with the pen. (a) Selecting text in a document, (b) Drawing, (c) Following hyperlinks

Users also have the ability to enter and edit text in their documents by using independent keyboards. Each user can have a personal keyboard, which can be assigned to the user's personal working area. Each keyboard has its own identification token that can be recognized by the system. The user must place the keyboard over the surface of the personal area so that the table can recognize the keyboard device and assign it to that user. After device recognition, the user is able to enter, select and edit information in their documents. Figure 43 shows a user typing with their personal keyboard to enter text in a web search engine.

Both ways of input are supported for multiple users that work simultaneously without affecting each other's work. Each user can use a single keyboard, and all implemented pointing interactions require a single pen from each user. In its current implementation, SMURF Table supports up to seven concurrent users, but is able to recognize a large number of simultaneous input pens and keyboards without problems.

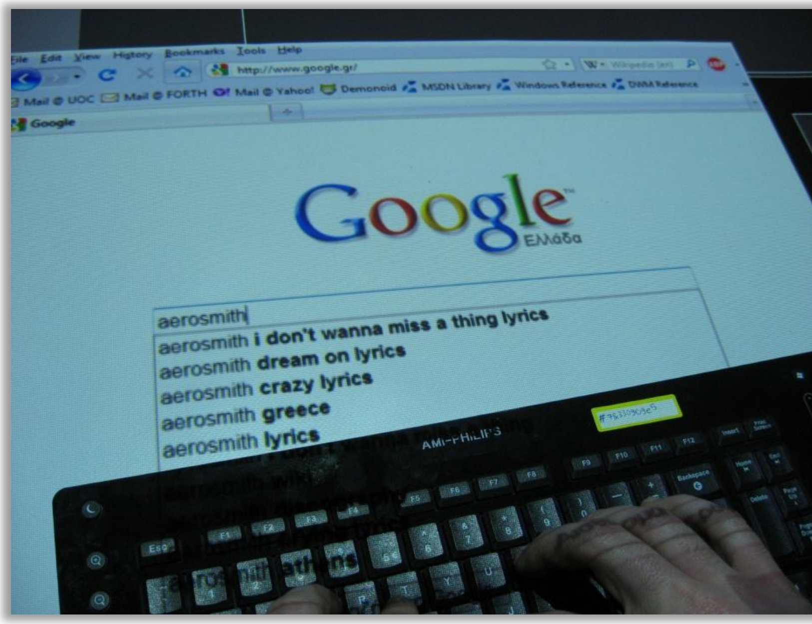


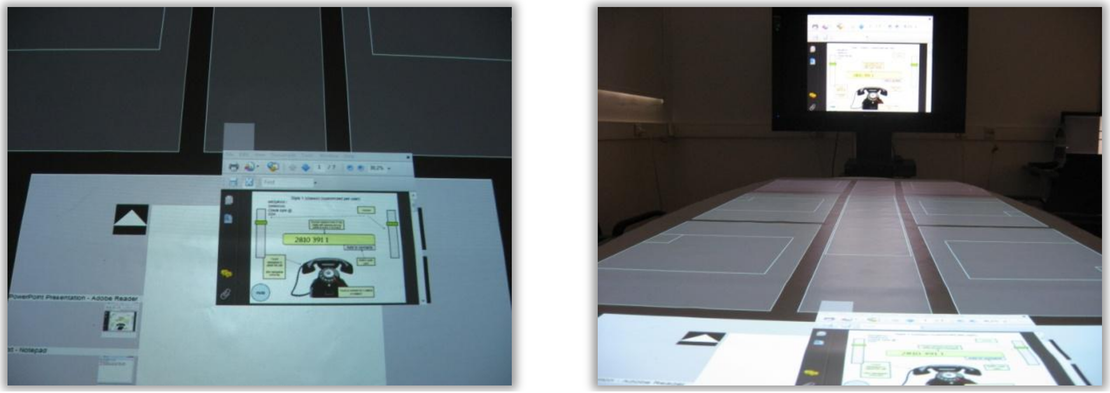
Figure 43. Using a personal keyboard to enter text

3.2.7 Public Working Area

Apart from the seven personal working areas, the SMURF Table surface also contains a public working area represented by a rectangular region in the space across the center of the table. This area is used for public display of any document. When a user needs to show a document to all users around the table, it would not be practical to drag it towards each of the other users separately to create copies of it. Instead, the user can drag the respective document and release its handle at any point within the public working area. This action creates a copy of that document that is presented on a large secondary display next to the table. The secondary display is large enough so that information is legible from an acceptable distance, while its position allows information to be easily viewable by anyone around the table, since it is placed across the side of the table that contains no working areas.

Figure 44 shows a user dropping a document's handle in the public working area and a large copy of that document appearing on the secondary display. When a document is presented in the secondary display and another document is dropped within the public working area, a large copy of the latter is created on the secondary display overlapping the former. In general, the order in which documents appear on the secondary display depends on the order in which the respective document handles are dropped within the public area.

Whenever a user no longer needs to publicly present a document, they can drag the respective document from the public area back to their personal area. This action will remove the copy of that document from the secondary display. In general, a document copy remains on the secondary display as long as the respective document handle is found within the public working area.



(a) (b)

Figure 44. Public display of documents. The user drops a document's handle within the public area (a), a large copy of that document automatically appears on the secondary display (b)

3.2.8 Working Area Deactivation

When a user no longer needs to work on the table, their personal working area can be cleared using their personal identification card. The user can place the card over the card slot of the current area to deactivate it. When an area is deactivated, it returns to its initial state and the user can pick up the card and leave the table. The interface elements of the deactivated area disappear and the space is then considered as unoccupied. When returning to its initial state, a working area also takes care of closing any documents its most recent user may have left open, no matter where they may be located on the table. This feature is very useful, since it allows the user to leave the table just by placing their card on the surface, without having to manually close any of their open documents.

3.3 SMURF Document Manager

The SMURF Document Manager is an application that has been developed in order to allow the users of SMURF Table to manage their documents through any other personal computer or laptop. It is a necessary complement to SMURF Table, though it does not require the SMURF Table application to be running in order to be used.

When a user opens the SMURF Document Manager application, a simple login window is presented. Figure 45 shows the login dialogue. Basically, the user only needs to enter username and password. This means that before using SMURF Table, the user must have obtained a unique username and password that are registered into the SMURF Table system by an administrator. The host address and database fields are automatically filled in by default and the user does not need to change them, except if the SMURF Table server system configuration changes. A detailed description of how the SMURF Document Manager communicates with the SMURF Table server is presented in Chapter 4. When the user has entered all the required information, they can click on the 'Connect' button to proceed.

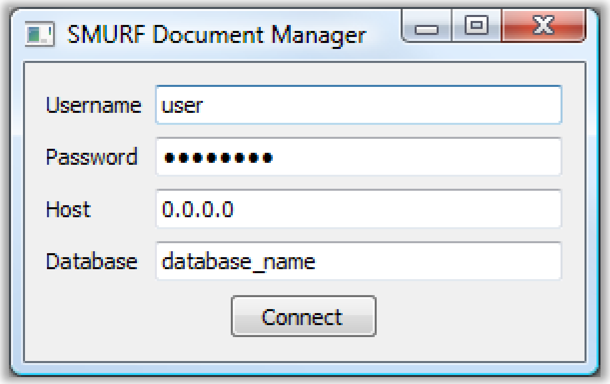


Figure 45. The SMURF Document Manager login dialog

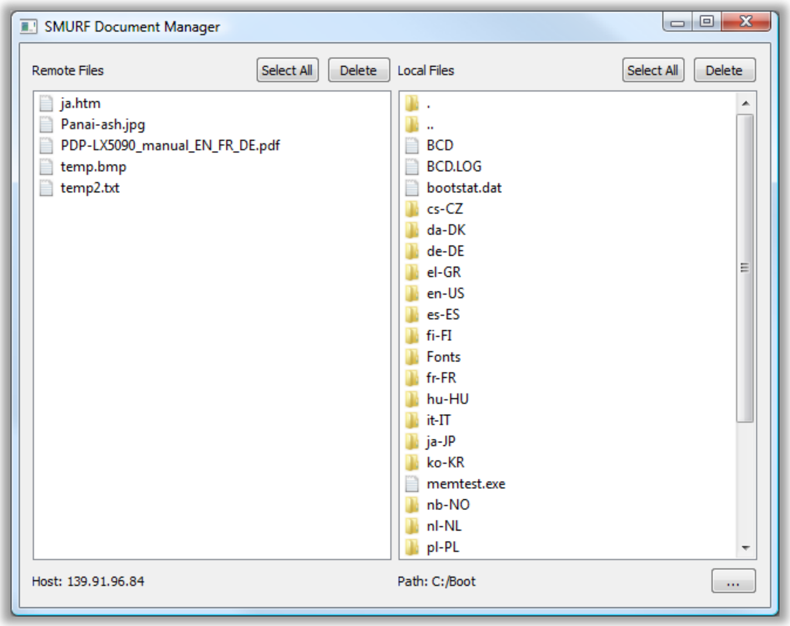


Figure 46. The SMURF Document Manager main screen

The user interface of the SMURF Document Manager consists of a single screen and is rather easy to interact with. Figure 46 shows the application's main window that provides everything the user needs to remotely manage their documents. The interface consists of two basic parts: the remote files pane and the local files pane.

3.3.1 Remote Files Pane

The remote files pane occupies the left half of the application's window. Above its top-left corner there is a label that informs the user that this part of the interface refers to their remote documents. The pane itself presents the user with a list of the documents they currently own on the SMURF Table. Each document is presented with its own icon and the respective file name. The user can click on any of the documents in the pane to select it and can also select multiple documents. Above the top-right corner of the pane there are two buttons. The 'Select All' button allows the user to select all of their remote documents with a single click, while the 'Delete' button can be clicked to delete the currently selected documents from the SMURF Table server. Below the bottom-left corner of the pane there is a label displaying the host address of the server where the user's documents are stored.

3.3.2 Local Files Pane

The right half part of the application's interface is occupied by the local files pane, as denoted by the label above its top-left corner. The pane contains a list of the user's local files and folders. By default, the initial directory displayed is the root directory of the user's local file system. Folders are presented with an icon different from the one that is used for files, so that the user can easily distinguish them. Similarly to the remote files pane, the user can click on any item to select it, as well as select multiple items. In order to browse the local file system, the user can double-click on any folder icon to view its contents in the pane. Above the top-right corner of the pane there are two buttons. Similarly to the remote files pane, the 'Select All' button allows the user to select all local files that are currently displayed in the pane, while the 'Delete' button is used to delete the currently selected files. The only difference with the 'Select All' function is that it filters the contents of the local files pane to select all the displayed files, excluding the folders. This feature is useful because users are not allowed to transfer folders to the SMURF Table, but distinct files. It also prevents the user from having to manually select each of the files one by one. Below the bottom-left corner of the pane there is a label displaying the full local path the user is currently browsing, which is updated each time the path changes. The local files pane also contains an extra browsing button below its bottom-right corner. When the user clicks on this button, a folder browsing dialog appears (see Figure 47) that allows the user to select a local folder using more advanced options, such as by typing a folder name or by using navigation buttons to point to previous, next and parent folders.

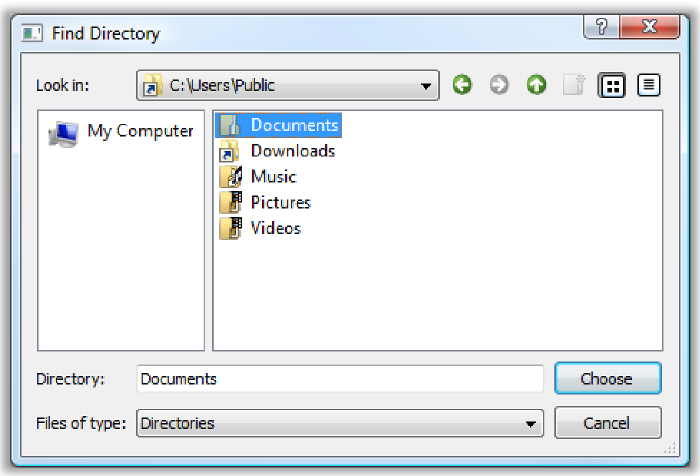


Figure 47. The folder browsing dialog

3.3.3 Document Downloading

The SMURF Document Manager allows users to download documents that they own on the SMURF Table to their personal computers. This way, users have the opportunity to update their documents regardless of their current location, as well as work on ideas that they shared with the rest of the users around the table.

The user has two ways to download files from the SMURF Table server to a local computer. To download a single file, the user can locate that file in the remote files pane and double-click on it to have it transferred to the local storage. The downloaded file will be stored in the folder that is currently selected in the local files pane, so the user must select the desired destination folder prior to downloading the file.

Additionally, the user can select one or more files from the remote files pane, drag them with the mouse and drop them on the local files pane. This will result in downloading the selected files and storing them in the folder that is currently selected in the local files pane. This allows the user to download multiple files at once without having to download them one by one.

Regardless of how files are downloaded, the application always informs the user of the downloading progress by displaying a progress bar for the file transfer process. Figure 48 depicts the progress bar dialog presented to the user during a downloading process. When the file transfer is finished, the local files pane is automatically refreshed, so that the user is ensured that the transfer was successful. At any time, the user is able to interrupt the downloading process by clicking on the 'Cancel' button provided by the progress dialogue.

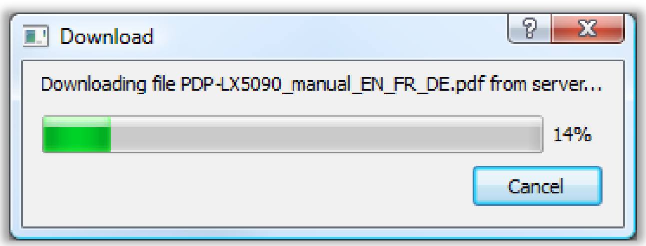


Figure 48. The file downloading progress dialog

3.3.4 Document Uploading

The SMURF Document Manager also allows users to upload files from their local storage to the SMURF Table server in order to use them when collaborating with other users around the table. The file uploading feature is very important, since it is the only way users have to add new documents to their collections on the table. Additionally, it allows users to remotely renew their documents by uploading updated versions while located elsewhere.

There are three different ways in which a user can upload files to the SMURF Table server. In order to upload a single file, the user first has to locate that file in the local files pane by browsing through the local file system in the ways described previously. When the file is located, the user can double-click on it, and it will be automatically uploaded to the remote server.

The user is also able to select one or more files from the local files pane, drag them with the mouse and drop them on the remote files pane. This way, all selected files will be automatically sent to the remote server.

Additionally, the user can transfer one or more files from their operating system's environment directly though the SMURF Document Manager. The user can select the desired files by selecting the respective file icons from their operating system's environment, drag them into the application's window and drop them on the remote files pane. The selected files will be automatically uploaded to the SMURF Table server. This uploading method also implies that the user is able to upload files without using the local files pane at all.

Similarly to the file downloading process, no matter which way the user chooses to upload files, the application displays a progress bar dialogue. Again, the user is able to cancel an uploading process at any time by clicking on the 'Cancel' button provided by the progress bar dialogue. When an upload process is finished, the remote files pane is automatically refreshed, containing the new uploaded files, so that the user receives confirmation of the successful transfer.

When the user has finished transferring files to or from the SMURF Table server, they can close the SMURF Document Manager application by clicking on the window's 'Close' button. This is the button at the top-right corner of the window, which most users are familiar with when closing applications. When closing the SMURF Document Manager, the application automatically disconnects the user from the remote server.

3.4 Hardware

This Section describes the hardware technology used to support the developed collaborative interface. SMURF Table uses special equipment for the virtual surface display, user identification, user input and visual input tracking. The hardware setup of the developed system is presented in detail in the following subsections.

3.4.1 Main Computer System

The SMURF Table application has been developed and tested on a standard desktop computer, equipped with a dual core processor running at 3.0 GHz and 4 GB of RAM. A combination of two similar graphics cards, each supporting multiple display output, was used in order to provide flexibility at using various output devices. The installed operating system is the 32-bit version of Microsoft Windows Vista, updated with the Service Pack 1.

3.4.2 Working Surface

The physical surface is a large meeting table, around which seven users can sit and work without interfering with each other's personal space. The table's physical dimensions are 250cm length, 115cm width and 72cm height.

On the ceiling right above the table, two projectors were placed next to each other in order to project the virtual workspace on the table surface (see Figure 49). Each of the projectors is connected to a separate output on the computer's graphics card and their projected areas are combined sequentially to form a single large area. Both projectors are full high definition capable, meaning that each one is able to project the computer's video signal at a resolution of 1920 x 1080 pixels, also supported by the equipped graphics card. The high resolution was chosen because the virtual surface must cover the table surface entirely, with the best possible detail. Also, the high definition projection allows virtual documents to be as legible as possible on the table, regardless of their size.

Figure 49. The projectors above the table surface

The natural color of the wooden surface is not a suitable choice for the projection surface, since it does not allow the projected image to appear clearly. A plain white cover was placed on top of the table surface in order to achieve the best possible image and color contrast.

3.4.3 User Identification

As previously described, each user has a unique identification card that allows the system to recognize them. User recognition is achieved by using Radio Frequency Identification (RFID) technology.

Seven RFID antennas are placed underneath the surface of the table. The arrangement of the antennas is similar to the layout of the seven distinct working areas of the virtual workspace. Each antenna corresponds to a single working area and is placed right beneath its surface. The physical surface material is non-conductive, so RFID transponder cards are identified without problems. Additionally, in order to avoid interference between adjacent antenna fields, special attenuators were added to each antenna in order to reduce their field range.

The antennas are all connected to an antenna multiplexer that takes care of querying each antenna for identification data when needed. The multiplexer itself is connected to an RFID reader that temporarily stores and processes the data received from the antennas. The data processed by the RFID reader are then sent to the computer through an Ethernet connection.

Besides user identification through RFID transponder cards, SMURF Table also recognizes individual keyboards using RFID technology. Each keyboard used on the table has previously been augmented with a unique RFID tag sticker, so the antennas can recognize keyboards as well. When a user needs to use a keyboard, they can place their keyboard over their working area, that is over the respective antenna, and the system will recognize the unique keyboard and will assign it to the respective user.

3.4.4 User Input

SMURF Table provides users with two ways of input, pointing and text entry. Pointing is achieved by using custom pens and Infra-Red (IR) technology, while text entry is possible through wireless keyboards.

As reported previously, users can have their own keyboards in order to edit text in their documents. In the current implementation of SMURF Table, the keyboards used are wireless and are connected to the computer via Bluetooth. Up to seven keyboards can be used concurrently, since a single keyboard is assigned to each user, but a large number of keyboards can be recognized as independent devices, through USB hubs or directly to the computer's available USB ports.

Figure 50 shows a custom pen that is used on the table for pointing interactions. The body of the pen comes from a standard marker usually used for writing on whiteboards. Its contents are emptied and a simple custom IR mechanism is placed in its interior instead. A small IR led is used as the marker's tip, while a small button is installed on the outer part of the marker's body.

Figure 50. A custom IR pen used on the SMURF Table

When the user presses the button, the IR led is activated and emits IR light, while a red spot is created on the table surface to inform the user of the exact position they are pointing at. The user can tap the button once, hold it down and move the marker around, or release the button, while pointing towards the table. On SMURF Table, these actions are equivalent to clicking the left mouse button once, moving the mouse around while holding the left button down and releasing the left mouse button respectively. All pointing interactions with documents and interface elements are done though these three basic actions. The way custom pens are tracked by the system is explained in the following subsection.

3.4.5 IR Tracking

The custom pens that users are provided with emit IR light in the form of IR spots when pointed towards the table surface. These spots are tracked by an IR-sensitive camera that is placed on the ceiling, right above the table (see Figure 51). The camera can capture a large number of simultaneous IR spots and track their position and movement over the surface. A connection between the camera and the computer allows the tracked data to be transferred to the computer that takes care of interpreting each IR spot as a distinct user interaction point. Because the camera itself cannot visually cover the entire table surface, a wide-angle lens was applied to the camera to enable it to track IR spots all over the table.

Figure 51. The IR camera above the table surface

3.4.6 Public Information Display

Public display of documents is achieved by using a 50-inch LCD television that is connected to the main computer system running the SMURF Table application. The television is placed across the side of the table that contains no working areas, so that it can be viewed by any user regardless of their position around the table. The size of the secondary display was mainly chosen in order to allow information to be viewed from an acceptable distance, while the resolution supported is full high definition (1920 x 1080 pixels) allowing information to be as detailed and legible as possible.

The hardware setup of the SMURF Table system can be seen in Figure 52.

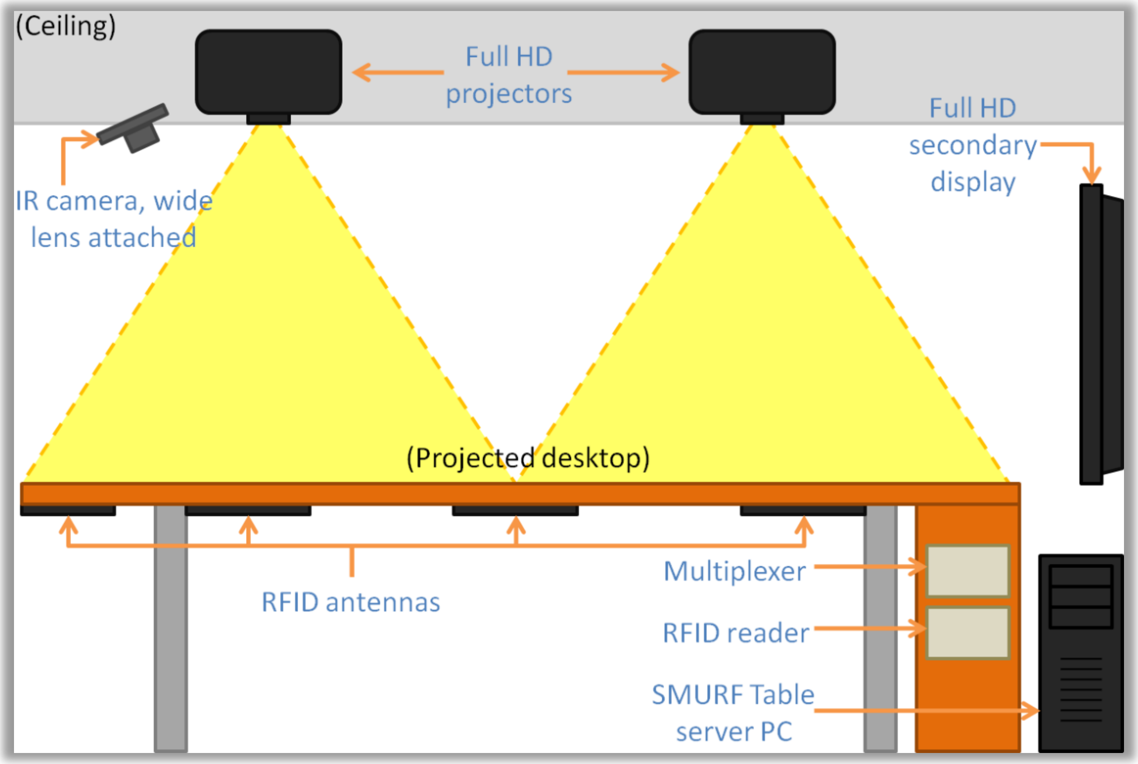


Figure 52. The SMURF Table hardware setup

4. Implementation

This Chapter describes the implementation of the SMURF Table system. The Chapter begins with a brief description of the AmI environment where SMURF Table has been integrated. A detailed explanation of the software architecture of the SMURF Table application follows, pointing out all the important modules comprising the application, as well as explaining how the SMURF Table interacts with the AmI environment. The Chapter closes with a description of the implementation of the SMURF Document Manager, explaining how remote computers connect to the SMURF Table server in order to transfer documents.

4.1 AmI Sandbox Architecture

SMURF Table has been developed and integrated in the AmI Sandbox, which is an area dedicated to Ambient Intelligence (AmI) research and development at the Institute of Computer Science (ICS) of the Foundation for Research and Technology – Hellas (FORTH).

The AmI Sandbox environment consists of various devices, such as computers, display monitors and projectors, as well as various types of sensors, such as cameras and antennas. The sensors are used to monitor the environment and forward relevant information to other devices, in order to process or display it. This infrastructure is supported by an internally developed middleware layer based on CORBA [57], which offers intercommunication facilities among the low-level services that control and monitor the various hardware components and the interactive applications. The middleware layer provides libraries and tools to enable software developers to create services with a true Object-Oriented Application Programming Interface (API). These services can be developed and used from any program written in any of the supported programming languages which include C++, .NET languages, Java, Python, and Flash ActionScript. The middleware effectively allows services to be distributed across the network, hiding the details of network connections and data serialization from the programmers.

The software units that comprise the AmI Sandbox can be seen in Figure 53. SMURF Table uses the libraries and tools provided by the middleware in order to obtain references to remote services and invoke the methods that are part of their API. In this context, a remote service invocation is indistinguishable from a method invocation on a local C++ object.

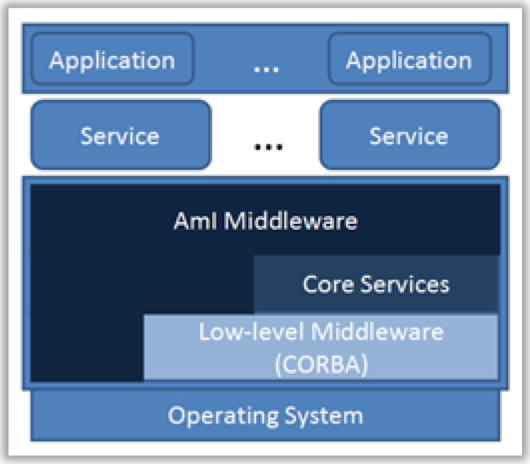


Figure 53. AmI Middleware architecture

4.2 SMURF Table

The SMURF Table application has been developed in order to provide a surface for multi-user collaboration in an AmI environment. Basically, multiple users interact with the application through the provided user interface. The application itself contains the appropriate modules that communicate with the environment to retrieve data from the IR camera and the RFID antennas, in order to identify users and provide interaction feedback.

The application is entirely written in C++ and the operating system used is the 32-bit version of Microsoft Windows Vista SP1. The operating system was chosen mainly because of the window management capabilities it provides. The Desktop Window Manager (DWM) is a module introduced in Windows Vista that takes care of how windows appear on a desktop surface, as well as how effects, such as transparency or the 'glass effect', are applied to windows in order to improve the users' visual experience. The DWM provides a variety of APIs that allow an application to control the graphical nature of windows. The SMURF Table application uses some of the DWM APIs, as well as various Win32 APIs from the Windows SDK, in order to communicate with the operating system and provide the desired functionality.

The main architecture of the SMURF Table application can be seen in Figure 54. The application consists of various software modules, each having a distinct role in providing the desired functionality to the users through the user interface. The role, features and functionality of all the software modules are explained in detail in the following subsections.

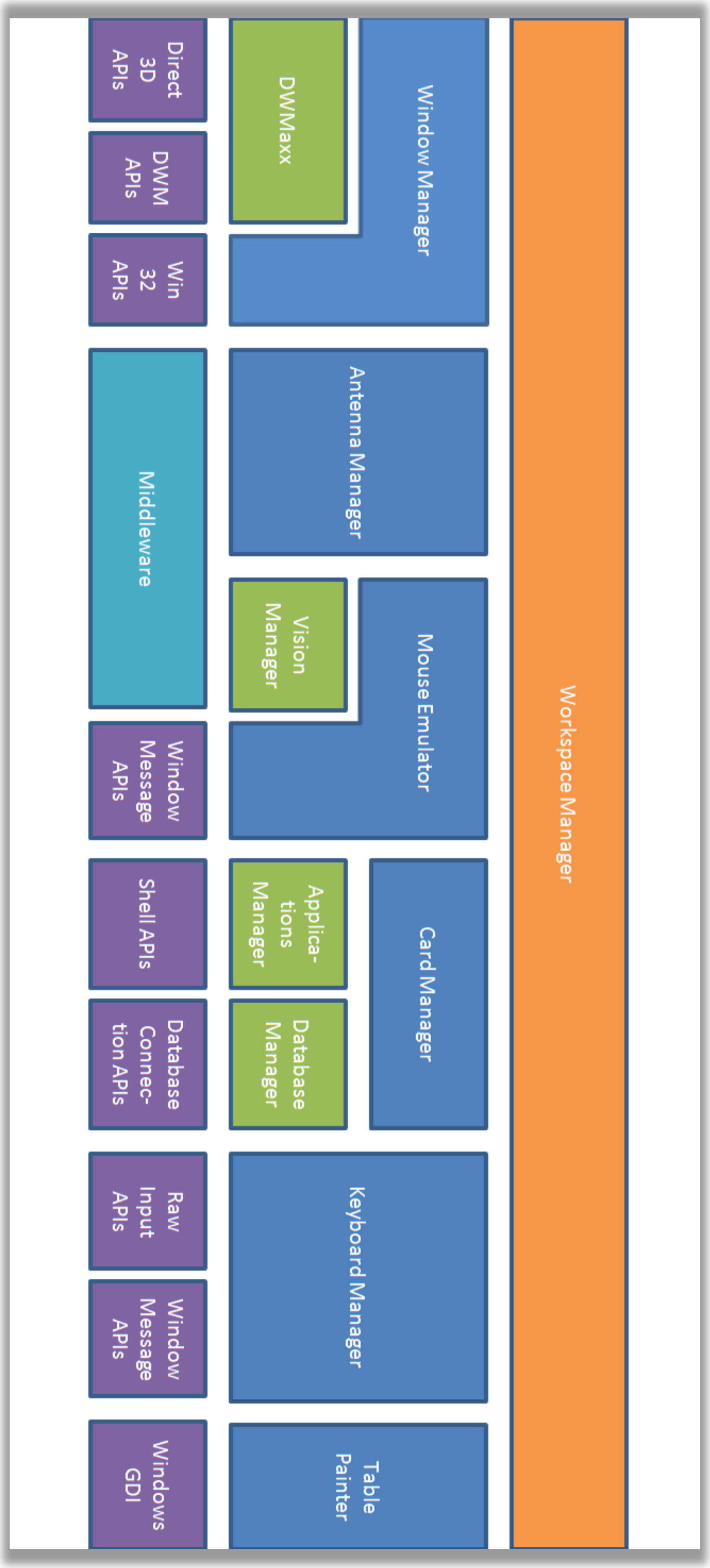


Figure 54. The software architecture of the SMURF Table application

4.2.1 Window Manager

The Window Manager is responsible for all the actions and monitoring operations concerning the manipulation of windows used in the application. The monitoring functions of the Window Manager include operations such as keeping track of a window's coordinates on the surface, retrieving a window's bounding box, getting information on child windows and checking a window's visibility state, among others. More complex functions include window creation, duplication, rotation and positioning.

Each document that a user opens is represented by a window of the application that the respective type of document is assigned to be opened with. For example, in the Microsoft Windows operating system, when a user chooses to open a simple text document, this is represented by a window of the Notepad application containing the text document. This happens because the operating system was configured to assign that application to that specific type of documents. The problem, though, with standard application windows is that they cannot be duplicated and shared with other users. To overcome this problem, SMURF Table uses live window thumbnails instead of standard windows.

The most important feature of the DWM is the ability to create live thumbnails of any standard window. These thumbnails are copies of standard windows, their content is updated in real time and their size can be adjusted freely. Live thumbnails always reflect the state of the original window anytime a change occurs. The Window Manager makes use of DWM's live thumbnails in order to create copies of standard windows for use on the table. The original window is first hidden off screen, since it won't be directly used, while its data is retrieved from the operating system. This information is then passed to the DWM that instructs the operating system to create a live thumbnail of the original window. The new window data is retrieved and can then be used by the Window Manager to manipulate it. The DWM allows creation of multiple live thumbnails of a single window, thus making it suitable for document duplication and sharing.

While live thumbnails may appear as normal windows, they provide no interaction capabilities at all. This means that if a user clicks at a specific point within a live thumbnail nothing will happen. This is due to the nature of live thumbnails, which act as 'live views' of an original window rather than normal windows. In order to allow users to interact with live thumbnails, the Window Manager applies a point-sensing and translation mechanism. Initially, the user interacts with a specific point within a thumbnail. The Window Manager retrieves the coordinates of the interaction point and translates it according to the coordinate system of the original window. The translation is done by taking into consideration the position of the original window and the ratio of the thumbnail's size to the original window's size. In case the thumbnail is rotated, its angle is also used to make the appropriate transformations between Cartesian and polar coordinates. The new point resulting from the translation can then be used as an interaction point in the original window to update its contents. The Window Manager then updates the live thumbnail view, providing the user with the appropriate feedback.

In order to support window rotation, an external open-source project was integrated into the SMURF Table application. The project, called DWMaxx[3], can perform 2D window rotation at any angle by using DWM and Direct3D APIs. DWMaxx works in Windows Vista SP1, but does not work on subsequent updates of the operating system. This is another reason why the specific version of Microsoft Windows was used for SMURF Table. Excluding the DWMaxx that does not work on newer versions of the operating system, the SMURF Table application itself should run properly in Windows 7 too, since there were no changes in the needed DWM and Win32 APIs.

Windows users are usually accustomed to moving windows around the desktop by clicking on the top border of windows and dragging the mouse until they reach the desired location. The windows that are used on SMURF Table are borderless, so users need another way to control their documents' position. The Window Manager overcomes this issue by creating a special handle for each open window. This handle is attached to the top-left corner of the window and retains its position whenever the window is moved or rotated. The handle is a simple semi-transparent window with no contents, whose basic functionality is to allow the user to hold, move and release it on the surface, while it takes care of carrying the attached window wherever it may be positioned.

As described in Chapter 3, each user has an individual thumbnail collection that acts as an index for personal documents. To avoid confusion with DWM's live thumbnails, the user's personal collection of document thumbnails will be referred to as documents collection from now on. Each item in a user's documents collection is created by the Window Manager as a live thumbnail of the document window it represents. These thumbnails are static windows, meaning that they do not need to be moved since their purpose is only to represent other windows. Because the items in a documents collection are small in size, a textbox with the document name is attached on top of each item to help users distinguish documents from each other.

When a user needs to open a specific document, they select the respective item from their documents collection. When the user points at an item, a semi-transparent copy of that item is created and follows the user's pointing path until they release it, as already described in Chapter 3. The Window Manager creates these ghost items by using live thumbnails, on which transparency is applied. The ghost items are temporary windows, meaning that they exist as long as the user drags them on the surface and they are destroyed as soon as the user releases them at any point. Figure 55 depicts the various state transitions of a document on SMURF Table.

[3] http://code.google.com/p/dwmaxx/

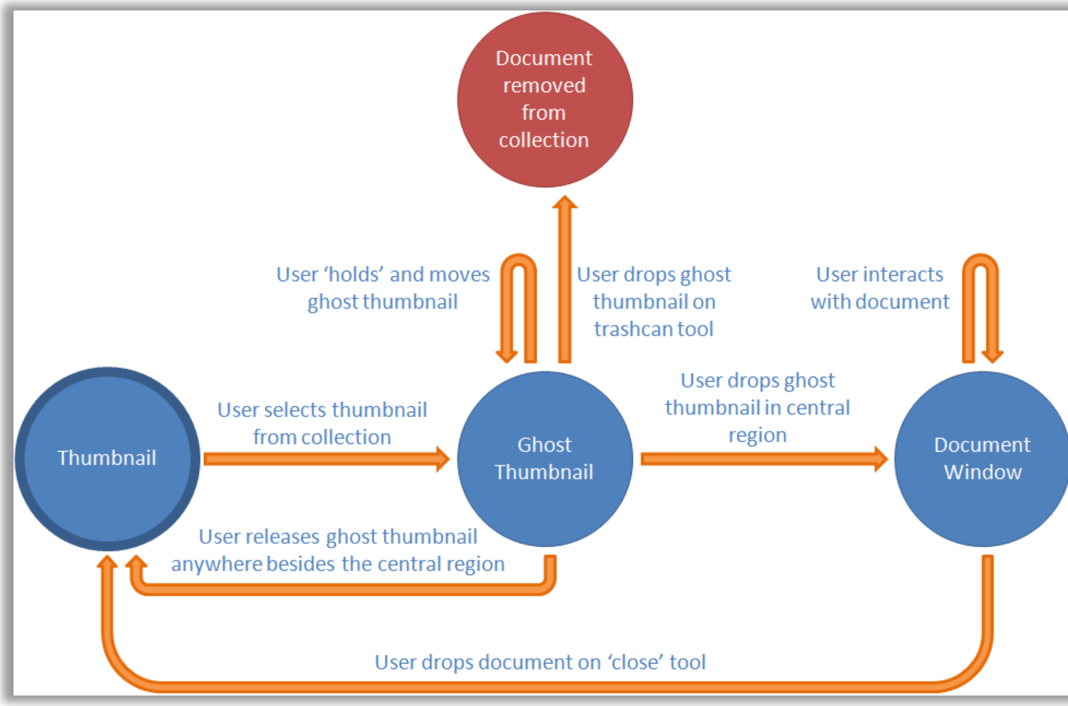


Figure 55. The various states of a document

The Window Manager internally stores the relations between all windows that are used in the application. Each window is recognized by a unique identifier provided by the operating system when the window is created. The Window Manager stores these identifiers in internal structures that allow the application to retrieve the identifier of an original window that a live thumbnail is based on, and vice versa.

4.2.2 Antenna Manager

The Antenna Manager is the software module that is responsible for the communication between the SMURF Table application and the RFID antennas installed underneath the table surface. The antennas are part of the AmI environment in which the SMURF Table system is integrated and the Antenna Manager takes care of handling events that are sensed by the antennas in order to forward them to the SMURF Table application.

A dedicated service has been implemented, that runs on a separate computer and captures events that occur on the antennas. The service queries all the antennas sequentially and checks whether there are RFID transponders present in any of the antenna fields. A unique number has been assigned to each of the seven antennas, namely from 1 to 7. The service basically captures two types of events, transponder down and transponder up. These events correspond to the user placing their personal card on the table and picking it up from the table, respectively. Each RFID card contains a unique serial number that is retrieved by an antenna when the card is placed within the antenna's field range. When an event occurs, the service creates a signal that announces whether a transponder was left on the table or picked up, accompanied by the respective antenna number and the card's serial number.

The Antenna Manager receives the signals emitted by the aforementioned service and the retrieved information is then forwarded to the application for processing. As mentioned above, a signal contains three distinct pieces of information: the event type, the antenna number and the transponder serial number. The event type allows the application to know whether a user needs to use or leave the table, expand or shrink their personal space, or use a keyboard, since keyboards are also tagged with RFID tokens. The antenna number informs the application where an event happened, so that the application can decide which actions should take place at the working area that the antenna belongs to. The transponder serial number allows the application to know whether the event refers to a user or a keyboard and, in the case of users, identify the user that the transponder belongs to. Figure 56 shows how the SMURF Table application communicates with the RFID antennas.

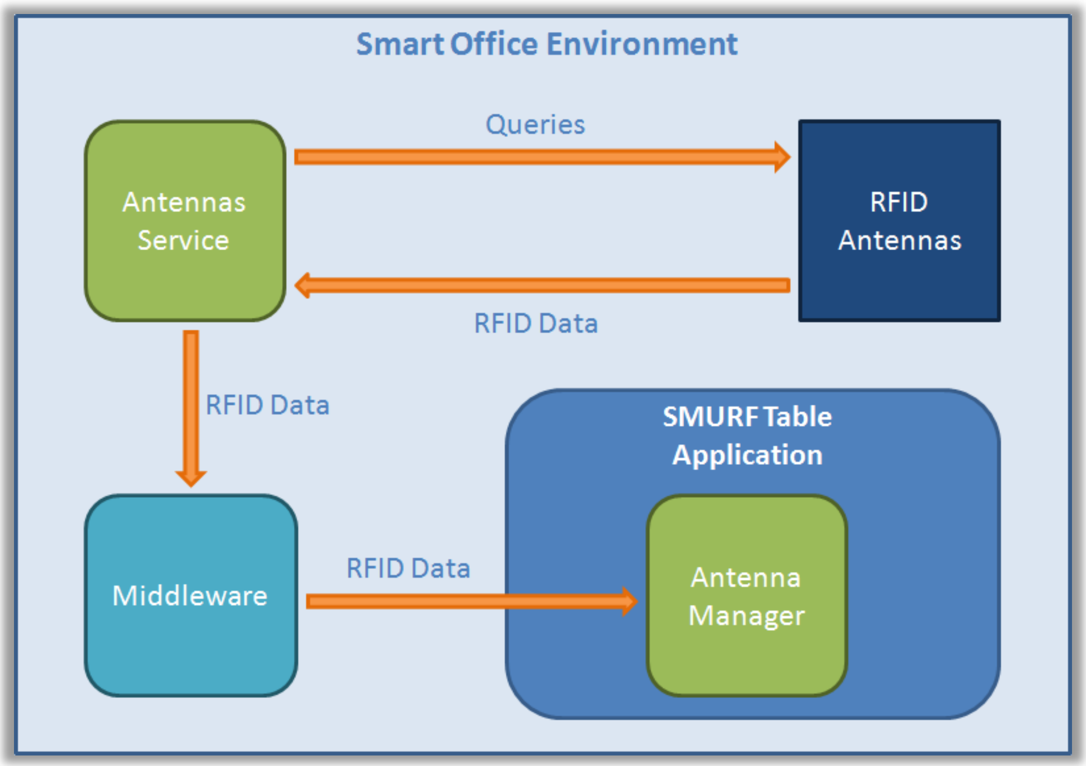


Figure 56. Communication between the SMURF Table application and the RFID antennas

4.2.3 Vision Manager

The Vision Manager is in charge of connecting the SMURF Table application with the IR camera above the table surface in order to retrieve information from the users' special pens. The camera is part of the AmI environment and the Vision Manager makes sure that all captured IR sources are properly recognized by the SMURF Table application.

In order to capture and monitor all IR spots on the table surface, a separate service has been implemented. The service runs on a dedicated computer and its purpose is to analyze data that is captured by the camera using computer vision techniques. Initially, the service is calibrated in order to specify the entire area it is supposed to monitor. The virtual surface of SMURF Table is rectangular, thus a four-point calibration is carried out that requires from a user to point with their IR pen at each of the four corners of the surface sequentially. When calibration is completed, the service can then infer the area that it is supposed to cover, as well as make the appropriate calculations whenever the captured data needs transformations to be appropriately represented. Calibration is not needed every time the service runs and should be carried out only when the position or angle of the camera changes.

When the vision service runs, the camera begins capturing continuous images of the table surface. Whenever one or more IR spots appear on the surface, the camera recognizes them and the respective data is retrieved by the service. The service then makes the appropriate transformations and produces a set of points, each corresponding to the coordinates of a distinct IR point on the surface. Each pair of coordinates is accompanied by a unique number that identifies the respective IR spot. This way the service can track the positions of distinct IR spots even when they are moved around the surface. An IR spot retains its identification number as long as it is visible and its moving speed remains under a specified threshold. When the speed of an IR spot exceeds that threshold, it is no longer considered as existing and a new IR spot is identified instead. This is due to the frame rate at which the camera captures the images of the surface. If the user moves their pen fast enough, the covered distance between two captured frames may be long enough for the service to infer that there are two different IR spots far from each other. In the current implementation, a capturing rate of 20 frames per second is considered suitable, since no glitches are caused in the application and the moving speed threshold is acceptable.

The Vision Manager module receives the data produced by the vision service and appropriately prepares the respective information for the application to use. The data sets created by the service contain two types of information: IR spot identification numbers and their respective coordinates. These coordinates are expressed relative to the coordinate system that the service uses, which is different from the coordinate system used in the SMURF Table application. The Vision Manager handles the translation of the received pair of coordinates to the application's coordinate system, creating a new set of information that the application can then use to create interaction points on the surface.

4.2.4 Mouse Emulator

The Mouse Emulator is the software module responsible for the creation of interaction points on the surface. Interaction points are the points on the table surface that users point at with their pens in order to interact with documents or other interface elements.

The Mouse Emulator module was named after its purpose, which is to emulate the behavior of the Windows operating system when a user interacts with windows by using a mouse. The operating system communicates with windows through a message sending mechanism. Each time a user interacts with a window, the operating system sends an appropriate message to that window to inform that it should be updated. For example, when a user clicks on the 'Minimize' button on the top border of a window, the operating system creates a 'Minimize' message and sends it to that window in order to announce that it should be minimized. The Windows operating system uses by default a white arrow cursor to inform the user of their current pointing position. This cursor is called the system cursor and it is the only way a user can interact with the operating system environment when using a mouse. The Windows operating system was designed to be operated by a single user on a computer. This means that the user may connect more than one mouse to their computer, but they will not be independent. Each mouse will be recognized as a separate device, but they will all control the same cursor, that is the system cursor. This means that multiple users are not supported by the operating system. This issue is overcome by using an independent message dispatching mechanism and completely discarding the system cursor.

The Mouse Emulator continuously communicates with the Vision Manager module to retrieve pointing coordinates from all users. Depending on the users' actions, three types of mouse messages can be created. When a user taps the pen's button once, a 'Left Mouse Button Down' message is created. When a user holds the pen's button down and moves the pen, a 'Mouse Move with Left Button Down' message is created. Finally, when a user releases the pen's button, a 'Left Mouse Button Up' message is created. The Mouse Emulator creates all messages according to the rules defined by the operating system, so there is no problem in dispatching them to windows. Each created message encloses the respective coordinates that are retrieved from the Vision Manager and, when a message is ready, the Mouse Emulator takes care of dispatching it to the appropriate location. Figure 57 shows how the SMURF Application retrieves the needed IR data from the camera.

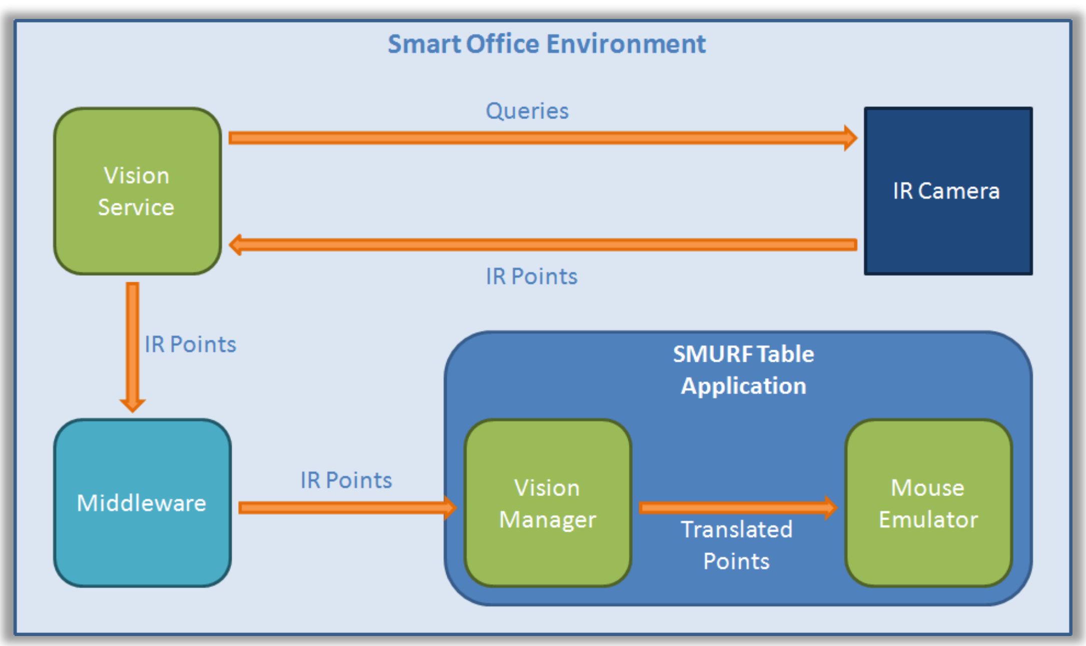


Figure 57. Communication between the SMURF Table application and the IR camera

4.2.5 Database Manager

All documents used on the SMURF Table are stored in the computer where the application runs. When a user works on the table, the application needs a way to know which documents are owned by that user. To achieve that, a database is used in order to allow the application to match documents with users.

The Database Manager is the module that connects the application with the aforementioned database to retrieve the desired information. In the current implementation a MySQL[4] database is used. A MySQL database server is installed and runs in the computer where the SMURF Table application runs. The database contains a table whose entries consist of two fields: a serial number and a document path. The first field is a RFID card serial number that identifies a user, while the second field is the exact path where a document owned by that user is stored on the computer.

Basically, the Database Manager is a database client that can execute simple queries on the SMURF Table database and retrieve the respective results. The supported operations include retrieving a user's documents, removing specific documents from a user's collection, retrieving all the owners of a single document and adding new documents to a user's collection.

4.2.6 Applications Manager

The Applications Manager module is in charge of managing the external applications that are used by documents on SMURF Table. The external applications are programs that are installed on the computer and that have been assigned to open specific types of documents. The assignment is usually done by the operating system automatically at installation time, but can also be configured by users through configuration interfaces provided by the operating system.

The functionality of the Applications Manager consists of two operations: launching and shutting down external applications on demand. Each time an application needs to be invoked, the Applications Manager uses the respective document's path to delegate its activation to the Windows shell. Basically, the Applications Manager does not need to remember or decide which applications are used for specific document types; the only requirement is that each document's path must be known. When a document path is available, the Applications Manager uses Windows shell commands to order the operating system to load the appropriate application. When an external application opens, the respective window identifier is retrieved so that it can later be used by the SMURF Table application. Whenever a document is no longer needed on the table, the Applications Manager uses shell commands to inform the operating system that the respective application should be shut down.

4.2.7 Card Manager

The SMURF Table application needs to have immediate access to users' personal resources in order to allow users to work efficiently. The Card Manager is a module that provides fast indexing of user resources and takes care of automatically loading user documents at run time.

[4] http://www.mysql.com/

When a user needs to work on the table, they place their personal card on the table surface. As already mentioned, each card contains a unique serial number that can be identified by the system. The Card Manager uses that serial number and communicates with the Database Manager in order to retrieve a list of document paths that correspond to that user from the database. The user serial number and the respective document names are internally stored by the Card Manager as long as the user works on the table. This way, the application gains fast access to user resources when needed, without having to query the database each time a check has to be performed. Each time a user adds or removes a document from their collection, the Card Manager instructs the Database Manager to update the database appropriately, updating at the same time the internally buffered user information to keep the system synchronized. When the Database Manager removes a document from a user's collection, the database is automatically checked to find out whether that document is owned by any other users. In case no other users own that document, the Card Manager is informed and takes care of completely removing the respective file from the computer. When a user leaves the table, the Card Manager erases all temporarily stored information related to that user.

When a user's list of documents has been retrieved from the database, the Card Manager instructs the Applications Manager to launch the applications corresponding to each of the documents. If a document is owned by multiple users and at least one user already uses it on the table, the Applications Manager is instructed not to launch the respective application, since it is already open and the user only needs a copy of the existing application window. When a user closes a document, the Card Manager checks whether it is also owned by other users who are still using it on the table. If no other user still uses that document, the Applications Manager is instructed to close the respective application, otherwise it remains open.

4.2.8 Keyboard Manager

As already discussed, each user on SMURF Table may use a personal keyboard to enter and edit text in their documents, thus the application needs a way to support multiple keyboard devices that will provide independent input for each user.

There are two main issues concerning multiple keyboard input. The Windows operating system, as mentioned previously, was designed for single-user operations on a single computer. When multiple keyboards are connected to the same computer, the system recognizes them as distinct devices but the operating system directs all keyboard input to the same destination. Additionally, the operating system restricts keyboard input to one window at a time. The only window that can accept keyboard input, called the focus window, is the last window the user has interacted with, either by keyboard or by mouse. Besides the focus window, all other open windows are considered inactive and cannot accept any kind of input by default.

The Keyboard Manager has been implemented in order to address the above issues and augment the application with multiple keyboard capabilities. A set of Win32 APIs, characterized as Raw Input, has been used to support the identification of separate keyboard devices. These APIs allow an application to retrieve a unique hardware identifier for each input device that is connected to the system. When a keyboard device is connected to the SMURF Table computer, the Keyboard Manager identifies it by retrieving and storing its hardware identifier. As discussed previously, each keyboard used on the SMURF Table has been equipped with a unique RFID token, so that the system can recognize it. When a user places a keyboard over their working area, the system uses its RFID serial number to assign it to that user, while the Keyboard Manager provides the keyboard's hardware identifier so that the application knows that input coming from this device was typed by that user.

The application does not make use of the default operating system focusing system; instead, a separate focus identifier is used for each user. When a user selects a document, either by selecting its handle or by pointing anywhere inside its interaction area, the respective window automatically becomes the user's personal focus and its identifier is internally stored. Whenever the user selects another document, the respective personal focus identifier is updated to refer to the new window.

Besides the hardware device identifier, the Raw Input APIs also allow an application to retrieve key sequences coming from a specific device. These key sequences may consist of either single key presses or key combinations. Each time a user types with their keyboard, the application receives the respective key sequence in the form of character codes. These character codes can then be used to infer which key combination the user has pressed.

When a hardware identifier and a key sequence are available, the Keyboard Manager can then redirect the input to the respective user's personal focus window. The operating system uses keyboard messages to send keyboard input to windows. The Keyboard Manager uses the character codes retrieved from the various keyboards to construct new messages that comply with the message specifications defined by the operating system. When a new message is created, the Keyboard Manager retrieves the focus window identifier of the user that the keyboard currently is assigned to and dispatches the message to that window.

4.2.9 Table Painter

The Table Painter module is in charge of drawing all the elements that comprise the SMURF Table virtual surface. The virtual surface itself is a window whose size covers entirely the table surface. The exact dimensions of the virtual surface are 3840 x 1080 pixels, that is the resolution of two full high definition displays placed next to each other. Basically, the virtual surface window covers exactly a Windows desktop surface that is expanded across two full high definition displays.

The Table Painter uses the virtual surface as a canvas for its drawing operations. Each distinct working area is represented by a set of rectangles that indicate its borders, as well as the contained subareas. The Table Painter retrieves the working areas' specifications from a configuration file that contains information such as the areas' positions on the surface and the subareas' positions within a working area. The virtual surface layout is fully configurable, since the specifications file allows the users to change the position and size of each working area, the layout within a working area, as well as various offsets, such as the distance between working areas. The Table Painter reads all the needed information from the configuration file and dynamically draws the respective elements at initialization time.

The aforementioned drawing operations take place at the application's initialization time. The Table Painter is also used at run time to update parts of the surface, depending on specific user actions. When a user places their card over a working area, the area is activated in order to be used. This is visualized by updating the area's colors so that it can be distinguished from the rest of the surface. The Table Painter uses colors that are brighter than the initial ones to indicate that an area is activated.

Besides visualizing the various areas and subareas on the surface, the Table Painter is also responsible for displaying images where needed. The images representing the briefcase and trashcan tools, as well as the browsing arrows of a documents collection, are actually bitmap images, whose destination positions are retrieved from the configuration file in order to be displayed on the surface.

When a user expands their personal area, the Table Painter takes care of appropriately visualizing the expansion by drawing the entire portion covered by the conjunction of the merged areas. Similarly, when a user shrinks their personal area, the Table Painter fills the new active area with the respective colors, while the released areas' colors are restored to the initial ones. Finally, when a user deactivates their personal area, the Table Painter restores its initial colors to indicate that it is inactive.

4.2.10 Workspace Manager

The Workspace Manager is responsible for the proper functionality of all areas that comprise the SMURF Table surface. It is also the software module that communicates with most of the previously described modules in order to synchronize their operations and ensure interoperability. The functionality of the Workspace Manager and the communication among all the software modules is described in the following examples of common actions that users can take on SMURF Table.

Application Initialization

When the SMURF Table application is initialized, the Workspace Manager reads the surface specifications configuration file (see subsection 4.2.9) and internally stores all territorial information for each working area. The Window Manager is used to create the surface window and the Table Painter is then instructed to actually draw the virtual surface.

Working Area Activation

When a user places their card over a working area, the Antenna Manager forwards the antenna and serial number information to the Workspace Manager. The Table Painter highlights the activated area and draws the area's documents collection region, briefcase, 'close' and trashcan tools. The RFID information is then passed to the Card Manager that identifies the user through the Database Manager, while the Workspace Manager automatically assigns the area to that user. A list of the user's documents is retrieved from the database and the Card Manager passes that list to the Applications Manager in order to launch the appropriate external applications. When an external application window opens, its identifier is retrieved and the Workspace Manager passes it to the Window Manager in order to move the respective window off the surface. When all external applications are launched and moved off the surface, the Window Manager is instructed to create the user's documents collection, by creating a thumbnail item and a title textbox for each document and eventually placing them in the documents collection region. The Workspace Manager also checks the position of the activated area and instructs the Window Manager to properly rotate each window after it is created. When items are added to the documents collection, the Workspace Manager automatically organizes them in pages that can be browsed through the provided arrows. Figure 58 depicts how the various software modules communicate with each other in order to activate a working area.

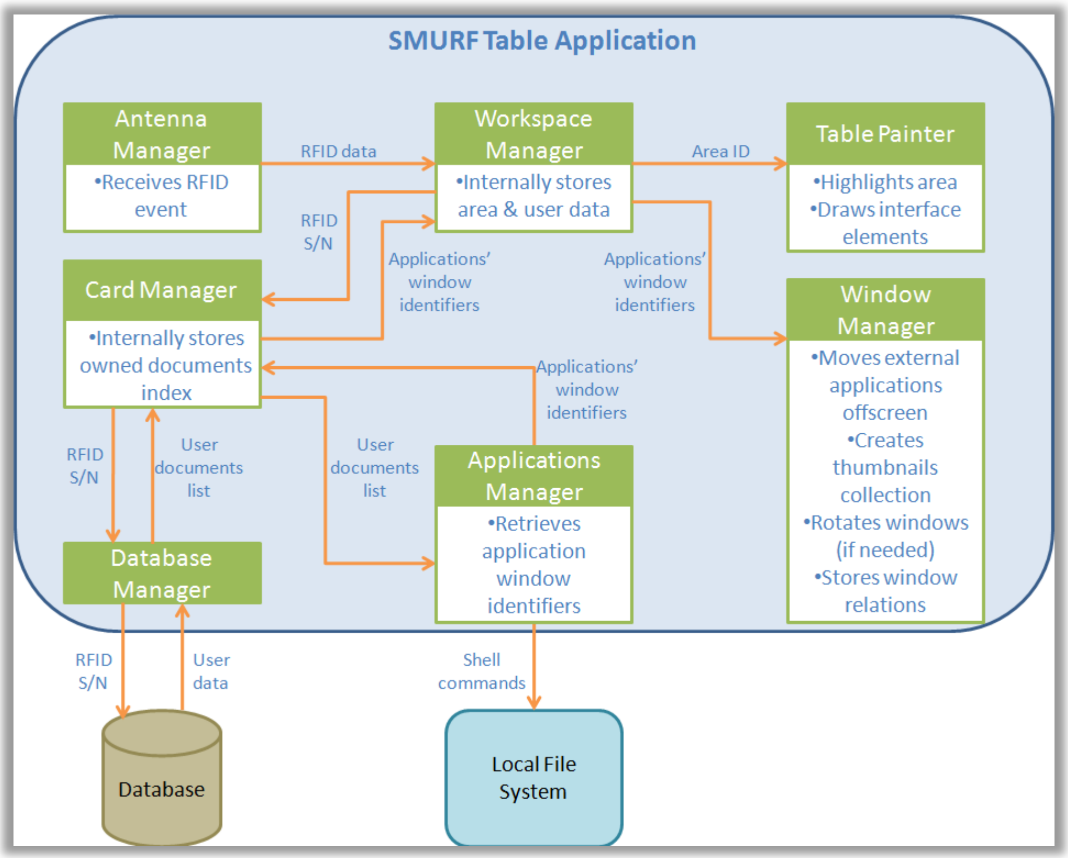


Figure 58. Communication among software modules for the activation of a working area

Document Opening

The user selects a document to open by pointing at the respective documents collection item with the pen. The Vision Manager captures the user's pointing location and the Mouse Emulator creates an interaction at that point. The interaction point is retrieved by the Workspace Manager that forwards it to the Window Manager in order to check whether it belongs to a window. The Window Manager confirms that the point belongs to an item from the documents collection and creates the respective ghost item that follows the user's pointer. When the user releases the ghost item, the Workspace Manager is informed by the Mouse Emulator that the user has dropped the selection at some point. The Window Manager is instructed to destroy the ghost item, while the Workspace Manager checks whether it was dropped within that area's central region. If so, the Window Manager creates a window of the respective document along with its handle inside the area's central region.

Document Sharing

To share documents with someone else sitting around the table, the user drags the respective window by its handle towards the other user's working area. While the window is dragged over the surface, the Workspace Manager checks its position through the Window Manager. If the window passes over another working area, the Workspace Manager checks the position of that area and instructs the Window Manager to dynamically rotate the window when needed. When the user drops the window by releasing its handle, the Workspace Manager checks if it is dropped within another user's working area. If so, the Window Manager creates an exact copy of the window based on the same original window, places it in the second user's central region and automatically rotates it to fit the user's perspective.

Keyboard Input

A user can place a keyboard over their personal area to be able to edit text in their documents. The Antenna Manager recognizes the keyboard by its RFID token, while the Workspace Manager instructs the Keyboard Manager to retrieve its hardware identifier. The identified keyboard is then assigned to the user's working area by the Workspace Manager and the user can start entering text. The user first selects the desired document by pointing at it with the pen, while the Workspace Manager automatically stores the respective window identifier as the user's current focus. When the user enters text, the Keyboard Manager receives the typing events and identifies the user by checking the hardware identifier of the keyboard that produced the input. When the user is identified, the Workspace Manager retrieves the user's current focus identifier and, when the input data is ready to be sent, the Window Manager is instructed to retrieve the identifier of the original window of that document. The retrieved identifier is passed to the Keyboard Manager that then sends the text data to the document's external application window. When the original window receives the input data, the Workspace Manager instructs the Window Manager to search for all copies of that window and automatically update them to display the new information, so that all users who share that document stay up-to-date. Figure 59 shows how independent keyboard input is achieved.

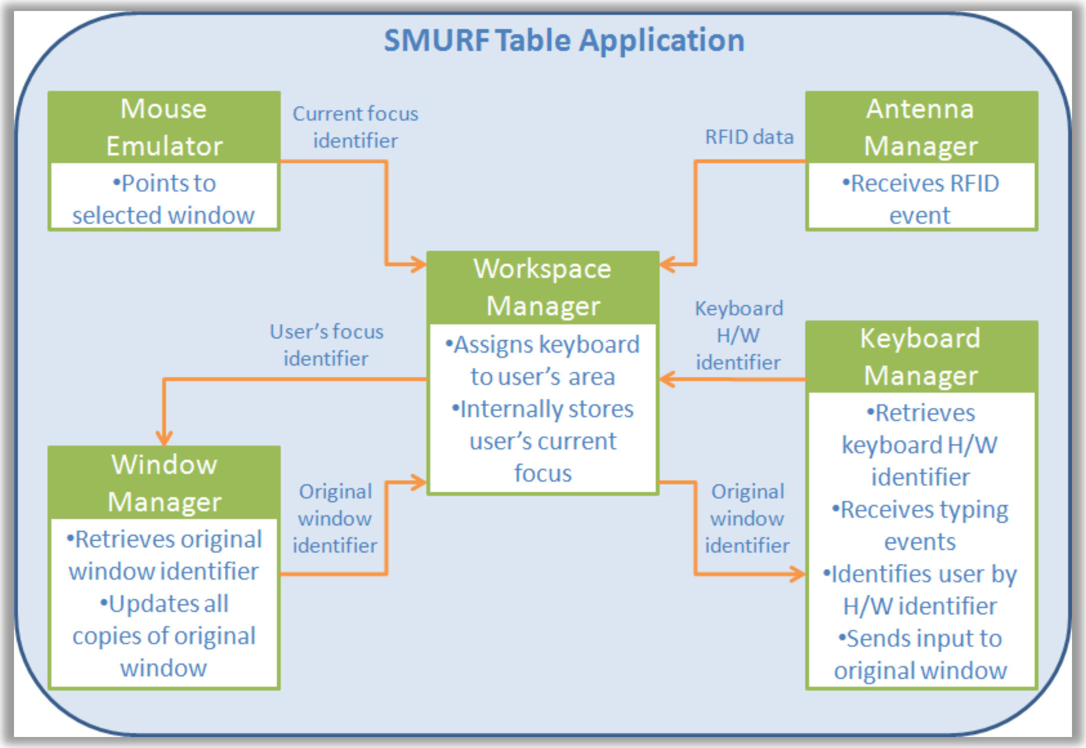


Figure 59. Communication of software modules to provide independent keyboard input

Working Area Deactivation

When a user no longer needs to work on the table, they place their card over their working area to deactivate it. The Antenna Manager retrieves the RFID serial number from the card and sends the respective data to the Workspace Manager that checks if the serial number actually matches the user that the area is assigned to. If so, the Window Manager is instructed to retrieve all windows that correspond to the user's open documents and close them. The Window Manager also automatically closes all items in the user's documents collection. The Workspace Manager then uses the Table Painter to erase the documents collection region, briefcase, 'close' and trashcan tools from that area. Finally, the Table Painter restores the working area's colors to the initial ones and the Workspace Manager internally clears all data related to that area in order to make it available to other users.

4.3 SMURF Document Manager

The SMURF Document Manager is an application that allows users to remotely connect to the SMURF Table server in order to manage the documents that they use on the table. The application is entirely written in C++ and has been successfully tested and used in Microsoft Windows XP, Vista and 7.

The architecture of the SMURF Document Manager application can be seen in Figure 60. As depicted, the application consists of three basic software modules: a Database Client, a FTP Client and a Remote Resource Synchronizer. The functionality and communication among these modules is described in the following subsections.

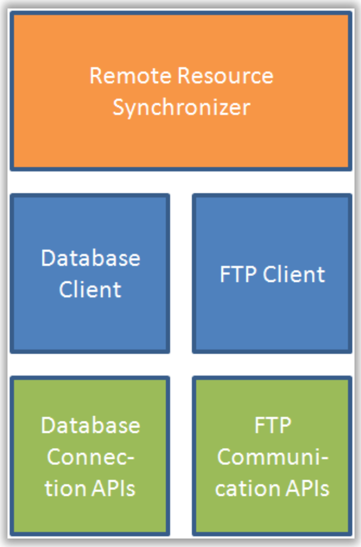


Figure 60. The software architecture of the SMURF Document Manager application

4.3.1 Database Client

The Database Client is responsible for all the transactions between the application and the SMURF Table database. It is the software module that connects to the database server running on the main computer system and communicates with it so that user and document related information can be retrieved on demand.

When the user runs the application, a simple login form appears requiring from the user to enter their username and password. When the user attempts to connect, the Database Client uses the provided credentials and checks the remote database to verify their integrity. Besides the table of documents and owners described previously, the database also contains a table of user-specific information that the SMURF Document Manager uses to identify users and provide access to the SMURF Table server. When a user successfully connects, their card serial number is retrieved from the database and can then be used by the Database Client to execute queries related to the user's documents.

The basic operations supported by the Database Client include retrieving a user's list of documents, as well as adding and removing documents from a user's documents collection.

4.3.2 FTP Client

When users connect remotely to the SMURF Table server, they need a way to transfer documents between the remote server and their local storage. In order to support efficient file transfer, a FTP Client has been implemented and integrated into the SMURF Document Manager application.

To support the remote end of the communication, a FTP server has been installed on the computer where the SMURF Table runs. In the current implementation, the open source

FileZilla[5] server has been chosen as the FTP server. To provide secure file transfers, the FTP server requires from each user to provide a username and password. The FTP Client uses the information that users provide when the application starts to connect to the remote FTP server.

User documents stored on the SMURF Table computer are organized in a centralized way. Basically, each user's documents are not stored separately from other users' documents, but are rather stored in a common space. This choice was made mainly in order to support the shared documents philosophy, that is to use a single document as a shared resource rather than using different copies of the same document that may contain inconsistent information due to individual content alterations. All user documents are stored in a single folder on the main computer and the FTP server is configured to use this folder for all file transactions.

The FTP Client performs several operations on remote files, including file uploading and downloading, as well as deleting files from the remote server.

4.3.3 Remote Resource Synchronizer

The Remote Resource Synchronizer is responsible for synchronizing the transactions of the Database and FTP Client modules. Its main purpose is to retain consistency between the indexing information (provided by the Database Client) and the actual data (provided by the FTP Client) when transactions take place.

When a user adds a new document to their collection, the Remote Resource Synchronizer first instructs the FTP Client to upload that file to the server. When the uploading is complete, the Database Client is then instructed to update the SMURF Table database by adding a new entry for the uploaded document (see Figure 61). In case the user removes a document from their collection, the Database Client first erases the respective entry from the database and then automatically checks whether that document has any other owners. If no other user owns that document, the FTP Client is then instructed to actually erase the respective file from the server.

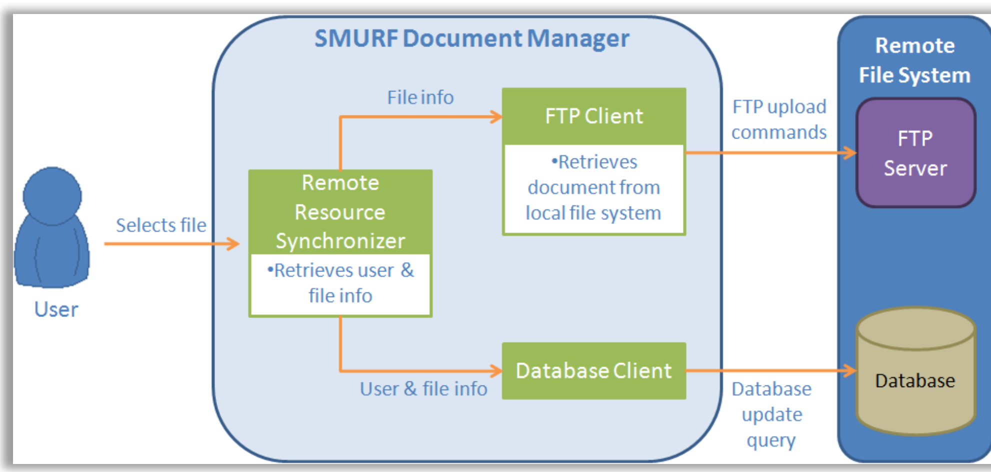


Figure 61. Uploading a file through the SMURF Document Manager

[5] http://filezilla-project.org/

In summary, the SMURF Table system provides users with two applications in order to manipulate their documents (see Figure 62): the SMURF Document Manager allows users to manage their document collections, while the SMURF Table application provides users with the means to interact with their documents.

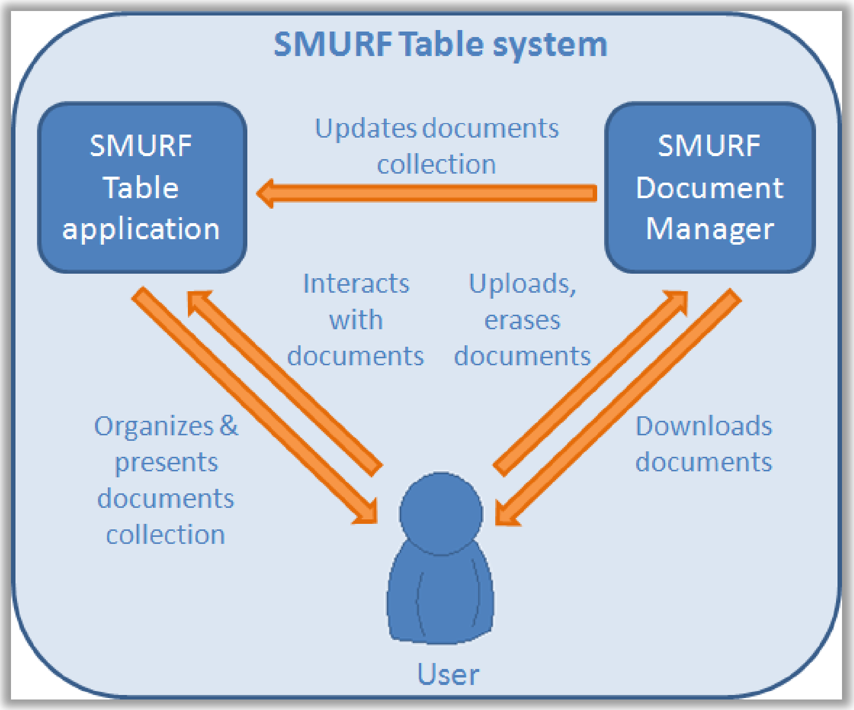


Figure 62. Basic system usage for document manipulation

5. Evaluation

An evaluation process was conducted in order to determine the developed system's usability against the four usability principles as presented by Booth [55]:

- *Usefulness* – does the SMURF Table application meet the users' needs? Does it help the user achieve their goals?
- *Effectiveness* (*ease of use*) – what is the degree of effort to use the application?
- *Learnability* – is the application easy to learn? Do the users use it more effectively after some amount of time?
- *Likeability* or *Satisfaction* – what do the users think about the application? How do they feel about it?

Brooke's System Usability Scale (SUS) [56] was used for the quantitative analysis. It is a simple ten-item attitude Likert scale and was mainly chosen because it covers a variety of aspects concerning system usability, such as the need for support, training and complexity. For the qualitative analysis the think-aloud method was followed, asking the users to comment on their experience during and after each task.

5.1 Evaluation Set-up

All the evaluation sessions took place in two separate rooms within the ICS-FORTH AmI Sandbox area. In one of the rooms a computer was placed exclusively for the needs of the evaluation experiment. The SMURF Document Manager application was installed on the computer, which was also networked in order to be able to communicate with the SMURF Table server. Some test files were also created in order to be used by participants in the evaluation process. The other room was the office in which the SMURF Table application and equipment are actually installed and run. Some test accounts were created in the SMURF Table, and each user was provided with a personal identification card for use on the table, along with a username and password corresponding to their account. A set of wireless keyboards and custom IR pens were also provided for each participant to use on the table. Along with each participant there was an evaluator in the room in order to coordinate the whole process. The evaluator first introduced each participant to the functionality of the system and presented a set of tasks that each participant should try to accomplish. Additionally, each participant was encouraged to directly ask any questions they might have as well as freely make any comments while using the system.

5.2 Participants

Six volunteers participated in the evaluation process, three females and three males, whose ages ranged from 23 to 35 years. All participants are expert users of computers, but none of them was familiar with the system's functionality or participated in the system's development process. Some of the participants are expert user interface designers.

5.3 Process

The evaluation process was conducted in two parts: in the first part the system was evaluated by one user at a time, while in the second part two collaboration sessions took place with three users at a time.

In the first part, each user was introduced to the functionality of the system and was presented with a set of tasks to be accomplished in order to complete the individual evaluation process. Each user was then asked to use a specific computer, on which the SMURF Document Manager application was installed. The interface of the document management application was demonstrated to each user, who was then asked to perform the following series of tasks:

- Task 1 – Log in to the system
- Task 2 – Upload a specific document from the desktop to SMURF Table
- Task 3 – Download a specific document from SMURF Table
- Task 4 – Remove the previously downloaded document from SMURF Table

After completing the document management application tasks, each user was directed to the room where the SMURF Table application was running. Each user was asked to freely choose the area they wanted to occupy on the table. The interface of the virtual surface was explained to each user and, afterwards, they were asked to perform the following series of tasks:

- Task 1 – Activate the chosen working area
- Task 2 – Open a text document from the personal documents collection
- Task 3 – Highlight some text in the document
- Task 4 – Assign a keyboard to the personal working area
- Task 5 – Type some text in the open text document
- Task 6 – Close the open text document
- Task 7 – Remove a document from their personal documents collection
- Task 8 – Expand the personal working area
- Task 9 – Open two different documents from the personal documents collection
- Task 10 – Place the open documents at any position within the expanded working area
- Task 11 – Shrink the personal working area
- Task 12 – Deactivate the personal working area

In the second part of the evaluation process, two separate collaboration sessions took place with a different group of three users participating in each session. The users were asked to occupy any area on the table and were then introduced to the collaborative capabilities of the system. After an explanation of all possible collaborative interactions, each of the users was asked to perform the following series of collaborative tasks:

- Task 1 – Share an empty image document with the rest of the users
- Task 2 – Draw a shape in the empty document to show to the rest of the users
- Task 3 – Ask another user to share one of their documents

- Task 4 – Store the received document copy in the personal documents collection
- Task 5 – Open a presentation document from the personal documents collection
- Task 6 – Present the document on the secondary display
- Task 7 – Progress the presentation by two slides
- Task 8 – Move the presentation document back to the personal working area

The following tables show how much time each test took to complete, excluding the time during which the system's functionality was explained to the users, as well as the time during which the users moved from one room to the other.

Participant	Time (in minutes)
User 1	6.2
User 2	7.1
User 3	6.5
User 4	5.7
User 5	5.9
User 6	7.2
Average Time	***6.4***

Table 1. The individual evaluation process completion times

Session	Participant	Time (in minutes)
1	User 1	4.5
	User 2	6.0
	User 3	5.1
	Average Time	***5.2***
2	User 4	5.3
	User 5	5.9
	User 6	5.0
	Average Time	***5.4***
	Total Average Time	***5.3***

Table 2. The collaborative evaluation process completion times

The first part of the evaluation took in average 6.4 minutes for each user to complete (see Table 1), while an average of 5.3 minutes was needed for each user to complete the requested collaborative tasks (see Table 2). During the evaluation process, all users' comments were noted down along with any spontaneous remarks the users made as a result to certain task accomplishments. At the end of the evaluation process, the participants were asked to fill out the System Usability Scale questionnaire (see Appendix) in order to help quantitatively analyze the system's usability.

5.4 Results

5.4.1 Quantitative Results

Figure 63 shows the System Usability (S.U.) calculated per user from the questionnaires that the participants filled out after the evaluation experiment. The mean S.U. is 89.2 (in a scale from 0 to 100), which is a rather positive outcome concerning the overall usability of the system.

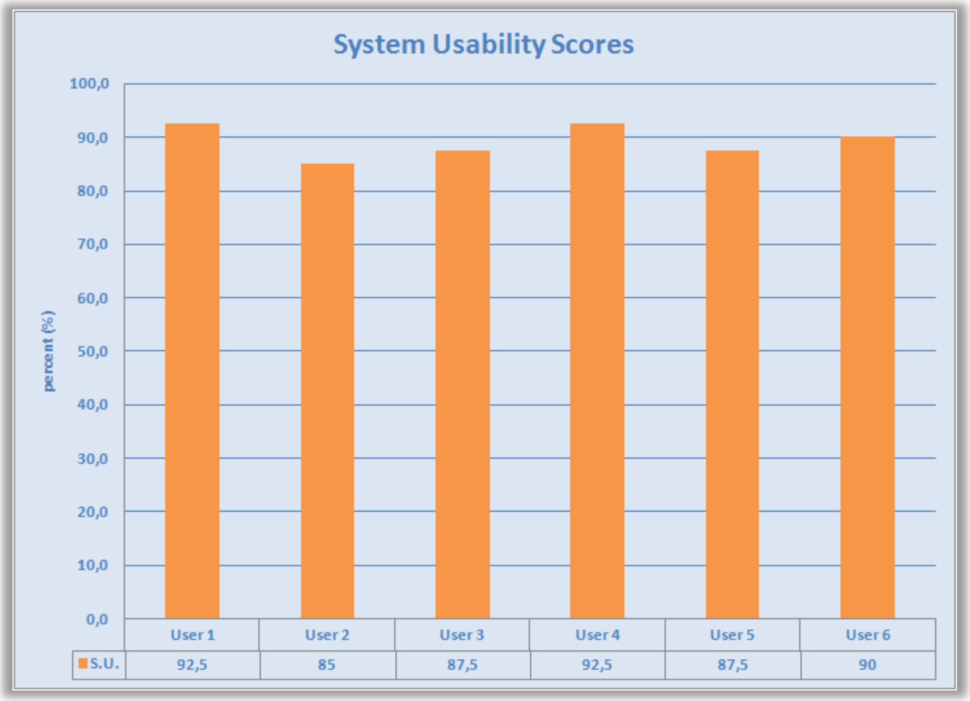


Figure 63. System usability calculated per user and mean system usability

Figure 64 shows the average S.U. per question. These results, in combination with remarks that users made, pointed out specific issues that are discussed in the following subsection. It is very possible that the overall S.U. can be improved when further system improvements are made.

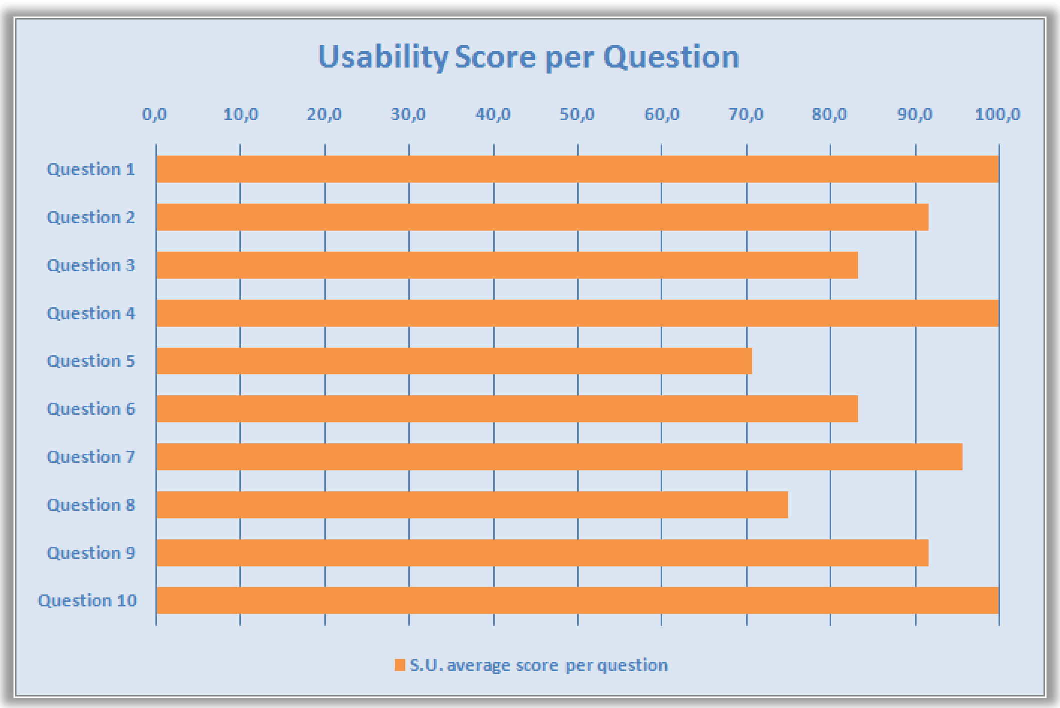


Figure 64. Average system usability scores per question

5.4.2 Qualitative Results

During the evaluation process, the think-aloud method was followed to encourage the participants to freely make comments and remarks while completing the requested tasks. The feedback received from the participants proved rather useful in identifying potential improvements. A summary of the gathered observations concerning the four usability principles is given below.

Usefulness

In general, all participants agreed that SMURF Table provides a very useful surface for meetings. Most of the users pointed out that one of the system's most useful features is that its use is not restricted to specific user groups, such as developers or designers, but can accommodate virtually any group's needs, e.g., a team of medical examiners sharing ideas on their patients' history records, while viewing an image of an X-ray on the secondary display. Some users also liked the fact that they can gather a team of colleagues and immediately start a presentation without having to carry a laptop or storage device with the presentation documents. All participants also agreed that the SMURF Document Manager is a very practical application, especially because it allows documents to be transferred to the SMURF Table regardless of the user's location, as well as because it relieves users from remembering to carry their documents with them when going to a meeting.

Effectiveness

The participants found that the system was rather easy to use and they completed the requested tasks without encountering difficulties. Some of the users' comments, though, pointed out a few interface related issues that should be taken into consideration in order to improve the respective system functions.

One of the issues is related to the 'close' tool of each working area. Some users were confused at times when they needed to close an open document and had to drag and drop it over the 'close' tool. They pointed out that they are accustomed to closing a document window by using the 'x' button that is on the top-right corner of windows in most operating systems, rather than dragging it to a certain area in order to close it.

Another issue was related to the trashcan tool. It seems that some of the users found that it is not clear that the tool accepts thumbnails in order to remove the respective documents from document collections. As observed during the evaluation process, some users dragged and dropped document windows on the trashcan in order to remove them from their document collections, action which had no result. The users who encountered such problem proposed that it would be preferable to make the function of the trashcan tool clearer. A potential solution to overcome this issue would be to place the trashcan tool on the left part of each working area, close to the documents collection region. This way the users should be able to easily associate the tool's use with documents collection items instead of document windows.

Learnability

The time required to introduce each user to the system's functionality was approximately 4 minutes for the individual task functions and less than 2 minutes for the collaborative tasks. All participants agreed that the system was neither complex nor cumbersome at all. Most of the participants liked the fact that document sharing is done in a 'natural' way, similar to dragging physical paper documents towards a colleague to share it with them. All participants also liked the fact that they did not need to think and recall how to use any of the provided functions, thus the system posed no mental load at all, while observations during the evaluation process showed that each user became familiar enough with the system after 1-2 minutes of interacting with it.

Likeability

In general, all participants were pleased with the capabilities of SMURF Table. Most of the users expressed the opinion that although the system's interface is very simple, it provides the necessary facilities for a group meeting. Some of the users also stated that they would like to use the system for their brainstorming sessions, since these sessions usually come up spontaneously and the system allows them to gather and quickly exchange ideas, without having to make any preparations beforehand, besides uploading their documents. Overall, the participants felt that SMURF Table provides a useful collaboration surface and they were rather encouraging against the prospect of an updated version enriched with more functions.

6. Conclusions and Future Work

Ambient Intelligence is an emerging field of research whose practice in recent years has shown promising ways to support personal and social interactions in smart spaces. It is very possible that in the next few years computers will become a vital part of various environments, rather than just being machines for personal use. These computer-augmented environments, also called smart environments, will be able to provide users with natural ways to accomplish various tasks, at the same time releasing them from the load of taking actions that can be sensed and accomplished by the environment itself. The SMURF Table system reported in this thesis has been developed in order to provide a collaborative surface that can be integrated in a smart office environment. It provides a tabletop workspace for meetings among multiple users who carry a single card rather than a briefcase of documents. Personal user files are automatically loaded for use on the table without requiring any user actions, while documents are presented through already existing applications that most users are accustomed to. On-the-fly document duplication and orientation allow multiple users to share information updated in real time, while each user is able to independently work on their personal documents without affecting others' work. The SMURF Document Manager provides users with the means to stay up-to-date no matter where they may be located, at the same time allowing them to add new documents that they can use and share on the SMURF Table.

As shown from the system's evaluation outcome, SMURF Table achieves efficient multi-user collaboration through a rather simple interface, without encumbering users with unnecessary actions in order to accomplish tasks. Several system enhancements are currently planned, aiming to enrich the SMURF Table's functionality, as well as to improve user experience even further.

As already reported in Chapter 3, the system's virtual surface contains a public working area that allows documents to be duplicated on a large secondary display. Apart from the seven existing RFID antennas, an additional RFID antenna is planned to be placed underneath the table surface, right below the public area. This will allow the system to recognize various RFID-augmented objects that could be used to provide additional public interactions and widgets (e.g., a projected clock or meeting timer).

A set of seven individual speakers is also planned to be added to the system in order to augment it with audio feedback. The speakers will be placed in front of each working area, around the edges of the table, thus they should be practical in size to avoid blocking the users' movements and decreasing their comfort. Each speaker's volume level should be individually adjustable and allow users to connect personal headsets to retrieve private audio feedback. The use of audio feedback would enforce the users' perception of various events, such as the activation of a personal keyboard or the duplication of a document in their area. It would also be useful to provide public audio feedback, in conjunction with the public working area, in cases when everyone's attention should be focused on a public event or presentation on the secondary display.

The evaluation process pointed out a couple of interface related issues that are planned to be resolved in the next version of the SMURF Table. Some control widgets are planned to be added to each document window, such as an 'x' button on the top-right corner that will allow users to close windows without dragging them onto a separate tool, as well as a resize tool on the bottom-right corner that will give users the ability to resize documents at will.

Another extension of SMURF Table could also be the ability to use hand gestures in order to interact with documents. A gesture mechanism could be implemented to provide additional interaction capabilities to the current system. Hand gestures could be used to perform actions such as document resizing, duplication and free rotation. This mechanism would require the appropriate additional hardware, such as a camera, as well as an extra service running in the AmI environment for the gesture recognition analysis.

There are also several limitations to the current implementation of SMURF Table, mostly due to restrictions imposed by the operating system. As already explained, all windows used on SMURF Table are actually live thumbnail windows provided by the Windows Vista DWM module. While they have the ability to be updated in real time according to the original windows they are based on, they are not capable of reproducing application context menus. This means that when a user selects an option from the menu bar of an application, the menu that appears in the actual application window is not visible in the respective live thumbnail. Thus, user interactions are limited to the available quick-action interface elements that each application may provide, along with the left mouse button events that the system emulates.

Another feature that was initially intended to be provided by SMURF Table was to allow users to remotely occupy a working area. Windows Vista provides a set of APIs that allow applications to have remote connection capabilities, referred to collectively as the Remote Desktop Protocol (RDP). While the RDP APIs could allow remote collaboration to be realized on SMURF Table, the operating system automatically deactivates the DWM module when a remote connection is activated, thus live thumbnails cannot be used in the application. This behavior is mandatory and cannot be programmatically overridden.

A further improvement of the system would be an alternative implementation of document orientation. As already discussed in Chapter 4, window rotation is currently supported through an integrated open-source application called DWMaxx. This application supports window rotation up to Windows Vista SP1 version, which is a concern since there might be a need to move the system to a newer operating system version. Further research should be conducted regarding this issue, since the SMURF Table application itself is able to run on Windows 7 with no alterations needed.

7. References

[1] Mish, F.C. and Morse, J.M. (eds.) (1999) Merriam-Webster's Collegiate Dictionary, tenth edition (Springfield: Merriam-Webster, Inc.), p. 36

[2] Alcañiz, M. and Rey, B. (2005) New Technologies For Ambient Intelligence. In *The Evolution of Technology, Communication and Cognition Towards the Future of Human-Computer Interaction.* Amsterdam: IOS Press; p. 3-15

[3] "ubiquity", In *Merriam-Webster Online Dictionary* (2010) Retrieved March 10, 2010, from http://www.merriam-webster.com/dictionary/ubiquity

[4] Weiser, M. (1993) Some computer science issues in ubiquitous computing. In *Communications of the ACM* 36 7, July, p. 75-84

[5] Buxton, W. (1995) Ubiquitous Media and the Active Office. In *Nikkei Electronics*, 3.27 (no. 632), p. 187-195

[6] Abowd, G.D. (2004) Investigating Research Issues in Ubiquitous Computing: The Capture, Integration, and Access Problem. Retrieved March 11, 2010, from http://www.cc.gatech.edu/fce/c2000/pubs/nsf97/summary.html

[7] Daniels, A. (1997) Ubiquitous Computing. Retrieved March 11, 2010, from http://www.cc.gatech.edu/classes/cs6751_97_fall/projects/gacha/daniels_essay.html

[8] Chin, D. N. (1991) Intelligent interfaces as agents. In *Intelligent user interfaces*, J. W. Sullivan and S. W. Tyler (eds.), New York: Addison-Wesley; p. 177-206

[9] Rich, E. (1979) User Modeling via Stereotypes. In *Cognitive Science*, vol. 3, 335-366

[10]Maybury, M. and Wahlster, W. (eds.) (1998) Readings in Intelligent User Interfaces. Morgan Kaufmann Press

[11]Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J. and Burgelman, J.C. (2001) ISTAG, Scenarios for Ambient Intelligence in 2010. In *Information Society Technologies*, European Commission

[12]Chou, P., Gruteser, M., Lai, J., Levas, A., McFaddin, S., Pinhanez, C., Viveros, M., Wong, D., and Yoshihama, S. (2001) BlueSpace: Creating a personalized and context-aware workspace. IBM technical report, RC22281

[13]Lai, J., Levas, A., Chou, P., Pinhanez, C. and Viveros, M. (2002) BlueSpace: personalizing workspace through awareness and adaptability. In *International Journal of Human-Computer Studies*, v.57 n.5, p. 415-428

[14]"Q – Case Studies – IDEO: Q For Steelcase – Mobile Workstation Concept", Retrieved March 12, 2010 from http://www.ideo.com/work/item/q3/

[15]Green, K.E., Gugerty, L.J., Walker, I.D. and Witte, J.C. (2005) AWE (Animated Work Environment): Ambient Intelligence in Working Life. In *Proceedings Conference on Intelligent Ambience and Well-Being (Ambience 05)*, Tampere, Finland, September 2005, p. 1-7

[16]Green, K.E. (2006) Architecture Plus: The Collaborative *Animated Work Environment* Design Research Project. In *Proceedings of the 2006 ACSA Conference*, Salt Lake City, Utah, April 2006

[17]Stewart, J., Bederson, B.B. and Druin, A. (1999) Single Display Groupware: A Model for Co-present Collaboration. In *Proceedings of the SIGCHI conference on Human factors in*

computing systems: the CHI is the limit, p. 286-293, May 15-20, 1999, Pittsburgh, Pennsylvania, United States

[18]Tang, J.C. (1991) Findings from observational studies of collaborative work. In *International Journal of Man-Machine Studies*, 34, p. 143-160

[19]Scott, S.D., Grant, K.D. and Mandryk, R.L. (2003) System Guidelines for Co-located, Collaborative Work on a Tabletop Display. In *Proceedings of the Eighth European Conference on Computer-Supported Cooperative Work*, 14-18 September 2003, Helsinki, Finland, p. 159-178

[20]Scott, S.D., Carpendale, M.S.T. and Inkpen, K.M. (2004) Territoriality in Collaborative Tabletop Workspaces. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW)'04*, November 6-10, 2004, Chicago, IL, USA

[21]Kruger, R., Carpendale, M.S.T., Scott, S.D. and Greenberg, S. (2003) How People Use Orientation on Tables: Comprehension, Coordination and Communication. In *Proceedings of GROUP '03*, November 9-12, 2003, Sanibel Island, Florida, USA, p. 369-378

[22]Ringel, M., Ryall, K., Shen, C., Forlines, C. and Vernier, F. (2004) Release, Relocate, Reorient, Resize: Fluid Techniques for Document Sharing on Multi-User Interactive Tables. In *CHI '04 extended abstracts on Human factors in computing systems*, April 24-29, 2004, Vienna, Austria

[23]Ringel, M., Ryall, K., Shen, C., Forlines, C. and Vernier, F. (2004) Beyond "Social Protocols": Multi-User Coordination Policies for Co-located Groupware. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, November 06-10, 2004, Chicago, Illinois, USA

[24]Morris, M.R., Cassanego, A., Paepcke, A., Winograd, T., Piper, A.M. and Huang A. (2006) Mediating Group Dynamics through Tabletop Interface Design. In *IEEE Computer Graphics and Applications*, 26 (5), p. 65–73, 2006

[25]Wang, X. and Maurer, F. (2008) Tabletop AgilePlanner: A Tabletop-based Project Planning Tool for Agile Software Development Teams. In *Proceedings of the 3rd IEEE Tabletop and Interactive Surface*, Amsterdam: IEEE Press, p. 129-136

[26]Piper, A.M. and Hollan, J.D. (2008) Supporting Medical Conversations between Deaf and Hearing Individuals with Tabletop Displays. In *Proceedings of the ACM 2008 conference on Computer supported cooperative work*, November 08-12, 2008, San Diego, CA, USA

[27]Morris, M.R., Piper, A.M., Cassanego, A. and Winograd, T. (2005) Supporting Cooperative Language Learning: Issues in Interface Design for an Interactive Table. Stanford University Technical Report, 2005

[28]Jordà, S., Geiger, G., Alonso, M. and Kaltenbrunner, M. (2007) The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, February 15-17, 2007, Baton Rouge, Louisiana

[29]Russell, D.M. and Gossweiler, R. (2001) On the Design of Personal & Communal Large Information Scale Appliances. In *Proceedings of UbiComp*, 2001, Atlanta, GA, U.S.A.

[30]Ashdown, M. and Robinson, P. (2005) Escritoire: A Personal Projected Display. In *IEEE MultiMedia*, v.12 n.1, p.34-42, January 2005

[31]"Perceptive Pixel – Multi-touch Wall and Multi-touch Workstation", Retrieved January 10, 2010, from http://www.perceptivepixel.com/solutions.html

[32]"Ideum – MT-50 Multi-touch Table", Retrieved January 10, 2010, from http://www.ideum.com/products/multitouch/

[33]Dietz, P. and Leigh, D. (2001) DiamondTouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, November 11-14, 2001, Orlando, Florida

[34]"Welcome to Microsoft Surface", Retrieved January 10, 2010, from http://www.microsoft.com/surface/en/us/Default.aspx

[35]Hollemans, G., Bergman, T., Buil, V., van Gelder, K., Groten, M., Hoonhout, J., Lashina, T., van Loenen, E. and van de Wijdeven, S. (2006) Entertaible: Multi-user multi-object concurrent input. In *Adjunct Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST06)*, ACM, p. 55–56

[36]"SMART – Front projection", Retrieved January 10, 2010, from http://www2.smarttech.com/st/en-US/Products/SMART+Boards/Front+projection/

[37]SMART Technologies Inc. (2003) DViT, Digital Vision Touch Technology. White paper, Retrieved January 10, 2010, from http://smarttech.com/DViT/DViT_white_paper.pdf

[38]Shen, C., Everitt, K.M. and Ryall, K. (2003) UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. In *Proceedings of UbiComp'03*, p. 281-288

[39]Shen, C., Vernier, F.D., Forlines, C. and Ringel, M. (2004) DiamondSpin: an extensible toolkit for around-the-table interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, p.167-174, April 24-29, 2004, Vienna, Austria

[40]Morris, M.R., Morris, D. and Winograd, T. (2004) Individual audio channels with single display groupware: effects on communication and task strategy. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, November 06-10, 2004, Chicago, Illinois, USA

[41]Morris, M.R., Paepcke, A. and Winograd, T. (2006) TeamSearch: Comparing Techniques for Co-Present Collaborative Search of Digital Media. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, p.97-104, January 05-07, 2006

[42]Na, S., Billinghurst, M. and Woo, W. (2008) TMAR: Extension of a Tabletop Interface Using Mobile Augmented Reality. In *Transactions on Edutainment I*, Springer, Berlin, p. 96-106

[43]Streitz, N.A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P. and Steinmetz, R. (1999) i-LAND: an interactive landscape for creativity and innovation. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, p. 120-127, May 15-20, 1999, Pittsburgh, Pennsylvania, United States

[44]Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K. and Tachibana, K. (2000) Virtual object manipulation on a table-top AR environment. In *Proceedings of the Third International Symposium on Augmented Reality (ISAR 2000)*, Munich, Germany. New York: IEEE Press, p. 5–6, October, 2000

[45]Izadi, S., Brignull, H., Rodden, T., Rogers, Y. and Underwood, M. (2003) Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, p. 159-168, November 02-05, 2003, Vancouver, Canada

[46]Wu, M. and Balakrishnan, R. (2003) Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, p. 193-202, November 02-05, 2003, Vancouver, Canada

[47]Hunter, S. and Maes, P. (2008) WordPlay: A Table-Top Interface for Collaborative Brainstorming and Decision Making. In *3rd IEEE Int. Workshop on Horizontal Interactive Human Computer Systems*, Amsterdam, the Netherlands, 2008

[48]Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N. and Steinmetz, R. (2001) ConnecTables: Dynamic Coupling of Displays for the Flexible Creation of Shared Workspaces. In *Proceedings of the 14. Annual ACM Symposium on User Interface Software and Technology (UIST'01)*, ACM Press (CHI Letters 3 (2)), 2001, p. 11-20

[49]Wilson, P.H., Duckworth, J., Mumford, N., Eldridge, R., Guilielmetti, M., Thomas, P., Shum, D. and Rudolph, H. (2007) A virtual tabletop workspace for the assessment of upper limb function in traumatic brain injury (TBI). In *Proceedings of the Virtual Rehabilitation Conference*, IEEE, p. 14-19

[50]Wilson, A.D. (2005) PlayAnywhere: a compact interactive tabletop projection-vision system. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, October 23-26, 2005, Seattle, WA, USA

[51]Haller, M., Brandl, P., Leithinger, D., Leitner, J., Seifried, T. and Billinghurst, M. (2006) Shared Design Space: Sketching ideas using digital pens and a large augmented tabletop setup. In *ICAT 2006, Lecture Notes in Computer Science 4282*, p. 948–959, Springer Verlag, 2006

[52]Kubicki, S., Lepreux, S., Lebrun, Y., Santos, P.D., Kolski, C. and Caelen, J. (2009) New Human-Computer Interactions using tangible objects: application on a digital tabletop with RFID technology. In *HCII 2009*, San Diego, CA, July 2009

[53]Hinske, S. and Langheinrich, M. (2007) RFIDice - Augmenting Tabletop Dice with RFID. In *4th International Symposium on Pervasive Gaming Applications (PerGames)*, Salzburg, Austria, 2007

[54]Floerkemeier, C. and Mattern, F. (2006) Smart Playing Cards - Enhancing the Gaming Experience with RFID. In *Proceedings of the Third International Workshop on Pervasive Gaming Applications - PerGames,* Dublin, 2006

[55]Booth, P. (1989) An Introduction to Human-Computer Interaction. Lawrence Erlbaum Associates, London

[56]Brooke, J. (1996) SUS: A "quick and dirty" usability scale. In *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester and A. L. McClelland, Eds. Taylor and Francis, London, p. 189–194

[57]Object Management Group (1998) The Common Object Request Broker: Architecture and Specification. Object Management Group, Framingham, MA

Appendix – System Usability Scale

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5