Task Completion by Flow

A new holistic interaction paradigm for browser based business software

Thesis for the degree of Master of Science in Human Media Interaction

December 2007

Corné Verbruggen
University of Twente, The Netherlands
corne@g-majeur.nl

Committee:
Dr. Udo Arend (SAP)
Dr. Betsy van Dijk (University of Twente)
Dr. Paul van der Vet (University of Twente)

Conducted at the department of Research & Breakthrough Innovation of SAP AG, St. Leon-Rot, Germany

Abstract

For various good reasons, many businesses are moving from traditional desktop applications towards web based applications. But especially in web based applications, the user system interaction lacks any fluidity that could keep the user in the task flow. Jumps and breaks reduce the usability. Furthermore, the transaction orientedness and static paths of actions in these applications reduce the usability even more.

We developed a new interaction paradigm, adopting the notion of an information space containing the business data. Landmarks, contextual information and animation were used to deliver good navigation and direct manipulation in this information space. A graph-based conceptual model was created, that resulted in the concrete interaction paradigm.

The paradigm was implemented in a personnel administration system. Qualitative evaluation of a high-fi prototype with a small group of participants demonstrated that the new interaction paradigm is very promising for the creation of business software with a more fluid user system interaction within the restrictions of a browser environment.

PREFACE

The title of this thesis is somewhat misleading when interpreted as a characterization of the process of completing this work; staying in the flow was quite a challenge every now and then. But actually, most jumps and breaks were interesting and I would not have missed many of them. Therefore I would like to thank those people that made the breaks worthwhile and enjoyable, as well as those – often the same people – that helped me get back into the flow. Thanks to the members of my graduation committee – Udo Arend at SAP and Betsy van Dijk and Paul van der Vet at the university of Twente – for their enthusiasm, critics and comments. Thanks to the colleagues at SAP for the highly creative (non)sense brainstorm sessions, the "Kicker" games (especially those on my last day ;-) and all else. Thanks to family and friends for staying in touch, or even visiting me in Große Kreisstadt Leimen! I really enjoyed that.

Corné Verbruggen

Groningen, December 2007

CONTENTS

Chapter 1

INTRODUCTION

Personnel administration is one of the components, or so-called *work centers,* of SAP's new Business ByDesign system, a complete Enterprise Resource Planning (ERP) suite, geared towards Small and Medium Businesses (SMB). This new "All-in-One" product is offered as a hosted service, enabling customers to manage all their business processes using a web browser, without having to worry about the technical and hardware related challenges. Therefore, it should be a good solution for companies with limited in-house IT resources.

ERP software in general can be pretty complex. Not only from a technological point of view, but especially from its user's point of view. In addition to a more or less clear skill set, it also requires patience and tolerance to be able to work with it. This will come as no surprise, as the processes that are dealt with simply are complex. The word *task flow* is often used to describe the steps a user of a system has to take to accomplish a certain task. But actually, *flow (*defined as the mental state of operation in which the person is fully immersed in what he or she is doing, characterized by a feeling of energized focus, full involvement, and success in the process of the activity[1]) is one of the experiences a user will seldom – if not never – encounter.

It is a challenge to overcome these limitations and design an interaction paradigm that eliminates the need to try to make things doable by training users and providing quick fixes that do not really solve the problem at its root. While this holds true for the whole suite of ERP software solutions, this thesis focuses on personnel administration as an example. It contains most of the important more or less generic task elements that can be found in ERP software.

1.1 Problem description

Based on oral reports from the SAP supervisor of this work, the problems with the current SAP personnel administration software can be grouped into three more or less overlapping categories. These are described in the next sections. More detailed information on the personnel administration task can be found in chapter 2.

1.1.1 Transaction orientedness

The user interface of the personnel administration software is heavily influenced by the transactional nature of the database system behind it. All user actions in fact result into a transaction within the database containing all personnel data. This is reflected in the user

interface. Most screens the user sees, correspond to one transaction. The data of an employee are displayed, they can then be edited and saved afterwards. This approach focuses on *declarative* knowledge while most tasks actually are *procedural* in nature. The effect of this transaction oriented, declarative approach is a user-system interaction consisting of *jumps*, and *breaks*, lacking any kind of *fluidity* that would do justice to the procedural nature of the tasks being performed. There is no *flow* in the interaction. For browser based applications, this problem is even more serious as the duration of these breaks are mostly longer due to network latency, bandwidth constraints and the repeatedly slow rendering of pages. We expected the user to prefer an interaction that better supports this procedural nature. The results support this expectancy, but it might be the case that some type of user prefers to use the more traditional approach or even some other. This question is interesting on its own for further investigation.

1.1.2 Statical paths of action

The way a user performs a task is dictated by the static nature of the system. Instead of giving the user the freedom to move through the information space the way that seems most sensible to him, predetermined jumps must be made to get where he wants to get. For example, when the user wants to process a work item from his work list, he is dropped in the middle of the information space, opened in a new window showing the data of the employee involved. Despite the notion of an *information space*, the user is not free to move through it according to good principles of wayfinding. This causes the user to easily get lost in the information space, getting off the path that leads towards successful completion of the task. While many windows and views are involved, it is hard to keep track of all of them and their interconnectedness, especially when being interrupted by e.g. a telephone call. Creating a more holistic way of interacting with the application and giving the user the freedom to move around freely and supporting him during his navigation, would be more supportive. The results indicate that users like such an approach. Of course, also with respect to this concern, it would be interesting to explore more possibilities and better investigate and identify different types of users with different preferences.

1.1.3 Not fun

It might seem inappropriate to talk about fun when looking at a serious business software solution. But in fact, most usability books and references mention the fun-factor as one of the most important success criteria for software. Fun, in this context, is interpreted as the quality of being enjoyable to work with. The previous two categories of problems already indicate that the user will not enjoy working with the personal administration software as much as is possible. The overall satisfaction is too low. Though this, in fact, is part of the other two problem categories described in the previous sections, we mention it here as a separate problem while the relevance of it is so important when discussing concepts like *flow*.

1.2 Goals

The goal of this thesis work is to develop a new interaction paradigm for the personnel administration application that solves the problems that were described in the previous section. To be more precise, the following research questions should be answered:

- How should the rather procedural tasks related to personnel administration be modelled
    - to identify crucial interactions between the user and the system,

  - and provide a cognitively plausible basis for the design of a user interface,
  - that optimally supports the user in performing these tasks?
- How can this model be translated to a concrete interaction paradigm
  - that solves the problems we mentioned
  - within the constraints a browser environment poses?

Much related work has been done in past years, but most of it focuses on one of these two broad topics: *navigation* on the one hand and *data manipulation* on the other. A major challenge of this thesis work will be to integrate these two, because both navigation and data manipulation are part of the personnel administration task.

With regard to modelling the task, the major challenge will be to find a suitable technique while most existing task flow modelling paradigms focus on the technical aspects of the system, and not on the user experience.

1.3 Approach & Thesis structure

The major source of input for this thesis is the personnel administration task description. Based upon information available at SAP, we first define a clear task description that identifies the major concerns from a user perspective (chapter 2).

As a lot of research has been done in the area of user interface design and related domains, there are good ideas, proven principles and rock solid theories available that provide valuable input to this thesis work. Therefore, chapter 3 provides a theoretical background, along with some examples demonstrating how the theory has currently been applied to probably useful products and interface concepts. The latter goes beyond the study of what has been implemented in mainstream software and other products. Good ideas do not always end up in real-life solutions. So, to prevent ourselves from reinventing the wheel, we have to see what solutions scientists and designers already came up with during the last decades.

These two main sources of information resulted in a conceptual model for the type of tasks identified and the subsequent design of a new interaction paradigm. This paradigm is concretized in the design of a new user interface for the personnel administration software (chapter 4). To be able to find out whether this new front-end really solves or reduces the problems we identified, a high-fidelity prototype is implemented using a combination of XHTML, CSS and JavaScript (chapter 5), and qualitatively evaluated with a small group of users (chapter 6).

This thesis is neither the closing run of the whole user experience business, nor the one and perfect solution for all user interactions with Enterprise Resource Planning applications. The results are discussed and some thoughts for future challenges and opportunities given in chapter 7.

Chapter 2

PERSONNEL ADMINISTRATION

This chapter describes the Personal Administration task that has to performed with the personnel administration software. In the SAP Business ByDesign system, personnel administration is one of the *work centers*, but it has connections with other work centers, especially with the Human Resource Time Administration work center and the Payroll work center. These two deal with – as can be guessed – the working time registration for employees and the payroll process. The personnel administration work center itself has the sole global purpose of giving its user the tools to

- *find* information about an employee,
- *edit* data of an employee,
- *add* a new employee and
- *delete* an employee.

Besides having this functionality, it can also act as a launching point into the other, associated work centers due to relatedness of information contained in the system. Exactly what functionality is available to the user depends on his position and role within the organization.

Before we describe the personnel administration task in section 2.2, we first define the employee business object, while it is the subject of all actions in the task.

2.1 The employee business object

Most data held in SAP's systems is perceived of as *business objects*, for example a customer business object or a salary business object. Every type of business object is described by its clearly determined set of attributes. For example, a customer business object has attributes like name, address and contact person. Depending on its type, a business object can have a few or lots of attributes. Attributes can also be mere pointers to other business objects, as is the case with the contact person attribute, which refers to a contact person object. This way, different types of business objects are interrelated.

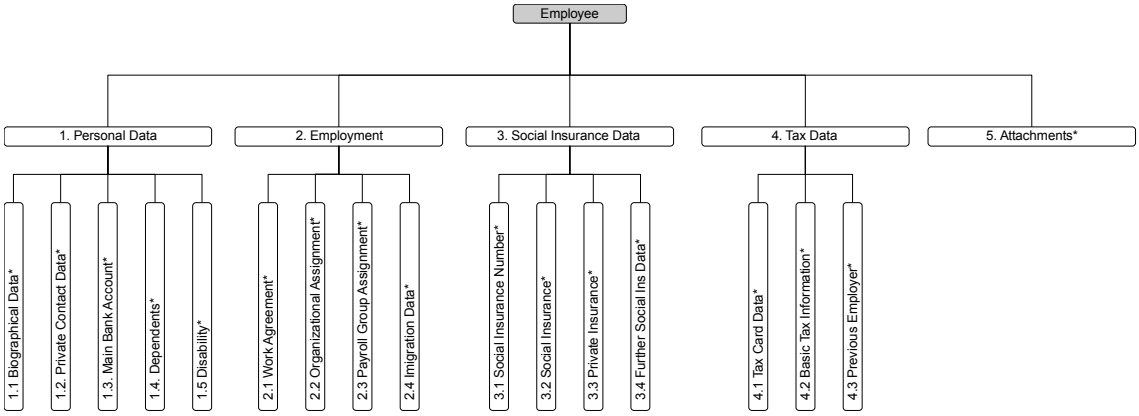


Figure 2.1: The employee business object and its categories of attributes

The most important business object in the personnel administration software is the *employee*. Employees can be added, removed, data can be modified and so on. This employee business object can best be described by its attributes. Attributes are grouped into five categories, each holding one ore more subcategories, as is depicted in figures 2.1 - 2.6. These subcategories contain the actual attributes. In total, an employee business object has over 150 attributes, spread over seventeen subcategories. In fact, it has more attributes, in the sense that every set of attributes corresponding to one subcategory is valid for a certain period of time. Thus, multiple entries for every set of attributes can be stored. For example, an employee might get a new bank account and prefer to receive his salary on this account starting the next month. In this case, a new set of attributes is created for the *Main Bank Account (1.3)* subcategory, having the first day of the next month as start date. The already present set of attributes gets the last day of the current month as end date. These types of attribute sets are *non-overlapping*. There also are attribute sets that are *overlapping*. For example, an employee can have a spouse and two children at the same time. Corresponding to this, multiple attribute-sets are stored under *Dependents (1.4)*.

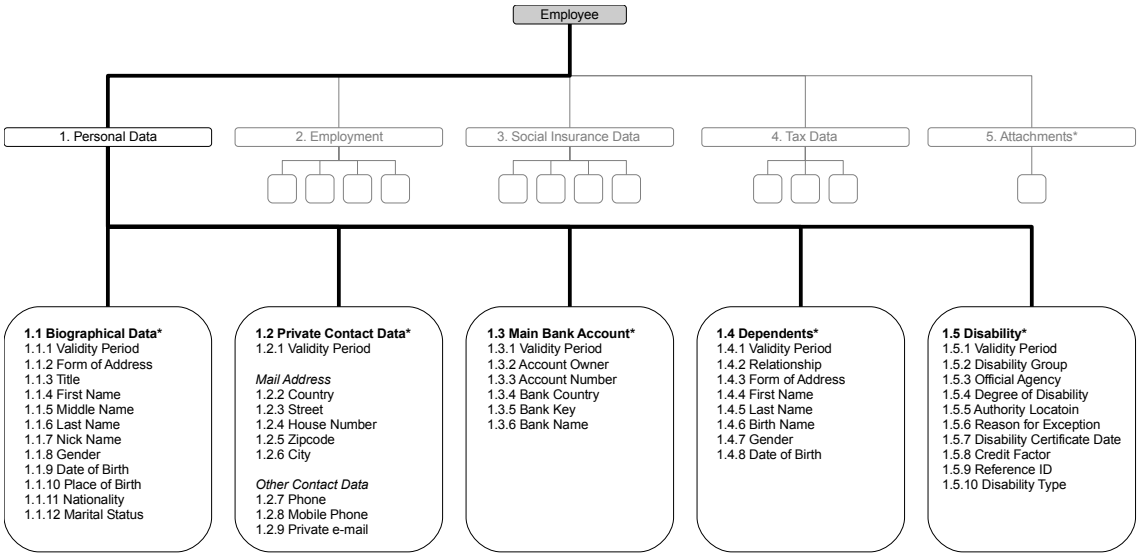


Figure 2.2: The employee business object's attributes in the five subcategories of the Personal Data category

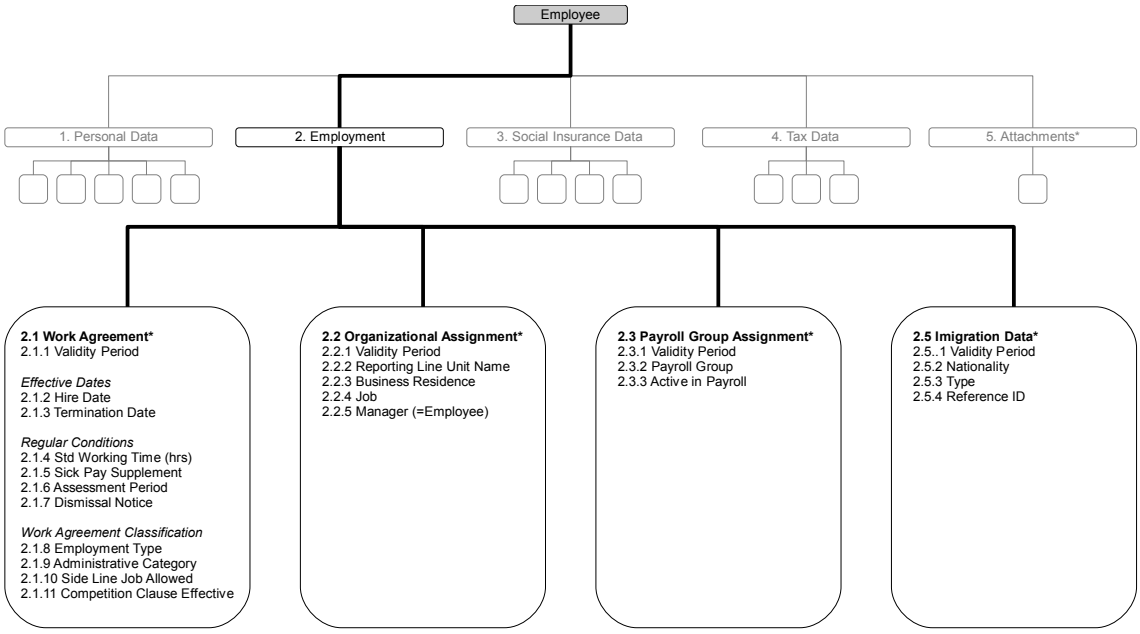


Figure 2.3: The employee business object's attributes in the four subcategories of the Employment category

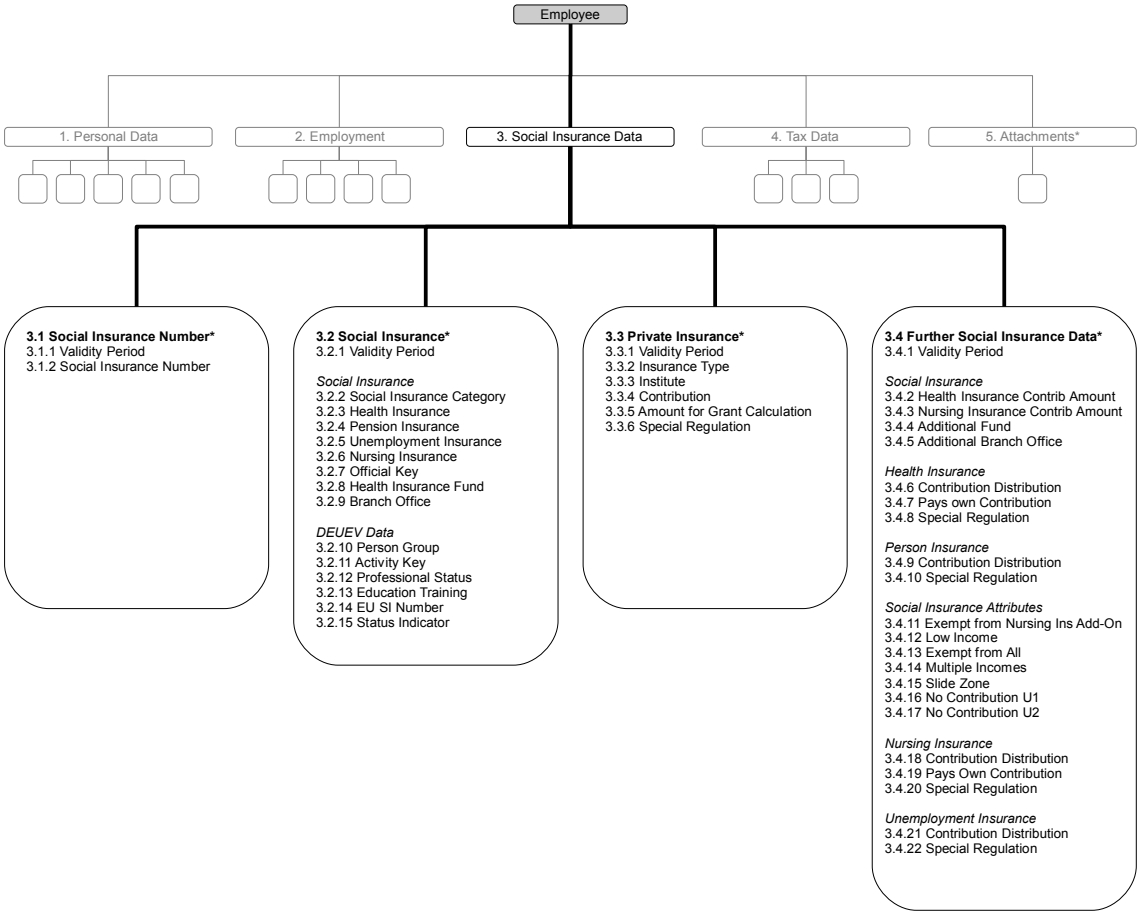


Figure 2.4: The employee business object's attributes in the four subcategories of the Social Insurance Data category

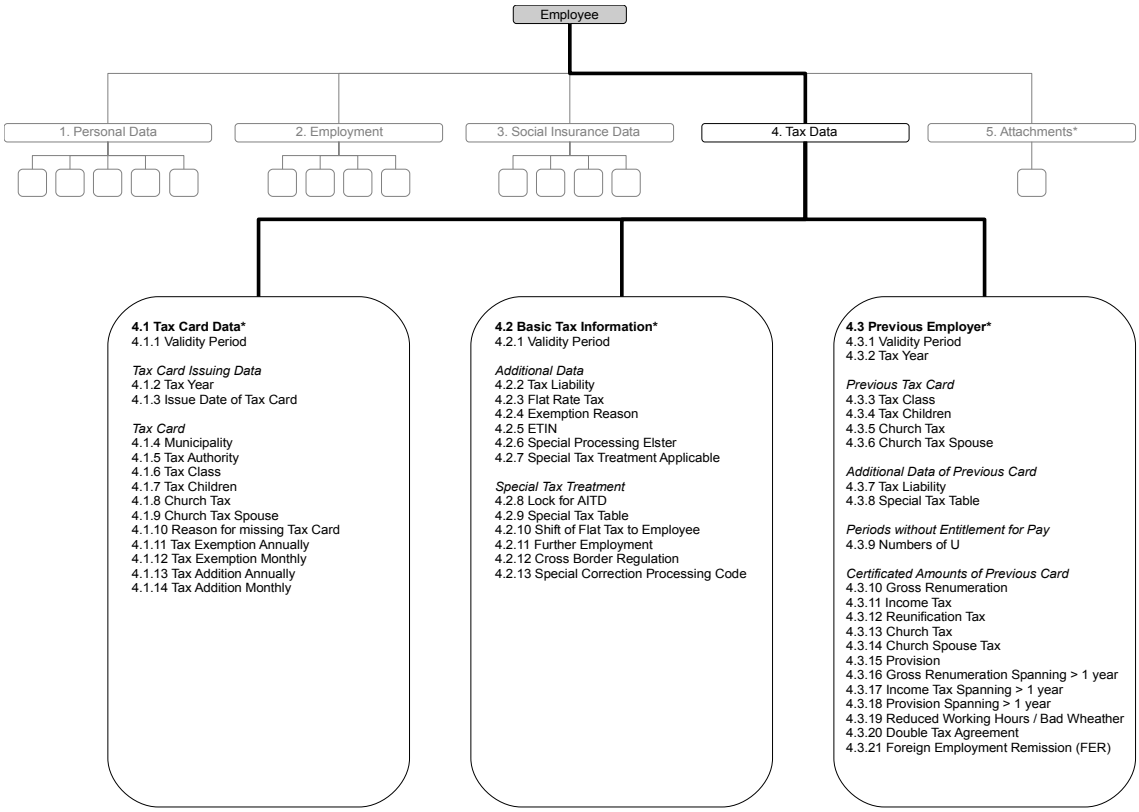


Figure 2.5: The employee business object's attributes in the three subcategories of the Tax Data category

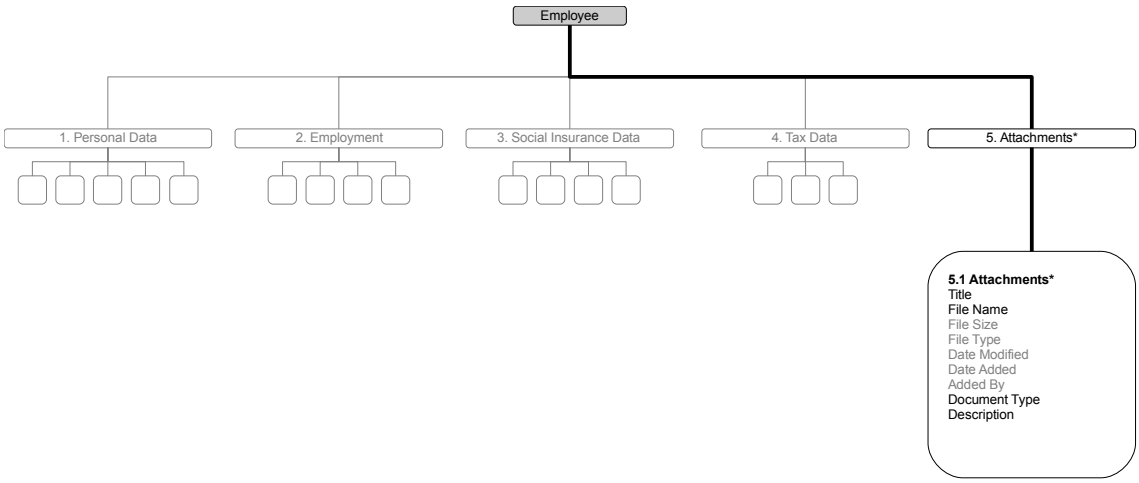


Figure 2.6: The employee business object's attributes Attachments category

2.2 The task

The personnel administration task can be perceived as consisting of several sub-tasks:

1. **Finding information about an employee.** First of all, the right employee business object has to be found. This is done by opening the list of employee business objects, searching for the business object and selecting it. Searching can be done visually, by scanning and browsing the list, or by using filters like name or id-number. When the correct business object is selected, the required information has to be found within the 150 attributes.

2. **Editing data of an employee.** To edit data (attributes) of an employee business object, the first steps are identical to those needed to find information about a certain employee. When the relevant attributes are found, new values must be entered and saved.

3. **Adding a new employee.** A new employee business object has to added. This requires a certain set of required values to be filled in. Afterwards, the business object is saved and thus added to the list of employee business objects.

4. **Deleting an employee.** First of all, the relevant employee has to be found in the list. When found, it is simply deleted. In reality, the data will still be kept on the server for historical reasons and undo-functionality. The delete action is thus more a kind of putting-aside action.

The user of the personnel administration system can start a task in several different ways:

1. **Own initiative.** The user has a task in mind that he wants to perform. This task can originate from multiple sources, for example a telephone call from an employee who wants to know whether his correct private phone number is stored in the system. Depending on the type of task and the preferences of the user, he can perform the task in two different ways:

   a) Directly act upon the list of business objects by adding, deleting and modifying entries.

   b) Initiate a task from a list of predefined *Personnel Events*, which are a kind of template tasks.

2. **A work item.** The system maintains a list of work items the user has to perform. These work items can be added by other people than the user himself, for example by a manager. This way, they act as *triggers*. Every work item corresponds to exactly one employee business object.

This thesis work focuses on 1a and 2. The first (own initiative, directly acting upon the objects) is interesting, as it gives the possibility to see whether the new approach establishes a good situational awareness within the information space (*navigation*). It also fits very well with the idea of *direct manipulation* (more on this in the next chapters). The second (triggers), in addition to this, is interesting, as it provides the possibility to see how we can use a kind of *shortcuts* into the information space, associating work items with business objects, while still maintaining situational awareness.

Situational awareness in the context of this thesis work should be interpreted as receiving cues from the information space through the system, and using them to make sense of the current

state of the system, parallel to how situational awareness is most often defined as a human factor[2]. Having good situational awareness thus means that the user knows *where* he is in the information space, *how* he got there, *why* he is there and *what* next actions are possible.

Chapter 3

THEORETICAL BACKGROUND AND RELATED WORK

The problem to be solved cannot be strictly positioned into one clearly defined research area or topic as it is multi-faceted and will have to draw upon results from multiple backgrounds. Integrating these results was the major challenge. We tried to design a new interaction *paradigm,* covering not only elementary tasks like *browsing online documents*, *searching a list* or *manipulating data*, but a combined total task flow supporting interface.

When using the notion of an information space containing all the data, *navigation* (section 3.1) and *object manipulation* (section 3.2) are two of the most important topics relevant for this thesis work.

Supporting the user by providing a good navigation system is crucial. A central element of this thesis work therefore is the usage of *landmarks* as navigational cues (section 3.1.1). The second topic related to navigation is *context* (section 3.1.2). Providing proper contextual information helps users to understand where they are in the information space. Since it is useful for the discussion of a new interaction paradigm, we explore a *conceptual model* (section 3.1.3) that incorporates these two navigation related topics. From the study of conceptual models, it becomes clear that moving around in the information space can be interpreted as a path consisting of a succession of moves. In addition to landmarks and proper contextual information, a topic of great importance to this "moving around" is the relevance of *optic flow* (section 3.1.4): continuous visual information that can be a significant aid to wayfinding.

The third section of this chapter (section 3.3) shortly discusses *enjoyment*, a topic related to the discussion in sections 3.1 and 3.2 through the usage of animation.

The next two sections give some background information on characteristics of web based applications (section 3.4) and some notes on the application of innovative user interface ideas in a business setting (section 3.5), both interesting topics because we are designing a new interaction paradigm for business software, within the limitations of a web browser environment.

Finally, in the latest section (3.6 - Wrap up) we try to integrate the theories and work discussed in the other sections of this chapter and see how they are relevant for the design of the new interaction paradigm.

Along with the more theoretical discussion, some sections of this chapter give an example of application of the theory in concrete concepts or designs.

3.1 Navigation

In this section, we give an overview of related work that might shed light on the navigation problems arising in the personnel administration software. A lot of this work is related to 3D navigation in virtual environments. It is suggested that such simulated environments can be used to study navigation [3]. The relevance of the results of these experiments for wayfinding in information spaces should of course be critically studied, but the results from this thesis work suggest that their application onto this domain is fruitful. We have the impression that these principles for navigating in virtual 3D environments give us insights into the perceptual and cognitive principles that also hold true for our domain of interest: navigating through information spaces.

3.1.1 Landmarks as navigational cues

When navigating in the real world, we use so-called *landmarks* to memorize locations. This helps us to find the same place or route again later. A distinction can be made between local and global landmarks [4]. Local landmarks are visible only from a small distance and provide *views* that can lead to a *view-triggered response*. The next movement can then be specified with respect to the current observer orientation, which is implied by the recognized view. The observer remembers something like "When facing the city hall, turn right". Global landmarks, on the other hand, function more or less like a compass. They are visible from a large distance and help the observer to orientate in case of *place-recognition responses*, like "When arriving at the market place, walk south". This dichotomy between *views* and *place*s is central to navigation and is a basic element of the *view graph theory of spatial memory* [5] which we will discuss in section 3.1.3 (Conceptual models for understanding the information space).

An experiment investigating the role of local and global landmarks pointed out that both types of landmarks are used in the decisions for way-finding tasks, but different participants rely on different strategies [4]. Some of them used only local landmarks, some used only global ones and some used both. However, the *salience* of a landmark influenced the choice of strategy. When a very salient local landmark was available, the local strategy was preferred, neglecting the compass-information provided by the global landmarks. This can be interpreted as a view-recognition-triggered response. The experiment also demonstrated that participants not only showed *route-like knowledge* afterwards, but also *map-like or survey knowledge*. Route-like knowledge is information about the sequence of actions required to follow a particular route, to find a certain goal, and is built by connecting isolated bits of knowledge about landmarks into chains. Map-like knowledge exceeds this by encompassing information that will enable one to determine the effects of a certain action at a certain decision point [4].

An extension to the functionality of landmarks can be realized by *trail creation*, encoding the history of movements through the environment. Just like the real physical world, non-linear information space gives its user the freedom to choose which path to follow. But this requires that the user somehow must keep track of where he is. Otherwise, he might get lost. Efficiency can be gained from paths followed earlier, especially if they are marked with breadcrumbs or pebbles [6]. Local landmarks could in some cases have this function, but this is not completely the same. A view of a certain place might differ depending on the direction one comes from. Trails are more explicit additional signs, created by the user explicitly or as a result of his actions.

Due to the nature of the system, possibilities are in some sense greater than in a real physical world. In a real world, only the last part of the trail is visible. In a graphical user interface, it is possible to give the user a view on a larger part of the trail. This gives him the freedom to jump

straight to a previous location in the trail, without travelling back step by step. In the prototype we developed, the latter idea is clearly present and it seems to work very well.

3.1.2 Context

Besides landmarks, which actually are a kind of context, the context in a broader sense is of great importance to the user. An experiment investigating the effects of two types of contextual navigation on the performance of tasks in differing web systems revealed that good context information significantly improves task performance [7]. A distinction is made between *spatial or structural* context on the one hand, and *temporal context* on the other. Spatial context is needed to answer the question "Where can I go from here?", whereas the temporal context should answer the question "How did I get here?". This contextual information explains the current situation to user. Lack of it results in what has been summarized as the *lost in hyperspace phenomenon*, which is classified as either *disorientation* or *cognitive overhead* [8, 9].

Disorientation is defined as "the tendency to lose one's sense of location and direction in a non-linear document" [10]. Four types of disorientation are particularly relevant in hypertext (non-linear) systems:

- Not knowing where to go next.
- Not knowing how one arrived at a particular node.
- Not knowing where the information is.
- Not knowing how to get there, that is, not knowing which path leads to the place containing the information.

The personnel administration system strictly speaking is not a plain hypertext system, consisting of documents connected by associative links, but it *is* non-linear. Especially regarding work items that refer to (or are associated with) employees, this non-linearity is clearly a point of concern. Therefore, the four types of disorientation were carefully kept in mind during the design of the new paradigm.

Cognitive overhead is defined as "the additional effort and concentration necessary to maintain several tasks or trails at one time" [10]. These tasks can consist of remembering sequences, searching target items, browsing topics and related items, comparing between items, moving from one item to another, and so on. Performing all these tasks simultaneously causes cognitive *overload*.

Context information provides the temporal and structural cues of location that can prevent disorientation. At the same time, users without context information tend to experience cognitive overload induced by cognitive overhead, because context information also provides valuable cues for users' actions and task flows. This problem is inherent to hypertext systems that, in contrast to linear documents, hinder the user to maintain good context information. It can be solved by providing the previously mentioned two types of context information: spatial and temporal. It is suggested that a web systems with context information is perceived as more convenient for navigation than those without it [7]. Fewer nodes are visited repeatedly. This holds true in different types of web systems. Structural context reduces navigation problems by *previewing* information, and temporal context reduces them by *reviewing* information. More in general, good *orientation cues* should solve the problems by enabling users to:

- Identify their current position with respect to the overall structure;
- Reconstruct the path that led to this position;

- Distinguish among different options for moving on from this position.

It is clear that landmarks in combination with a trail – as discussed in the previous section – can easily fulfil the function of orientation cue.

While it is clear that context is very important to the user of a system, the total amount of context a user can handle is another question. It is known that one can focus only on a very limited amount of items at the same time [2, 11] . In addition to this limitation in *number* of items one can focus on, there are also limitations to the *size* of the area that can have focus. The human visual system can only perceive detailed information in the foveal region (center of the visual field). The surrounding regions are only perceived in low resolution for daylight colour vision and night-time monochrome (contrast) vision [12]. A graphic user interface with too much context, or too much detailed information in the context might distract attention from the activity at the current location and raise the complexity-level of an application above reasonable levels. Besides, screen space is mostly very limited. Although the average screen space for personal computers increased during the last years, displays are still pretty small compared to dimensions of real physical spaces and objects. Therefore it is important to waste no space on needless contextual information, and save space for truly relevant context. Furthermore, usage of colour and contrast in contextual information, and leaving out small details, fits very well to the capabilities of the human visual system [12].

Different solutions have been proposed and applied to provide context in computer applications. A very popular approach is to show two views. The first holds the context, the second the details of one zoomed in item selected in the context view. Most popular e-mail clients like Microsoft Outlook and Mozilla Thunderbird use this approach. Nevertheless, this type of interface has some serious drawbacks. First of all, the spatially indirect relation between the context-view and the detail-view might strain memory and increase the time used for visual search. Second, this type of interface requires a lot more screen space than those without a context.

Still, users in most cases prefer to have the context-view. Its drawbacks can be reduced by using zoomable user interface. The main characteristics of zoomable user interfaces are (a) information objects are organized in space and scale, and (b) users interact directly with the information space, mainly through panning and zooming. Space and scale are thus the fundamental means of organizing information [13]. Different types of zooming can be distinguished. The simplest form is geometric zooming where the scale linearly determines the apparent sizes of the object. Zooming the view of, for example, a photograph, in many cases is geometric zooming. There is actually nothing more to see in the zoomed-in view, but what is visible is displayed larger and can thus be more easily perceived. A more complex relation between appearance and scale is realized with semantic zooming, where the amounts of detail and features depends on the scale [13, 14]. A good and well known example of this is the way online map systems[1] implement zooming in the user interface. When zoomed out, only countries and their respective names are displayed. When zooming in, the part of the map that is visible is reduced, but the amount of information available for the area in view is increased. Cities, smaller towns and rivers are added. More meaning is added to the map. The goal is to give the most meaningful presentation at each scale.

Changing scale in zoomable user interfaces can be done in two ways. *Jump zooming* changes the scale instantly, without a smooth transition. In *Animated zooming*, on the other hand, the transition from the old to the new scale is smooth. It has been demonstrated that animated zooming leads to better reconstruction of the topology of the information space [15]. Animation is discussed in more depth in section 3.1.4.

1 Two popular examples are Google Maps (http://maps.google.com) and Yahoo Maps (http://maps.yahoo.com).

The work on Fisheye views [16, 17] and menus [18] demonstrate a type of zoomable user interface. The fundamental motivation is to provide a balance of local detail and global context, trading off so-called *a priori* importance against distance. This means that only increasingly important features further and further away are shown. This approach has proven to be especially effective in, but not limited to, browsing tasks. The requirement is that distance and some display-relevant notion of *a priori* importance can be defined. This is the case for general graphs, which makes this approach even more interesting in combination with the *view graphs* discussed in section 3.1.3. The See-Through Interface [19], discussed in more detail in section 3.2, also borrows in some extent from this approach.

3.1.3 Conceptual models for understanding the information space

Although, as described in the previous sections, landmarks and navigational cues (context) seem to be very important aids in wayfinding, it is suggested that the user's ability to understand the structure of the information is even more important [20]. People differ greatly in the way they conceptualize (and thus understand) an information space and it is impossible to create *the* model that resembles the *mental model* users maintain, simply because we do not know what it looks like. Results from [20] indicate that information spaces are conceptualized differently by different users. This suggests that the users' mental models also differ greatly. Nevertheless, it is suggested that certain factors can positively or negatively influence the ease with which users can construct *a* mental model of non-linear documents, whatever that model is like[9]. Two factors in particular are crucial in this respect: *coherence* as positive influence and *cognitive overload* as negative influence[9]. A document is considered coherent if a reader can construct *a* mental model from it that corresponds to facts and relations in a possible world [21].

Since describing *the* mental model is not possible, the best thing we can do is create a *conceptual* model that seems to make sense to the users. This model can be used to explain how the information space is ordered, what it consists of and how various elements are connected. When users prove to be able to easily understand and remember it, it is a good model. But that does not take away the fact that for different (types of) users, different conceptual models might be needed to improve the ability for them to understand the information space, as the results from [20] already showed.

One type of conceptual models is the graph-based model of space [22] which describes an environment by means of nodes and edges, roughly corresponding to places and their spatial relations. Such a model can be used to explain wayfinding behaviour such as exploration and route planning. The graph concept serves several purposes in spatial cognition:

- Graphs are conceptual models for the representation of environments, serving as working hypothesis for the structure, format and content of spatial memory which can be empirically tested.
- Graphs can describe the set of movement actions available at a given place.

Furthermore, single edges of graphs can be labelled or weighted to include distance and direction, allowing for metric navigation abilities such as *short-cutting behaviour.* Additionally, non-spatial information like emotional and episodic information can be attached to nodes. In short, graph-like mental representations of space, according to [22], are ecologically plausible, sufficient for the explanation of behaviour and they also fit well to the neural structure of our brains. Whether or not the latter is correct, we assume that graphs can be very useful as a conceptual model to describe the information space, which is what we did (see the next chapter). The results do not contradict this assumption and the graph we created is at least helpful as a basis for the creation and explanation of the actual interaction design.

A distinction is made between *place graphs* and *view graphs* [5, 22]. In place graphs, nodes correspond to single places or positions within an environment and edges describe the connectivity between nodes. In fact, they are purely topographical representations of space. In view graphs, on the other hand, nodes correspond to pictorials snapshots (views) of the environment as seen when navigating a given place transition. Nodes are connected by edges if the corresponding views can occur in immediate sequence while walking through the environment. View graphs were proposed as a minimal spatial memory model [5], and have successfully been used for the explanation of human navigation behaviour [4]. In fact, this is what the *view-triggered-response* is about (section 3.1.1). The path along the edges of a view graph lead from one view to another, resulting in a chain of actions triggered by these views. It is argued that the view graph is a concept which is both more natural and more powerful with respect to its theoretical content than the place graph of a maze [5]. Three arguments are given:

- Exploration of a maze produces a sequence of views, and not of places.
- The view graph contains all the information of the place graph.
- Two views can be connected by a unique movement decision, whereas for two places, this does not work.

Since the distinction between places and views is somewhat artificial in the case of this work (see section 4.1.2), the bottom-line is that creating proper views greatly facilitates the view-triggered-response.

3.1.4 Optic flow

When following a certain path, we not only just see landmarks and the other surroundings as static images. While we *move* trough our environment, we see motion. The surrounding scenery is constantly changing as we move forth on our way. Already back in 1958, it was pointed out that an animal's locomotion through the environment could be guided by visual information that results from these movements [23]. This visual information, which is the ensemble of optical velocities (interpreted as continuous visual signals) that surround the moving animal, is called *optic flow*. This research was extended into the domain of human navigation in synthetic environments [24]. Although different definitions exist for optic flow, in this work we define it as *the visual conditions that produce perceived motion*. This means that motion sampled at a too low rate is not considered to produce optic flow, as it demands temporal integration that lies outside the capability of the visual system. The latter type of successive visual samples is called *no-motion*.

In two experiments, the effects of optic flow on navigating in a synthetic (virtual 3D) environment were investigated [24]. The results indicate that optic flow helps participants to learn a series of left and right turns and spatial locations while navigating through synthetic environments. Although participants seemed to rely heavily on symbolic coding strategies (remembering a route as a succession of left and right turns) in one of the experiments, the absence of optic flow resulted in participants becoming disoriented and getting lost within the virtual mazes. The presence of (local) visual landmarks at maze junctions masked the effect of optic flow on performance. This is not too surprising as good landmarks already provide pretty good cues to learn a route. Other results of the experiments show that memory of spatial locations is better when the environment is navigated with fluid optic flow than when optic flow is choppy. When other cues (landmarks) are not available, optic flow can be a significant aid to wayfinding. Salient optic flow can facilitate the learning of specific locations in synthetic environments. Additionally, this optic flow aids in path integration and in forming mental representations of spatial environments.

Another experiment, focusing on mere 2D data visualization, led to a set of design guidelines. If a task requires subjects to know something about objects' spatial position, and the viewpoint is changed, then animating that change in viewpoint appears to help users. Therefore, it is suggested that movements in and between documents should be animated [15].

Research on the effects of animation on decision making supports the idea that animation can improve the user's mental model of the task. To be useful, the animation must be smooth, simple, interactive and explicitly account for the appropriateness of the user's mental model. Therefore, the design of the animated interface should be based on both the characteristics of the task domain and the structure of the task the user performs. Defining the task structure correctly enables an interface designer to decompose the task into animation segments and understand the sequence and type of interactivity that would allow the design of more comprehensible animation, thus supporting the user's awareness of the events and accounting for human memory capabilities [25].

Animation takes time. Very responsive systems capable of generating a new view instantly might provide the user with a new view faster than those using animation. Nevertheless, this does not automatically result in better task performance. It has been demonstrated that animation reduces the cognitive workload required to integrate the two views, needed to re-establish situational awareness. The time needed to integrate the two views can surpass the time needed for the animation to complete. Integration of two views and re-establishing situational awareness is not needed when animation is properly used, as the primary visual system is capable of maintaining it without the need for complex cognitive reasoning [15].

3.2 Object Manipulation

Object manipulation in this thesis is about the manipulation of so-called business objects, Employees, Salaries and Managers, for example. These objects have attributes. An employee has – among other things – a name, an insurance number and a salary. Changing these values is what we will call object manipulation. Thus, we can see a parallel with manipulation of a broader range of objects: objects in graphical software (circles, polygons) with attributes like colour, width and height, and objects in software desktops (windows, panels) having attributes like position and visibility. Although these objects and the ways they can be manipulated through a graphical user interface (GUI) differ greatly, there is at least one critical phenomenon that can be an issue in all of them: *object constancy* during manipulation.

Object constancy refers to the condition in which the user can use his visual perception system to interpret the change of an object during manipulation, allowing the cognitive system to stay focused on the task at hand. This perceptual phenomenon of object constancy enables the user to track structural changes with little cognitive effort [26, 27]. Object constancy can be realized by using animation to smooth the abrupt changes in appearance by gradually changing the screen's appearance from the old state to the new one.

A shortcoming in animating graphical objects upon manipulation is found in simple feedback techniques based on drawing object outlines. This tends to give the user the impression that a direct manipulation operation is happening at a surrogate object, rather than a real thing. Giving objects solidity by drawing them full-featured does not solve this issue as it can still fail to convey a sense of substance. Somehow it seems too easy to manipulate the object. To be convincing, solid-seeming objects must do more than look solid; they must also feel solid. Cartoon animation techniques realizing this sense of substance lead to the definition of four new principles for animating direct manipulation operations [28]:

1. **The principle of attachment** states that the objects being manipulated should at all times remain attached to the pointer, which maintains the impression that the user is always in control of the action.

2. **The principle of reluctance** states that objects should, in general, seem reluctant to change, which reinforces the illusion of substance by suggesting that changing an object requires effort on the part of the user.

3. **The principle of smoothness** states that objects must change in a continuous fashion, which reduces cognitive load by avoiding large and unexpected changes in visual information presented to the user.

4. **The principle of anticipation** states that the result of a user's action must be obvious at all times, which reduces cognitive load by supplying additional visual information and minimizing the use of short-term memory.

Although the experiment in which these principles were evaluated focused on direct manipulation operations as translating, scaling, rotating and shape editing, the insights derived from it might be beneficial to direct manipulation of business objects too. The principle of reluctance, for example, might be applied when the user of the personnel administration system tries to change one ore more attributes to invalid values; applying some sort of shake-animation could indicate that the business object is reluctant to have it's attributes changed to the new values. The possibilities of applying these ideas are an interesting topic on their own. The principles of anticipation and smoothness are easier to apply to business objects. Anticipation can be used in the form of highlighting an object when the mouse is moved over it, to indicate that this object will be opened when clicked. Smoothness is strongly related to the topic of animation already discussed extensively in previous sections. At least, the results of the experiment[28] clearly indicate again that, when sensibly applied, animation can make an interface more graceful and enjoyable to use.

Manipulation of objects requires a certain toolset to be available to the user, mostly visualized using buttons. Screen space mostly is very limited and context determines what tools might be needed (see also p.23). The approach of See-Through Interfaces [19] is very interesting with regard to this. This type of interface uses an overlay interface based on spatial modes, that does not use additional screen space. In fact, the tools that are available for manipulating an object are only shown as a kind of hover-effect when the mouse pointer is placed on top of the object. At the same time, its context-dependentness results in only relevant tools being available and irrelevant tools being absent. This makes the user interface cleaner and less cluttered and also nicely obeys Fitt's law, as the buttons needed for a manipulating action are always within a range of few pixels.

3.3 Enjoyment

We already discussed the relevance of animation in the previous sections on navigation and object manipulation. But it seems that animation also has the property of smoothing the rough edges and abrupt transitions in many GUI's in general. This results in a better overall feeling of direct manipulation (not only regarding business objects, but also the UI-components). In fact, effective animation smooths the animation of objects on a computer screen and simulates more closely the user's expectation of the physical world, leading to a readier acceptance of what is shown on the screen [28]. Animation is believed to make the interaction more engaging. Immediacy of response being both fun an useful. These are all *visceral experiences* according to

[29], where "visceral" means "intuitive" or "non-rational", "obtained through intuition rather than from reasoning or observation"[2].

Aesthetic appearance is also supposed to influence enjoyment. It is suggested that interface aesthetics are closely related to *apparent* usability[30]. As apparent usability seems to greatly influence *perceived* usability for a long time during product usage, it would be wise to design aesthetically pleasing interfaces. But what exactly is aesthetically pleasing is culturally dependent [30]. So determining how to make the interface aesthetically pleasing is not so straightforward, especially when designing a product for international usage. A more formal analysis of aesthetics is needed to be able to determine more precisely what makes an interface aesthetically pleasing.

One aesthetic notion, though, is especially interesting: simplicity. Simplicity not only is an classic aesthetic notion that has been praised for many centuries [31], it also is believed to be a key characteristic for good user interface design for the web, because people are extremely goal-driven and do not tolerate anything between them and their goal, according to [32]. Whether the latter is always true might be doubted somewhat, the goal-driven aspect holds true for a personnel administration system, even when it is no website. Although it is hard to tell what makes a user interface aesthetically pleasing, at least it seems to be true that simplicity is something to strive for anyway. And chances are, that many people will also judge a simple interface to be more enjoyable to use.

In the end, the discussion about enjoyment and satisfaction related to beauty seems to be far from providing answers other then stating that usability as well as visual appeal both need to be taken seriously [33].

3.4 Web based

During the last years, more and more applications are run over the web. This has several advantages and makes certain applications possible that were impossible with local applications. A distinction has to made when we talk about "run over the web". Using network connections to provide users all over the world access to the same data, stored on servers somewhere, is one thing. The other is the usage of an ordinary internet browser as a front-end to these data, rather than a more traditional desktop application. The former has advantages like availability, maintainability, and manageability of data, but the latter comes with some other attractive features of which the most appraised one is the fact that no local installations of software are required. Every personal computer comes already with a browser installed. So installing, maintaining and updating the front-end software (which is very important in a network setting due to continuous discovery of security holes) is done once on the server, and every user immediately can use it. This is big plus for security and financial reasons.

The major drawback is often taken for granted: a very limited set of interface and interaction elements resulting in serious problems from a user perspective[34]. For example, intuitive actions like drag-and-drop and editing data directly in their context are impossible. To accomplish this kind of tasks, multiple additional clicks and complete page loads are needed, often requiring the user to remember or reconstruct the context in which the action actually takes place. Some have argued that browser based applications are fundamentally flawed [34]. Yet, the trend is clear: more and more software solutions are implemented as internet applications with a browser as a front-end. SAP also joins the race with its new enterprise resource planning solution. If user satisfaction is key to success of software, how successful will all these new solutions be?

2 WordNets definition of "visceral" (http://wordnet.princeton.edu/perl/webwn?s=visceral)

But recently, some things have changed. "The Web" seems to be revolutionized in several aspects. The so-called "Web 2.0" was there, and would soon make the web a very pleasant and enjoyable place to be. The toolkit for user interface designers was, and still is, being greatly extended with new features that mainly are composed of existing techniques, but applied in a innovative way. This takes away some of the limitations of the traditional browser based applications and at the same time introduces some new ones [35]. From the perspective of this thesis work, the *increased responsiveness* due to AJAX [36] and the usage of JavaScript to create new, more dynamic widgets, is the most eye-catching feature. The discontinuities typical for browser based applications are no longer disrupting the interaction and thus, the overall user experience can greatly be enhanced.

Although these new technologies bring much improvement, a lot of challenges have to be faced. Standardization and the usage of consistent interface patterns [37] as – in the end – was more and more the case with traditional desktop applications, has to be established for the new web-based interfaces.

A major drawback of browser based applications remains the lack of hardware accelerated graphics in browsers[3], which will limit the extent to which animation can be applied. Nevertheless, a fair amount of useful animation can be implemented. Attention has to be paid to the performance of the application, as disruptions in animation might seriously degrade the user experience and usability of the product [24]. Only animations that run smoothly in the browser should be considered for acceptance.

3.5 Application in a business setting

Applying innovative user interface ideas in the more-or-less traditional domain of business software, like a personnel administration system, has several problems. First of all, user centered design is a phenomenon that is still pretty new and has to establish itself more clearly in most software companies. Although progress is being made, in practice there seems to be a not-so-easy-to-cross border between user interface designers and software developers. Not all stakeholders are yet convinced of the need of good user interface design for overall success of the product. Second, the enormous multitude of current users of existing business software got used to the *status quo* regarding the graphical front-end of the systems they work with. Switching from the rather static "traditional" point-and-click interface concepts somehow seems to bring a risk with it; the user might not want to learn to work with a new type of interface (or its employer is not eager to spend time and money on it), or even unlearn inefficient workarounds needed in the old type of interface they got used to.

Whatever the reasons may be, very few of the concepts discussed in the previous chapter have found their way into real business software.

3.6 Wrap up

The previous sections discussed more ore less distinct problems, theories, insights and solutions that might be relevant for the design of a good interaction paradigm for – among others – personnel administration software. Below we discuss how this contributes to the development of the new interaction paradigm we describe in the next chapter.

We adopt the notion of *information space* as a structure containing all the data available through the program. Key concerns are *navigation* and *direct manipulation*. Navigation implies movement

3 This regards browsers without additional plug-ins like a Flash or Java Runtime Environment. With the latter, it is possible to use accelerated graphics, but in that case it is merely a separate application that is downloaded and only *displayed* in a browser window.

through the information space. The perception of movement, called *optic flow*, is crucial in human navigation and wayfinding. We therefore use animation to support the user in maintaining sound situational awareness. Animation must be smooth, properly timed, simple and interactive and explicitly account for the conceptual model used.

In addition to the perception of movement, *landmarks* are important signs for humans in acquiring route-like and map-like knowledge. Using global landmarks as well as salient local landmarks accounts for the fact that people differ in the wayfinding strategy they prefer. Thus, using both types of landmarks supports users preferring a local strategy as well as those preferring a global strategy.

The usage of *trail creation* extends the functionality of landmarks, by visualizing the history of movements through the environment.

To further prevent the *lost in hyperspace phenomenon,* proper *temporal* and *spatial* context must be provided to the user. Trail creation accounts for temporal context, answering the "How did I get here?" question to the user. The local landmarks answer the question "Where can I go from here?", thus providing spatial context. The preview characteristic of a landmark is very important in this respect. Care should also be taken to limit the number of directions or operations the user can perform, as the user can only focus on a limited number of items. Usage of colour and contrast can further support the user, using the characteristics of his visual system to discern multiple items based on global features.

When providing context information to the user, a zooming interface might be better than an interface with two separate views (context and details) as it preserves the direct relation between the context and the details. Animation, according to principles already stated in this paragraph, should be applied to really preserve this relation and prevent additional cognitive load.

A *view graph* and a *place graph* can be used to conceptually model the information space. A loose combination of both of them, complemented with some new features to incorporate the idea of direct manipulation will be described in the next chapter.

The notion of an *information space* requires object manipulation to be direct in the sense that the user can directly operate on data he sees in the interface. The information is modelled as objects whose attributes can be manipulated. *Object constancy* is required to enable the user to stay focused on the task at hand. This can be realized by using animation to smooth the abrupt changes in appearance by gradually changing the screen's appearance from the old state to the new one. Cartoon animation techniques can create a sense of object substance, preventing the user to have the impression that the direct manipulation is happening on a surrogate object.

See-through interfaces can support direct manipulation by providing context-dependent tools to the user. This results in a cleaner user interface and further supports the notion of direct manipulation.

Besides the positive characteristics of animation already mentioned, it also seems to contribute to an overall better feeling of direct manipulation in general. Effective smooth animation simulates more closely the user's expectation of a physical world, leading to readier acceptance of what is shown on the screen. In addition to that, animation is believed to make the interaction more engaging.

Aesthetic appearance seems to influence perceived usability. Therefore, care should be taken to make the interface aesthetically pleasing, though it is difficult to determine exactly what pleases

users. *Simplicity* is something to strive for anyway. In addition to it's aesthetic notion, it seems to make interfaces more enjoyable to use, by supporting the user in goal directed tasks.

When designing a new interaction paradigm for web-based applications, possibilities are limited to the what the browser platform can provide. These possibilities currently are quite extensive, including animation. Nevertheless, care should be taken to verify that the user interface complies with all requirements described so far, especially those related to animation.

Chapter 4

A NEW WAY OF INTERACTING

Based on the theoretical background described in chapter 3 and with the problems and goals identified in chapter 1 in mind, we will now describe how a new integrated interaction paradigm is designed. Before the actual design is done, the task is modelled in a way that especially takes the user experience into account, instead of the technological inner workings of the system. Based upon this model, section 4.2 provides the actual design by providing mock-ups and describing behaviour of the system as an effect of user interaction.

4.1 Modelling the task for optimal task flow

The key inspiring factor for the model we now will describe is the notion of *situatedness* within the application, especially within the information space consisting of all employee business objects and the connections between them. We model the information space as consisting of *places* and *connections* between them. The following section metaphorically illustrates our approach. The second section gives a more formal description of the resulting model.

4.1.1 Metaphorical explanation

As discussed in the previous chapter, a lot of research has been done in the domain of wayfinding and situational awareness. The fact is that those projects dealt with finding a route from one location to another. The only possible actions on every place are turning and moving. In our model, we will also have to deal with an additional *dimension:* the *manipulation* of the places. When a business object is perceived as a *place*, finding a route to this place is the first step in most tasks. Finding the values of certain attributes of this object can also be seen as wayfinding within the limited area of one object. But manipulating these attributes is not a wayfinding problem anymore. It is an object manipulation problem. To integrate these two problems into one model is the key towards deliverance of a good mental model for the personnel administration task.

Our approach toward this integration problem is not too complicated. With the risk of failing – every metaphor fails in the end – we introduce the comparison between the collection of employee business objects and a huge block of houses, each having multiple rooms. The parallels are defined like this:

Block of houses in a street	↔	The collection (list) of employee business objects
House	↔	A single employee business object
Room	↔	Category of attributes of the employee business object
Piece of furniture	↔	Subcategory of employee attributes
Properties of the piece of furniture	↔	Attributes of the employee business object
Window	↔	Limited view into other, *related* objects
Building an additional house	↔	Adding an employee business object
Demolishing a house	↔	Deleting an employee business object
Attic	↔	Store of old sets of attributes

All houses have their front door at the street. Thus, when standing in the street, in front of the block, all front doors can be seen. Though, the nearest front doors, of course, are better and in more detail visible than doors farther away. To be recognizable and have some unique visual appearance, every house has some distinctive feature on the front side, for example a unique front-door colour (Figure 4.1). Parallel to this, the employee business object can have a portrait of the employee it represents. These can function as *landmarks*, supporting way-finding. The visualization of this in the user interface can be done by using a Fisheye view [16], providing more details of the object, while showing and preserving the relation with its context. The business object under the mouse-pointer has the largest size. Other business objects are shown with decreasing size, depending on the distance from the pointer. Moving the pointer moves the focus (walking to the left or the right in front of the block). This should be done in a fluid manner, properly using animation to establish optic flow.

Figure 4.1: Facing a block of houses

When entering a house, it appears to have multiple rooms, each having specific purposes. This can be seen immediately when entering the house. Doors of the rooms might be decorated with some sort of icons, or they have windows in it, through which a rough impression can be gained from what is inside (Figure 4.2). Actually, the latter is most useful, as it not only provides cues about what *type* of room it is, but also shows already some more detail of what's inside. The personnel administration parallel for this could be the fact that, once a business object is selected, you can see the categories of attributes and from some of the most important attributes also the value.

The next step will be to enter a specific room, unveiling the presence of different objects in it. A living room, for example, will have a sofa, some other seats and a cupboard in it. These pieces of furniture have certain characteristics. The sofa has a colour, a texture and a number of legs. The cupboard has height, width, depth, number of sections and so on. The type of room can be compared to a category of attributes of the business object – e.g. 'Personal Data' – and the pieces

of furniture can be compared to the differing subcategories – e.g. 'Biographical Data'. The characteristics of the pieces of furniture can then be compared to the attributes of the business object and it's corresponding values, e.g. first name is “John”.

We also have to deal with the fact that every set of attributes within one subcategory has a period for which it is valid and that multiple entries can be stored. That means, also entries from the past. These past entries are not most relevant, but must be available for archiving and later inspection if needed. The parallel is clear: old furniture is stored in the attic, so old entries are stored in one or more 'attics' from which the details can not immediately be seen.

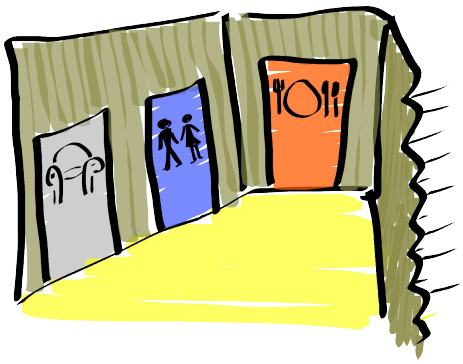


Figure 4.2: Entering a house

Removing a complete business object should not require one to enter it first. If the outside has enough unique identifying information, it can be determined from the outside that this is the correct object to be removed. This can be reflected in the visual design by using a see-through tool. Looking toward an object means having the object in focus and being able to operate on it directly (direct manipulation). It is definitely not intuitive to run towards to outer left- or rightmost part of the block to perform operations on the house in front, as has to be done in many web based applications (clicking on an object, then move the mouse to the top or bottom where the 'edit' and 'delete' button is located).

When really a lot of employee business objects are stored in the system, it is impossible to see them all at the same time. Compare this with a really huge, stretched block of houses of which you actually can not see the first and last ones anymore. Here, the comparison is probably facing its limitations: in the personnel administration-list, filtering could be used to decrease the number of objects in the list. This will increase the visible size of the objects. Maybe it could partly be compared to getting closer to the block, limiting the number of houses within sight. But it is clear that the metaphor ends here and one should not push metaphorical design beyond the level where it only results in limitations.

One important aspect of this metaphor is *movement*. Entering a house or a room is a continuous process, resulting in optic flow. This should be reflected in all movements in the information space as well. Proper usage of animation can support the user to maintain situational awareness and make the application more enjoyable to use.

4.1.2 The model: place and view graphs with manipulation extension

The comparison with a block of houses is not interpreted metaphorically in the sense that the new user interface will have “doors, rooms, sofa's” and so on. It is just a source of inspiration and explanation. We assume that, while people are capable of finding their way in buildings when given good cues and a sense of context, wayfinding can serve as a nice conceptual model of how people will find their way in the information space and overall structure of the personnel administration software.

The *view graph* model, discussed in section 3.1.3, indicates that *views* are to be selected over *places* when creating a model for wayfinding, while people store routes as series of views and movements. Views can best be given salient, recognizable properties that can function as *local landmarks*. Good salient local landmarks can result in local wayfinding strategies that also lead to map-like knowledge in physical navigation (see 3.1.1). For the problem at hand, we also assume that usage of good local landmarks reduces the need of global landmarks and still

results in good awareness of the layout of the information space. For the employee business object and its (sub)categories, we have to define what can serve as good local landmarks. The best way is to determine what attributes in each category are the most important. These can serve as identifying landmarks, each corresponding to a view in a view graph.

Figure 4.3 shows a place graph as defined for the personnel administration tasks. Although it is the visualization of a model, the similarity to Robertson *et al*'s Cone Trees [26] visualization of hierarchical information spaces is clear. But we will first focus on the model and describe visualization in the next section.

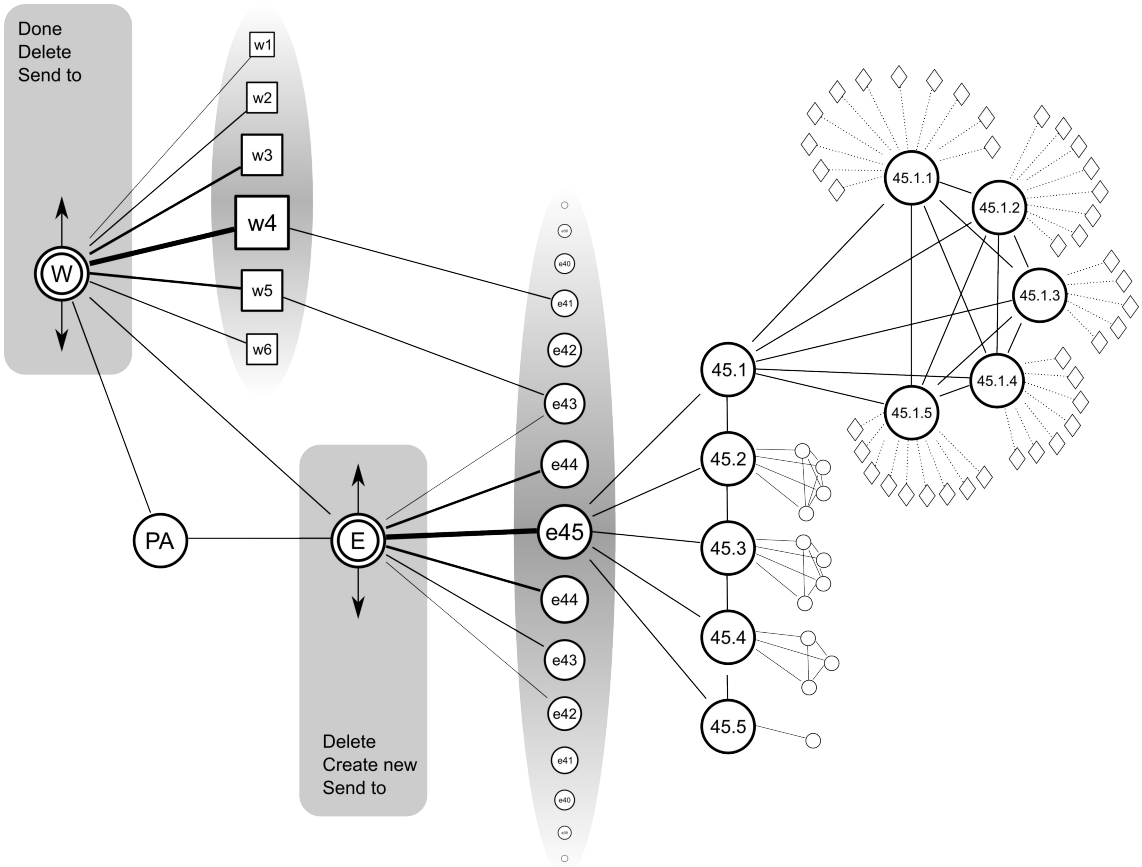
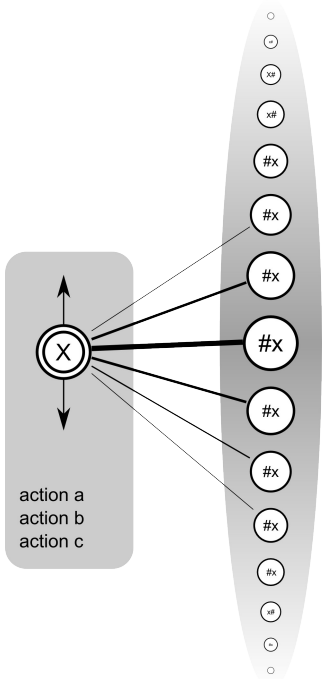


Figure 4.3: Partial place graph with object manipulation extension

Every node in the graph is depicted using a circle or a square with an identifier in it, and defines a *place* in the information space. Every edge is depicted using a line. Edges define transitions between two places and have directedness. As all edges in Figure 4.3 are bidirectional, arrows are left out of the diagram to keep it clean and uncluttered. A succession of nodes and edges represents a route through the personnel administration information space (see chapter 2 for a description). From every node in the graph, several views can be perceived. Every edge leaving a node corresponds to a view. A view provides some representation of the place one will reach when the transition in its direction is chosen. As there is a one-to-one relation between views and places, the nodes in the figure can also be perceived as the *views* one sees when looking into the direction of the place associated with it. It could be argued that the view of a certain place differs depending on the direction from which one comes, but in the case of the current information space this does not seem to make sense.

Node types	
x#	Business object (base object) of type x. *Example*: Employee business object
x#	Meta object of type x. *Example*: Work item.
#.#	Category. *Example*: Personal data, a category of an employee business object.
#.#.#	Subcategory. *Example*: Biographical data, a subcategory of the Personal data category
◊	Attribute. *Example*: Last name, an attribute in the subcategory Personal data.
X	View-point. *Example*: View upon the list op all employee business objects

Relations	
	Business object, category or subcategory having an attribute that is visible and editable. The dotted line indicates action of manipulation (not a movement).
	Bidirectional path between two business objects or (sub)categories . The line indicates a view from one node to the other, and the related movement between them.
	Directed (one-way) path between two business objects or (sub)categories line indicates a view from one node to the other, and the related movement between them.
	Bidirectional path between a meta-object and a business object or (sub)category.

Complex		
	One view-point node (X) relates to all the nodes (on the right) in a collection, providing a view upon them. Nodes in the center of the the view are better visible than those farther away. From the view-point, operations can be performed upon a node in the list. The view-point can also be moved, to focus on another part of the list. Compare with a rotated view of Figure 4.1:	

The actual *view graph* corresponds to the place graph in a fairly straightforward manner. In the view graph, the nodes represent views, and the edges represent navigating actions. A path through the view graph is a succession of views and navigating actions. Navigating actions in maze navigation tasks (as in [5]) differ somewhat from the navigation task in the virtual space in the personnel administration tasks as several views are integrated into one. In a maze, when looking in one direction and perceiving the corresponding view, one has to *turn-left* or *turn-right* to perceive the view associated with one of the other corridors (edges). In the current task, it is possible to perceive, for example, all views corresponding to places 45.1 – 45.5 in Figure 4.3 when located in place e45. As long as the number of these views is fairly limited, they fit on a single screen and can be perceived as a whole, without the need for turning or moving.

When a node has a huge amount of edges connected to it, though, it is no longer possible to perceive all views associated with all the nodes at these edges' endpoints as one view. Some movement is needed to focus on a subset of it. view graph theory is fairly discrete in nature. Navigating actions are *move-right*, *move-left*, *turn-left* and *turn-right* for maze wayfinding. Every navigating action results in a new view. But for the navigation in the task at hand, we introduce a somewhat more continuous type of navigation to be able to model the mentioned focusing on subsets: *move-up* and *move-down* in the list. In Figure 4.3 we have two lists. One representing a list of *work items*, all associated with one Employee business object, and one representing the list of all Employee business objects. We use the move-up and move-down navigating actions in view-points E and W to change the point of focus on the list. Moving is continuous, so it actually results in continuously changing views. But these views consist of separate views, each associated with one employee business object. Remember the metaphor depicted in Figure 4.1 and compare it with the representation of lists in Figure 4.3, which can be perceived as a 90 degrees rotated view of each other.

In addition to mere places, views and transitions (navigation), we introduced the additional notion of *manipulation actions* which is depicted using grey boxes. From the places associated

with the nodes in a grey box, the actions mentioned can be used upon the objects or attributes visible from it. So, when located in view-point E, having the focus on a certain employee (e45 in the figure), one can directly act upon it with the actions listed in the grey box: delete, send to and create new.

Moving through the information space is fairly unrestricted, as can be seen by the amount of edges between nodes. This is very useful for the tasks related to existing business objects. The task of creating a new Employee business object is modelled using the place graph as given in Figure 4.4. The difference is that the path is more guided to make sure that all required attributes are given a value.

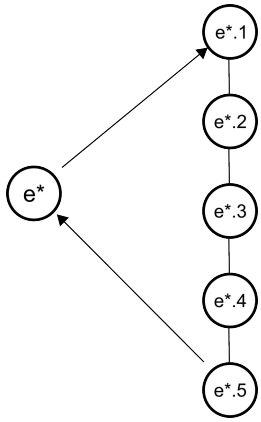


Figure 4.4: Place graph for creating a new employee business object. The route is guided.

4.2 From model to visualization

This section describes how the model can be mapped to a graphical interface, including the way it reacts on user behaviour. The issues and opportunities identified in the model will be addressed. It will result in screen mock-ups and descriptions of how they will respond to user input.

4.2.1 Basic layout of the new graphical user interface

The basic layout of the GUI as shown in Figure 4.5 identifies four global elements, each corresponding to one of key questions for navigation. These four elements will now be discussed in more detail.

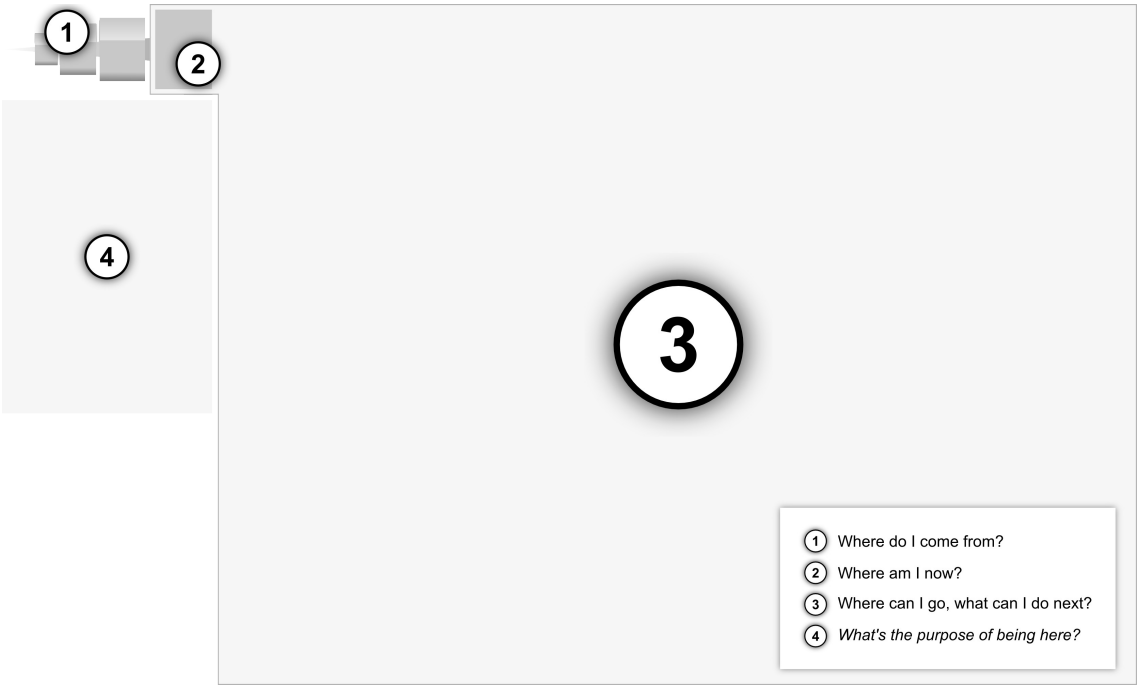


Figure 4.5: Basic layout of the new graphical user interface

1 Where do I come from?

The path the user followed towards the point where he finds himself, is visualized using a trail of clickable landmarks. These landmarks are the scaled-down versions of the ones used in the views of possible further steps the user can take. As with visual perception, landmarks of places further away are perceived smaller than the ones closer to the point of view. Every navigating action of the user results in an additional landmark being added to the trail from the right, thereby pushing the other landmarks in the trail to the left, reducing their sizes. The new landmark originates from the element 2 ("Where am I now?"), and changes from a display-only visual element into a clickable button. Clicking on it will bring the user back to the place corresponding to its place.

The model as described in the previous chapter does not include something like a trail explicitly. It only provides a view towards the last place the user visited. On the other hand, the model does assume the creation of a path in the user's memory, facilitated by the use of landmarks. By explicitly visualizing this trail of landmark, the user is supported in the creation of this in-memory path without taking much screenspace. In addition to that, it gives the user the possibility to move back to a point earlier in the trail, without disrupting the situational awareness. Every movement, back and forth, is animated, supporting this awareness.

This element of the user interface prevents the *lost in hyperspace phenomenon*, because it provides good temporal context.

2 Where am I now?

This element holds the landmark corresponding to the place the user currently finds himself. Before entering this place, this landmark was visible in element 3 ("Where can I go?"). After the user chose to go to this place, it is transferred from its location in element 3 towards element 2 using animation, parallel to the animation revealing the new location. As it does not make sense to move from the current place to the current place, this landmark is not clickable and therefore not visualized as a button, opposed to the landmarks in elements 1 and 3. This is also clarified by visualizing elements 2 and 3 as one, using element 2 as a kind of label on the current view.

3 Where can I go, what can I do next?

The largest part of the screenspace is used to visualize the possibilities for further action. These possibilities are either of the type *navigation* or of the type *manipulation/viewing*. Navigation possibilities are visualized using landmarks that give a clear cue about what place the user will enter after clicking it. In addition to this more or less iconic landmark, some textual information can be added. Both the iconic landmark and the textual information must provide some sort of *global features* of the place they represent. Additional usage of different colours even improves the discernibility of different places. Together, these features provide the spatial context that prevents the *lost in hyperspace phenomenon*.

Clicking a landmark takes the user to the place it represents by revealing it using animation, giving a sense of moving within the information space and maintaining object constancy.

The other type of possibilities, *manipulation/viewing*, is available when the user arrives at a place holding attributes of a business object as expressed in the model (see Figure 4.3). While trying to create a sense of direct manipulation, every attribute can be modified by clicking it and inserting a new value. This is major shift away from the popular pattern where the user has to click an edit-button before any value can be changed and pressing the save-button afterwards. In this new paradigm, new values are saved immediately upon manipulation. The evaluation revealed that this is a point of concern for more detailed investigation, especially how this could best be visualized when integrating undo functionality (see the discussion of results related to this on page 54).

4 What is the purpose of being here?

The fourth element in the user interface is optional. While navigating in the information space, viewing information and browsing for some information, the user has a clear goal stored in his memory. This corresponds to the type of task the user starts on his own initiative (see section 2.2). In addition to this scenario, the user can be provided a work-item, modelled as a meta-object. This work-item refers to one business object and describes what kind of administrative manipulations should be performed upon this business object. Such a task description is too lengthy to be kept in memory and therefore is provided to the user as a kind of shopping list. This gives the user two possibilities:

- He can re-read the task description any time he wants.
- When the task or parts of it are done, he can mark them as finished immediately.

4.2.2 Visualization of actions

This section describes how the interface and actions performed by the user are visualized. It shows parts of the interface that are relevant for the visualization being described, along with the corresponding part of the place model. The next section (4.2.3) shows integrated mock-ups.

1 Moving

Two types of movement are identified. The first one deals with moving in a list. The second one is about moving into one of the possible directions available. These two will now be discussed in more detail.

1. Moving in a list (move-up and move-down), that is: moving the focus. This is visualized using a fish-eye technique. In addition to scale animation, increasing the size

of items where the pointer is, we also use semantic zooming. The item in the center of the focus point not only is displayed larger than any other, making it more readable, it also contains some additional information (e.g. the employee's role and department).

Figure 4.6: Moving the focus on a list, using a fisheye technique.

The text in every row clearly indicates which business object it represents and might already provide some minor landmark functionality. The shapes of words can be perceived and recognized. But the more salient landmarks, required for the establishment of good route and map-like knowledge, are implemented using a photographic representation of the employees: a portrait. In the example in Figure 4.6, these portraits differ greatly. Not only the people already differ enormously, also the background colours of the photograph do. It might even be the case that especially these background colours provide the *saliency* that is required. The true value of this landmark will in practice depend on the kind of photographs being used. It must be noted that to make this type of interface really useful in a realistic setting of organizations with lots of employees, additional functionality like filtering should be added.

2. Moving in the direction of one of the available views at a certain node, along an edge connecting two places in the graph. Only paths modelled in the place graph (Figure 4.3) can be followed. The move is visualized using an animated fold-out of the item selected, thereby unveiling the details it contains. What exactly is unveiled depends on the type of end node. Figure 4.7 illustrates what happens when the user clicks a preview of a business object. The following four animations happen synchronously:

   - The landmark (in this case, the portrait) moves from it's original position to the end of the trail visualizing the path that lead to the current position.
   - All other elements of the trail move to the left, and shrink to a smaller size. This resembles the fact that these past steps now are farther away.
   - The preview moves to the top, and out of view.
   - From the center of the preview, the new view unfolds and fills the “where can I go” part of the screen.

   The way back is simply the reverse of this, resulting in the same view as encountered before.

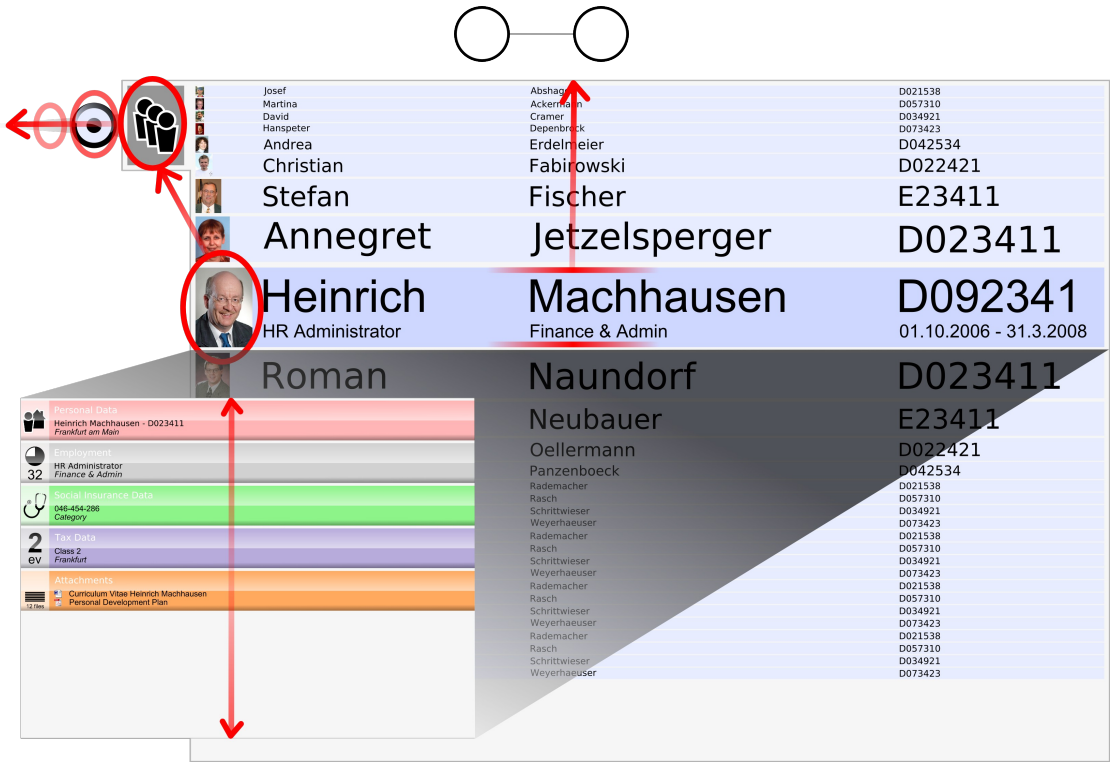


Figure 4.7: Move from one node to another

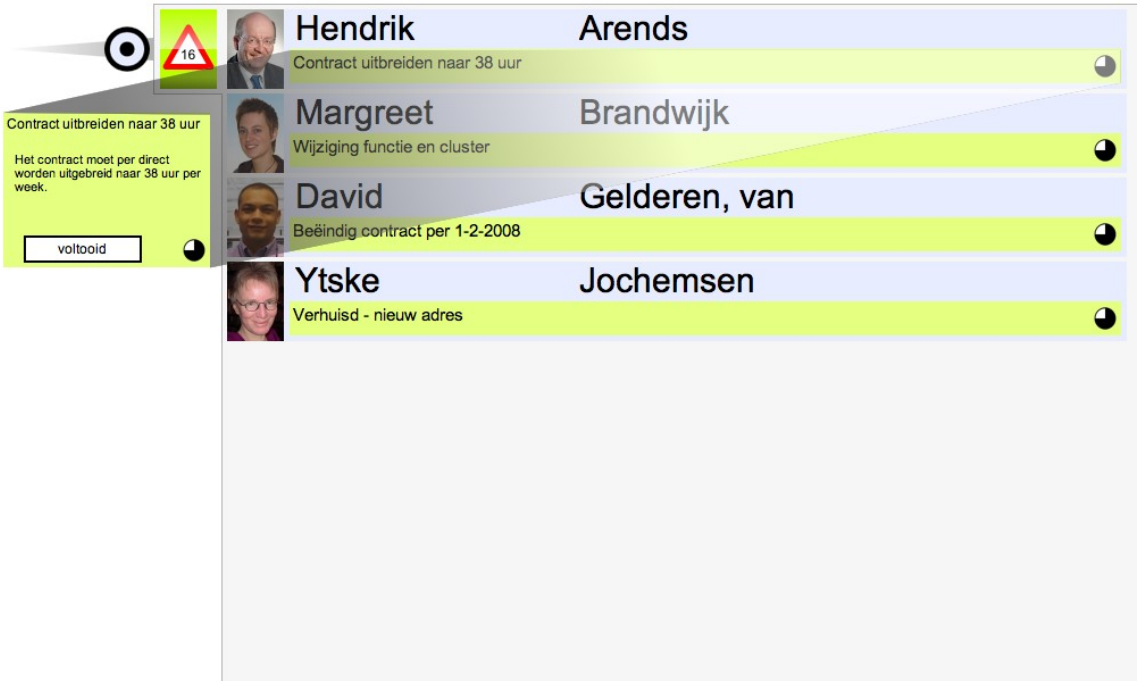


Figure 4.8: After selecting a work item meta object, the task description is moved to the left and shown completely.

Selecting a meta object results in additional animation. Work items are implemented as meta objects and refer to employee business objects. In addition to showing the preview of the related employee, it also shows the title of the task that has to be performed on the related employee business object. When clicking the meta-object, the title of the task moves to the "what is the purpose of being here?" area (Figure 4.8). While moving to that place, it grows and shows the

complete task-description, including a button that can be used by the user to let the system know that the task is complete. Clicking on this button takes the user back to the work list.

2 Manipulation

Several types of direct manipulation are used. Although not all are implemented in the prototype, we mention them for completeness. Those features that are not in the prototype are discussed in chapter 7, as they are interesting for future investigation. The following is the complete set of manipulating actions, divided in two types:

1. Remote, from a list, using a see-through tool:
    - Deleting a whole business object (not in prototype).
    - Sending the data of a business object to some person (not in prototype).
    - Creating a new business object (not in prototype).
    - Open a business object, which actually is a move.
2. From within the object:
    - Directly manipulating attributes with in-place editors.

Type 1 is concerned with acting directly on the list of business objects. The focus determines which business object is under control and can be acted upon. For both types, refer to the next paragraph or the actual prototype or the screencast, to have an impression what this all looks like.

4.2.3 Integrated mock-ups

This section introduces the integrated visual mock-ups of the new system. As animation is hard to visualize on static paper, it is recommended to consult the actual prototype or the screencast[4] for a better impression.

4 http://g-majeur.nl/thesis/screencast.wmv

Figure 4.9: Work Center, the starting point for the personnel administrator.

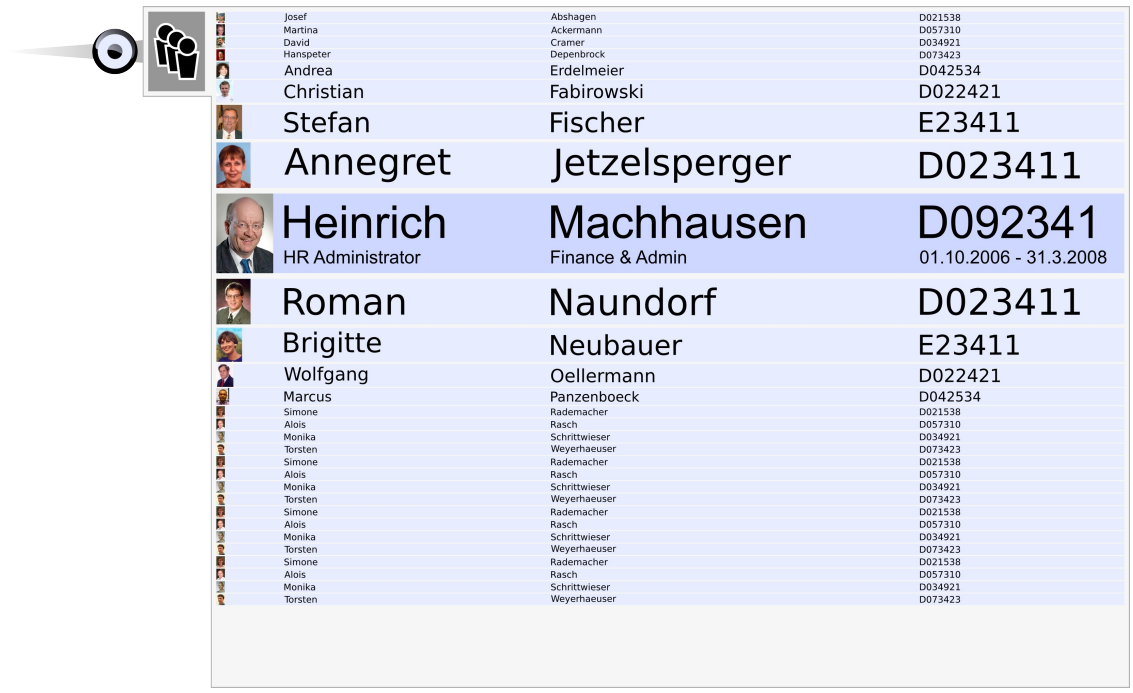


Figure 4.10: Employee List, showing previews of all employees using a fisheye technique

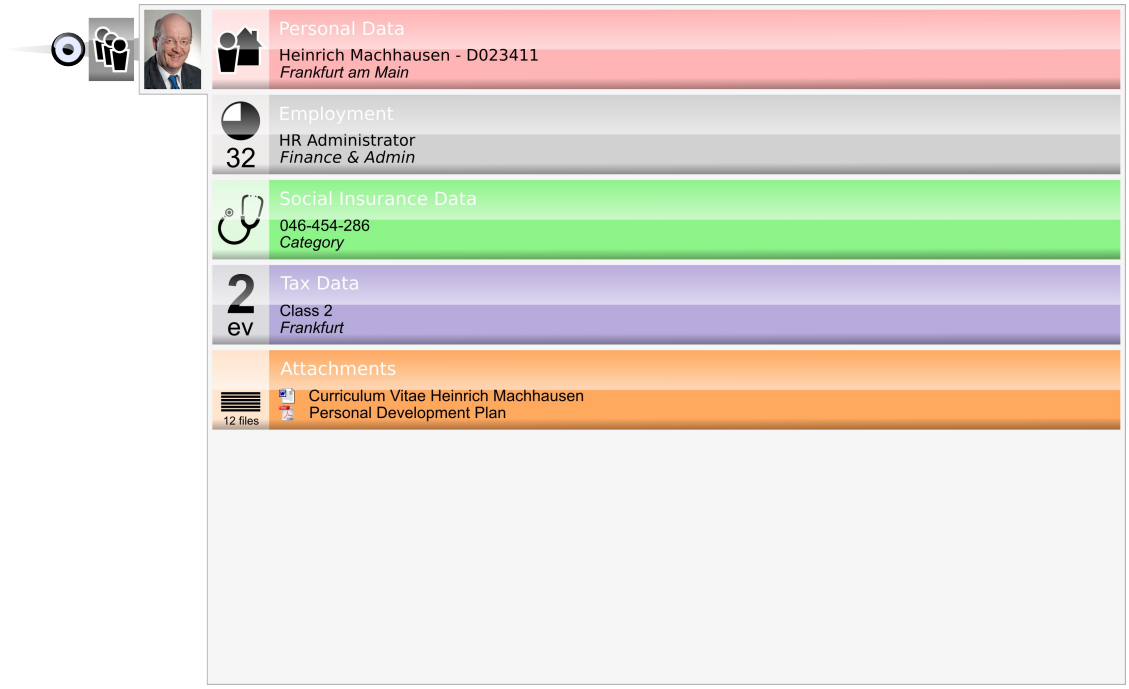


Figure 4.11: Employee overview, showing previews of the five categories of attributes.

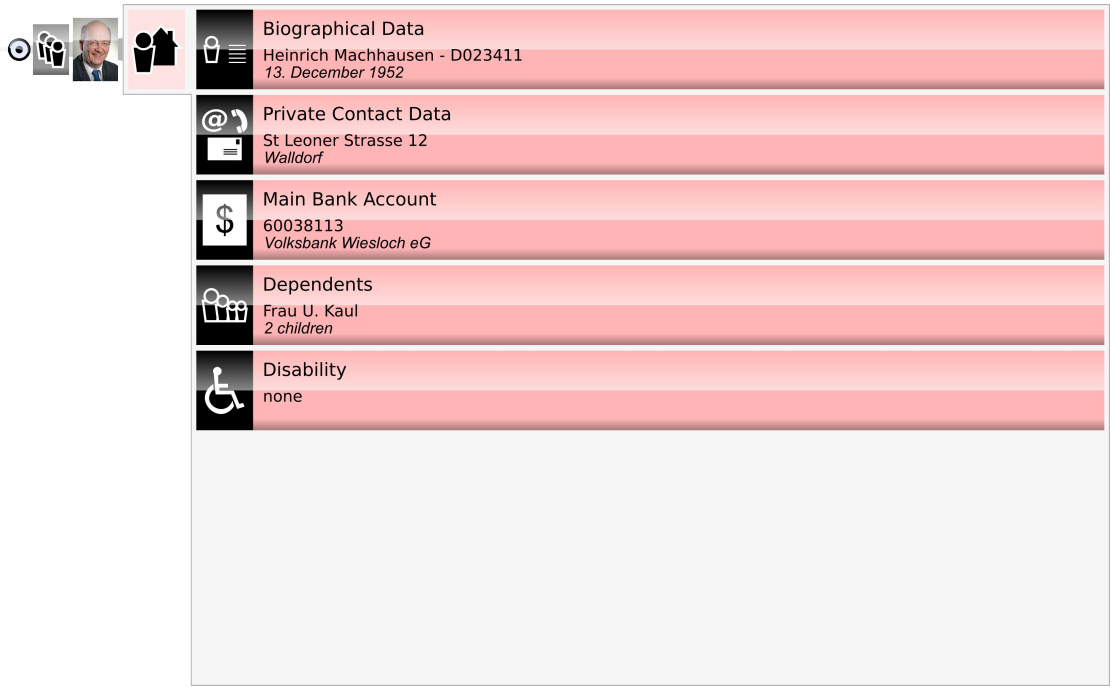


Figure 4.12: Employee's personal data, showing five subcategories of attributes.

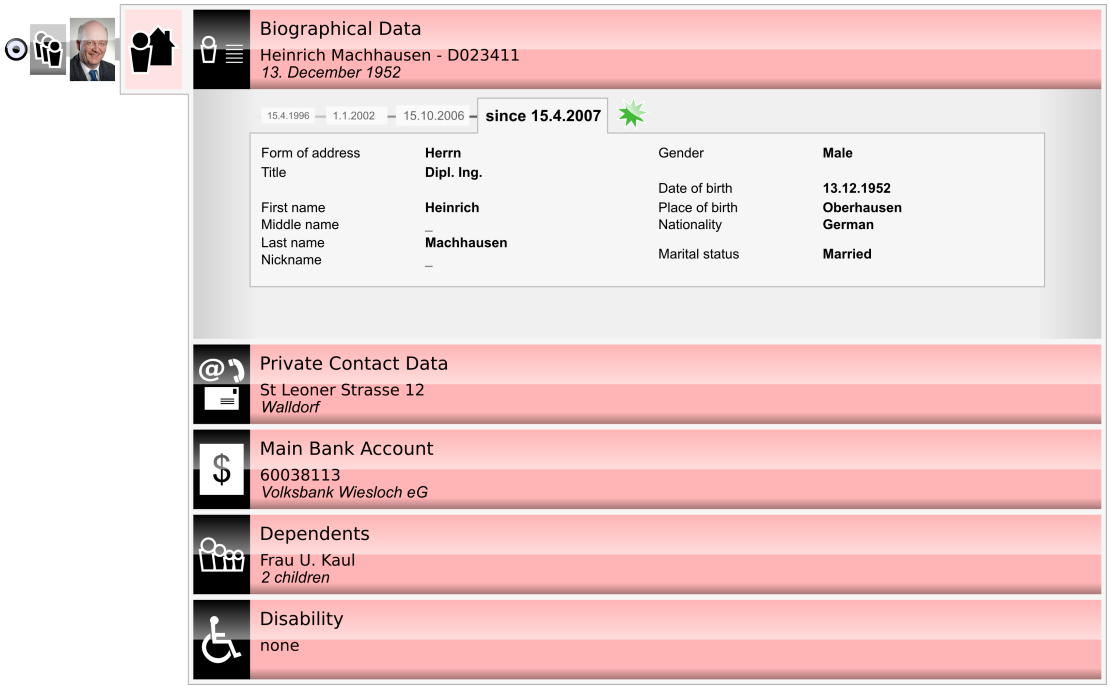


Figure 4.13: Employee's attributes in the Biographical Data subcategory.

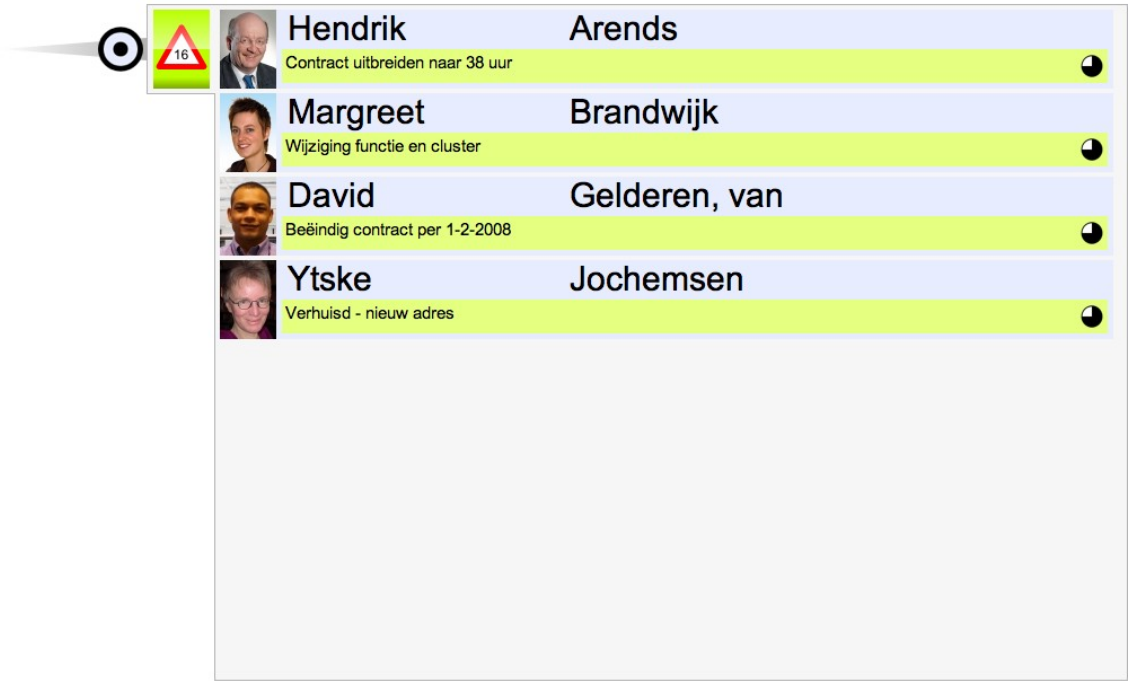


Figure 4.14: The work list showing four work items as meta objects referring to employees.

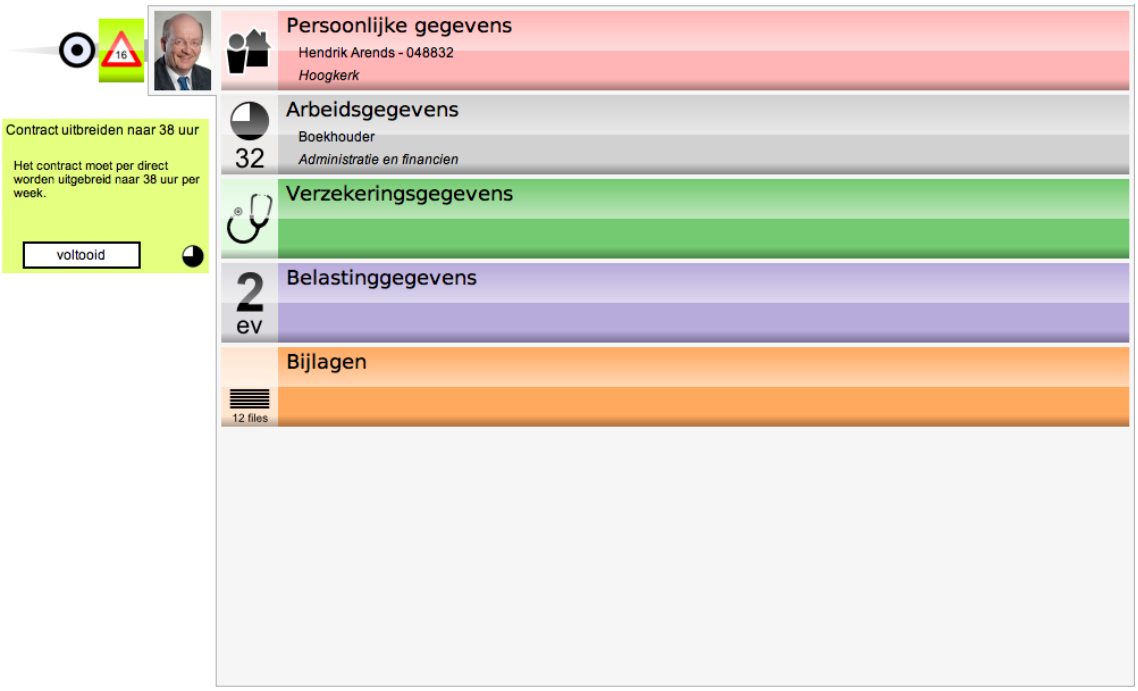


Figure 4.15: Entering a business object view after selecting a task related to it in the work list.

All integrated mock-ups are implemented according the pictures above, except for one feature shown in Figure 4.13; the "time-line" on top of the set of attributes visualizes the fact that every set of attributes is valid for a certain period in time. Here, like in the navigation trail, size is used to indicate distance in time. Although probably very useful, this feature was not implemented due to practical reasons.

4.2.4 Example tasks

To illustrate and clarify the ideas formulated so far, a screencast[5] is created which illustrates the first two subtasks mentioned in section 2.2 are present: *finding information about an employee* and *editing data of an employee*. The functionality required for the other two subtasks is left out of the prototype we designed. Also, both types of task initiation are demonstrated: *own initiative* and *triggers* (work items). In fact, the tasks performed in the screencast are exactly those that the participants had to perform during the evaluation. Therefore, it is useful to take a look at paragraphs 7, where the evaluation tasks are discussed in more depth.

5 http://g-majeur.nl/thesis/screencast.wmv

Chapter 5

IMPLEMENTATION

The design described in the previous chapter is implemented in a prototype. Some features are left out of the prototype for practical reasons; the timespan of this project is too short to implement and evaluate the complete new paradigm. Features left out are the following:

- See through tool
- Cartoon animation techniques in error handling
- Client-server interactions
- Time-line feature

These features and their relevance are discussed in chapter 7 in more detail.

This chapter describes the process of implementation. After presenting the global technical layout of the program (5.1), the selection of software and development tools is discussed (5.2) followed by some notes on rendering performance in browsers (5.3).

5.1 Global technical layout

The key ingredients of any browser based application are XHTML[6], JavaScript[7] and Cascaded Style Sheets (CSS[8]). This also holds true for the prototype we developed. Figure 5.1 below illustrates the way these components interact and deliver an integrated user experience in the browser. The actual code or relevant excerpts can be found in the appendix starting at page 73.

6 http://www.w3.org/TR/xhtml1
7 http://en.wikipedia.org/wiki/JavaScript
8 http://www.w3.org/Style/CSS

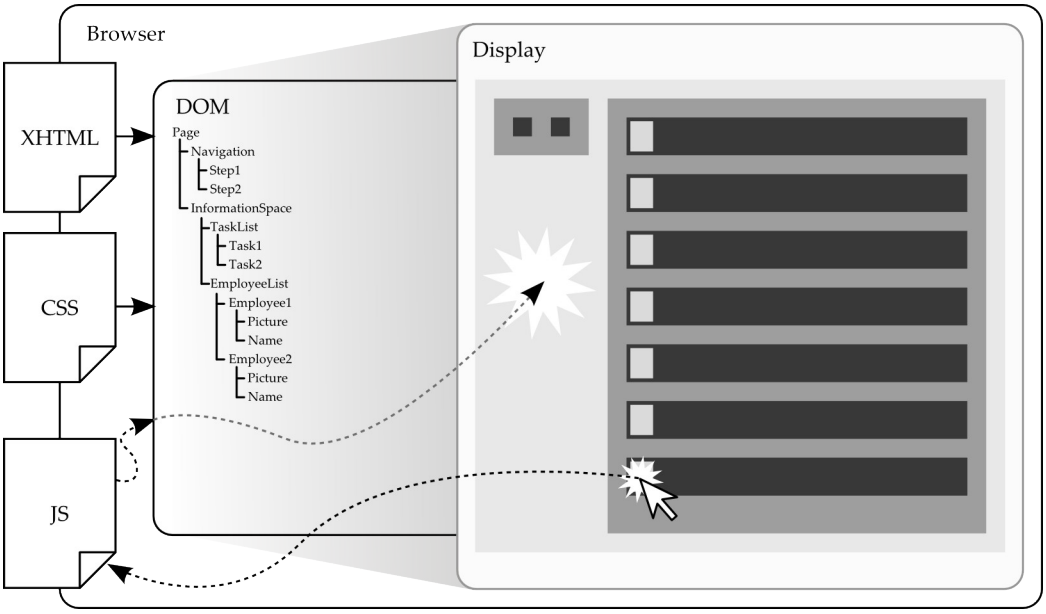


Figure 5.1: Basic technical layout of browser based application

First of all, the XHTML file is loaded. This contains the basic structure of the application, which is used by the browser to create the Document Object Model (DOM[9]). It also contains pointers to the required CSS files and the required JavaScript files. These are loaded by the browser and applied to the DOM. The CSS files contain information on the positioning and styling of all elements in the DOM. The JavaScript files are parsed when loaded, thus creating the actual application which can react on user behaviour. In the initial phase, the JavaScript code adds so-called event handlers to the various elements in the DOM. For example, all "preview"elements of the employees have a *hover* handler, which takes care of properly reacting on the user putting the mouse over a preview (highlighting). They also have *click* handlers which take care of executing appropriate functions, like folding out the elements holding the details of the employee.

The XHTML file in this case holds a template for a employee business object. An additional JavaScript file contains the data of all employees. At the end of the initialization process, a JavaScript function is called to uses the template to generate all the business objects, using the template. In a real-life situation, the data would be pulled from the server in the background and integrated into the application.

5.2 Selection of software

The software used for this project is discussed in the next sections. Two categories can be discerned: the software that the actual prototype consists of, and the software used during development (tools).

5.2.1 JavaScript libraries

As the goal of this thesis is to develop a new interaction paradigm *within the limitations of a web browser*, without requiring special plug-ins, it is clear that XHTML, CSS and Javascript have to be used. These three technologies or standards provide everything that is needed to build the prototype. Although it is possible to build the whole prototype – and especially the JavaScript – from scratch, a readily available tool-kit is used that provides a lot of functionality out of the box. These days, there are many tool-kits available. Which one to use is partly a matter of taste

9 http://www.w3.org/DOM

or personal preference. In the end, most things can be done with most tool-kits. But there are also differences in the type and amount of functionality already build into it. As this prototype uses a lot of animation, the tool-kit of choice should ideally already provide some sort of animation functionality. The jQuery[10] library fits this requirement nicely. It has a very flexible animate function that can be used to animate CSS attributes like size, position and colour of elements. It also makes *chaining* of functions possible. Thus, multiple effects can be applied to one element, requiring only one line of custom code.

5.2.2 Development tools

Though not too relevant, for sake of completeness we mention the tools we used to build the prototype:

- Firefox[11] webbrowser.
- Firebug[12], a free and very useful debugger for use with Firefox.
- Trimpath Template[13], a template system used for the creation all business object elements in the application based on a JavaScript data file.
- Aptana IDE[14], an integrated development environment for web applications that integrates the debug functionality of Firebug.
- Safari[15], Apple's web browser for the Microsoft Windows platform. We used it for testing purposes in the end. See the next paragraph.

5.3 Rendering performance in browsers

To build a web application that is available to anybody anywhere means that the application should run in all major web browsers. The problem is that there are minor differences between browsers that can make the development of a web application that runs smoothly in *all* of them a pity. Especially incomplete or faulty implementation of W3C web-standards like XHTML and CSS is debit to this. In the end, most applications can be implemented such that they do run in all modern browsers. It only takes some additional development time to write hacks that “solve” these problems. But wile this project only deals with a prototype, it is not needed to achieve this level of compatibility. Therefore, development was simplified by focusing on the Firefox browser. The choice for this browser originates from the fact that the a free debugger (Firebug) is available for it, which is of great help.

When it became clear that the prototype did not run completely fluently within the Firefox browser, a short test pointed out that the actual performance is dependent on many factors, including the platform. The Windows version of Firefox performed much better than the Linux version on the same hardware. But even the Windows version had some disruptions in the animations. It might be useful to investigate possibilities to optimize the prototype so it runs smoothly in this browser, but another quick solution was found in Apple 's Safari web browser for Windows. This was just lately released as a beta version, but turned out to be way better in rendering and animating. Animations run without disruptions and font and image rendering is

10 http://jquery.com
11 http://www.mozilla-europe.org/nl/products/firefox
12 http://www.getfirebug.com
13 http://code.google.com/p/trimpath/wiki/JavaScriptTemplates
14 http://www.aptana.com
15 http://www.apple.com/nl/safari

better also. This is especially visible in the zoomed out images. In Firefox these look jerky. In Safari they are interpolated more neatly, resulting in smooth images.

One performance problem remained, though. The zooming in the fishy-eye list lagged behind the mouse movements. This is solved by reducing the number of renderings to 50% of the initial amount. Basically this means that the list is not rendered for every pixel the mouse moves, but only for half of it, reducing the computational power needed.

The final prototype is available for exploration[16], but the limitations mentioned above should be taken into account.

16 http://g-majeur.nl/thesis/prototype/index.html

Chapter 6

EVALUATION

The prototype described in the previous chapter has qualitatively been evaluated to see whether the new interaction paradigm does indeed provide a good and valuable new way of interacting with business data. This chapter describes the participants (6.1), the setup of the evaluation (6.2) and the results that are obtained from it (6.3).

6.1 Participants

The participants were were selected on having the following primary characteristics:

- Experience using a computer, especially in using a mouse.
- Knowledge about, or affinity with employment and employment related relevant information.

In addition to these characteristics, all participants were of age 25 – 35 year and highly educated. The number of participants was seven; 3 male and 4 female.

Although the new interaction paradigm has been evaluated with a prototype of a personnel administration system, the participants were no personnel administrator in their daily life. This type of participants can be coined "novice"as they do most likely have no experience with a typical personnel administration system. Using novices for the evaluation is considered an acceptable strategy for the following reasons:

- No conflicting situations arise due to strong habits expert users have learned after working for many years with a certain personnel administration system. Basically, less *un*learning is needed.
- The task the participants had to perform during the evaluation (see 6.2.2) is fairly straightforward. Although a novice might not know the details about a certain type of tax rate, he definitely does know most common things about employment, salary, health insurance and the common data like name and address. Therefore it was easy to construct a task that addresses everything needed to determine whether the new interaction paradigm is successful or not, without requiring advanced knowledge in the domain of personnel administration.

- It is easier to find novices. Due to the space and time frame limitations of this project, using novices was more feasible.

Of course, using novices also has some drawbacks. Since they do not work with a personnel administration system in their daily life, they do not belong to the target audience and can thus not provide feedback about certain elements in the task that might require better support in the application.

The small number of participants seriously limits the conclusions that can be drawn from the evaluation, but using a proper setup (described in the next section) will give at least some valuable insights.

6.2 Setup

A realistic scenario was created that the participants were expected to understand easily: the personnel administration of a high school. The employees are teachers, assistants and other positions commonly available at high schools. Since all participants themselves have been in such a school for many years, this scenario makes the "common-sense argument" described in the previous section even more credible. The next sections describe the steps taken in every evaluative session.

6.2.1 Introduction

Before the participant started with the task, he was told about the high school scenario and asked to imagine being the personnel administrator of such a school. No further information about the program was given, as the intuitiveness of the new interaction paradigm is a key element of the evaluation. Participants were asked to think aloud and to say anything they thought about the program and the way it reacted. Finally, participants were asked whether they agreed upon capturing everything that happened on the screen and recording audio during execution of the task.

6.2.2 The task

All participants were given the same set of tasks, described and printed out on a sheet of paper. Actually, this set consists of *questions* and *assignments*. The participants were asked to answer the questions by speaking out the answer and to perform the assignments. The full task description can be found in the appendix on page 65 and consists of the first two types of sub-tasks described in section 2.2 (The task), *finding information about an employee* (questions) and *editing data of an employee* (assignments). Task initiation according to section 2.2, was a *trigger* or *own initiative*. These correspond to a work item in the work list on the one hand, and a task given on the sheet of paper on the other hand. The latter type of task initiation actually is also a type of *trigger* as the participant in this case is executing a task described in the task description provided on the sheet of paper. The difference nevertheless is clear: the *trigger* type of tasks provide the user with a "What is the purpose of being here" kind of task description (see page 37).

To be able to see whether the idea of landmarks was useful, multiple tasks were related to the same employees. Testing of the situational awareness was done by providing tasks in an order that required the participant to go back to previous places multiple times.

6.2.3 Observation

Besides asking the participant to think aloud, as much feedback as possible was required to improve the quality of the final conclusions. As this project stresses animation and direct manipulation, valuable information could be extracted from everything that happened on the screen. To be able to analyse this behaviour in more detail afterwards, the screen was captured during task execution and audio was recorded in parallel. Complete analysis was not the purpose of capturing this data, but it is obvious that it would be a useful addition to the questionnaires.

6.2.4 Questionnaire

After performing the tasks, the participant was asked to fill out a questionnaire with 18 questions on a 1 to 5 scale and two open questions. This questionnaire was custom created for this project, but inspired by well known standard questionnaires for assessing usability of software and websites (USE[38], SUMI[39], CUSQ[40]). As is pointed out in [41], creating custom questionnaires has some drawbacks that could make *quantization* of the results impossible. As the evaluation at hand is of a more qualitative nature anyway and fairly limited, this drawback was accepted.

The full questionnaire can be found in the appendix on page 67. As the evaluation was done among Dutch participants, the questionnaire is in the Dutch language. It incorporates five sub-scales, each containing two or more questions related to it:

- Ease of use (1,3,8,13,16)
- Learnability (7,9,12)
- Satisfaction (2,6,15,18)
- Efficiency (5,10,11,17)
- Effectiveness (4, 14)

The two open questions at the end of the questionnaire asked the participant to mention the three most negative and the three most positive characteristics of the program.

6.2.5 Post questionnaire interview

In addition to the questionnaire, the participants were asked for further remarks if this was needed to clarify their answers. This happened at the same place, enabling the participant to use the prototype to make clear what he wanted to say.

6.3 Results

The results of the questionnaire can be found in appendix C on page 69. Question 10 ("Wanneer ik een fout maak tijdens het gebruik van het programma, kan ik die snel en eenvoudig herstellen" / *"When I make an error while using the program, I can correct it fast and easily")* is left out of the analysis, as it was ambiguous and participants thus interpreted it differently. Furthermore, this question turned out to be irrelevant as it was related to error correction; no serious errors were made by the participants and the prototype did not contain any functionality to indicate errors in their input. As a matter of fact, the latter is clearly a limitation of the prototype.

We used a 5 point scale questionnaire with value 1 as the minimum and 5 as maximum. The mean values of the results are rather high: around 4.5 for all subscales, with standard deviations between 0.3 and 0.6 (see appendix C, page 69). Although the sample size is way too small to draw statistically significant results from it, it does provide the impression that the program scores high on all five subscales (ease of use, learnability, satisfaction, efficiency and effectiveness).

Very interesting results can be obtained from the two open questions in which the participant was asked to mention most negative and most positive characteristics of the program. Wording of answers on this questions differed greatly, but could easily be aligned to a rather limited set of positive and negative characteristics (see appendix C). These are given in tables 1 and 2, together with the number of participants that mentioned them. These tables also provide counts of the number of participants that, based on the observation and analysis of the screen captures, also perceived the characteristic as negative or positive. Characteristics referring to limitations of the prototype or unexpected labelling of certain elements are left out of the analysis, as they do not give any insight into matters relevant for this thesis.

	Negative Characteristics	$\mathbf{n_q}$	$\mathbf{n_{obs}}$
1	The fisheye list of employees is too sensitive, resulting in confusion, frustration and lack of control.	**6**	7
2	The "Where am I now" icon provokes users to click on it, although it is not clickable.	**1**	3
3	Absence of undo functionality accompanying the "save" button on text areas.	**1**	2
4	The "What is the purpose of being here" sticky note is not always noticed.	**1**	2
5	The subcategory panels holding the attributes fold out when their preview is clicked, but do not fold in when clicked again.	**1**	3
6	It is not clear whether the data are instantly saved after changing them.	**0**	2

Table 1: Most negative characteristics of the program according to participants (n_q indicates number of participants mentioning this in the questionnaire, n_{obs} the observation based number of participants).

It is clear that the most negative characteristic of the program is the fisheye list of employees. It appeared to be too sensitive. Participants were clearly out of control, especially when entering the list for the first time and when trying to enter the list from one of the sides, expecting to point the mouse cursor over a certain entry. Nevertheless, three participants said that working for a longer time with this list might improve their performance. This is in accordance with the observations and analysis of the screen captures; most participants, while still being irritated, gain more control over the list during the session. Two participants, who turned out to be experienced CAD designers, were clearly better in control of the list than the others. Although they also had trouble with it, they were much more successful at finding and clicking entries in the list and reading the information in the previews. Most notably, they moved the mouse much more carefully and precisely, resulting in a more smooth operation.

It is to be expected that the effect of the confusion resulting from the fisheye list might have influenced the performance of the participants negatively. At least for two participants, they seemed to be so involved with trying to control the list, that they were no longer able to note the additional information in the preview (employee function and department).

For the second negative characteristic, there are two possible explanations and related solutions.

- The prototype showed the "pointer" type of mouse cursor on the "Where am I now" icon, which should not be the case as it is used to indicate that the item under the cursor

is clickable. This can easily be solved be defining the normal cursor to be show on this icon.

- The position of the “Where am I now” icon is sub optimal. It might have been perceived as part of the “Where do I come from” element, despite usage of colour and borders (see 4.2.3, “Integrated mock-ups“).

What mostly contributes to the described behaviour is not clear, but defining the correct cursor might already prevent the user from clicking, as most participants hover over most interface elements in the beginning in an explorative manner. When the correct cursor is used, they will no longer be provoked to click.

The third negative characteristic was reported by fewer participants; they wanted to undo their changes to a textual attribute, thus returning to the original value. No button for this is available. The functionality is available, though, using the browser's undo functionality which works on forms. Drawback of this built-in undo functionality is that it is rather unknown and limited. Besides, undo functionality should be tightly integrated into the program. This issue could partly be solved by adding a typical undo icon (as found in most office applications) to every text area. Just like the “save” button, it should only be displayed when working on the attribute it relates to. A complete solution should also include the possibility to go back more than one step: an undo stack. This is an interesting topic for further investigation.

The fourth negative characteristic is interesting as it is related to one of the major topics of this thesis; while most users noted the task description moving from the preview (with only a task title) to the “what is the purpose of being here” area (with a full task description), some clearly did not. When it was pointed out, they said they focused so much on the centre of the screen, that the animated move of the task description did not get their attention. Once they knew about this feature, they had no problems with it anymore, although one participant stated that the animation was too fast to keep track of it. The reason they initially did not notify the full task description, might be found in the fact that they tried to focus as much as possible on the task title (which actually was already a task description in case of the first task they had to perform), because they thought they had to remember it themselves. In that case, this negative characteristic is not too big a problem as it is solved as soon as the user knows how it works. Actually, most participants were very enthusiastic about this feature, as will be discussed below.

The fifth negative characteristic mentioned by one participant and observed in two others, is the fact that the panels holding a subcategory's attributes does not fold in when its preview is clicked again. It only folds in when another subcategory of the same employee is selected, but some participants wanted to be able to fold them in theirselves. This behaviour can easily be added, without further changes to the interaction.

In addition to the negative characteristics reported by participants in the questionnaire, there is one other characteristic that seemed to be negative for some participants, based only on observation (including the thinking-aloud): it is not clear whether the data are instantly saved after changing them (negative characteristic 6). As this characteristic was not reported in the questionnaire, it seems to be a minor issue. Otherwise, it should have been mentioned explicitly. Nevertheless, it deserves attention as it possibly points at interesting problems strongly related to the idea of direct manipulation. Two participants wondered whether they should press a save button some where after they changed some attributes. It might be a matter of some explanation and getting used to, but further investigation is needed.

Overall, the positive characteristics reported by the participants greatly outnumber the negative ones. When filling out the questionnaire, most of them had hard times thinking of negative features (apart from the fisheye list), but seemed to have less trouble thinking of the

characteristics and features they liked. Table 2 gives a summary of this. The participants' remarks again are aligned to a limited set of characteristics (see appendix C). The numbers based on observation are not always available, as certain remarks simply cannot be derived from behaviour or the words participants say.

	Positive Characteristics	$\mathbf{n_q}$	$\mathbf{n_{obs}}$
1	The task description sticky note is very useful. You immediately see that it is there and do not have to remember yourself what has to be done. The visualization to show that a task is done is also very convenient.	**3**	5
2	Information in preview is useful, as you can see it quickly without clicking any further.	**3**	6
3	Clean, clear and logically ordered structure and layout.	**6**	7
4	Intuitive, easy to understand without any explanation.	**3**	7
5	Overall colour usage (in previews, pictures and icons).	**4**	
6	Good navigation, easy to find information, sometimes even without a need to click.	**4**	7

Table 2: Most positive characteristics of the program according to participants (n_q indicates number of participants mentioning this in the questionnaire, n_{obs} the observation based number of participants).

In addition to the six characteristics shown in table 2, some remarks are also worth consideration. The results from the 5 point scale questions already provide an overall estimate of the usability of the new interaction paradigm, but the following utterances are very illustrative and might further support the high scores from the questionnaire:

- *"It's very clean and ordered... no irrelevant information is shown, but it is very easy to make other information visible... if you need to. You instantly know where to go."*
- *"The colours are also very nice... I like that. It's like everything being more... smooth... pleasant for the eyes. No hard edges, as with the SAP software we use at my company."*
- *"It's suitable for dummies."*
- *"That's cool!"* (when navigating back to the starting point)
- *"The way it shows that the task is finished is very nice. It gives a feeling of... satisfaction, a feeling of making progress... tidiness".*

The next section describes what conclusions can be drawn from these results and how this all relates to the goals of this thesis.

6.4 Conclusions

The results described are promising. Because the group of participants was rather small, no statistically valid results can be obtained. Nevertheless, the results give a good impression. The next sections each draw conclusions related to the major topics of this thesis work.

6.4.1 Navigation

Navigation works very well. It is easy to find the way back. The trail used to visualize the path that leads to the current position is very intuitive. The idea of landmarks seems to work well too. Most participants used the information that could be obtained from it. The colours were also appreciated. They were useful to remember places and types of information. Usage of animation seems to contribute to it too.

The fish-eye list showing the employees did not work out very well in its current form. Some training might help users to work with it effectively, but it is clear that the sensitivity is too high and users thus do not have enough control over it. Nevertheless, the functionality it provided trough semantic zooming was used and appreciated. It reduces the amount of clicks needed to find information, while keeping the interface simple and uncluttered.

6.4.2 Direct manipulation

The idea of direct manipulation works well. Some users will have to get used to it, as it is different from conventional software. Especially the fact that there is no need to explicitly save manipulated data raised some questions. The presence of the green save-button when manipulating a textual attribute made some participants click it when they finished manipulating the corresponding attribute. But some other participants ignored it, most probably because it disappeared without any notifications after using the keyboards tab-button to navigate to the next field or after clicking anywhere outside the text area. The latter is default browser behaviour and cannot be circumvented. The addition of the green save-button to the design actually was an anticipation to the confusion that was expected to arise within users not used to using the tab-button to navigate in forms, or those not knowing about clicking anywhere outside a textual input to release the focus on it.

6.4.3 Fluid interaction

The new interaction paradigm seems to be a step forward into the direction of more fluid interaction, lacking the jumps and breaks that are common to many business software systems. Usage of animation helps to establish a sense of an holistic experience. It seems to make working with the application fun as well.

6.4.4 General usability

The results of the questionnaire clearly indicate that the new interaction paradigm is very usable. The same conclusion can be drawn based upon the observations and the remarks of the participants. When discussing the question whether it is “fun” to work with an application based on the new interaction paradigm, we strongly have the impression that people really find working with it enjoyable and satisfying.

6.4.5 Browsers

The prototype did not run smoothly in the most used browsers (Firefox and Internet Explorer). The fact that it does run smoothly in the latest Safari browser, indicates that it should be possible to make a browser perform better, thus making fluent animation possible. It is not known how browsers will improve the next years, but it is not very unrealistic to expect overall improvements.

Apart from browser characteristics, the prototype itself could be optimized better. During the development of it, not too much time was available for optimization.

Chapter 7

CONCLUSIONS & FUTURE WORK

This thesis work resulted in a new interaction paradigm for browser based business software. From the evaluation it can be concluded that it indeed seems to deliver fluid interaction, lacking the jumps and breaks common to many web based business software applications. The notion of an information space, good navigation and a sense of direct manipulation seems to be fruitful and properly addressed in the paradigm. The limitations of the browser environment are still somewhat problematic, but might be solved the next years.

Due to the very limited number of participants, the results of the evaluation only provide the rough impression that the direction of this work is promising. More extensive research with larger groups of participants is needed to evaluate the paradigm more rigorously.

In addition to that, the prototype developed was somewhat limited in the level of realism it reflected. It provided only the means to perform a subset of the tasks the personnel administrator needs to perform. More generally, actions like deleting business objects and creating new ones were not possible, but are needed in a realistic setting.

Another interesting question is how the paradigm can be extended to incorporate connections between multiple more or less separated information spaces.

Some ideas presented in this thesis were not worked out in detail and not implemented in the prototype. Nevertheless, some of them are interesting and deserve further attention in future work. The same is true of a lot of topics related to this work. Below we provide a list of topics that are interesting for further investigation.

- **See-through tool** – A see-through tool could be used to add functionality to directly act upon a list of objects, improving the idea of direct manipulation. The focus of the mouse pointer determines which business object in the list is under control and can be acted upon. It keeps the interface clean and uncluttered, as no additional buttons are needed for every object.

- **Cartoon animation techniques** – This technique could be applied to improve the sense of object constancy during the manipulation of business objects. Every time the user wants to move, a check could be performed to see whether the attributes in the current place are valid. If so, an animated move to the requested place is made. If not, the view of the current place is animated according to the *principle of reluctance* (see page 25). This

must be done so, that the user perceives the fact that the place does not let the user go, as its data are not yet correct.

- **Time-line** – This feature is already shown in Figure 4.13. It adds the possibility to view and edit attributes in sets relating to different periods of time, which would be needed to account for the visualization of the non-overlapping sets of attributes discussed in 2.1. For the visualization of overlapping sets of attributes, a slightly different approach would be needed.

- **Improved and augmented fish-eye list** – From the evaluation, it is clear that the fish-eye list is too sensitive. Nevertheless, some participants thought that some experience with it might improve their performance. The question is what exactly is *too* sensitive and what is just fine. More qualitative research could answer this question. Another point of concern is additional functionality making this kind of list more useful in a realistic setting (many more employees). The fish-eye list should be augmented with dynamic filtering options. This could, for example, give the user the possibility to search for employees with names starting with certain characters.

- **Undo functionality** – The current prototype did not include any undo functionality, apart from what the browser provides. The latter is limited to undoing changing values in a form and is rather limited and unknown to most users. It would be interesting to investigate how a more advanced undo system could be designed that fits within the paradigm and adheres to the principles we used, thus making it easy and understandable for the user to undo certain manipulations.

- **User research and grouping** – As already pointed out in section 1.1 (Problem description), it would be interesting to do more extensive user research to clearly identify the user needs and how these can differ among different types of users. Distinction might be made between novices and experts, and between users working with the application very often and those only using it incidentally. Also regarding the best way to model the information space, differentiation might be required, based on more extensive use research (see section 3.1.3).

REFERENCES

[1] Mihaly Csikszentmihalyi. *Flow. The psychology of optimal experience*. Harper Perennial, New York, 1990.

[2] Christopher D Wickens, Sallie E. Gordon & Yili Liu. *An introduction to human factors engineering*. Addison Wesley Longman, New York, 1998.

[3] P. Péruch & F. Gaunet. Virtual environments as a promising tool for investigating human spatial cognition. *Cahiers de Psychologie Cognitive*, 17(4-5), pages 881-899, 1998.

[4] Sibylle D. Steck & Hanspeter A. Mallot. *The Role of Global and Local Landmarks in Virtual Environment Navigation (Technical Report No. 63)*. Max-Planck-Institut für biologische Kybernetik, Tübingen, Germany, 1997.

[5] B. Schölkopf & H.A. Mallot. View-based cognitive mapping and path planning. *Adaptive Behavior*, 3(3), pages 311-348, 1995.

[6] Eelco Herder. *Forward, Back and Home Again. Analyzing User Behavior on the Web*. [Ph.D. thesis], University of Twente, 2006.

[7] Joonah Park & Jinwoo Kim. Contextual Navigation Aids for Two World Wide Web Systems. *International Journal of Human-Computer Interaction*, 12(2), pages 193-217, 2000.

[8] Y.L. Theng, H. Thimbleby, M Jones. "Lost in hyperspace": Psychological problem or bad design?. In *Proceedings First Asia Pacific Conference on Computer Human Interaction (APCHI'96)*, pages 387-396. IEEE, Piscataway, NJ, 1996.

[9] Manfred Thüring, Jörg Hanneman & Jöorg M. Haake. Hypermedia and Cognition: Designing for Comprehension. *Communications of the ACM*, 38(8), pages 57-66, 1995.

[10] Jeff Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 20(7), pages 17-41, 1987.

[11] George A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 101(2), pages 323-352, 1956.

[12] H.L. Resnikoff. *The Illusion of Reality*. Springer-Verlag, New York, 1989.

[13] Kasper Hornbæk, Benjamin B. Bederson, Catherine Plaisant. Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview. *ACM Transactions on Computer-Human Interaction*, 9(4), pages 362-389, 2002.

[14] Benjamin B. Bederson & James D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 17-26. ACM Press, New York, NY, USA, Marina del Rey, California, United States, 1994.

[15] Benjamin B. Bederson & Angela Boltman. Does animation help users build mental maps of spatial information?. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 28-35. IEEE, Piscataway, NJ, 1999.

[16] George W. Furnas. Generalized Fisheye Views. In *Human Factors in Computing Systems CHI'86 Conference Proceedings*, pages 16-23. ACM, New York, 1986.

[17] George W. Furnas. A fisheye follow-up: Further reflections on focus + context. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 999-1008. ACM, New York, 2006.

[18] B.B. Bederson. Fisheye menus. In *Proceedings of ACM Symposium on User Interface Software*

and Technology (UIST'00), pages 217-226. ACM, San Diego, CA, 2000.

[19] Eric A. Bier, Maureen C. Stone, Ken Oier, William Buxton & Tony D. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73-80. ACM, New York, 1993.

[20] Stephen T. Kerr. Wayfinding in an electronic database: The relative importance of navigational cues vs. mental models. *Information Processing & Management*, 26(4), pages 511-523, 1990.

[21] P.N. Johnson-Laird. Mental Models. In *Foundations of Cognitive Science*. M.I. Posner (Ed.). 1989. pp. pages 469-499.

[22] Gerald Franz, Mallot, Hanspeter, A & Jan M. Wiener. Graph-based models of space in architecture and cognitive science - a comparative analysis. In *Architecture, Engineering and Construction of Build Environments*, pages 30-38. Max-Planck-Institut für biologische Kybernetik, Baden-Baden, 2005.

[23] J.J. Gibson. Visually controlled locomotion and visual orientation in animals. *British Journal of Psychology*, 49(), pages 182-194, 1958.

[24] Matthew P. Kirschen, Michael J. Kahana, Robert Sekuler & Benjamin Burack. Optic flow helps humans learn to navigate through synthetic environments. *Perception*, 29(7), pages 801-818, 2000.

[25] Cleotilde Gonzalez. Does animation in user interfaces improve decision making?. In *Conference on Human Factors in Computing Systems - Proceedings* , pages 27-34. ACM, New York, 1996.

[26] George G. Robertson, Jock D. Mackinly & Stuart K. Card. Cone Trees: Animated 3D visualizations of hierarchical information. In *Proceedings of the ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 189-194. ACM, New York, NY, 1991.

[27] Jack D. Mackinly, George G Robertson & Stuart K. Card. The perspective wall: detail and context smoothly integrated. In *Proceedings of the ACM CHI'91 Conference on Human Factors in Computing Systems*, pages 173-179. ACM, New York, NY, 1991.

[28] Bruce H. Thomas & Paul Calder. Applying cartoon animation techniques to graphical user interfaces. *ACM Transactions on Computer-Human Interaction*, 8(3), pages 198-222, 2001.

[29] D.A. Norman. Emotionally centered design. *Interactions*, 13(3), page 53+71, 2006.

[30] Noam Tractinsky. Aesthetics and Apparent Usability: Empirically Assessing Cultural and Methodological Issues. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 115-123. ACM Press, Atlanta, GA, USA, 1997.

[31] Kristiina Karvonen. The Beauty of Simplicity. In *Proceedings on the 2000 conference on Universal Usability*, pages 85-90. ACM Press, New York, NY, USA, 2000.

[32] Jakob Nielsen. *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, Berkeley, California USA, 2000.

[33] Gitte Lindgaard & Cathy Dudek. What is this evasive beast we call user satisfaction?. *Interacting with Computers*, 15(), pages 429-452, 2003.

[34] Mark S. Silver. Browser-based applications: popular but flawed?. *Information Systems and E-Business Management*, 4(4), pages 361-394, 2006.

[35] Corné A. Verbruggen. *Usability of Rich Internet Applications*. University of Twente (unpublished)

[36] Jesse James Garrets. *Ajax: A New Approach to Web Applications* [web essay] February 2005, `http://adaptivepath.com/publications/essays/archives/000385.php`. [Accessed January 13th, 2007]

[37] Jennifer Tidwell. *Designing Interfaces - Patterns for Effective Interaction Design*. O'Reilly Media, Inc, Sebastopol, CA 95472, 2006.

[38] Arnold M. Lund. *Measuring Usability with the USE Questionnaire* [web page] , `www.stcsig.org/usability/newsletter/0110_measuring_with_use.html`. [Accessed October 9th, 2007]

[39] *SUMI, The de facto industry standard evaluation questionnaire for assessing quality of use of software by end users* [web page] `sumi.ucc.ie/index.html`. [Accessed November, 2007].

[40] J.R. Lewis. IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. *International Journal of Human-Computer Interaction*, 7(1), pages 57-78, 1995.

[41] Kasper Hornbæk. Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human Computer Studies*, 62(2), pages 79-102, 2006.

Appendix A - Evaluation task description

	Vragen en Opdrachten
1	Welke functie heeft Denise Hollenbach?
2	Wat is het privé-telefoonnummer van Gerrit Klompenhouwer?
3	Denise Hollenbach is verhuisd. Haar nieuwe adres is: Hoofdweg 12 9723 AR Groningen Tel: 050 856 23 78 Werk haar gegevens bij.
4	Voer de eerste taak van de takenlijst in het programma uit.
5	Gerrit Klompenhouwer is getrouwd. Werk zijn gegevens bij.
6	Wat is het bankrekeningnummer van Christiaan Verhoeven? En op welk adres woont hij?
7	Wat is de functie van Raymond Keetelaars? Hoeveel uur per week werkt hij?
8	Gerrit Klompenhouwer wil ontslag nemen. Wat bedraagt zijn opzegtermijn?
9	Voer de overige taken van de takenlijst in het programma uit.

Appendix B - Evaluation questionnaire

		Helemaal mee oneens	Enigszins mee oneens	Niet mee eens, niet mee oneens	Enigszins mee eens	Helemaal mee eens
		- 1	2	3	4	5 +
1	Over het geheel genomen ben ik tevreden met hoe eenvoudig het is om met dit programma te werken.					
2	Het programma ziet er prettig uit.					
3	Het programma maakt het uitvoeren van de opdrachten eenvoudig.					
4	Dit programma helpt me de opdrachten volledig en naar tevredenheid te uit voeren.					
5	Ik kan de opdrachten met dit programma vlot uitvoeren.					
6	Ik voel me op mijn gemak tijdens het gebruik van dit programma.					
7	Het was eenvoudig om met dit programma te leren werken.					
8	Het is eenvoudig om de informatie te vinden die ik nodig heb.					
9	Ik geloof dat ik met dit programma snel produktief ben.					
10	Wanneer ik een fout maak tijdens het gebruik van het programma, kan ik die snel en eenvoudig herstellen.					
11	Het programma reageert niet snel genoeg.					
12	Ik had al vrij vlot door hoe het programma werkte.					
13	De ordening van informatie op het scherm is duidelijk.					
14	Ik kan de opdrachten goed uitvoeren met dit programma.					
15	Ik vind het leuk om met dit programma te werken.					
16	Ik had niet echt het idee dat ik het programma onder controle had.					
17	Het programma maakt het uitvoeren van de taken ingewikkelder dan nodig is.					
18	Over het geheel genomen ben ik tevreden met dit programma.					

Noem de meest *negatieve* eigenschappen van dit programma:

1.

2.

3.

Noem de meest *positieve* eigenschappen van dit programma:

1.

2.

3.

Appendix C - Questionnaire Results

		P1	P2	P3	P4	P5	P6	P7	Mean	StdDev
EoU	**Q1**	5	5	4	4	4	5	5	4.57	0.53
	Q3	5	4	4	3	5	5	5	4.43	0.79
	Q8	5	4	5	5	4	5	5	4.71	0.49
	Q13	4	4	4	5	4	5	5	4.43	0.53
	-Q16	5	5	5	4	3	5	5	4.57	0.79
	Mean	*4.8*	*4.4*	*4.4*	*4.2*	*4*	*5*	*5*	***4.54***	***0.4***
L	**Q7**	5	4	5	5	4	5	5	4.71	0.49
	Q9	5	3	5	4	4	5	5	4.43	0.79
	Q12	5	4	4	4	4	5	5	4.43	0.53
	Mean	*5*	*3.67*	*4.67*	*4.33*	*4*	*5*	*5*	***4.52***	***0.54***
S	**Q2**	4	5	5	4	4	5	5	4.57	0.53
	Q6	4	2	5	5	4	5	5	4.29	1.11
	Q15	5	5	4	4	4	4	5	4.43	0.53
	Q18	5	5	4	5	4	5	5	4.71	0.49
	Mean	*4.5*	*4.25*	*4.5*	*4.5*	*4*	*4.75*	*5*	***4.5***	***0.32***
Effi	**Q5**	5	5	4	4	4	5	5	4.57	0.53
	-Q11	5	5	5	4	3	5	5	4.57	0.79
	-Q17	5	5	3	4	4	5	5	4.43	0.79
	Mean	*5*	*5*	*4*	*4*	*3.67*	*5*	*5*	***4.52***	***0.6***
Effe	**Q4**	5	4	4	4	5	5	5	4.57	0.53
	Q14	5	5	4	4	4	5	5	4.57	0.53
	Mean	*5*	*4.5*	*4*	*4*	*4.5*	*5*	*5*	***4.57***	***0.45***

Table 3: Results of the 5-point scale questions (EoU: Ease of Use, L: Learnability, S: Satisfaction, Effi: Efficiency, Effe: Effectiveness)

Participant	Negative features	Alignment
1	Verspringende balkjes met namen.	1
1	[Start-point icon] -> roept op om op te gaan klikken, terwijl al hoofdmenu.	2
2	Sorteren van gegevens.	L
2	Geen kruisje voor ongedaan maken.	3
3	De flexibele zoekfunctie van personen.	1
3	Hokje met taken aan linkerzijde valt niet op. Beweegt te snel naar links.	4
4	Wijze van weergave namenlijst.	1
5	Namen scrollen te snel: lastig aanvinken.	1
5	Nummers rechts: functie?	W

Participant	Negative features	Alignment
6	Lijst met medewerkers: als je met de muis over de lijst heen beweegt vind ik het een beetje lastig om een medewerker te selecteren door de "gevoeligheid". Dit maakt het gebruik wat onrustig.	1
6	In subcategorieën zou het logischer zijn geweest dat je na een 2e muisklik het tabblad weer inklapt.	5
7	Gegevens verschuiven als je ze wilt aanklikken.	1

Table 4: Negative characteristics as described by the participants in the questionnaires (see Table 6, L: Due to limitation of prototype, W: Due to wrong labelling in prototype).

Participant	Positive features	Alignment
1	Overzichtelijk & geordend.	3
1	Makkelijk werkbaar, zonder uitleg te snappen.	4
1	Kleuren zijn erg handig en helder, prettig.	5
2	Overzichtelijk.	3
2	Handig van taak -> persoon.	1
3	Kleurigheid v/h programma.	5
3	Eenvoudige en gestructureerde opzet.	3, 4
3	Toepassing van foto's.	2
4	Overzichtelijke weergave, rustig.	3
4	Kleurgebruik.	5
4	Logische indeling van informatie, duidelijke navigatie.	6
5	Logisch overzicht.	3
5	Makkelijk gegevens te vinden.	6
5	Prettig bij taken -> opdracht blijft staan.	1
6	Navigatie erg gebruiksvriendelijk. Qua geschiedenis, oplichten van het geselecteerde, kleuren.	6, 5
6	Overzichtelijk. Irrelevante informatie is niet in beeld, maar het is wel gemakkelijk die info weer tevoorschijn te toveren.	3, 2
6	Gele briefjes: overzichtelijk. Taken die voltooid zijn wegstrepen, geeft een gevoel van voldoening, opgeruimd gevoel.	1
7	Gegevens zijn snel toegankelijk, zelfs zonder door te klikken.	6, 2
7	Makkelijk aan te leren.	4

Table 5: Positive characteristics as described by the participants in the questionnaires (see Table 7).

	Negative characteristics alignment
1	The fisheye list of employees is too sensitive, resulting in confusion, frustration and lack of control.
2	The “Where am I now” icon provokes users to click on it, although it is not clickable.
3	Absence of undo functionality accompanying the “save” button on text areas.
4	The “What is the purpose of being here” sticky note is not always noticed.
5	The subcategory panels holding the attributes fold out when their preview is clicked, but do not fold in when clicked again.
6	It is not clear whether the data are instantly saved after changing them.

Table 6: Negative characteristics alignment

	Positive characteristics alignment
1	The task description sticky note is very useful. You immediately see that it is there and do not have to remember yourself what has to be done. The visualization to show that a task is done is also very convenient.
2	Information in preview is useful, as you can see it quickly without clicking any further.
3	Clean, clear and logically ordered structure and layout.
4	Intuitive, easy to understand without any explanation.
5	Overall colour usage (in previews, pictures and icons).
6	Good navigation, easy to find information, sometimes even without a need to click.

Table 7: Positive characteristics alignment

Appendix D - Implementation code

XHTML code excerpt

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Personeelsadministratie</title>

    <!-- include the stylesheet: -->
    <link rel="stylesheet" href="css/pa.css" type="text/css" media="screen" charset="utf-8" />

    <!-- include javascript libraries: jquery, jquery dimensions plugin, jquery interface plugin, personnel data file,
                        the custom javascript code and the template library -->
    <script src="js/jquery.js" type="text/javascript" charset="utf-8"></script>
    <script src="js/jquery.dimensions.js" type="text/javascript" charset="utf-8"></script>
    <script src="js/interface.js" type="text/javascript" charset="utf-8"></script>
    <script src="js/pa_data_nl.js" type="text/javascript" charset="utf-8"></script>
    <script src="js/pa.js" type="text/javascript" charset="utf-8"></script>
    <script src="js/trimpath-template.js" type="text/javascript" charset="utf-8"></script>
  </head>
  <body>
    <!-- TEMPLATE DEFINITIONS -->

    <!-- Template for employee -->
    <textarea id="employee-template" style="display:none"">
      {for e in employees}
        <div id="e-${e_index}-preview" num=${e_index} class="employee-preview preview" altColor="#ced7ff">
          <span class="landmark-wrap">
            <img id="e-${e_index}-landmark" src="img/portraits/${e.portrait}" class="landmark" />
          </span>
          <span class="first-name">${e.firstName}</span>
          <span class="last-name">${e.lastName}</span>
          <span class="id-nr">${e.id}</span>
          <span class="extra">${e.func} - ${e.department}</span>
        </div>
        <div id="e-${e_index}" class="employee view">
          <div altColor="#e36565" class="category preview" style="background-color: #ffb5b5;">
            <span class="landmark-wrap">
              <img src="img/landmark_employee_personal_data.png" class="landmark" />
            </span>
            <span class="title">Persoonlijke gegevens</span>
            <table>
              <tr>
                <td class="extra1">${e.firstName} ${e.lastName} (${e.id})</td>
              </tr>
              <tr>
                <td class="extra2">${e.city}</td>
              </tr>
            </table>
          </div>
          <div id="e-${e_index}-personal" class="category view">
            <div altColor="#e36565" class="sub-category preview" style="background-color: #ffb5b5;">
              <img src="img/landmark_employee_personal_data_biographical_data.png" class="landmark" />
              <span class="title">Biographische gegevens</span>
              <table>
                <tr>
                  <td class="extra1">${e.firstName} ${e.lastName} - ${e.id}</td>
                </tr>
                <tr>
                  <td class="extra2">${e.dob}</td>
                </tr>
              </table>
            </div>
            <div class="sub-category view">
              <table class="left">
                <tr>
                  <td class="lbl">Aanhef</td>
                  <td><input type=text size=8 value="${e.foa}" /></td>
                </tr>
                <tr>
                  <td class="lbl">Titel</td>
                  <td><input type=text size=8 value="${e.title}" /></td>
                </tr>
                <tr>
                  <td> </td>
                  <td></td>
                </tr>
                <tr>
                  <td class="lbl">Voornaam</td>
                  <td><input type=text size=20 value=${e.firstName} /></td>
                </tr>
                <tr>
                  <td class="lbl">Achternaam</td>
                  <td><input type=text size=20 value=${e.lastName} /></td>
                </tr>
```

```
            <tr>
              <td class="lbl">Tussenvoegsel</td>
              <td><input type=text size=8 value=${e.midName} /></td>
            </tr>
            <tr>

            </tr>
              <td class="lbl">Bijnaam</td>
              <td><input type=text size=20 value=${e.nickName} /></td>
            </tr>
          </table>
          <table class="right">
            <tr>
              <td class="lbl">Geslacht</td>
              <td>
                <select>
                  <option {if e.gender} selected {/if}>
                    Man
                  </option>
                  <option {if !e.gender} selected {/if}>
                    Vrouw
                  </option>
                </select>
              </td>
            </tr>
            <tr>
              <td class="lbl"> </td>
              <td></td>
            </tr>
            <tr>
              <td class="lbl">Geboortedatum</td>
              <td><input type=text size=20 value="${e.dob}" /></td>
            </tr>
            <tr>
              <td class="lbl">Geboorteplaats</td>
              <td><input type=text size=20 value="${e.pob}" /></td>
            </tr>
            <tr>
              <td class="lbl">Nationaliteit</td>
              <td><input type=text size=20 value="${e.nat}" /></td>
            </tr>
            <tr>
              <td class="lbl"> </td>
              <td></td>
            </tr>
            <tr>
            </tr>
              <td class="lbl">Burgelijke staat</td>
              <td>
                <select>
                  <option {if e.ms == "nm"} selected {/if} >
                    Ongehuwd
                  </option>
                  <option {if e.ms == "m"} selected {/if}>
                    Gehuwd
                  </option>
                  <option {if e.ms == "rp"} selected {/if}>
                    Geregistreerd partnerschap
                  </option>
                </select>
              </td>
            </tr>
          </table>
        </div>
          :
          :
          :
          :
     {/for}
</textarea>

<!-- template for work-item -->
<textarea id="work-item-template" style="display:none">
  {for e in employees}
    {if e.task}
    <div id="e-${e_index}-preview" num=${e_index} class="work-item-preview preview" altColor="#ced7ff">
      <span class="landmark-wrap">
        <img id="e-${e_index}-landmark" src="img/portraits/${e.portrait}" class="landmark" />
      </span>
      <span class="first-name">${e.firstName}</span>
      <span class="last-name">${e.lastName}</span>
      <div id="${e_index}-task-preview" class="task-preview">
        ${e.task}
        <div class="task-desc">
          ${e.taskDesc}
        </div>
        <div class="task-done">voltooid</div>
        <img class="task-icon" src="img/icons/${e.taskIcon}" />
      </div>
    </div>
    <div id="w-e-${e_index}" class="employee view">
      :
      :
      :
      :
    </div>
    {/if}
  {/for}
</textarea>

<!-- END OF TEMPLATE DEFINITIONS -->
```

```
    <!-- container element containing all other interface components -->
    <div id="container">

    <!-- status area, displaying history/path -->
    <div id="status" style="display: none; position: absolute; background: yellow; width: 100px;
      height: 20px;"></div>
      <div id="past">
        <img id="work-center-landmark" oriTop=5 oriLeft=136 class="landmark" src="img/landmark_work_center.png" />
      </div>

      <div id="current">
      </div>

      <!-- the actual work-center element -->
      <div id="work-center">

        <!-- work-items list -->
        <div id="work" class="wrap">
          <div id="work-list-preview" class="preview" style="background-color: #e5ff80;" altColor="#bed956">
            <span class="landmark-wrap">
              <img id="work-list-landmark" class="landmark" src="img/landmark_worklist_shine.png" />
            </span>
            <span class="title">Taken</span>
            <table>
              <tr>
                <td class="left-col">Nog niet uitgevoerd:</td>
                <td class="left-col" id="tasks-todo-count">3</td>
              </tr>
              <tr>
                <td class="left-col">
                  Totaal:
                </td>
                <td class="left-col" id="tasks-total-count">
                  3
                </td>
              </tr>
            </table>
          </div>
          <div id="work-list" class="view">
            <!-- FILLED BY TEMPLATE SYSTEM (instantiations of work-item template) -->
          </div>
        </div>

        <!-- employee list -->
        <div id="employees" class="wrap">
          <div id="employee-list-preview" class="preview" style="background-color: #f6f6f6;" altColor="#cccccc">
            <span class="landmark-wrap">
              <img id="employee-list-landmark" class="landmark" src="img/landmark_employee_list_shine.png" />
            </span>
            <span class="title">Werknemers</span>
            <table>
              <tr>
                <td class="left-col">Totaal:</td>
                <td class="left-col">26</td>
              </tr>
              <tr>
                <td class="left-col">
                  Met taken:
                </td>
                <td class="left-col">
                  3
                </td>
              </tr>
            </table>
          </div>
          <div class="view" id="employee-list">
            <!-- FILLED BY TEMPLATE SYSTEM (instantiations of employee template -->
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Cascaded StyleSheet

```
/* File Name: pa.css */
body {
          font-size: 12px;
          font-family: sans-serif;
}

.title {
          font-family: Bitstream Vera Sans, sans-serif;
          font-size: 18px;
          margin: 7px;
          line-height: 26px;
}
#container {
          position: absolute;
          width: 1000px;
          height: 600px;
          background: transparent;
}
#work-center {
          position: absolute;
          right: 0px;
          top: 0px;
          width: 808px;
          height: 598px;
```

```
        background: #f6f6f6;
        border: solid 1px #b4b4b4;
}
#past {
        position: absolute;
        left: 0;
        top: 0;
        width: 130px;
        height: 80px;
        background: transparent url(../img/past_gradient.png) 0 50% no-repeat;
}
#current {
        position: absolute;
        left: 130px;
        top: 0;
        width: 60px;
        height: 78px;
        background: #f6f6f6;
        border: solid 1px #b4b4b4;
        border-right: none;
        z-index: 1;
}
#work-center-landmark {
        position: absolute;
        z-index: 100;
        top: 5px;
        left: 136px;
}
.preview {
        position: relative;
        float:left;
        margin: 0 4px;
        background-image: url(../img/preview_shine.png);
        cursor: pointer;
        border-top: solid 4px #f6f6f6;
        width: 800px;
}
.preview table {
        margin-left: 60px;
}
#work-list-preview, #employee-list-preview {
        height: 70px;
}
.landmark {
        cursor: pointer;
}
.preview .landmark {
        height: 100%;
        float: left;
}
.left-col {
        width: 150px;
}
.view {
        clear: both;
        position: absolute;
        top: 0; /* will be set dynamically on click */
        height: 0;
        width: 100%;
        overflow: hidden;
        background: #f6f6f6;
        z-index: 1000;
}
.employee-preview {
        height: 10px; /* initially in fisheye list */
        font-size: 10px; /* initially in fisheye list */
        position: relative;
        overflow: hidden;
        background: #e7ecff;
        clear: both;
}
.sub-category.view {
        position: static;
        background: transparent url(../img/sub_category_gradient.png) repeat-y;
        margin: 0 4px;
}
.landmark-wrap, .first-name, .last-name, .id-nr{
        float: left;
        height: 100%;
        overflow: hidden;
}
.work-item-preview {
        width: 792px;
        font-size: 30px;
        background: #e7ecff;
}
.task-icon {
        position: absolute;
        right: 0;
        bottom: -5px;
        height: 40px;
}
.task-preview {
        font-size: 15px;
        background: #e5ff80;
        position: absolute;
        bottom: 5px;
        padding: 3px;
        right: 5px;
        width: 726px;
        height: 24px;
        overflow: hidden;
```

```
}
.task-desc {
          position: absolute;
          top: 35px;
          left: 10px;
          font-size: 11px;
}
.task-done {
          position: absolute;
          background: white;
          width: 100px;
          height: 20px;
          text-align: center;
          right: 60px;
          top: 108px;
          line-height: 20px;
          cursor: pointer;
          border: solid 2px black;
}
.task-done:hover {
          background: lime;
}
.inline-buttons {
          float: right;
          cursor: pointer;
          margin-right: 10px;
}
.first-name, .last-name, .id-nr {
          margin-left: 10px;
}
.landmark-wrap {
          width: 50px;
}
.first-name {
          width: 240px;
}
.last-name {
          width: 300px;
}
.id-nr {
          width: 180px;
}
.extra {
          display: block; position: absolute; top: 45px; left: 60px; font-size: 14px;
}
.extra1 {
          font-size: 12px;
}
.extra2 {
          font-size: 12px;
          font-style: italic;
}
/* input */
table.right {
          width: 380px;
          float: left;
          margin-top: 12px;
}
table.left {
          float: left;
          width: 380px;
          margin-top: 12px;
}
.employee td {
          padding: 2px 6px;
}
.lbl {
          text-align: right;
          width: 120px;
}
input, select, option {
          background: transparent;
          border: solid 1px white;
          font-weight: bold;
          width: 190px;
}
option {
          background: white;
}
```

Data file excerpt

```
/**
 * File name: pa_data_nl.js
 *
 * This file contains some test data for the Dutch PA prototype.
 */

var data = {
  employees : [
    {
      firstName: "Hendrik",
      midName: " ",
      lastName: "Arends",
      nickName: " ",
      gender: true,
      ms: "m",
      id: "048832",
      portrait: "man_01.jpg",
      foa: "Dhr.",
```

```
      dob: "12-4-1956",
      pob: "Maastricht",
      nat: "Nederlandse",
      street: "Eikenlaan",
      number: "23",
      zip: "9812 AK",
      city: "Hoogkerk",
      phone: "050 852 42 34",
      mobile: "06 435 253 93",
      email: "henkdrik56@hotmail.com",
      accountOwner: "H.K. Arends",
      accountNr: "9186631",
      bank: "Postbank N.V.",
      bankKey: "PSTBNL21",
      bankCountry: "Nederland",
      partner: "Mevr. H. Jansen",
      kids: "2",
      startDate: "1-5-1985",
      endDate: " ",
      hours: "32",
      func: "Boekhouder",
      department: "Administratie en financien",
      task: "Contract uitbreiden naar 38 uur",
      taskDesc: "Het contract moet per direct worden uitgebreid naar 38 uur per week.",
      taskIcon: "hours.png"
    },
    {
      firstName: "Arie",
      lastName: "Badenhuijsen",
      id: "D048832",
      portrait: "man_02.jpg",
      func: "Rector",
      department: "Directie"
    },
    {
      firstName: "Margreet",
      lastName: "Brandwijk",
      id: "028870",
      portrait: "woman_01.jpg",
      func: "Docent Frans",
      department: "Onderwijs onderbouw",
      midName: " ",
      nickName: " ",
      gender: false,
      ms: "m",
      foa: "Mevr.",
      dob: "12-4-1972",
      pob: "Maastricht",
      nat: "Nederlandse",
      street: "Eikenlaan",
      number: "23",
      zip: "9812 AK",
      city: "Hoogkerk",
      phone: "050 852 42 34",
      mobile: "06 435 253 93",
      email: "margreet923@hotmail.com",
      accountOwner: "M. Brandwijk",
      accountNr: "9186631",
      bank: "Postbank N.V.",
      bankKey: "PSTBNL21",
      bankCountry: "Nederland",
      partner: "Dhr. H. Jansen",
      kids: "1",
      startDate: "1-5-1995",
      endDate: " ",
      hours: "32",
      task: "Wijziging functie en cluster",
      taskDesc: "Wijzig functie in 'Docent Engels' en het cluster in 'Onderwijs bovenbouw'",
      taskIcon: "hours.png"
    },
    {
      firstName: "Frank",
      lastName: "Cornelissen",
      id: "D048832",
      portrait: "man_03.jpg",
      func: "Technisch onderwijs assistent",
      department: "Diensten en Faciliteiten"
    },
      :
      :
      :
      :
    {
      firstName: "Annemarie",
      lastName: "Zandstra",
      id: "D048832",
      portrait: "woman_13.jpg",
      func: "HR Administrator",
      department: "Finance and Admin"
    }
  ]
}
```


Custom JavaScript code

```
/**
 * File name: pa.js
 *
 * All other JavaScript libraries should already be included in the main XHTML file:
 * - jquery.js
```

```
 * - jquery.dimensions.js
 * - interface.js
 * - pa_data_nl.js
 * - pa.js
 * - trimpath-template.js
 */

/* A stack for storing the path/history with a roll-back action associated with every step */
var revertStack = new Array();

/* Variable used to determine whether rendering the fish-eye list is needed */
var render = true;

/* Revert interface back to target position in history */
function revert(target) {
  if(target < revertStack.length) {
    var pastEls = $("#past").children();
    pastEls.each(function(i){
      if(i < pastEls.size() - 1) {
            $(this).animate({
              width: $(pastEls[i+1]).css('width'),
              height:$(pastEls[i+1]).css('height'),
              left: $(pastEls[i+1]).css('left'),
              top: $(pastEls[i+1]).css('top')
            },
            500);
      }
    });
    x = pastEls.eq(pastEls.size() - 1);
    x.animate({left: x.attr('oriLeft'), top: x.attr('oriTop')}, 500);
    revertStack.pop()(function(){x.remove(); revert(target)});
  }
}

/* jQuery function defining all operations that need to be performed to initialize the application. This code is
 * executed as soon as the XHTML file has been parsed.
 */
$(function() {
  /* Instantiate the work-item and employee templates */
  $('#employee-list').html(TrimPath.processDOMTemplate("employee-template", data));
  $('#work-list').html(TrimPath.processDOMTemplate("work-item-template", data));

  /* Make form elements "editable" */
  $("input[@type=text]").focus(function(){
    $(this).after('<img src="img/icons/save.png" class="inline-buttons" style="display: none" />')
  });
  $("input[@type=text]").focus(function(){
      $(this).css({background: '#ffffdd'});
    $(this).next().show();
  });
  $("input[@type=text]").blur(function(){
    $($(this).attr('id') + "-buttons").remove();
      $(this).css({background: "transparent"});
    $(this).next().hide();
  });

  /* Initialize the start-icon with click-response to revert to position 0 */
  $('#work-center-landmark').click(function(){revert(0)});

  /* Set startColor attribute of previews to use it for the hover effect */
  $('.preview').each(function(){$(this).attr({startColor: $(this).css('background-color')})});

  /* Add hover-effect (highlighting) to previews, except employee-previews and work-item previews for performance
   * reasons. Might be tweaked.*/
  $('.preview').not('.employee-preview').not('.work-item-preview').hover(
    function(){
      $(this).animate({backgroundColor: $(this).attr('altColor')}, '100');
    },
    function(){
      $(this).animate({backgroundColor: $(this).attr('startColor')}, '100');
    }
  );

  /* Add click-responses to all previews, except for sub-category views, as these have different behaviour */
  $('.preview').not('.sub-category preview').click(
    function(e){
      if($(this).is('.work-item-preview')) {
        var tOriLeft = $(this).children('.task-preview').offset().left - 8;
        var tOriTop = $(this).children('.task-preview').offset().top - 4;
        var stickyOriTop = $(this).children('.task-preview').offset().top;
        var stickyOriLeft = $(this).children('.task-preview').offset().left;
        var task = $(this).children('.task-preview').attr('id');
        var todo = parseInt($('#tasks-todo-count').html());
        $(this).children('.task-preview').
          clone().
          attr({id: 'sticky', task: $(this).children('.task-preview').
          attr('id'), oriTop: stickyOriTop, oriLeft: stickyOriLeft}).
          css({zIndex: 4000, top: tOriTop, left: tOriLeft}).appendTo('#container').
          animate({top: 100, left: 2, width: 176, fontSize: 12, height: 130}).
          children('.task-done').
          click(function(){
            $('#tasks-todo-count').html(todo - 1);
            $('#' + task).css({textDecoration: "line-through", opacity: 0.5, background: "#aaaaaa"});
            revert(1)
          });
      }

      var pos = revertStack.length + 1;
      var el = $(this).next('.view').attr('id');
      var oriLeft = $(this).offset().left - 8;
      var oriTop = $(this).offset().top - 4;
      if($(this).is('.work-item-preview')) {
```

```
        revertStack.push(function(callback){
          $('#' + el).animate({height: 0, top: oriTop + 35}, 500, null, callback).
          children('.preview').not('.employee-preview').animate({height: 0});
          $('#sticky').animate({top: $('#sticky').
          attr('oriTop'), left: $('#sticky').attr('oriLeft'), width: 726, height: 24, fontSize: 15}, null, function(){
            $('#sticky').remove()
          });
        });
      }
      else {
        revertStack.push(function(callback){
          $('#' + el).animate({height: 0, top: oriTop + 35}, 500, null, callback).children('.preview').
          not('.employee-preview').animate({height: 0});
        });
      }
      $(this).children('.landmark-wrap').children('img').clone().attr('id', 'past-' + $(this).attr('id')).
      css({position: 'absolute', top: oriTop, left: oriLeft, zIndex: 4400}).
      attr({oriLeft: oriLeft, oriTop: oriTop}).
      appendTo('#past').click(function(){revert(pos)}).animate({left: 136, top: 5}).prev('.landmark').
      animate({left: 87, top: 12, width: 40, height: 56}).prev('.landmark').
      animate({left: 52, top: 17, width: 32, height: 45}).prev('.landmark').
      animate({left: 29, top: 28, width: 20, height: 28});
      $(this).next('.view').
      css({top: oriTop + 35}).
      animate({height: 598, top: 0}, 500).children('.preview').not('.employee-preview').animate({height: 70}, 500);
    }
  )

  /* Add accordion-like click response to sub-category previews */
  $('.sub-category.preview').click(
    function(){
      target = $(this).next().animate({height: 220}, 500);
      $(this).siblings('.view').not(target).animate({height: 0}, 500);
    }
  )
  $('.employee-preview').mousemove(
    function(e) {
      /* Only render on 50% of all mousemouve events for performance reasons */
      if(render) {
        /* Calculate cursor position relative to the top-border of the current entry */
        var delta = e.pageY - ($(this).offset().top);

        /* Set current entry to maximum zoom level */
        $(this).css({height: 70, fontSize: 40});

        /* Adjust previous 4 entries */
        $(this).prev().prev().css({height: 70 - delta/2.3333, fontSize: 40 - delta/7}).prev().prev().css(
          {height: 40 - delta/7, fontSize: 30 - delta/7}).prev().prev().css(
          {height: 30 - delta/7, fontSize: 20 - delta/7}).prev().prev().css(
          {height: 20 - delta/7, fontSize: 10}
        )

        /* Adjust next 4 entries */
        $(this).next().next().css({height: 40 + delta/2.3333, fontSize: 30 + delta/7}).next().next().
        css({height: 30 + delta/7, fontSize: 20 + delta/7}).next().next().
        css({height: 20 + delta/7, fontSize: 10 + delta/7}).next().next().css({height: 10 + delta/7, fontSize: 10});

        /* Reset all other entries to minimal zoom level */
        $(this).siblings('.employee-preview').lt(parseInt($(this).attr('num')) - 4).css({height: 10, fontSize: 10})
        $(this).siblings('.employee-preview').gt(parseInt($(this).attr('num')) + 4).css({height: 10, fontSize: 10})

        /* Don't render on the next mousemouse */
        render = false;
      }
      else {
        /* Render on the next mousemouse */
        render = true;
      }
    }
  )

});
```