

# Information Representations, Context, and the Design Process

Gavin Doherty  
Department of Computer Science,  
Trinity College,  
Dublin, Ireland.  
Gavin.Doherty@cs.tcd.ie

## Abstract

The advent of ubiquitous, context aware computing has made the problem of choosing how to present information to the user significantly more complex. The designer must consider the level of detail and abstraction determined by the physical form factor, the integration of information from other external and communicated information sources (and representations), and the use of represented information in collaborative acts and communications. Furthermore, it is a design decision which must be made in the context of the wider development process. In this paper, we re-evaluate the processes and models with which we can choose information representations, and put forward a framework based on the Resources Model. We consider how this can be done while obtaining leverage from modeling and requirements analysis effort carried out for other parts of the system development.

## 1. Introduction

Information representations have an important influence on the usability of interactive systems, as they affect the way in which users perceive information, how they reason with it, and hence how they perform tasks and solve problems [9,17,22]. Such external representations of

information can be provided by a computer system (eg. a handheld device), by the environment, or (increasingly) both. Many factors affect the usefulness of a representation (see Figure 1); it is a complex design decision, hence we would like to understand the factors involved, and take a reasoned and methodical approach to evaluating the merits of different presentation designs.

Perceptual aspects of information displays have received attention in the past, in the setting of desktop applications, but the variety of environmental and other factors that can affect the appropriateness of a given representation is particularly important when we consider ubiquitous and mobile systems where the device providing this representation can move through a variety of physical and informational contexts.

Hence we have a greater challenge in addressing this particular design problem, but also in terms of carrying out this activity within a more complex system development. As there is a growing awareness of the need to consider human computer interaction within the Software Engineering community, it is important to try to take advantage of (and gain additional leverage from) modeling and analysis activities which are carried out in the rest of the development process. For the moment, we consider relatively focused and application specific displays – ambient displays may have a different set of design criteria.

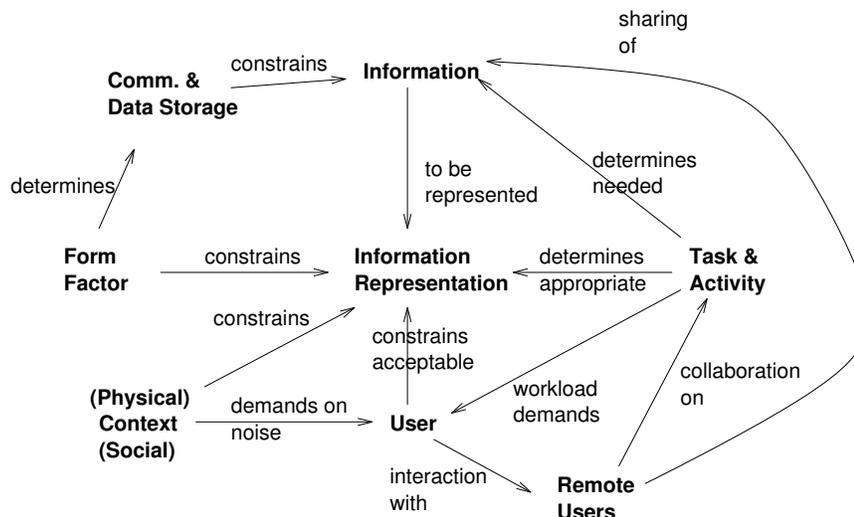


Figure 1 - Factors influencing appropriate representation

## 2. Requirements for Effective Representations

The question of what determines an *effective* representation is more complex than it first appears; in an effort to identify the different factors at play, we have identified the following general requirements for display designs:

1. *The display should make available (and give prominence to) the information most relevant to the task and current activities of the user.*
2. *The display should support the most common operations (eg. comparison, correlation, synthesis) on the information, which would be involved in carrying out the task.*
3. *The information sources provided or mediated through the system or device should complement information sources in the environment (context) of the user.*
4. *The display should use representations consistent with the representations of information outside the device (ie. in the environment).*

Interactions obviously consist of more than the user perceiving information, so we must tie this work into a framework for viewing the entire interaction, including actions performed by the user and system responses, and changes in context which may result as a consequence of these actions. We examine the resources model of Wright Fields & Harrison [21] as a basis for providing this framework for reasoning about contextualised interaction.

## 3. Analysing Representations

There is a diverse body of work dealing with the topic of representational analysis. Much of the work has a psychological emphasis, looking at the influence of representation on perception and problem solving. For example, Zhang [23] focuses purely on representational analysis of relational information (eg. scatterplots, barcharts, pie charts and so on), using a formalisation of scales of measurement from Stevens [18]:

- Nominal scales, distinguished by simple equality
- Ordinal scales add magnitude comparison
- Interval scales support equality of intervals
- Ratio scales have an “absolute zero” and hence support ratio operations.

Different forms of relational display embody (possibly several) different scale types and hence provide varying degrees of perceptual support for certain task-related comparisons. However this relates only to one type of interface element and (given the psychological background) stops short of proposing a method for even this restricted subset. A more concrete approach with a

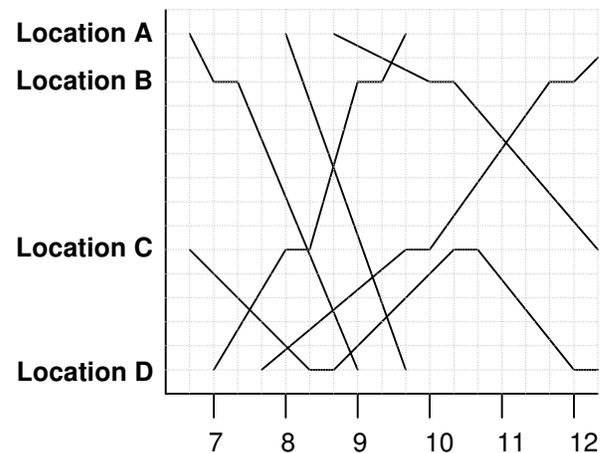
psychological background is the display structuring guide of May et al. [14], which draws on the ICS cognitive model [2] and the gestalt principles of organisation.

Purchase [16] puts forward some interesting work on graph drawing visualisations, however this is restricted to a particular form of information display (graph based), and graph based tasks. The work provides concrete results on the influence of different graph drawing approaches (minimise bends, minimise edge crossings etc.) on generic graph-display operations such as:

- How long is the shortest path between two given nodes?
- What is the minimum number of nodes that must be removed in order to disconnect two given nodes such that there is no path between them?

An informally derived set of guidelines and a large number of examples are gathered in Tufte [20]; simple guidelines concern maximising the use of the available space, minimising “non-data-ink” and so on. Interestingly, many of the examples illustrate the capability of well designed representations to facilitate complex computations and comparisons as relatively simple perceptual operations, consistent with the Distributed Cognition viewpoint.

An example of an interesting representation which enables a number of complex operations to be

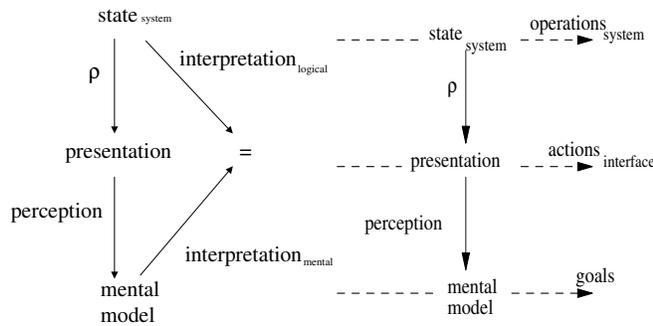


**Figure 2 - Graphical railway timetable**

performed in a simple perceptual fashion is given in Figure 2 (after an illustration in [20]). Here, we have a representation of a train timetable over two axes – location and time, both proportional (ratio) scales. Each line represents a train. As we have proportional time and distance, the speed of the trains is represented by their slope. The exact time and position where trains pass or overtake one another are shown by line crossings. Direct

trains can be seen as uninterrupted lines, and so on. Thus we can see that a well designed display can make a significant difference on the cognitive load placed on the user. Beyond this, the work is interesting as it gathers and attempts to explain a number of good and bad designs, but again does not constitute a general method or approach to analysis.

A formalization of the mapping between application, task, and perceptual operations is given in [7] (Figure 3, left), along with a real-time extension of the model [5] (Figure 3, right). Checking whether these relationships between the logical, perceptual and task levels hold can uncover hidden assumptions about the operational constraints on the system and the perceptual capabilities of the user. In terms of the overall design process this form of analysis is tied closely to task analysis.



**Figure 3 – Models for validating perceptual-logical operator mapping**

In the area of visualisation there is a body of highly task specific work, with some evaluations for particular domains; however there is little consideration of the processes involved in general interactive systems development, or indeed the match between effective displays and user tasks and activities.

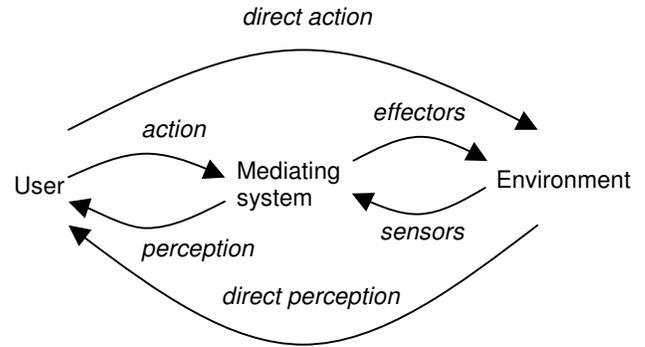
Thus we have a number of different forms of low level analysis, very often specialised for a particular form of display, with very little to guide us on how to drive the analysis, and how to integrate it into the wider development process.

A wider perspective on the role of information representations on cognition and collaboration is provided by the Distributed Cognition view, and particularly the work of Hutchins [9,10]. Here the entire work system is taken as the unit of analysis, including the human and system actors, technological artifacts, and external information representations. Analysing the role of information representations (particularly external representations) in achieving the goals of the entire system is one of the main facets of the approach. While in a sense this sounds familiar to a software engineering approach, the analysis is entirely post-hoc, and ethnographic or

observational in nature. While it gives us a way of thinking about the role of representations, it does not constitute a design method.

#### 4. Framework

As a first step, we establish a setting within which a device is used. At a low level of granularity, we can consider the information display as a set of *resources for action*. What do we mean by an *action*? There is a distinction between actions that affect the environment



**Figure 4 - Direct and indirect interaction**

directly and actions concerning, or mediated by the system (possibly affecting environment via effectors). Likewise perception may be of information sensed by the system or directly from the environment (see figure 4).

In the list of requirements above, we have a need to examine the way information within the environment (or context) can affect the usefulness of a display (Requirement 3). Beyond this “static” view, *context changes* affect the usefulness of information, since we may have a different set of information sources in the new context. An example of a design issue related to context changes is the notion of *plasticity* [19], an aspect of which would be the degree of small changes in the context which can be tolerated before a major change (e.g. mode change or change of modality) must be made by the application in order to account for the different context.

A further complication is that in ubiquitous computing applications, the device itself will likely be responding to the changes in the context. This means the application will be maintaining some form of context model, usually based on abstractions over sensors in the environment. We would expect this context model to be a subset of the context considered for the purposes of interaction design, but once a decision has been made concerning such context awareness, we must take the new “device” context into account if it changes the information available to the user, the way information is presented, the

actions available to the user, or the behaviour of the application..

#### 4.1 The user in context – the resources approach

The resources model [21] is an interaction model whose aim is to support Distributed Cognition style analysis at an early stage of design, rather than as a means of understanding existing systems or prototypes. The model itself is simple – it views interaction as involving a number of information resources which can be characterised in terms of a number of different categories, depending on the role the information may play in interaction. Different *interaction strategies* on the part of the user (eg. goal-matching, plan following) will exploit different types of information, and place a different emphasis on such information. For those interested in developing mobile and context aware systems, the model is attractive since the information resources considered may be located in the environment, or provided by the system. Information in the environment can take the form of static, unchanging information, it can take the form of continuously changing streams reflecting the state of some entity, or it can take the form of events. The categories used to characterise information resources are as follows:

- **Plan:** sequence of actions, events or states that should be carried out
- **Goal:** required state of the world.
- **State:** collection of relevant values of objects that feature in an interaction
- **Action-effect:** relation between an action or event and its effect on the interaction.
- **History:** actions, events or states already achieved in the interaction.
- **Possibilities:** set of possible next actions of the user.

As an example, we could see a trail based application such as a museum tour guide as providing an external representation of plan information, whereas displaying a print icon when viewing a document is making available possibility information. The model also gives us a new representation requirement:

5. *The information displayed, and the manner in which it is displayed, should reflect the role it plays in the interaction strategy of the user.*

We can use the resources model to drive and provide a context for representational analysis (see section 5), but in order to do so, we must be able to answer questions concerning user tasks and goals, application behaviour, and the different contexts the application is concerned with. In the following sections we consider the models which may be available within the interactive system development to answer these questions.

## 4.2 User Models

We can see traditional task models which may be produced as part of the system development process as forming part of our User Model, as they are abstractions representing behaviours of the user which lead to the achievement of user goals. To deal with context issues, we need a richer model, as the user will be responding to context changes by changing or altering the scope of their tasks, and altering the strategies used for performing those tasks (and consequently via resources model the exploited information resources). An example would be a user who changes their interaction strategy when in a different context.

User models such as the Model Human Information Processor [6], or EPIC [11] may be of some use in assessing the complexity and “cost” of certain cognitive and perceptual operations. Models such as ICS [2] can also shed light on issues concerning multi-modal interactions. However the specialist skills needed to develop and interpret the models means that they are unlikely to be used extensively in other phases of the system development.

## 4.3 Application Models

There has been much work in the past on modeling the behaviour of applications such that the models can be used as the basis for reasoning about interactive applications [1,8,13]. As we do not necessarily wish to construct complete models, we should identify which aspects/abstractions are important for our purposes.

- that part of the application responsible for generating presentation (eg. changing interface modality)
- that part of the application responsible for sensing and maintaining context model
- that part of the application responsible for reacting to context.

The most attractive option is obviously to reuse existing Software Engineering models wherever possible to answer these questions. This might mean however that analysis which relies on such models is harder to carry out in the earliest stages of system development. Another open question is to what extent standard software engineering models (eg. expressed in a notation such as UML [3]) can answer these questions in sufficient detail.

## 4.4 Context Models

In terms of the requirements above, we see three main effects of changes in context.

- May result in change in task (requirements 1,2)

- May result in change in strategy (requirement 5)
- May result in change in externally available information (requirements 3,4)

A useful resource then when analysing representational issues would be a context model which allows us to reason about these issues. Such a model may be used for other parts of the development process. A number of different modeling approaches have been taken to the problem of modeling context, often with different goals.

- State Transition approaches can capture characteristics of discrete changes in context, together with simple specifications of the context aware systems, allowing some dynamic behaviour to be explored.
- Hybrid modeling approaches may allow modeling of certain continuous processes in the environment, as well as the concurrent operation of a system sensing and reacting to these processes.
- Some models focus on the ontological aspects of context (semantic networks, object models etc.) (Object models may also have a state transition definition of behaviour associated with them).
- Some technology driven models exist in order to provide a toolkit of sensor abstractions for building context aware applications.

A difficulty is that no single type of model above will capture all the relevant aspects of context. Many existing “semantic” and technology-driven models take a strictly hierarchical view, whereas other would argue that significant aspects of context are non-hierarchical in nature. To address the more “static” elements of context, and particularly the role elements of the context may play as information resources, we must have a model which incorporates the ontological or semantic aspects of context. Concurrently, in order to address the effect changing context has on the most appropriate information resources and representations, we must also be able to account for the dynamic aspects of context. This motivates an approach which exploits both ontological and dynamic context models.

As mentioned above, modern systems may maintain an internal model of the context, based on sensed information, which helps determine appropriate behaviours for the system. This raises an interesting modeling issue regarding models of systems which react to context (perhaps as dependencies between environmental and system aspects of context). Behavioural properties may require detailed modeling approaches [13], whereas we simply want to factor context into resources model. However, one of the lessons of HCI in recent years has been the need to understand the user’s context for all applications, as an integral part of

the software development process, as it has such an important effect on the activities of the user and how best to support them. As such, we would maintain that there is a strong need within the software engineering of context aware applications to model context beyond that sensed by the application.

## 5. Representational Analysis in the Development Process

In this section we consider how analysis of representations may be driven and informed by activities and requirements gathering which form part of the overall development process.

### 5.1 What information should be presented?

The decision on what information should be presented has many dependencies on other aspects of the development process, concerning the availability (and possibly timeliness and quality) of information, decisions on application functionality, and resources (bandwidth, storage, networked information sources). Working within these constraints, the first step is identifying information necessary to carry out the user’s tasks. A common technique to aid in making such development decisions would be Hierarchical Task Analysis. Using the output of the task analysis (again, attempting to re-use artifacts produced by other parts of the design process) in order to help decide what information should be provided will identify required information in the form of decision variables.

Particular use cases which have been documented as part of the requirements analysis can be used to further direct this analysis. For example, let us consider a “courier” scenario where a mobile user is having difficulty matching directions on a mobile device to the appropriate route in the real world, and makes a query to a central controller with access to a detailed display. While one considers the task facing the pair: the mobile user must include in the query information regarding what they can perceive in the environment; this must be mapped to an orientation by the controller, who can then relate this to the detailed display in order to generate an appropriate response for the mobile user. We can represent this aspect of the system with a use-case:

Use Case: Request guidance	
Scenario: Courier is having difficulty interpreting directions given by device.	
Environment	Courier is on street, and can see the possible turns, and surroundings of current position.

Situation	Courier has been following directions given by device up to this point
Sequence of events	1. Courier requests guidance
	2. Dispatcher identifies mobile user, and locates them on display
	3. Courier describes surroundings and possible turns
	4. Dispatcher deduces couriers orientation and reference points
	5. Dispatcher communicates desired turn with respect to couriers reference points and orientation.

While this form of representation allows us to identify some of the required information and note its source (possibly in the environment) it does not provide a technique for how to do it.

However, information which should be provided can sometimes be less direct, such as encoding rules of a domain when carrying out task [22]. For example, in an application involving control of a physical process (eg. a chemical plant), certain operations will not be possible due to the current state of the system (eg. if the action would violate a safety constraint). A well designed display might not suggest affordances which correspond to these unwise operations. The Resources Model can be used as a means to identify such indirect information requirements. Hence by considering the interaction strategy the user is expected to take, we can identify the most important types of information resource. In the chemical plant example mentioned above, by considering *possibility* and *action-effect* resources, we can identify the need to either disallow or distinguish such illegal operations (depending on automation decisions, which is beyond the scope of this part of the analysis).

For both of the above forms of analysis, we have the added complication of changing resources in the environment due to context changes. Information may be present in the context, and so the information presented should complement (rather than always duplicate) this information.

It is appropriate at this point to ask what form of early design representation might be appropriate for supporting such analysis, in addition to HTA and use-case information. A development activity which may produce useful design artifacts (for our purposes) is Allocation of Function analysis. Allocation of function is concerned with deciding which functions to automate or partially automate. Such analysis has a new scope and saliency in the ubiquitous and context aware arena, since much of the context awareness and “intelligent” behaviour of such

systems are effectively allocating “traditionally” human functions to the system.

For functions which may be partially automated one can produce an IDA-S template [4]. The template has four components – Information, Decision, Action and Supervision, allowing a function to be decomposed into a number of elements, which can themselves be characterised (information collection, decision selection etc.). In practice templates may be generated for a number of design alternatives. For example, returning to the “courier” use case above, we have the IDA-S template below.

<i>Choose a direction using guidance</i>		
<i>information</i>	<i>collection</i>	GPS location, what the user sees (used to produce description of landmarks), detailed map
	<i>integration</i>	Control centre compares description with map of location
	<i>Initiate response</i>	User perceives ambiguity in interpreting directions, seeks guidance.
<i>decision</i>	<i>proposal</i>	Description of route to take relative to mapped features and description
	<i>evaluate/modify</i>	Evaluates instructions relative to surroundings
	<i>select</i>	Chooses interpretation of instructions
<i>action</i>		Takes direction
<i>supervision</i>	<i>Identify completion</i>	System indicates user has reached desired location (user confirms).

The *usage* of information is quite explicit in this template, and as such is a very useful resource to drive representational analysis. As an example, consider steps that are concerned with *information integration*: this is a specific type of perceptual and cognitive task, for which certain types of analysis (consistency of the representations, amount of searching, complexity of comparisons) are appropriate. Also, the fact that different design alternatives may be explored allows a

representational analysis to provide some input to such decisions by highlighting eg. highly dense information displays supporting a number of simultaneous decisions, or alternatively the lack information for certain decisions due to the automation or partial automation of other functions. A perceived weakness of strongly task-focused approaches is that important factors such as situational awareness may be missed. The IDA-S template may be of use in picking up on such issues, as it may highlight the integration of strongly task focused information with “situation” information in decision making.

From the perspective of wider development, it is desirable to reuse the results of modeling carried out in order to make function allocation decisions. For this form of analysis, the Resources Model and postulated interaction strategy for a given context form the User Model. The context model constitutes the information resources in the context, plus relevant portions of the application model (concerning sensing and reacting to context), which again are of relevance to the wider development process.

## 5.2 What representation is most appropriate?

As discussed in section 3 above, from the literature [22,16,12,17] we can see that it is possible to select representations that directly support a given set of tasks. Doherty, Campos and Harrison [7] put forward an approach for doing this based on models of the presentation and task operations. We note that detailed application models are not required for this approach, which has a focus on perceptual issues in its User Model.

An additional dimension we can consider is whether the representation matches the interaction strategy the user is expected to take in a given context. For example a given information representation might make it easier to perceive possibilities rather than plans, or planning could be used within the device to filter presented possibilities, without an explicit representation. This seems to add a relatively subtle effect to the decision on appropriate representation. Furthermore, the appropriateness of entire modalities may be affected by changes in context.

The other requirement – choosing a representation that is compatible with representations present in the context, seems fundamental and problematic. A challenging research question is how can we take a systematic approach to the identification of “compatible” representations. Beyond perceptual compatibility, there are issues concerning the timeliness of different information sources. For example an application where information is intermittently sensed or downloaded from a remote source may display a set of values for different parts of the state of the system which do not refer

to the same point in time, and may apparently violate system constraints. Generally, compatible representations can be seen as reducing the gulf of evaluation - allowing the user to readily evaluate the presentation in terms of his goals [15].

## 6. Conclusions

Choosing what information to represent and how to represent it is a difficult and complex issue, particularly as some design decisions may be tied up with the functionality of the application and the structure of the interaction with the user. For this reason, it is desirable to deal with information representation within a framework which can accommodate reasoning about the overall interaction and application design. For pragmatic reasons, and also to help achieve a more integrated development, we also need to reuse design artifacts from other design activities, including those with a software engineering focus.

In this paper, we have reconsidered the design of information representations, using the resources model to provide the interaction framework. We have considered the types of model potentially available to answer questions about user behaviour, application behaviour, and context. We would emphasise that context is not an isolated issue - we must look at the entire interaction to see how context information can be used as a resource in the interaction. While models of user behaviour are mostly used for those designing the interactive parts of the system, application and context models may well be available from other parts of the development process. For the most problematic situations we may need a dynamic behavioural model of the application sensing and reacting to context in order to explore the dependencies between such a system and the user’s own reactions to changes in context. In terms of what we require from a context model in order to make such choices, we have the following basic requirements:

- What information is available within the context.
- How the relevant information is represented in the context.

We have looked at how other design artifacts (such as use case descriptions and IDA-S templates) can be used to help drive the choice of information to represent, and also certain forms of representational analysis.

By considering resources and context together, we can help make an informed choice on what information to present (which is tied up with other aspects of interaction design, and requires answers to questions concerning user behaviour, application behaviour and context) and how to present it, which we can see as a relatively independent design issue, once sufficient

information is available to drive the representational analysis.

## 6. References

- [1] G.D. Abowd, Formal Aspects of Human Computer Interaction, D.Phil. Thesis, University of Oxford Computing Laboratory, 1991.
- [2] P.J. Barnard. Interacting Cognitive Subsystems: modelling working memory phenomena within a multi-processor architecture. In: A. Miyake & P. Shah, (Eds) *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, pp. 298-339, Cambridge University Press, 1999.
- [3] G. Booch, I. Jacobson and J. Rumbaugh, The unified modeling language user guide, Reading, Mass.:Addison-Wesley, 1999.
- [4] A. Dearden, IDA-S: A Conceptual Framework for Partial Automation, in People and Computers XV - Proceedings of HCI-IHM 2001, pp.213-228, Springer, 2001.
- [5] J. Campos and G. Doherty, "Reasoning about Dynamic Information Displays", Proceedings of Design Specification and Verification of Interactive Systems, Lecture Notes in Computer Science volume 2844, Springer, 2003.
- [6] S. Card, T. Moran and A. Newell, The psychology of Human Computer Interaction, Lawrence Erlbaum Associates, 1983.
- [7] G. Doherty, J.C. Campos and M.D. Harrison, "Representational Reasoning and Verification", *Formal Aspects of Computing*, 12(4), pp.260-277, 2000.
- [8] G. Doherty, A Pragmatic Approach to the Formal Specification of Interactive Systems, D.Phil. Thesis, University of York, 1998.
- [9] E. Hutchins, *Cognition in the Wild*, MIT Press, 1995.
- [10] E. Hutchins, How a Cockpit Remembers its Speeds, *Cognitive Science*, 19:265-288, 1995.
- [11] D. Kieras and D. Meyer, An overview of the epic architecture for cognition and performance with application to human-computer interaction, *Human-Computer Interaction* 12(4), 1997.
- [12] J.H. Larkin and H.A. Simon, Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11, pp. 65-99, 1987.
- [13] K. Loer, Model-based Automated Analysis for Dependable Interactive Systems, D.Phil. Thesis, University of York, 2003.
- [14] J. May, S. Scott and P.J. Barnard, *Structuring Displays: A Psychological Guide*. Eurographics Tutorial Notes Series EACG: Geneva, 1995. Appeared as Amodeus report B04.
- [15] D. Norman, *The psychology of everyday things*, Basic Books, New York, 1988.
- [16] H.C. Purchase, Effective information visualisation: a study of graph drawing aesthetics and algorithms, *Interacting with Computers*, 13, pp.147-162, 2000.
- [17] M. Scaife and Y. Rogers, External Cognition: how do graphical representations work? *International Journal of Human Computer Studies*, 45:185-213, 1996.
- [18] S.S. Stevens, On the theory of scales of measurement, *Science* 103:677-680, 1946.
- [19] D. Thevenin, J. Coutaz Plasticity of User Interfaces: Framework and Research Agenda, Proceedings of Interact '99, vol. 1, Edinburgh: IFIP, IOS Press, pp. 110-117.", 1999.
- [20] E.R. Tufte, *The visual display of quantitative information*, Graphics Press, 1983.
- [21] P.C. Wright, R. Fields and M.D. Harrison, Analysing Human Computer Interaction as Distributed Cognition: The Resources Model, *Human-Computer Interaction*, 15(1), pp.1-42, 2000.
- [22] J. Zhang and D.A. Norman: Representations in distributed cognitive tasks. *Cognitive Science*, 18:87-122, 1994.
- [23] J. Zhang, A representational analysis of relational information displays, *International Journal of Human Computer Studies*, 45, 1996.