The D.EU.PS Model: A Tool for Studying Effects of Information Systems Design

Emma Eliason & Pär J. Ågerfalk

Dept. of Informatics (ESI)
Örebro University
SE-701 82 Örebro
Sweden
E-mail: eaen@esi.oru.se, pak@esi.oru.se

Abstract
The aim of information systems design is to change or support changes in the way people act. An information system creates an action space by enabling some actions and making other actions impossible to perform. A designer cannot escape from the responsibility that follows with such a design. Still, it is impossible to predict exactly the actual effects of a design – the action space as perceived by actual users. The paper presents a model, referred to as the D.EU.PS. model (pronounced 'dupes'), which is suggested as a tool for analysing effects of information systems design by focusing on user perceptions of an information system.

Keywords: Design effects, Intention, User perception, Usage, Information systems design, Functionality

1 Introduction

When performing action, including inaction, people intervene in the world. Such interventions bring about changes, both physical changes (such as the relocation of physical objects) and changes in the actor's social world (such as the creation of commitments). This is to say that actions have consequences, which arise triggered by the action (Weber 1978). It is possible to distinguish consequences that are direct results of an action from the possible effects that will eventually arise. Effects are something that is beyond an actor's control while control is maintained with respect to the result (Goldkuhl & Ågerfalk 2002). Let us consider an example. A person posting a note on a billboard controls the content of the note, its physical appearance and its position on the billboard. The note displayed on the billboard is a *result* of the person's action. On the other hand, the person posting the note does not directly control possible *effects* arising when other people read the note and act upon its message. Certainly, there might be (and probably are) intentions behind the posting and the person wants to achieve certain effects, for example, in all friendliness inviting colleagues to a party with the hope that they will show up and have a good time together. Such intended effects are not necessarily the same as those actual effects eventually arising. People may interpret the message differently than intended, and may do completely different things than the posting person intended. We can, for example, imagine that a colleague who dislikes the person throwing a party regards it only as 'yet another showing off his beautiful home', with the effect that the dislike increases and that she is definitely not going to that party, – 'how could he even think that I, or anyone else for that matter, would come to his stupid party?' This, in turn may lead to this second person's decision to throw her own party, just to make people not to go to the first. An effect certainly not intended by the first person, and also an effect that should not have arisen should the original invitation not have been made.

This dichotomy between results and effects may be useful when analysing the design of an information system (IS) and its effects on users and their behaviour. Designers are in control of their design until the IS is put to use. The IS is a result of a design process and effects will emerge during IS use – effects outside the designer's control, for which the designer is, at

least partially responsible. A parallel can be drawn to mass communication theory (e.g., McQuail 2000) which distinguishes many types of effects of media. These are seen as consequences, or outcomes, from working with, or being exposed to, mass media, which may or may not be intended. There can be planned as well as unplanned, short- or long-term effects.

The need for a dialog about the responsibility that follows with a design has been emphasized in the design literature (Stolterman & Nelson 2000; Shneiderman 1998; Löwgren & Stolterman 1998). The reasoning is based on the view that we design new technology that will eventually change society, and that it is impossible to exactly predict the actual consequences of a design. Such consequences include providing a better life for various stakeholders and also consequences of unintended effects (perhaps making life worse for some). A designer should feel responsible for their design because an artefact is the result of purposeful actions by a designer. On the other hand, not every property of a design is the result of conscious design decisions, and designs may lead to unintended effects.

The aim of information systems design is to change or support changes in the way people act. An IS creates an action space by making some actions possible to perform and making other actions impossible, and there is no possibility for a designer to escape from the responsibility that follows design. Stolterman and Nelson (2000) suggest that the majority of designers do not feel responsible for the consequences of their design, that designers consider themselves as agents for the clients doing only what they are told to do. But how do you go about to study effects of information systems design?

There are theoretical models and frameworks to support such enquiry. These can, somewhat incisively, be broadly categorized as those that are too abstract or general to describe specific properties and functions of an IS, and those that are too restricted in their view of IS functionality. In the first category we find general frameworks elaborating on IS use and IS success such as 'technological frames' (Orlikowski & Gash 1994), 'task-technology fit' (e.g., Goodhue & Thompson 1995), 'cognitive absorption' (Agarwal & Karahanna 2000), and the classical IS success taxonomy of DeLone and McLean (1992). These models do not seem to offer practical support for assessing specific IS properties, but may serve as important general frameworks for understanding the relationship between technological, organizational and individual aspects of IS use. The most prominent example of the opposite category is probably the International Organization for Standardization's model of usability (ISO 9241-11 1998), which may serve as an important tool to understand and measure the effectiveness, efficiency and subjective satisfaction of information systems (Bevan & Macleod 1994; Maguire 2001). Even though important qualities, effectiveness, efficiency and satisfaction alone are too restrictive a notion to bring an understanding of the interplay between what designers intend and what users perceive, and it does not stress that understanding of an IS evolves over time (Ågerfalk & Eliason 2003).

In this paper we elaborate on a model referred to as the D.EU.PS. model (pronounced 'dupes'), first introduced in a slightly different form by Ågerfalk *et al.* (2002). This model builds on the underlying ideas of the models and frameworks mentioned above, and intends to fill the gap between the two categories. Specifically, we discuss how the model can be used as a tool for analysing effects of information systems design by focusing on user perceptions of an IS. The aim is to show how the D.EU.PS. model can be used to classify information systems functionality as a means to studying effects of IS design.

2 Research Approach

The research approach adopted in this work can be characterized as a reflective qualitative case study approach (Alvesson & Sköldberg 2000). Through theoretical reasoning, parts of the model were formulated and tried on existing information systems. Based on empirical findings, the model was extended to capture aspects not initially thought of. This way, the model extended from consisting of two categories with three classes (the two categories and their intersection) to its current status (five categories constituting eighteen classes), see below.

In this paper we discuss how the model, in its current form, was used in a case study at a Swedish manufacturing company. The study was conducted as an evaluation of the company's intranet (hereafter referred to as The Intranet). The Intranet is used for disseminating information and booking resources, such as rooms and overhead-projectors. The study involved expert-based evaluation of The Intranet as well as a number of interviews with representative users. The study had a larger scope than merely trying the D.EU.PS. model as it was part of a larger project concerning the development of a contextual IS evaluation approach (see Ågerfalk *et al.* 2002). This approach involves also, for example, the use of a set of principles targeted at the evaluation of information systems as systems for communicative action (see Ågerfalk 2003). In this paper we focus on the D.EU.PS. model *per se* and discuss its usefulness for studying effects of IS design. The main units of analysis (Patton 1990; Yin 1994) in this work have been The Intranet (its function and form, see below), user's perception of The Intranet and the D.EU.PS. model as a tool for understanding and analysis. The evaluation process is detailed in Section 4.

3 The D.EU.PS. Model in Theory

The D.EU.PS. model consists of eighteen classes derived from the combination of five high-level categories of IS functionality. To understand these classes and categories we take our point of departure in a model of action spaces proposed by Berglind (1990). When being in a specific situation, an actor may or may not act depending on how the situation is perceived. According to Berglind (1990), there are two basic considerations when a person is to decide whether to take action or not. Either you can do something, given the particular circumstances, or you cannot, and either you want to do something or you do not. If you can do what you want to do, then everything is fine. Sometimes, however, people desire things they cannot achieve, they want but cannot. You may also occasionally act because you feel that you have to, even though you may not really want to. On the other hand, sometimes you may choose not to act despite your will to do so because acting involves something you believe you should not do. These two dimensions (can and want) together span an action space, and their combinations form incentives for action (including inaction), as illustrated in Figure 1. The grey fields in the figure illustrate that you do not always know what you want and can.

	Can	**?**	**Cannot**
Want			
?			
Does Not Want			

Figure 1: Action space as a combination of can/cannot and want/do not want (Berglind 1990).

The action space model of Figure 1 gives rise to questions such as: What action possibilities do I have? What are my intentions? What restricts me from achieving my goals? What forces me to do things I do not want to do? These questions must be considered with an understanding of the IS as an important part of creating the action space (Goldkuhl & Ågerfalk 2002). When action is mediated by an IS (Kuutti 1996; 1999), two interrelated contextual aspects, the function and the form of the IS, must be considered as they together shape the action potential delivered by the IS (Ehn 1995).

If there is something that you want to do with the IS, there is a desired functionality. On the other hand, things that a user does not want constitute undesired functionality. The can-dimension in Figure 1 assumes that the functionality is perceived in use – that the user sees and understands the possible actions to perform. This dimension gives rise to questions such as: What action possibilities and what restrictions does the user experience when using the IS? In order to believe that you can use a function, you have to perceive that function. A function perceived as something you cannot do with the IS is something experienced as a restriction – an action potential not offered by the IS. On the other hand, perceiving something as undoable does not necessarily mean that you do not perceive the corresponding functionality as such. It may, for example, be the case that you cannot use a function since you do not know how to use it.

Ehn and Löwgren (Ehn 1995; Ehn & Löwgren 1997) emphasize that 'form' is often misleadingly used to mean an artefact's physical appearance alone, and stress that form is not a property of an IS but of the relation between system and user. This view accords with Gibson's (1977; 1979) concept of affordances. Gibson argues that in order to use, for example, a door, the actor has to perceive its properties, whether the door, for example, affords push or pull – is 'pushable' or 'pullable'. Affordances are properties in relation to actors. They exist in external objects, but they are not mere physical properties – they emerge only in relation to an actor. It is thereby important that properties signalling affordances are possible to perceive. The central issue is not whether certain affordances exist, but if they are possible to perceive in relation to oneself and one's intentions to act. A cup is graspable if its size matches that of a grabbing person's hand, it is neither the hand nor the cup that constitutes this affordance, but their combination (Bærentsen & Trettvik 2002).

Building on the notion of affordances, artefacts can be said to have intrinsic properties of form and function but it is only in use that these properties can be evaluated. The properties are expressed in use. In the design process, an IS may have been given properties to promote certain effects, for example, that the user will perceive it as effective and efficient in use. These properties are features of the IS (as they are results of the design process). On the other hand, whether a property gives rise to intended effects can only be evaluated with respect to the relation between system and user (effects emerge in use). It can result in an intended or unintended effect; the user can perceive a property differently than what was intended during design. As Norman (1999) points out, the art of the designer is to ensure that desired, relevant actions are readily perceivable.

It is also possible to live under the false impression that certain functionality exists when it in fact does not. A user can, for example, believe that she uses a function when she in fact does something else than what she thinks she does. In that case, from a designer point-of-view, the user utilizes functionality that she does not perceive. This seemingly controversial position can be understood based on Aronsson (1990) who distinguishes between subjective and objective action space. The subjective action space is what you believe you can do, and the

objective action space is what is really possible to do. In practice, the distinction is not so easy to make. The relationship between subjective and objective is characterized by movement. Each individual changes his or her objective action space through, for example, increased knowledge, and so may increase his or her subjective action space accordingly. Thus, it is the perception that controls which actions you believe you can do. There may exist action space that is not perceived and perceived action space that does not exist. Of course, there is also action space that is both perceived and existing.

If we apply this argumentation to information systems design, the objective corresponds to what is in the system and the subjective is what users perceive and designers intend. We may therefore distinguish the categories existing functionality (as intended by the designer) from that which users perceive to exist. It is important to recognize not only what functions the IS implements but also the actions that the user perceives as possible to perform. We may also distinguish desired functionality from functionality that eventually exists in the system. The categories desired and existing may not overlap completely. There may be actions that users want to perform but cannot because the desired functionality is not implemented. Furthermore, it is a well-known fact that many systems or parts thereof are not used to the extent anticipated during design (e.g., Davis 1989). It is therefore essential to additionally distinguish between what exists, perceived or not, and what is actually utilized. Davis (1989) also points out that it is important to not confuse desired functionality with satisfactory. Even if users believe an application to be useful (as in desired), they might still consider it too hard to use (as in unsatisfactory). On the other hand, people may perceive an undesired function as satisfactory: 'sure, it works fine, but I don't need it'. Utilization of existing functionality is thereby, in addition to desired functionality, influenced by perceived ease of use (as in satisfaction). Satisfaction thus relates to the like or dislike of the actual implementation of a function while desired functionality regards holistically the desirability of a function.

Altogether, this discussion leaves us with five high-level categories of IS functionality: Desired (D), Existing (E), Utilized (U), Perceived (P), and Satisfactory (S). Figure 2 shows how these categories relate to the action space model of Figure 1.

		Can		**?**	**Cannot**	
		E P	E ¬P		¬E P	¬E ¬P
Want	D	Situation 1 US, U¬S, ¬US, ¬U¬S	Situation 3 U, ¬U		Situation 5 ¬US, ¬U¬S	Situation 7 ¬U
?						
Does Not Want	¬D	Situation 2 US, U¬S, ¬US, ¬U¬S	Situation 4 U, ¬U		Situation 6 ¬US, ¬U¬S	Situation 8 ¬U

Figure 2: Action space as utilized and non-utilized functionality.

From Figure 2 we see that in Situations 1 and 2, if a user perceives (P) an existing function (E), that function can be utilized (U) even though it may not be (¬U), regardless of the user's will to use it (D) or not (¬D). Whether used or not, the function may be considered satisfactory or unsatisfactory (S or ¬S). In Situations 3 and 4 we can see that even if a user does not perceive (¬P) an existing function (E), that function can still be utilized (U or ¬U), the user, in that case, is not aware of using it and so it cannot be satisfactory (¬P → ¬S). Still, it may be a function that the user desires (D) or does not desire (¬D). All these situations (1–

4) involve the use or non-use of existing functionality – functionality that can be used, functionality that the designer put into the system and that users may or may not perceive. This is different from the remaining Situations 5–8, which all constitute situations in which the particular functionality cannot be used (¬U). If we look at situation 5 and 6, the user perceives (P) a function that does not exist (¬E). The user believes that it exists, desired or not, but obviously he or she does something else than what he or she believes. The subjective understanding of the user diverges from that of the designer. Nonetheless, since perceived, the function can be considered satisfactory (S) or unsatisfactory (¬S). Finally, Situations 7 and 8 constitute non-existing (¬E) functionality that the user does not perceive (¬P). This may be functionality the user desires (D), hence a missing user requirement, or a function the user does not want (¬D).

An obvious goal with a design is that users' perception of the system's functionality correspond to the intended (Norman 1988; Orlikowski & Gash 1994). That is, making the subjective understanding of the designer match with that of the user so that intersubjectivity is reached and maintained. Through analysing users' perceived action space, the designer has the possibility to identify and implement missing user requirements (desired non-existing functionality). Additionally, it may assist in making users aware of functionality that has so far been used unknowingly. This is where the D.EU.PS. model comes into play.

The D.EU.PS model is a comprehensive model of IS functionality in use and represents combinations of the five high-level categories of IS functionality: Desired, Existing, Utilized, Perceived, and Satisfactory. *Desired functionality* is functionality believed to enable a user to accomplish intended effects. *Existing functionality* is functionality which is implemented and accessible in the system. *Utilized functionality* is the subset of existing functionality that constitutes functionality which is actually being used (something that is utilized always exist). *Perceived functionality* is functionality that users believe to exist. Finally, *Satisfactory functionality* is the subset of perceived functionality which is considered satisfactory (something that is satisfactory is always perceived). The model (see Figure 3) provides a means to classify the functionality of a system into these five different but overlapping categories, constituting eighteen classes. Note that the respective sizes of the classes in the figure do not correspond to any quantitative empirical measure. They have been chosen simply to facilitate presentation. Note also that since utilized is regarded a subset of existing, and satisfactory a subset of perceived, we do not use all five letters in all class in the model; the classes can be uniquely identified anyway. This subset constitution of the model is also the reason for calling it D.EU.PS.

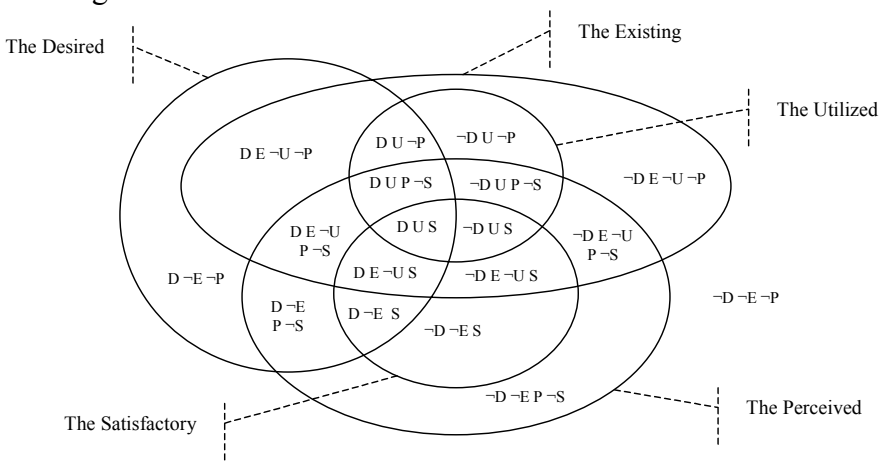

Figure 3: The D.EU.PS. model (adapted from Ågerfalk *et al.* 2002). Capital letters refer to the corresponding category and the symbol ¬ represents Boolean not; that is: D = Desired, ¬D = Undesired.

The eighteen classes of the D.EU.PS. model are described in Table 1 and Table 2. To facilitate reading, the description of the classes is divided into two groups: desired functionality (Table 1) and undesired functionality (Table 2). A few examples of these classes are discussed in Section 4, further examples are found in (Ågerfalk & Eliason 2003).

Table 1: Desired functionality in the D.EU.PS. model.

Class	**Description**
D U S	This is good! This is they way all functions should be like.
D U P¬S	This is good but could be better.
D U ¬P	Users perform this function even though they are not aware of its existence. Probably a result of unattended action, possibly related to learnability or education.
D E¬U S	Sure, but another way works even better.
D E¬U P¬S	Its so bad users cannot really use it even though they would like to.
D E¬U ¬P	Users do not perceive this function so they do not use it, but they would have if they understood it.
D ¬E S	Users want this function and believe that it exists and that it works fine, but obviously they do something else than what they think they do.
D ¬E P¬S	Users want this function and believe that it exists even though it could be better, but obviously they do something else than what they think they do.
D ¬E ¬P	This should really be there. A missing user requirement

Table 2: Undesired functionality in the D.EU.PS. model.

Class	**Description**
¬D U S	It works well even though users do not really want it.
¬D U P¬S	Not only forced to use it, it is unpleasant as well.
¬D U ¬P	This function is used even though users are not aware of its existence, and they do not want it. This may be a result of unattended action (e.g., trial and error) and should, in that case, be avoided. It may also be the case that the system does something automatically without the user's knowledge and approval.
¬D E¬U S	Sure it works fine but users do not need the functionality.
¬D E¬U P¬S	An undesired function that users do not need and regard as unsatisfactory.
¬D E¬U ¬P	This function is neither desired nor used, and not even recognized. It can probably be removed.
¬D ¬E S	Users believe that this undesired function exists and that it works fine, but it does not, and we better leave it that way. This is probably related to learnability or misleading instructions/education.
¬D ¬E P¬S	Users believe that this undesired function exists but that it could be better, but it does not exist at all, and we better leave it that way. This is probably related to learnability or misleading instructions/education.
¬D ¬E ¬P	An undesired function that does not exist in the system, everyone agrees to this, and there is no reason for implementing it either.

4 The D.EU.PS Model in Practice

We claim that the D.EU.PS. model captures important aspects to consider when studying user interaction with information systems. In this section the model is presented in terms of how it as an analytic tool can support an evaluator to direct attention to such key aspects. We also present how it can be used, and indeed has been used, as a tool to study effects of information systems design. To this end, we present examples from the case study introduced in Section 2. As described in Section 2, the case study was conducted on an intranet mainly used for dissemination of information and resource booking. The focus in the presentation is on booking of resources. This is a part of The Intranet where users (Bookers) have the possibility to book, for example, a room at another department in another city. Every resource has one or several Resource Owners who are responsible for the resource. You can for example contact a Resource Owner if you have any questions about a particular resource, for example, how far a particular room is from the coffee machine, which The Intranet information usually not tells.

The overall process that was used in the case study is described in Table 3 in terms of four steps. In the remainder of this section, these four steps are discussed in detail together with reflections on our part regarding the applicability of the concepts introduced so far in the paper, concepts that constitute the foundation for the D.EU.PS. model, as well as the model *per se*.

Table 3: The evaluation process adopted in the case study.

What to do?	**How?**
Step 1: Identify existing functionality; what you can do with the IS.	Documentation analysis, studying the IS and interviewing designers.
Step 2: Design an evaluation protocol.	The evaluator relates the identified (existing) functionality to the five categories.
Step 3: Identify phenomena (Evaluation).	Observe and interview users. Classify observations with respect to the D.EU.PS. classes.
Step 4: Analysis.	Relate observations to causes and effects. Problem relations, Strengths relations: Causes and effects.

4.1 Identifying Existing Functionality

The aim of this initial step, identifying existing functionality, was to identify the existing functionality of The Intranet. Trivial as it may seem, this aim raised questions such as: What characterizes existing functionality? Is it the designer's intended functionality that constitutes the existing functionality? If users use a function in a different way than what was intended, does the original function still exist? How do you determine the existing functionality? Can the objective action space of designers and users change as knowledge of how the IS is used increases?

When an evaluator seeks to understand an IS, the existing functionality corresponds to that functionality which the evaluator understands to be in the system – what the evaluator believes users can do with the system. That is, the evaluator's understanding of the system is what defines the existing functionality, and the evaluator thus has the authority to decide on what exists and not. Such understanding may be reached through eliciting designers' intentions and studying the IS and its documentation. The evaluator's authority is thus based on an interest in creating a shared understanding about the system and its use. The class of existing functionality typically extends over time as the evaluator learns more about the

system and its use. Perceived functionality, on the other hand, is what the user believes exist, the user's understanding. Both perceived and existing functionality may differ from what the designer intended; i.e., from the functionality the designer believes exist. Particular properties of an IS that are regarded as an existing function may be used in a completely different way than what was intended, and thus constituting a further perceived but non-existing function. As discussed in Section 3, the goal of a design should be to create a shared understanding of the IS so that intended, perceived and existing functionality coincide; that designers, users and evaluators agree on what the system can do. (Of course, sometimes a designer may also act as evaluator)

The distinction between existing and perceived functionality makes it possible to identify existing functionality initially without involving actual users. In The Intranet case, this identification was achieved by studying the purpose of the artefact and the functionality it delivered. As discussed above, the functionality is the benefit of the artefact, what you should be able to do in a specific context (Holmlid 2002). An IS implements certain functions. These functions together constitute the action potential of the IS – what one can possibly do with it. The action potential of the IS, in turn, spans an action space for the user to act within.

According to Löwgren & Stolterman (1998), functions tend to be expressed with two words: a verb followed by a noun, representing an action and an object acted upon, for example, 'register customer'. A function thus corresponds to an action that the IS supports. The functionality of an IS goes hand-in-hand with its form. Functions are made visible and actable through screen documents (Goldkuhl & Ågerfalk 2002) – screen documents are what give functions a form. A screen document can bee seen as multifunctional in the way that it can support the conduct of an action (informing), it can function as an action media in the execution of an action, and it can contain results of action for users to interpret (Cronholm & Goldkuhl 2002).

During evaluation of The Intranet we identified several functions through analysing system documentation, studying and experimenting with the IS, and interviewing people at the IT support department (representing the designers). These are described in Table 4 together with business roles of the actors performing and interpreting the actions. Since all Resource Owners can also make bookings, a Resource Owner can also perform actions described as performed by a Booker in Table 4.

Table 4: Example of supported actions and corresponding functionality found in The Intranet.

Action	**Performer**	**Directed Towards**	**System Function**
Search Resource	Booker	Booker	Search for and display the name of a resource, based on different criteria
Show Information about Booking	Booker	Booker	Display information about a particular booking (a particular resource at a particular time)
Request Resource (demand)	Booker	Resource Owner (and other Bookers)	Mark the resource as 'request pending' for the requested time and notify Resource Owner
Cancel Booking	Booker	Resource Owner (and other Bookers)	Mark the resource as 'available' and notify Resource Owner
Respond to Request (Acceptance or Denial)	Resource Owner	Booker (all Bookers, and specifically the requesting)	Mark the resource as 'booked' or 'available' depending on decision and notify requesting Booker
Recall Accepted Request	Resource Owner	Booker (all Bookers, and specifically the requesting)	Mark the resource as 'available' and notify requesting Booker

As an example, Figure 4 shows a screenshot from The Intranet: the screen document used to perform the 'Request Resource' action from Table 4.

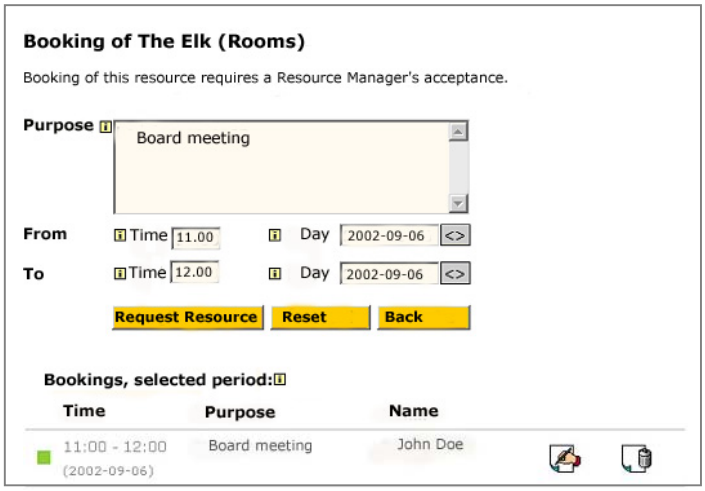

Figure 4: Example screenshot from The Intranet showing information about a particular booking.

Identifying existing functionality makes it possible to discover situations were users use a function differently than what was intended by the designer. Knowledge of how the IS is used leads to the designer changing his perception of the user's action space. This, in turn, may lead to the designer changing his subjective understanding of the IS as well, conforming to the user's view, or that the designer through, for example, education tries to change the user's perception of the function. This way, what is characterized as existing functionality and perceived functionality is dynamic and contingent upon users' and designers' experiences of the IS in use. Of course, such dynamics cannot be identified without involving users.

4.2 Designing Evaluation Protocol

When the existing functionality had been initially identified, a protocol was created in which it was possible to make notes about phenomena and relate them to the five different D.EU.PS. categories (see Table 5). This way, the protocols used during evaluation were explicitly based on the D.EU.PS. model. The protocol supported us in directing attention to key aspects and reminded us to ask questions such as: Is this functionality desired? Is it satisfactory? Is there anything you lack in the system? An example of the use of the protocol is given in Section 4.4.

Table 5: Structure of the protocols used to note observations during evaluation.

Function	Desired	Existing	Utilized	Perceived	Satisfactory
Name of function and/or corresponding supported action	Classification with comments	Classification with comments	Classification with comments	Classification with comments	Classification with comments
Description of Phenomenon					
Verbal description of observed phenomenon that justifies the classification and gives a hint about possible redesigns.					

4.3 Identifying Phenomena

The evaluation performed in the case study included both observations and interviews. In that way, the evaluation could be extended to include not only what was actually in the system,

but also to what users believed to be in it and what they thought it should provide. This way we could work closely with users to learn both the users' tasks and the ways they used the IS. Beyer and Holtzblatt (1998) describe such contextual enquiry as a setting that tries to resemble a 'new on the job' situation, between user and evaluator.

Berglind (1990) stresses that it is through talking with people about what they want, do not want, can, cannot, and what restricts them that you can get an understanding of their experience. In our case, the goal was to enable users to show and describe how their actions were performed, and to express reasons for action within their actual work context. That is, to express their knowledge in action (Schön 1983). Our approach, which was basically a combination of thinking aloud and user observations, was favoured because it enabled users, at least to some extent, to express tacit knowledge (Polanyi 1983). Together with expressed thoughts and observations, we had the opportunity to understand what was problematic or unproblematic (which could be observed from user actions) and to learn why and how it caused effects. Data collection was thus based primarily on positive and negative aspects encountered in the work practice. This way, different phenomena (problems and strengths observed in relation to The Intranet) and their causes and effects could be identified. Table 6 summarizes this third step in the evaluation: Identify Phenomena.

Table 6: Summary of Step 3: Identify Phenomena.

Objectives	What to Study?	How?	Documentation
Identify phenomena (problems and strengths) their causes and effects.	IS in use	Data collection as a combination of observations and interviews.	Protocol (see Table 7)
Understand which class a problem or strength is related to, which gives a basis for assessment and possibly change proposals.	IS in use	Classification of observed phenomena in accordance with the D.EU.PS. model.	Protocol (see Table 7)

Table 7 shows an example of this step as it was performed in the case study. The observed phenomenon was that a Resource Owner did not find the function Recall Accepted Request satisfactory because the function did not provide an opportunity to explain why he recalled accepted requests. The function, which existed in the system, was desired and used but not with satisfaction. In this particular case, the D.EU.PS model thus helped us to pinpoint how the function was used and why. The function as such was desired but it did not give the user the required action space: to give an explanation to the turned down booker.

Table 7: Example protocol of an observed phenomenon related to the function Recall Accepted Request in The Intranet.

Function	Desired	Existing	Utilized	Perceived	Satisfactory
Recall Accepted Request	D	E	U : The function is used, but only after a phone call to the booker has been made.	P	¬S: The IS function does not permit noting why a reservation has been recalled.
Description of Phenomenon					
'I believe that you should call and check with the person who has booked the resource […] Perhaps they have booked a lot of people for a course […] You want to know why your reservation is recalled not only that it is recalled.' (A Resource Owner, authors' translation from Swedish)					

4.4 Analysis of Causes and Effects

Throughout the evaluation, identified strengths and problems were analysed in terms of effects and causes, and classified according to a schema based on the classes of the D.EU.PS. model. This classification made it possible, for example, to discuss who would be affected by a problem and the importance of correcting it. By classifying phenomena as strengths and problems, and relating them to each other, a better understanding of the effects of the IS can be achieved. This is in line with the approach to Change Analysis suggested by Goldkuhl and Röstlinger (1993) and results in directed graphs of problems and strengths showing what problems cause other problems, and what strengths contribute to other strengths. This way, the most important problems to correct and the most significant benefits could be pinpointed. Table 8 summarizes this final step in the performed evaluation.

Table 8: Analysis of Causes and Effects.

Objective	What to study?	How?	Documentation
Identify relations between strengths and problems.	Documentation from the evaluation conducted so far.	Data analysis: Relation of observation to causes and effects. Relationships between problems and between strengths; causes and effects.	Observation tables (see Table 9). Problem Graphs and Strength Graphs.

The D.EU.PS. model made it possible to discuss the functionality of a system in terms of what is desired and what is not, what exists and what is missing, what is actually utilized and what is needless, what is believed to exist, and what can be used with satisfaction. This in turn made it possible to identify and discuss different users' perceptions of and attitudes towards the system, and how these perceptions and attitudes changed over time. The analysis was summarized as shown in Table 9.

Table 9: The Intranet function Recall Accepted Request (Quotations are authors' translation from Swedish).

User(s)	Class	Cause	Phenomenon	Effect
User 1 A Resource Owner	D U P ¬S	The possibility to give information about why the reservation is recalled is missing in the IS.	'I believe that you should call and check with the person who has booked the resource […] Perhaps they have booked a lot of people to a course […] You want to know why your reservation is recalled not only that it is recalled.'	The function is used, but only after a phone call to the booker has been made. This is because the IS function does not permit noting why a reservation has been recalled.
User 2 A Resource Owner	D E ¬U ¬P	The Resource Manager is not aware of the possibility to recall accepted request.	'I cannot recall a reservation, even if I know that the room is empty, as if the meeting has been cancelled.'	The room is marked as booked in The Intranet, when it in fact is available.

From the example in Table 9 we can see that an observed phenomenon was that a Resource Owner (User 2) tended not to cancel bookings even though the resource was not any longer needed. The reason was that the Resource Owner was not aware of the possibility; she did not perceive the function. This can be due to that User 2 was recently promoted from Booker to Resource Owner. The direct effect of this user's perception was that the room remained marked as booked in the IS when it in fact was available for booking. User 1, on the other hand, perceived the function but used it unwillingly due to the function's limitation in creating

the required action space. The Resource Owner (User 1) considered that the IS did not support him to communicate the reason for the recall of the booking to the affected Booker.

5 Discussion

Prates *et al.* (2000) argue that a system can be perceived as a 'discourse deputy' for the designer. This means that the IS as deputy communicates to the users what the designer has predicted and that users can usually only communicate with the designer's deputy, not with the designer. Therefore effects of information systems design can only be identified in the users' perceptions of the IS, in terms of the D.EU.PS. model. Effects also exist as consequences of the design for user's actions based on that perception. For example, User 1 (see Table 9) regards the function Recall Accepted Request as not satisfactory and therefore decides to take action outside of the system by making a phone call before using it. This effect may eventually lead to unintended business effects such as the resource booking is not being used altogether (if, for example, the Resource Owner cannot get hold of the Booker).

A consequence (in this case a non-intended effect) of the inaction of User 2 (see Table 9), i.e., to not recall accepted requests, is that other users of the system believe the room to be booked. Thus, effects arise in system usage that in turn can have serious effects on business effectiveness and efficiency. Therefore it is important to see information systems as mediating tools for communication between different actors (Goldkuhl & Ågerfalk 2002).

A final note regards functionality that is undesired (¬D), non-existing (¬E), and unperceived (¬P); thus corresponding to Situation 8 in Figure 2. This would, for example, be the case if a new feature were proposed in response to identified problems during evaluation and we were to decide on whether to implement it or not. That is, to decide if it is desired or not. In set theoretical terms, we can think of such a class as the complement of the union of desired, existing and perceived functionality. The D.EU.PS. model can thus be used to discuss alternatives and 'new' functionality in terms of 'newness' (does it in fact already exist, even though maybe not perceived?), as well as 'desiredness' (is this function really desired?), and 'perceivedness' (if we were to implement this function, how would we make it readily perceivable to users?).

6 Conclusion

In this paper we have shown how the D.EU.PS. model can be used to study effects of information systems design. Effects arise at different levels, the perception of the action space spanned by the action potential of an IS, *per se*, is one such effect. Actions based on that perception could, in turn, lead to business effects; i.e., effects that involve other actors and that affect the effectiveness and efficiency of the business. Such effects can be intended or not by the designer. The important thing is to make the designer aware of potential effects of her design. To that end, the D.EU.PS. model may serve as an important analytic tool. Based on identified effects the designer can initiate re-designs, such as transforming not perceived functionality to perceived, and desired non-existing functionality to existing. D.EU.PS. based analysis may also make designers more aware of how a design affects the whole.

7 Acknowledgements

The D.EU.PS. model is developed within a project funded by the Swedish Agency for Innovation Systems (VINNOVA). We are grateful to the other project members' contributions to the development of the model: Stefan Cronholm, Göran Goldkuhl and Jonas Sjöström.

References

Agarwal, R., Karahanna, E. (2000) "Time Flies When You're Having Fun: Cognitive Absorption and Beliefs About Information Technology Usage", *MIS Quarterly,* Vol 24, No.4, pp 665–694.

Ågerfalk, P. J., Sjöström, J., Eliason, E., Cronholm, S., Goldkuhl, G. (2002) "Setting the Scene for Actability Evaluation: Understanding Information Systems in Context", In *Proceedings of the 9th European Conference on IT Evaluation (ECITE 2002),* 15–16 July 2002, Université Paris-Dauphine, France, Brown, A and Remenyi, D (Eds), MCIL, Reading, UK, pp 1–9.

Ågerfalk, P. J. (2003) *Nine Principles for Actable Systems Design,* Accepted for the 10th International Conference on Human-Computer Interaction (HCI International 2003), Crete, Grece, 22–27 June 2003.

Ågerfalk, P. J., Eliason, E. (2003) *Eighteen Classes of Functionality: The D.EU.PS. Model of Information Systems Use,* Accepted for the 10th International Conference on Human-Computer Interaction (HCI International 2003), Crete, Grece, 22–27 June 2003.

Alvesson, M., Sköldberg, K. (2000) *Reflexive Methodology: New Vistas for Qualitative Research,* SAGE, Thousand Oaks, CA.

Aronsson, G. (1990) "Handling Och Kontroll", In *Handling Och Handlingsutrymme,* Aronsson, G and Berglind, H (Eds), Studentlitteratur, Lund, pp 69–93.

Bærentsen, K. B., Trettvik, J. (2002) "An Activity Theory Approach to Affordance", In *Proceedings of the Second Nordic Conference on Human-Computer Interaction (NordiCHI 2002),* 19–23 October 2002, Aarhus, Denmark, ACM Press, New York, NY, pp 51–60.

Berglind, H. (1990) "Att Kunna Och Att Vilja: Om Villkoren För Mänskligt Handlande." In *Handling Och Handlingsutrymme,* Aronsson, G and Berglind, H (Eds), Studentlitteratur, Lund, pp 23–40. In Swedish.

Bevan, N., Macleod, M. (1994) "Usability Measurement in Context", *Behaviour & Information Technology,* Vol 13, No.1/2, pp 132–145.

Beyer, H., Holtzblatt, K. (1998) *Contextual Design: Defining Customer-Centered Systems,* Morgan Kaufmann Publishers, San Francisco, CA.

Cronholm, S., Goldkuhl, G. (2002) "Actable Information Systems: Quality Ideals Put into Practice", In *Proceedings of the 11th International Conference on Information Systems Development (ISD2002),* 12–14 September 2002, Riga, Latvia.

Davis, F. D. (1989) "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology", *MIS Quarterly,* Vol 13, No.3, pp 319–340.

DeLone, W. H., McLean, E. R. (1992) "Information Systems Success: The Quest for the Dependent Variable", *Information Systems Research,* Vol 3, No.1, pp 60–95.

Ehn, P. (1995) "Informatics: Design for Usability", In *The Infological Equation: Essays in the Honor of Börje Langefors,* Dahlbom, B (Ed) Gothenburg studies in information systems, Gothenburg University, Gothenburg, Sweden, pp 159–174.

Ehn, P., Löwgren, J. (1997) "Design for Quality-in-Use: Human-Computer Interaction Meets Information Systems Development", In *Handbook of Human-Computer Interaction,* Helander, M*, et al.* (Eds), Elsevier Science B.V., Amsterdam, pp 299–313.

Gibson, J. J. (1977) "The Theory of Affordances", In *Perceiving, Acting, and Knowing: Toward and Ecological Psychology,* Shaw, R and Bransford, J (Eds), Lawrence Erlbaum Associates, Hillsdale, NJ, pp 67–82.

Gibson, J. J. (1979) *The Ecological Approach to Visual Perception,* Houghton Mifflin Company, Boston, MA.

Goldkuhl, G., Röstlinger, A. (1993) "Joint Elicitation of Problems: An Important Aspect of Change Analysis", In *Human, Organizational, and Social Dimensions of Information Systems Development,* Proceedings of IFIP WG8.2 Working Group Information Systems Development: Human, Social, and Organizational Aspects, 17–19 May 1993, Nordwijkerhout, The Netherlands, Avison, D E*, et al.* (Eds), North-Holland, pp 107–125.

Goldkuhl, G., Ågerfalk, P. J. (2002) "Actability: A Way to Understand Information Systems Pragmatics", In *Coordination and Communication Using Signs: Studies in Organisational Semiotics 2,* Liu, K*, et al.* (Eds), Kluwer Academic Publishers, Boston, pp 85–113.

Goodhue, D. L., Thompson, R. L. (1995) "Task-Technology Fit and Individual Performance", *MIS Quarterly,* Vol 19, No.2, pp 213–236.

Holmlid, S. (2002) *Adapting Users: Towards a Theory of Use Quality,* Doctoral Dissertation, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

ISO 9241-11 (1998) *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs), Part 11: Guidance on Usability,* 1st ed., 1998-03-15*,* International Organization for Standardization, Geneva.

Kuutti, K. (1996) "Activity Theory as a Potential Framework for Human-Computer Interaction Research", In *Context and Consciousness: Activity Theory and Human-Computer Interaction,* Nardi, B A (Ed) MIT Press, Cambridge, MA, pp 17–44.

Kuutti, K. (1999) "Activity Theory, Transformation of Work, and Information Systems Design", In *Perspectives on Activity Theory,* Engeström, Y*, et al.* (Eds), Cambridge University Press, Cambridge, UK, pp 360–376.

Löwgren, J., Stolterman, E. (1998) *Design Av Informationsteknik: Materialet Utan Egenskaper,* In Swedish*,* Studentlitteratur, Lund. In Swedish.

Maguire, M. (2001) "Context of Use within Usability Activities", *International Journal of Human-Computer Studies,* Vol 55, No.4, pp 453–483.

McQuail, D. (2000) *McQuail's Mass Communication Theory,* 4th ed.*,* SAGE, London, UK.

Norman, D. A. (1988) *The Psychology of Everyday Things,* Basic Books, New York.

Norman, D. A. (1999) "Affordance, Conventions, and Design", *Interactions,* Vol 6, No.3, pp 38–43.

Orlikowski, W. J., Gash, D. C. (1994) "Technological Frames: Making Sense of Information Technology in Organizations", *ACM Transactions on Information Systems,* Vol 12, No.2, pp 174–207.

Patton, M. Q. (1990) *Qualitative Evaluation and Research Methods,* 2nd ed.*,* SAGE, Newbury Park, CA.

Polanyi, M. (1983) *The Tacit Dimension,* Peter Smith, Gloucester, MA.

Prates, R. O., de Souza, C. S., Barbosa, S. D. J. (2000) "A Method for Evaluating the Communicability of User Interfaces", *Interactions,* Vol 7, No.1, pp 31–38.

Schön, D. A. (1983) *The Reflective Practitioner: How Professionals Think in Action,* Basic Books, New York.

Shneiderman, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* 3rd ed.*,* Addison Wesley Longman, Reading, MA.

Stolterman, E., Nelson, H. (2000) "The Guarantor of Design: (G.O.D.)", In *Doing IT Together,* Proceedings of the 23rd Information Systems Research Seminar in Scandinavia (IRIS 23), 12–15 August 2000, Lingatan, Sweden, Vol. 2, Svensson, L*, et al.* (Eds), University of Trollhättan Uddevalla, Uddevalla, Sweden, pp 1177–1186.

Weber, M. (1978) *Economy and Society,* University of California Press, Berkeley, CA.

Yin, R. K. (1994) *Case Study Research: Design and Methods,* 2nd ed.*,* SAGE, Thousand Oaks, CA.