CASSM Taxonomy Of User-Interface-System Dependencies Plus Example Analyses

Ann Blandford & Iain Connell

March 2004

This document outlines the complete set of user-interface-system dependency values used by CASSM (Concept-based Analysis for Surface and Structural Misfits), and the taxonomy by which analysts may be guided when identifying a user-system misfit. Two example analyses (tables of dependencies), of the London Underground Ticket Vending Machines (TVMs), follow.

The document originated as appendices to Connell et al. 2004 (submitted to Behaviour & Information Technology in 2004).

Table 1 shows the complete set of user-interface-system dependency values used by CASSM.

USER	INTERFACE	SYSTEM
Present (P)	Present (P)	Present (P)
Difficult (D)	Difficult (D)	Absent (A)
Absent (A)	Absent (A)	

Table 1: CASSM user-interface-system dependency values.

In all cases, 'present' means clearly represented. We assume that underlying system concepts are either Present (P) or Absent (A), whereas for the user or at the interface there are concepts that are present but not clear, designated Difficult (D).

For users, difficult concepts are most commonly implicit – ideas they are aware of if asked but not ones they expect to work with.

There are various reasons why a concept my be represented at the interface but in a way that makes it difficult to work with. We have earlier created a more detailed taxonomy of types of difficulties that interface objects may present:

Disguised: represented, but hard to interpret;

Delayed: represented, but not available to the user until some time later in the interaction;

Hidden: represented, but the user has to perform an explicit action to reveal the state of the entity or attribute; or

Undiscoverable: represented only to the user who has good system knowledge, but unlikely to be discovered by most users.

Which of these apply in any particular case – i.e. why the interface object might cause user difficulties – is a further level of detail that can be annotated by the analyst.

We now summarise the situation for each combination of **User-Interface-System** possibilities :

P-P-P: No difficulties – good fit between user and system.

P-P-A: This is an unlikely combination, unless the analyst chooses to encode interface objects that only affect the display but not the underlying system representation in this way. Such a combination is unlikely to cause user difficulties.

P-D-P: This combination is likely to cause some user difficulties, depending on the exact reason why the interface representation causes difficulties.

P-D-A: Like present-present-absent, this is an unlikely combination, and it is up to the analyst to consider why they have encoded a concept in this way and what the likely difficulties might be.

P-A-P: The lack of interface representation would mean that users need to manipulate system concepts indirectly, which is likely to cause serious difficulties.

P-A-A: This is a common situation and is one of the three first-level cases. It causes difficulties.

D-P-P: In this case, the user may be forced to be explicit about some concept that they would naturally not mention. These are only problematic if the user is required to set or change values, not if the user only views pre-set system settings.

D-P-A: This is another unlikely combination, and is generally unlikely to cause user difficulties.

D-D-P: This combination is likely to cause user difficulties, depending on the exact reason why the interface representation causes difficulties. In this case the user is probably required to make explicit some information they would not normally work directly with.

D-D-A: Like present-present-absent, this is an unlikely combination, and it is up to the analyst to consider why they have encoded a concept in this way and what the likely difficulties might be.

D-A-P: The lack of interface representation would mean that users need to manipulate system concepts indirectly, which is likely to cause serious difficulties.

D-A-A: This is another unlikely combination. If it occurs, the analyst should consider the consequences.

A-P-P: As discussed above, these are concepts that the user has to learn; however, they are clearly represented at the interface.

A-P-A: This is an unlikely combination, unless the analyst chooses to encode interface objects that affect only the display but not the underlying system representation. These will be interface objects that are (presumably) easy to learn and only affect surface aspects of the interaction, and are therefore unlikely to cause great difficulties.

A-D-P: This combination is likely to cause user difficulties: these are important system concepts that are poorly represented at the interface in some way, but that the user has to learn to work with.

A-D-A: This is such an unlikely combination that it is equivalent to being absent from all three situations, which should never arise.

A-A-P: This is a source of user difficulties: something the user has to learn about if they are to work effectively with the system, but which cannot be accessed or manipulated through the interface.

Example: Two Independent CASSM Analyses (User-Interface-System Dependencies) of the London Underground Ticket Vending Machines (TVMs)

Tables 2 and 3 show the CASSM tables of User-Interface-System dependencies produced independently by two analysts. In both tables, analysts have distinguished between entities (e.g. Ticket Vending Machine, Table 2; Station, Tables 2 and 3) and attributes pertaining to each entity (e.g. Machine type, Table 2; Zone, table 3).

Concept	USER	INTERFACE	SYSTEM
Ticket Vending Machine	P	P	P
Machine type	D	P	P
Payment method	P	D	P
Payment status	P	D	P
Ticket	P	P	P
Ticket type	P	D	P
Date of issue	P	D	P
Time of issue	P	D	P
Period of validity	P	D	P
Price	D	D	P
Status (age of traveller)	D	A	P
Payment (transaction) method	P	D	P
Station	P	P	P
Name	P	P	P
Groundedness	D	P	P
Line	P	A	A
Name	P	A	A
Line colour	P	A	A
Zone	D	D	P
Name (number or letter)	D	D	P
Boundaries	D	D	P
Journey	P	D	P
Day of travel	P	A	P
Time of travel	P	A	P
Source station	P	A	P
Source zone	D	A	P
Destination station	P	D	P
Destination zone	D	P	P
Destination Line	D	A	A
Means of travel	P	A	P
Traveller	P	A	P
Age of traveller	P	A	P
(Dis)ability	P	A	A
Purchaser	P	A	A
Status (age)	P	A	A

Table 2: Analyst 1 CASSM analysis (table of User-Interface-System dependencies for the TVMs system).

Concept	USER	INTERFACE	SYSTEM
Station	P	A	P
Zone	A	P	P
Place (in London)	P	A	A
Distance	P	A	A
Transaction	D	P	P
Number of tickets	P	P	P
Ticket	P	P	P
Price	P	P	P
Type	P	P	P
Peak/off-peak	A	D	P
Duration of validity	D	P	P
Travel method validity	A	A	P
Current time	D	A	P
Change	D	P	P
Date	D	A	P
Ticket machine	P	P	P
Payment method	D	P	P
Payment method acceptance status	A	P	P
Capability	D	D	P
Traveller	P	P	P
Traveller age	D	A	A
Adult / child	P	P	P
Photo ID	P	D	P
Payment acceptance status	A	P	P
User payment method	D	P	P

Table 3: Analyst 2 CASSM analysis (table of User-Interface-System dependencies for the TVMs system).

Reference

CONNELL, I.W., BLANDFORD, A.E. and GREEN, T.R.G. CASSM and Cognitive Walkthrough: usability issues with ticket vending machines. Submitted to *Behaviour & Information Technology* in 2004.