



Concept-Based Analysis for Surface and Structural Misfits (CASSM *formerly known as OSM*)

Ann Blandford, Iain Connell & Thomas Green

Concept-Based Analysis for Surface and Structural Misfits
(CASSM) focuses on the user's and system's conceptual structures, and on
misfits between the two.

Aims of this tutorial

- You should become familiar with the basic principles underpinning CASSM, a declarative approach to evaluation.
- You should be able to apply CASSM competently.

The tutorial will combine presentation components with plenty of hands-on practice.

Structure of these notes

1. Background and overview
2. Focus on data gathering and preliminary analysis
3. Extension to full surface-level analysis
4. Extension to structural analysis

Part 1:

Background: Other techniques and the origins of CASSM

Overview of CASSM concepts and process

Introductory example: a drawing tool

Part 2:

Gathering user data – principles and techniques

Practical example: heating controller

Part 3:

Going the distance: all aspects of surface-level analysis

Practical example: e-commerce site

Part 4:

Introduction to Cognitive Dimensions

Analysing for viscosity, premature commitment, etc

Practical example: back to the drawing board

Summary and wrap-up

Part 1: Background & Overview

Compare and contrast

- Cognitive approaches (e.g. GOMS, CW) focus on user cognition and (mental) tasks.
- Task centred approaches (e.g. HTA, TKS) focus on (physical) processes.
- Heuristic approaches (e.g. HE) focus on surface features of design.
- CASSM focuses on concepts, and relationships between them.

GOMS stands for Goals Operators Methods and Selection rules. The approach assumes that users' tasks are hierarchically structured and that the key issues revolve around how long each mental process or physical action takes. Users are assumed to be experts, and to know how to work with the device. More details can be found in:

Card, S. K., Moran, T. P. & Newell, A. (1983) *The Psychology of Human Computer Interaction*, Hillsdale NJ: Lawrence Erlbaum.

John, B. & Kieras, D. E. (1996) Using GOMS for user interface design and evaluation: which technique? *ACM ToCHI* 3.4. 287-319.

CW, or Cognitive Walkthrough, is based on a similar kind of theory, but makes contrasting assumptions about users – namely that they are novices with clearly articulated goals but who are exploring the device, so that they have to work out what to do next at every step. See:

Wharton, C., Rieman, J., Lewis, C. & Polson, P. (1994) The cognitive walkthrough method: A practitioner's guide. In J. Nielsen & R. Mack (Eds.), *Usability inspection methods* (pp. 105-140) New York: John Wiley.

HTA (Hierarchical Task Analysis) and TKS (Task Knowledge Structures) are task-based approaches that consider the physical actions users have to perform and (to some extent) the knowledge users have to have to perform those tasks. See:

Johnson, P. (1992) *Human-Computer Interaction: Psychology, Task Analysis and Software Engineering*. London: McGraw-Hill.

Heuristic approaches, such as Nielsen's Heuristic Evaluation, provide a kind of check-list against which the design can be assessed – e.g. asking questions like whether users get appropriate feedback at every step. See:

Nielsen, J. (1994) Heuristic Evaluation. In J. Nielsen & R. Mack (Eds.), *Usability Inspection Methods* (pp. 25-62) New York: John Wiley.

All of these approaches have things to commend them, and all have a potential place to play in design and evaluation. In practice, the limited evidence there is indicates that few techniques have been taken up widely in industry – the exceptions probably being HTA and HE.

What's CASSM?

- Focuses on the core concepts and structures that define a “domain”
- Goes into no more detail than necessary
- Is a description that focuses on particular aspects, leading to identification of ‘misfits’
- Takes the user’s perspective.

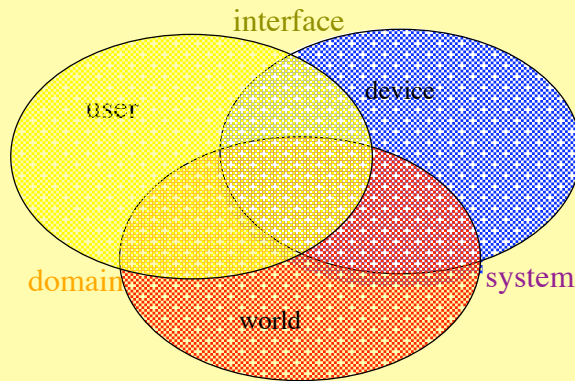
A CASSM model is an attempt to describe the ideas and concepts that a user needs to know about when using a system to achieve things in the world. It's a description of the concepts and 'things' in the system and in the world and how they are related, not a recipe for how to use the system. Rather than referring to “the world” in this document, however, we'll talk about “the domain” — that is, the little bit of the world that the system is designed to make changes to (whether that's a word processed document, a drawing, or the tuning of a radio station).

The description is sketchy. Computer scientists might prefer the term “partial” : you don't have to describe things in more detail than necessary for the purpose. Of course, the “down” side of this is that you have to judge for yourself exactly how much detail will be useful for a given analysis.

We're using the word “model” to mean: a description of something that focuses on particular aspects of it, and that allows one to reason about selected properties and behaviours of it.

There is one more feature of CASSMs that is really important: they describe the domain and device from an individual's perspective. We are not producing an abstract domain model, but one that takes a focused view. E.g. if we are analysing a spreadsheet program from the perspective of a lecturer who wanted to use it for collating students' marks, we would want to consider entities such as marks and marking schemes, and how well they can be represented and worked with in the spreadsheet, whereas if we were considering the same spreadsheet program for use in a small business, we would need to consider product types and costs, bills and receipts, etc.

Systems and domains: a sketch



OSM Interact tutorial

6

The Venn diagram above is intended to sketch out what we mean by ‘system’ and ‘domain’. Basically, the ‘domain’ is the user’s understanding of the world in which they are working, while the ‘system’ is the device and organisational representation of that world. Occasionally, there are concepts within the organisational system that the user has to work with that are not represented within the device; this is why the purple shading extends outside the world-device intersection.

User and device have to work together through the interface.

Shared concepts lie within the central (brownish) area.

If this diagram doesn’t help you, don’t worry about it.

Why?

- Consider user and system as ‘equals’.
- Identify **misfits** between user and system representations.
- Avoid ‘death by detail’.
- Work towards automatic identification of misfits...

Core concepts

- **Entities:** *concepts* the user is working with that can be created or deleted, or that have attributes.
- **Attributes:** properties of entities.
- **Actions:** change the state of an entity or attribute.
- **Relationships & constraints** that hold within the system.

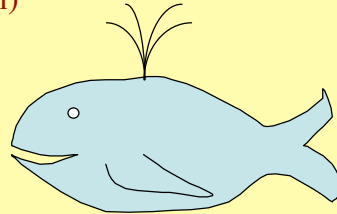
CASSM is a structured description of the entities, attributes and actions that a user needs to work with. It contrasts with nearly all the other usability evaluation techniques that we know of by focusing primarily on entities and attributes and structures, rather than on tasks, action sequences or scenarios. As you will see, there are cases where this is a better way of working, and cases where it is less effective. We don't claim to be providing a universal evaluation tool that will replace all others, but to be providing one that does well some things that the others don't.

An CASSM describes the:

- **entities** that the user has to work with. These entities, or “things” may be relevant to the domain or the device, or both. An entity is something that the user can create or delete, or change the attributes of.
- **attributes**, or properties, of entities. Attributes cannot exist in isolation (for example, you couldn't have “*italicised*” without having some text to apply it to!), so they cannot be created or deleted; they can only be changed.
- **actions** that effect a change to the state (i.e. creation or deletion of an entity, or setting of an attribute value).
- **relationships and constraints** that hold (e.g. that one entity *consists of* other entities, or that an action *affects* the state of another entity).

Example

- Entity:
 - Domain object (e.g. 'whale's tail')
 - Device object (e.g. 'curve')
- Attribute of domain object:
 - Colour(s)
 - Shape
- Action:
 - Change colour by using 'fill' on device object
- Relationship:
 - Domain object *consists of* device objects



Actions can only be performed on device concepts – one of the key issues addressed by CASSM is how the user makes domain-significant changes when domain changes always have to be mediated by device changes.

Origins of CASSM

- Entity Relationship Modelling for Information Artifacts (ERMIA: Benyon & Green)
 - Focus on structures, but demands detail
- Programmable User Modelling (PUM: Young & Blandford)
 - Focus on declarative knowledge, but demands detail
- Cognitive Dimensions (Green *et al*)
 - Useful but imprecise

CASSM's choice of what to represent draws on previous work, much of it our own, that explores representations of misfits. To predict misfit, an explicit description of the device or the information structures is needed. The representation of ERMIA can exhibit some kinds of misfits between users' conceptual models and the structures imposed by the device; CASSM builds on that but is more expressive, while also being deliberately sketchier and less rigorous in detail. Since CASSM includes the consequences of actions, it can be used for analysing a wider range of misfits than can ERMIA.

To represent the user's knowledge CASSM draws on Programmable User Modelling. For a given device, PUM models of goal-based behaviour include a detailed list of operators and their preconditions. Cognitive modelling techniques are applied to the modelled user's knowledge to determine likely behaviours. CASSM demands much less rigour than a full PUM knowledge analysis, and allows a wider range of concerns to be expressed within the model. Also, CASSM makes explicit the need for knowledge elicitation from users, thereby grounding it much more firmly in real-world data.

Finally, various types of misfit need to be identified and analysed. Many of the 'cognitive dimensions of notations' are examples of misfits. However, the cognitive dimensions were presented entirely informally, and the method of analysis was no more than asking a designer to think carefully about each dimension in turn. See Blackwell & Green (2003) for a more structured view of CDs analysis. With CASSM, some of the dimensions can be evaluated algorithmically, once a description of the artefact has been constructed. In other words, many of the Cognitive Dimensions can be expressed precisely in CASSM terms.

Compared to our past work, CASSM takes a middle position, being less rigorous than either ERMIA or PUM but more rigorous than Cognitive Dimensions.

Attributes of entities and attributes

- | | | |
|-------------|-------------|-----------|
| • User: | • Interface | • System: |
| – Absent | – Absent | – Absent |
| – Difficult | – Difficult | – Present |
| – Present | – Present | |

We need to define all these terms:

Absent: not directly represented.

Present: directly represented.

Difficult: there are many ways in which a user or interface concept might present difficulties. One kind of user difficulty might be that the concept is implicit – i.e. the user would recognise this entity / attribute as being important if asked, but wouldn't naturally name it. Another might be that it is difficult for users to learn.

There are various ways in which an interface concept might pose difficulties. For example, it might be:

Disguised: the information is immediately available, but only experienced users will recognise it as such.

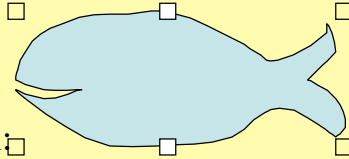
Delayed: the state of the entity/attribute is not immediately accessible to the user, but may be shown at some later time.

Hidden: the user can access the information but it's not immediately available. That is: the user has to perform an explicit action to reveal the information.

Undiscoverable: the information is available, but so well hidden that few users will find it.

Examples

- DomainObject in drawing tool:
 - System: Absent
 - User: Present
 - Interface: Present (*probably*)
- ObjectHandle in drawing tool:
 - System: Present
 - User: Absent for a novice (user has to learn)
Present for most users (user has to know about it to be able to manipulate it)
 - Interface: Difficult [hidden]



Some of these attributes are difficult to be sure about – that is one of the challenges of modelling. Some will also be different for different users, and for the same user at different times.

Actions

- Actions are linked with entities and attributes:
 - Entities can be created or deleted (or these actions may not be possible)
 - Attributes can be set (sometimes) or changed
- Actions may be:
 - Easy
 - Hard – e.g. moded or disguised
 - Indirect
 - Impossible
 - Done by the system (not this user, anyway)
 - ... but we only worry about this level of detail when quite far into an analysis

An action is moded if the same physical action has different effects in different circumstances, and disguised if it is difficult for the user to discover the action or its effect.

An action is indirect if the only way to change the current concept of interest is by changing something else and having this happen as a side-effect.

Relationships & constraints

- Consists_of
 - E.g. a London Underground zone *consists_of* stations
- Constrained_by_goal
 - E.g. user may need to keep document to a fixed length or number of pages
- Constrained_by_device
 - E.g. memory capacity of a disk
- Affects
 - E.g. changing page width affects number of lines of document
- Maps-onto
 - E.g. in the drawing tool, the distinction between DomainObject and DeviceObject is important, but an early analysis may confuse them

For some analyses, relationships are central. For example, they are key to the identification of Cognitive Dimensions such as viscosity and premature commitment. For others, the main misfits lie simply in the mismatch of surface-level representations, so relationships are of comparatively little importance. It really depends on the kind of system and the level of detail of analysis.

Finally, notes

- In many situations, the activity of constructing a model provides more insights that inspecting the resulting model.
- CASSM recognises this by providing a ‘notes’ area where the user can add any important observations about usability issues.

Similarly, in an electronic diary system, we found that the user’s concept of a ‘meeting’ was subtly but importantly different from the system concept implemented in the electronic diary. The user’s concept typically centred on who would be there and what the purpose was, whereas the system concept centred on what other users of the same shared diary system were invited to the meeting, when it starts, when it finishes and where it would be held. These apparently small differences resulted in substantial usability difficulties.

Exercise

- Work in pairs
- Use the information below, with your general understanding of drawing programs, to generate an CASSM description
- See next page for screen-shot
- Use tables in workbook if it helps ... but at this stage the main point is to identify concepts, not to organise them.

JSketch, by John Rieman.

1994

Some parts copyright Semantec Corp (Think C).

It's copyrighted, but it's free.

Lines are formed with Bezier curves. In "Drag" mode, new points are put down four times a second while the mouse button is down and the mouse is moving. After the first four points, each new point triggers the drawing of a new curve. Start a new set of curves (a new "line") by releasing the mouse button and putting it down again.

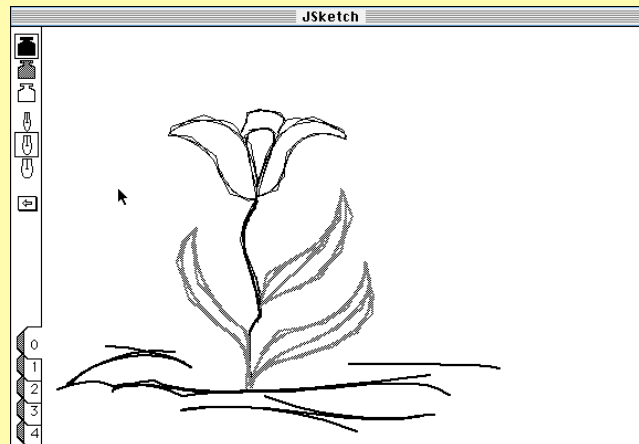
In "Click" mode, you put your own points down wherever you want them. Beware of repeated motion injury! To start a new set of curves in Click mode, press the space bar or change the pen or ink.

Delete on the keyboard or the Backspace button on-screen will remove the last point and undraw the associated curve, but it only backs up through the most recent set of curves.

Tabs on the lower left control layers. Click on main tab area to bring layer to the top, so you can draw on it. Click on the tab edge to make the ink on that layer visible or invisible.

To print, use "FlashIt" or some other screen snapshot package to grab the screen, then print from MacDraw or whatever.

Examples screen



OSM Interact tutorial

17

This tool is a research prototype; the description on the previous page is of version 1; a more recent version (not described) is easier to use — but correspondingly less interesting to describe in CASSM.

The tools on the left are as follows:

- Three 'inkwells' containing different colours: black, grey and white;
- Three pen-nibs to give different thicknesses of lines;
- A 'backspace' button to undo the last point laid down; and
- Five tabs to determine the current drawing surface (of the five sheets available). The grey left-hand edges of these tabs can set the ink on that sheet to be visible or invisible. So, for example, if you wanted the leaves of the flower to appear and disappear, you'd draw them on a different sheet from everything else and then click on the tab-edge for that sheet to make them invisible.

Group discussion

- What were key concepts?
- What misfits did you detect at this stage?
 - These will be primarily user concepts that aren't represented within the system, and system concepts that are hard for the user to learn and work with.

Part 2: focus on data gathering and preliminary analysis:

first, the user view

- (Paradoxically) some of the detailed analysis can be done without gathering real user data, but high-level analysis can't
- Data gathering must involve users talking!
 - Focus groups
 - Interviews
 - Contextual Inquiry observations & interviews
 - Think-aloud protocols

OSM Interact tutorial

19

There are no prescribed data collection methods. Indeed, triangulation across different methods is encouraged. What matters for the user data is that it is gathered from representative users from the particular population for which the current use of the system is being considered.

E.g. if we are analysing a spreadsheet program from the perspective of a lecturer who wanted to use it for collating students' marks, we would want to consider entities such as marks and marking schemes, and how well they can be represented and worked with in the spreadsheet, whereas if we were considering the same spreadsheet program for use in a small business, we would need to consider product types and costs, bills and receipts, etc. And in each of these cases we would gather data from users in that particular user population.

What data gathering techniques are chosen will depend on the situation, as well as your personal preferences.

Information on data gathering techniques can be found in:

Preece, J. Rogers, Y. & Sharp, H. *Interaction Design*. Wiley, New York (2002)

Information specifically about Contextual Inquiry is included in:

Beyer, H. & Holtzblatt, K. *Contextual Design*. San Francisco : Morgan Kaufmann. 1998.

Focus groups

- Gather together group of people who have relevant experience of a set of tasks the system is to support.
- Lead discussion on what their job entails, how they work, experience of current systems, requirements for new systems
 - Efficient
 - Users cannot demonstrate points
 - Decontextualised from work setting

Interviews

- More intensive than focus groups
- Address similar topics
- Can deal with issues in depth
- Like focus groups, interviews are usually decontextualised (but may be held in the work environment)

Contextual Inquiry

- Observe user in work-place
- When appropriate, interrupt to ask questions that probe the nature of the work, the use of technology (including low-tech paper, phones, etc.), the broader context, etc.
- Context 'comes for free', but must be noted
- Not practical for all working situations

Think-aloud protocol

- User ‘thinks aloud’ while working.
- Some users are more articulate than others!
- Some tasks are too demanding

However you gather data...

- Record it (audio is best – notes is sometimes the only option; notes are faster to analyse!)
- Transcribe it (maybe selectively)
- Systematically highlight all concepts users refer to.
 - Also key actions and relationships, if possible (this is harder)
- Assign as entities and attributes when you feel ready to do so

The device view

- Documentation is a good place to start. Particularly user manuals
- Supplement where possible with use of system
- Analyse in the same way as the user side.

Constructing an CASSM

- Start with data
 - System data from system documentation, interviews with developers, inspection of system, etc.
 - User data from think-aloud protocols, contextual inquiry, interviews, focus groups, etc.

Organise data

- Organisation of data involves working through it, identifying all important concepts (entities, attributes, actions, relationships) and laying these out.
- We use tabular paper-based forms for the early stages of this process.
- The Cassata tool (prototype) helps stop cheating as the analyst moves towards a more precise description.

The Cassata tool is not available yet as it is still undergoing development and testing. It will be available from the project website in due course...

Example

Entity	Create by	Delete by	Notes
Drawing object	Drag mouse pointer using a palette tool	Select using pointer and delete	The appropriate tool must first be selected from the palette (tool bar), thus heavily moded

Attributes	Set by	Change by	Notes
Type	Select tool from palette	Impossible	Later versions of Word and other applications have increasingly varied sub-tool sets (eg basic shapes, arrows) and line configurations (eg freeform, scribble).

Even before you get to this point, you may have just listed key concepts on the user and device side – as you did in the earlier exercise. But note that these descriptions here are for a more traditional object-based drawing package with different tools to draw ellipses, rectangles, lines, etc.

NB, for space reasons on powerpoint slides, the user, interface and system columns are omitted from this table.

Group working

- The second example we'll work on is a simple heating controller. We've chosen this because:
 - We have a simulation we can demonstrate!
 - We expect you all to have experience of similar devices

The device

<http://www.thomas-green.ndtilda.co.uk/workStuff/devices/controllers/HeatingA2.html>



Summary Instructions

These controls determine three periods of heating for each weekday. By clicking the ADVANCE button you can move from day to day and within each day you can move through the ON and OFF times for each period. The PLUS and MINUS buttons alter whichever panel is currently highlighted. **Don't try typing in the boxes - just use the PLUS and MINUS buttons.**

When the day of the week is highlighted, PLUS moves on to the next day, and ADVANCE brings up the settings for the day, while you can use the COPY button to copy yesterday's settings to today.

Process

- Where possible, work in threes.
- To identify the **user view**, each of you interview one of the other for 5-10 minutes. The third person should take as detailed notes as possible, focusing on concepts.
- To identify the **system view**, consider the device description (previous page). Identify core concepts from that.

When interviewing, questions you might ask include:

Describe a controller you're familiar with

Describe how you use it

Are there particular occasions when it doesn't work as you'd like, and how do you get around problems with it?

What other functions would you like it to include?

Process (2)

- Bring together your three sets of interview notes to create a common user description
- Construct a device and interface description
- Bring the two together to construct a first CASSM
- Do a misfit analysis (*see notes*)

Do a misfit analysis by identifying:

- User concepts that are missing from the device
- Device concepts that the user needs to learn
- Interface concepts that are difficult in any way
- Anything else you noted while working on the analysis

A comment on ‘the system’

- The heating controller only controls part of the home heating system
- The user will probably consider it as a total system – whose function is to keep the place at a comfortable temperature at reasonable cost
- Try in your analysis to focus mainly on the parts of the system controlled by the timer

...and whatever you do, don't get engrossed in the physics of heating systems!

However, the fact that the heating controller is of limited scope may mean that your analysis flags several user concepts that are not part of the device model, but that nevertheless aren't a problem in reality. You need to use your skill and judgement as an analyst to decide which problems are real and which are 'ghost' ones.

How did you get on?

- What were your key findings?
- In particular, what misfits did you detect?
- What aspects of analysis did you find particularly difficult?

There is a full worked analysis in a separate document available from the project website, against which you can compare your answer. We will discuss 'structural misfits' later this afternoon.






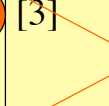
Part 3: all aspects of surface-level analysis

- 3 stages:
 1. Ignore the interface; just consider the user and the system
 2. Include the interface, to give a much richer taxonomy of possibilities
 3. Include actions and their potential difficulties

Within ‘surface’, there are three further levels: first, just identifying system and user concepts, with little reference to the interface between the two (we have done analyses – for example of an ambulance control system – where this was the only level analysed); second, more detailed analysis in terms of how well each concept is represented by the user, interface and system; third, analysis in terms of what actions are needed to change the system, and whether there are problems with actions. At this point, concepts are usefully sorted into entities and attributes to support thinking about what can be set, changed, created or deleted.

It is only when considering structural misfits that relationships really come into play, so they can often be ignored until quite late on into analysis.

User and system

System \ User ->	Present	Absent
Present		[2]  
Absent	[1] 	 [3] 

OSM Interact tutorial

36

These misfits fit broadly into three classes:

[1] **User concepts that are not represented within the system**, and hence cannot be directly manipulated by the user. A very simple example is the use of land-line telephones: the user is normally interested in speaking to a particular individual, but the only means of doing that is to place a call to a location where they are likely to be (e.g. their home or office), because the telephone system does not map handsets to individuals. They also often force users to introduce work-arounds, as users are unable to express exactly what they need to, and therefore use the system in a way it wasn't designed for.

[2] **System concepts that the user has to know about** but that are not naturally part of their initial understanding, and therefore need to be learned. An example is the 'layers' of most drawing tools that users need to be aware of and work with as they manipulate drawing objects. For users, these misfits may involve simply learning a new concept, or they may involve the users constantly tracking the state of something that has little significance to them.

[3] **User- and system concepts that are similar but non-identical**, and which are often referred to by the same terms. One example in the domain of diaries is the idea of a 'meeting'. The difference between these concepts is not great, but the user of the system who has lots of external meetings does not find the meetings facility useful. These misfits may cause difficulties because the user has to constantly map his / her natural understanding of the concept onto the one represented within the system, which may have a subtly different set of attributes that the user then has to work with.

Include the interface: a) system Present

<i>Interface</i>	User		
	Present	Difficult	Absent
<i>Present</i>	Good	Difficult (e.g. forced explicitness)	Difficult, but should be easy to learn
<i>Difficult</i>	Difficult	Difficult	Difficult
<i>Absent</i>	Difficult	Difficult	Difficult

We use the coding: user-interface-system :

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

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

P-A-P: This is an unlikely combination. The lack of interface representation means that users need to manipulate system concepts indirectly, which will cause serious difficulties.

D-P-P: 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 values, not if the user only views preset system settings.

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

D-A-P: This is an unlikely combination. The lack of interface representation means that users need to manipulate system concepts indirectly, which will cause serious difficulties.

A-P-P: These are concepts that the user has to learn; however, because they are clearly represented at the interface, they are unlikely to cause serious 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. The example of layers in a drawing package has been discussed.

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. This is a fairly unusual situation, but does occur; for example, drawing packages such as JSketch implement Bezier curves that can be drawn but not subsequently manipulated. The user therefore has to work with the Bezier curve points, but they are not accessible through the interface.

Include the interface: b) system Absent

	User		
<i>Interface</i>	Present	Difficult	Absent
<i>Present</i>	Good	–	–
<i>Difficult</i>	–	–	–
<i>Absent</i>	Difficult	–	–

Again, we use the coding: user-interface-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 – e.g. a ‘zoom’ function that is clearly represented at the interface with a recognisable magnifying glass. Such a combination is unlikely to cause user difficulties.

P-D-A: 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-A: This is a common situation and it causes difficulties.

D-P-A: This is an unlikely combination, but is unlikely to cause user difficulties.

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

A-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. These will be interface objects that are easy to learn and only affect surface aspects of the interaction anyway, and are therefore unlikely to cause great difficulties.

A-D-A: This is a very unlikely combination which should never arise.

A-A-A: There is absolutely no point in encoding these concepts within CASSM!

Information about actions

- Actions may be:
 - Easy
 - Hard (e.g. undiscoverable, moded, delayed effect)
 - Indirect
 - Impossible
 - Done by the ‘system’

Indirect actions are ones where the user has to change *something else* in order to achieve the intended effect.

Impossible actions are actually not always problematic. For example, entities that are listed simply because they have attributes that can be changed are impossible to create or delete, but this doesn't cause difficulties.

Considering actions: be alert to:

- Irreversible settings or changings
- Side-effects the user may not intend or notice
- Difficult settings / changings
- Undiscoverable actions
- Moded actions
- Unpredictable actions

Because CASSM is primarily concerned with conceptual and structural misfits, actions are of only secondary concern. The analyst is encouraged to define how actions change the existence of entities or the values of attributes, as a further step of analysis. In particular, the analyst should be alert to:

- Changes that are impossible. For example, are there actions whose effect is irreversible, or very difficult to undo?
- Side-effects of actions on other aspects of the system state. This leads to the identification of 'affects' relationships that are considered in more detail below.
- Settings and changings that are just more difficult than one might reasonably expect. These aren't deep misfits but they nevertheless cause user difficulties.
- Actions that the user is unlikely to discover for some reason. For example, on a Mac it is possible to select a sentence in MS Word by pressing the 'apple' key and clicking anywhere in the sentence; not many people know this!
- Actions that are context-dependent – e.g. depend on the system being in a particular mode.
- Actions that have unpredictable effects for the user (and if so, why?).

Action-related misfits are not automatically analysed, and the analyst should therefore make a note of misfits identified as working through the analysis. This is one of the reasons for having a 'notes' area!

Group working

- This time, you don't have an explicit device description so you're going to have to infer it from screen shots from a (fictitious) e-commerce site.
- You should interview each other about the concepts involved in on-line grocery shopping before analysing the site.
- Go through all three levels of CASSM analysis.

This example is intentionally less structured than those from this morning. You should devise your own interview questions as well as inferring system concepts from a 'walkthrough' of a partial task.

To remind you: the three levels are:

1. Broad user-system concepts and misfits between them.
2. More detailed analysis including the interface, and ideas such as 'implicit' and 'difficult'.
3. Add in actions and identify difficulties related to them.

We'll move on to consider relationships and constraints, and structural misfits, in the last session.

Welcome to the site!

Returning
customers log in



New customers
have a look around



Buy groceries!

1 hour delivery slots!

Next-day delivery

Free delivery

Special offers!

- New customer, so have a look around

This time, you are not being given an abstract device description, but are being provided with a walk-through of one task. We know this interface has some serious deficiencies – that's no accident!

Food & Drink	Toiletries & Baby	Household & Pet Care	Special Offers	Recipes
Bakery, Biscuits & Cereal		Confectionery & Crisps		Delicatessen & Cold Meat
Dried, Cooking & Jams		Drinks & Alcohol		Fresh Ready Meals
Fruit, Vegetable & Salad		Healthy		Meat, Fish & Poultry
Milk, Dairy & Frozen		Organics		Tinned

[home](#) [help](#) Message of the day

[search](#) [Book delivery](#)

[register](#) [Log in](#)

[Contact us](#)

Your trolley:

Your trolley is empty

- Click on 'organics'

Food & Drink	Toiletries & Baby	Household & Pet Care	Special Offers	Recipes
Bakery, Biscuits & Cereal		Confectionery & Crisps		Delicatessen & Cold Meat
Dried, Cooking & Jams		Drinks & Alcohol		Fresh Ready Meals
Fruit, Vegetable & Salad		Healthy		Meat, Fish & Poultry
Milk, Dairy & Frozen		Organics		Tinned

home	help	Baby products	Your trolley: <i>Your trolley is empty</i>
search	Book delivery	Bakery	
register	Log in	Confectionery	
Contact us		Frozen	
		Fruit & vegetables	
		Meats	

- Click on 'fruit & vegetables'

Note that clicking on 'Fruit, Vegetables & Salad' might be an alternative, but the user would then have to find the specific organic produce within that broad category.

Food & Drink	Toiletries & Baby	Household & Pet Care	Special Offers	Recipes
Bakery, Biscuits & Cereal		Confectionery & Crisps		Delicatessen & Cold Meat
Dried, Cooking & Jams		Drinks & Alcohol		Fresh Ready Meals
Fruit, Vegetable & Salad		Healthy		Meat, Fish & Poultry
Milk, Dairy & Frozen		Organics		Tinned

[home](#) [help](#) [Fruit](#)
[search](#) [Book delivery](#) [Herbs](#)
[register](#) [Log in](#) [Salads](#)
[Contact us](#) [Vegetables](#)

Your trolley:

Your trolley is empty

- Click on 'salads'

Food & Drink	Toiletries & Baby	Household & Pet Care	Special Offers	Recipes
Bakery, Biscuits & Cereal		Confectionery & Crisps		Delicatessen & Cold Meat
Dried, Cooking & Jams		Drinks & Alcohol		Fresh Ready Meals
Fruit, Vegetable & Salad		Healthy		Meat, Fish & Poultry
Milk, Dairy & Frozen		Organics		Tinned

home	help
search	Book delivery
register	Log in
Contact us	

Cucumbers
£0.89 each



+

Lettuce
£0.59 each

Your trolley:
Your trolley is empty

- Scroll down to find tomatoes

Food & Drink	Toiletries & Baby	Household & Pet Care	Special Offers	Recipes
Bakery, Biscuits & Cereal		Confectionery & Crisps		Delicatessen & Cold Meat
Dried, Cooking & Jams		Drinks & Alcohol		Fresh Ready Meals
Fruit, Vegetable & Salad		Healthy		Meat, Fish & Poultry
Milk, Dairy & Frozen		Organics		Tinned

home	help
search	Book delivery
register	Log in
Contact us	

Tomatoes
£1.15/kg



+

Your trolley:
Your trolley is empty

- Click on ‘+’

Food & Drink	Toiletries & Baby	Household & Pet Care	Special Offers	Recipes
Bakery, Biscuits & Cereal		Confectionery & Crisps		Delicatessen & Cold Meat
Dried, Cooking & Jams		Drinks & Alcohol		Fresh Ready Meals
Fruit, Vegetable & Salad		Healthy		Meat, Fish & Poultry
Milk, Dairy & Frozen		Organics		Tinned

[home](#) [help](#)

[search](#) [Book delivery](#)

[register](#) [Log in](#)

[Contact us](#)

Before you can purchase goods you need to register or log in

Your trolley:
Your trolley is empty

- Click on 'register'

To check whether we deliver to your area, please enter your postcode:

home

help

register

Contact us

OK

- Click on the box, then type postcode then press 'OK'

Please select the full address of the property:

home	1 Acacia Avenue	OK
help	10 Acacia Avenue	
register	11 Acacia Avenue	
Contact us	12 Acacia Avenue	
	13 Acacia Avenue	
	14Acacia Avenue	

- Scroll down to correct number

Please select the full address of the property:

<input type="button" value="home"/>	18 Acacia Avenue ▲	<input type="button" value="OK"/>
<input type="button" value="help"/>	19 Acacia Avenue	
<input type="button" value="register"/>	2 Acacia Avenue	
<input type="button" value="Contact us"/>	20 Acacia Avenue	
	21 Acacia Avenue	
	22 Acacia Avenue ▼	

- Select correct number then press 'OK'

Please confirm address:

home

help

register

Contact us

2 Acacia Avenue
Newtown
Beds
BE1 1AB

OK

- Press 'OK'

Please enter contact details:

<input type="button" value="home"/>	Name:	<input type="text"/>
<input type="button" value="help"/>	Email:	<input type="text"/>
<input type="button" value="register"/>	Tel:	<input type="text"/>
<input type="button" value="Contact us"/>		<input type="button" value="OK"/>

- Click on each box then type in details. Press 'OK'

Food & Drink	Toiletries & Baby	Household & Pet Care	Special Offers	Recipes
Bakery, Biscuits & Cereal		Confectionery & Crisps		Delicatessen & Cold Meat
Dried, Cooking & Jams		Drinks & Alcohol		Fresh Ready Meals
Fruit, Vegetable & Salad		Healthy		Meat, Fish & Poultry
Milk, Dairy & Frozen		Organics		Tinned

home	help	Welcome, Jessica Bloggs! Message of the day	Your trolley: <i>Your trolley is empty</i>
search	Book delivery		
register	Log in		
Contact us			

- Click on 'organics'...

Over to you. Note that this site has both CASSM-style misfits and also user difficulties that are more to do with the task structure. If you spot these too then great! But they're not the focus of this exercise.

FAQs

1. Am I going to get this right first time?
 - No, nobody could do that.
2. Am I *ever* going to get right ?
 - Who says there's a right answer?
3. How do I know when to stop?
 - Use common sense and focus on the questions you're trying to answer
4. Should an CASSM be a complete description?
 - No. It's only partial.

- 1) You have to chew it over and work things out.
- 2) This is a description of what *you* see as important.
- 3) Try to get down all the new concepts a learner would need. Don't bother with stuff they'd know already, such as what a mouse is or how a menu works.

FAQs (2)

1. There are lots of different commands, but they're all a bit similar. Do I need to describe every one of them?
 - No. Focus on what matters.
2. What's the difference between an entity and an attribute?
 - Entities can be created and deleted; attributes can only be changed. But initially, just think of them all as concepts.

1) Again, no. Usually you can just note that an entity can be created and can be altered. If there's something specially interesting about it, make a special note.

For instance, you wouldn't record all the ways of changing the style of a word. But you might give some attention to how to change the widths of table columns, because the learner has to discover it the hard way.

2) We've given a definition above: that an entity can be created or deleted, or has attributes that can be modified; that attributes cannot be created or deleted, but "attach themselves" to entities. But as we have also shown, sometimes things can usefully be treated as both. At early stages of analysis, the distinction is not important – just consider 'concepts'.

Summary

- What are the key misfits in the design of the e-commerce site?
- How would you use your insight to guide re-design?
- What were the most difficult things about using CASSM on this example?

Session 4: structural misfits

- Structural misfits are based on Green's Cognitive Dimensions (CDs)
- You have already unwittingly analysed for some of the CDs
 - E.g. **visibility** refers to the user's ability to view components easily.
 - **Abstractions** will already have been identified
 - whether they relate to the user's or the device's view

This is not a tutorial on Cognitive Dimensions, so don't expect much detail!

Abstractions: Abstractions include both hard domain abstractions (e.g. mathematics) and hard device abstractions (e.g. style sheets). For example (again in MS Word), understanding heading styles is necessary to be able to create a table of contents for a document. Device abstractions that the user had to work with in order to work effectively with the device (such as style sheets) are concepts the user must learn, and are therefore easy or difficult to learn according to how well they are represented at the interface.

Visibility: this refers to the user's ability to view components easily. This relates directly to the interface representation; concepts that are hidden, disguised, delayed or undiscoverable – all of which are coded as 'difficult' in the interface representation – are likely to cause difficulty in terms of visibility.

Provisionality: this is simply an attribute of some concept the user is working with. For example, when people are working with diaries, they often label events as provisional with additional notation such as '?' or '(tbc)'. If provisionality of a concept matters, in CASSM the question is a simple one: does the system provide an explicit representation of that?

Secondary notation: A secondary notation provides extra information besides the formal syntax. For example, the colour of a road sign can communicate further information like what kind of road you are on or what kind of place the sign directs you to. The usability of a secondary notation depends on the interface representation and how easy it is for the user to interpret these secondary aspects. For example if understanding the document structure is important for the user then the clarity of the interface representation will be of concern in the CASSM description; however, this aspect of usability pertains primarily to notations, rather than interactions.

Viscosity

- ‘Viscosity’ is resistance to change.
- We are aware of two kinds of viscosity:
 - **Repetition viscosity**: something that is conceptually simple requires many device actions
 - **Knock-on viscosity**: changing one thing causes side-effects that then need to be fixed too

1) Repetition viscosity: Changing attribute A of entity E, E(A), needs many actions if:

E(A) is not directly modifiable
 P(Q) affects E(A)
 P(Q) is modifiable
 E consists-of P

For example, we get repetition viscosity on figure numbers in a document because whenever a figure is added, deleted or moved, a range of figures need to be re-numbered one by one. Similarly, we get repetition viscosity for changing figure numbers from Arabic to Roman numerals (“number-type”). Stated more formally:

set-of-figures(number-type) is not directly modifiable
 figure(number-type) affects set-of-figures(number-type)
 figure(number-type) is modifiable
 set-of-figures consists-of figure

2) Knock-on viscosity: Changing E(A) has possible knock-on if:

E(A) is modifiable
 modifying E(A) affects P(Q)
 there is a domain constraint on P(Q)

The first example of repetition viscosity also suffers from knock-on viscosity because of the domain constraint that the figure numbers should be in numerical order through the document. Similarly, there is knock-on viscosity for inserting a new section in a document, but a powerful word-processor will typically include a feature to automatically re-number other sections, so in this case repetition viscosity is avoided.

Viscosity (2)

- Generate 3 examples of viscosity from your own experience, including at least one example of each type.
- Formalise them as per the definitions on the previous page.

Premature commitment

- The basic idea is that the user has to commit to specifying something earlier than is natural or easy. We have recognised three kinds:
 - **Non-modifiability premature commitment:** An attribute cannot be changed after it has been set.
 - **Abstraction-based premature commitment:** To make something easy to change later, conditions have to be set so that it can be changed via an *affects* relationship.
 - **Device-constraint premature commitment:** Setting an attribute of one entity constrains the way new instances of another entity can be created.

Non-modifiability premature commitment can be defined as follows:

E(A) is settable

E(A) is not modifiable

We have examples of this kind of premature commitment in JSketch – for example, in setting line width and colour.

Abstraction-based premature commitment is achieved by creating an abstraction — i.e. establishing a grouping of entities in some way for which an attribute can be changed simultaneously. This grouping will normally be implicit for the user, but the user will have to make it explicit in order to work with it, and will then have to set up consists-of relations within the system to define the groupings.

E has attribute A

Set-of-E consists-of E

Set-of-E has attribute A

Changing A for set-of-E causes A to change for all E such that (set-of-E consists-of E)

Examples of this kind of premature commitment include applying styles to paragraphs so that the appearance of all those paragraphs can be changed together.

Device-constraint premature commitment:

E(A) is settable

P(Q) is not settable

There is a device constraint between E(A) and P(Q)

It is possible to add more P's

One example of this is drawing a map on the back of an envelope: once the user has started drawing the map, each new instruction should be drawn in the appropriate direction on the envelope, but the envelope may not be big enough in that direction.

Premature commitment (2)

- Generate an example of each kind of premature commitment and formalise it according to the definitions on the previous page

Hidden dependencies

- Important links between concepts are not visible (or otherwise readily available to the user).
 - Spreadsheets contain many hidden dependencies, as changing a value or formula somewhere in a sheet can have unanticipated (and often untraceable) knock-on effects elsewhere in the sheet.

This is formalised simply, but not very elegantly:

Changing C affects D
The relationship is not visible

Hidden dependencies (2)

- Generate another (non-spreadsheet) example of hidden dependency
- Formalise it!

Relationships revisited

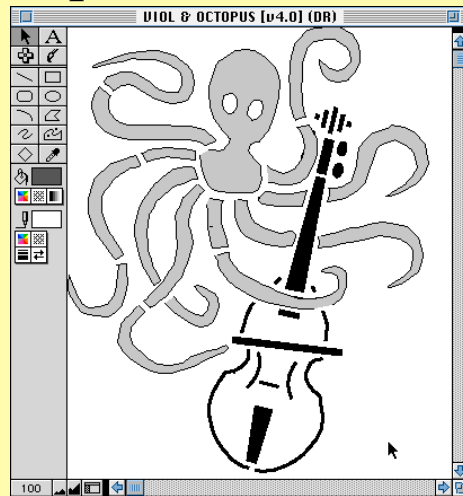
- It is in the definitions of structural misfits that relationships become central:
 - **Affects** and **consists-of** for repetition viscosity
 - **Affects** and **domain-constrains** for knock-on viscosity
 - **Affects** and **consists-of** for abstraction-based premature commitment
 - **Device-constrains** for device-constraint-based premature commitment
 - **Affects** (and visibility of relationship) for hidden dependency

Direct modifiability (or lack thereof) also plays a part in the definitions of several of the CDs.

Back to the drawing board

- Look back at the description you produced of the JSketch drawing tool. Extend your analysis of it to cover action-based misfits and structural misfits.
- Consider the more familiar (to most!) type of drawing tool illustrated on the next page. Identify further *structural* misfits for this tool.

An example of the use of Claris



OSM Interact tutorial

67

(Note: picture courtesy of Thomas Green)

In the manual, we find:

“To create a blank draw document: choose New from the File Menu, select Drawing and click OK [...] To draw objects ... select a drawing tool from the tool panel and drag the pointer in the document ... you create and arrange objects of various shapes, sizes and colours. When you overlap objects, the hidden part still exists. Every object has a border (which you can hide) and a fill (the area within the border)”

Consider what happens if you want to redraw the picture with (for example) the tentacles in different positions.

Consider where you might have started if you wanted to draw a picture like this.

What other changes might you want to make to a drawing that would cause user difficulties? Are they expressible in terms of structural misfits?

A note on structural misfits

- Structural misfits don't occur for e-commerce sites. Why not?
- List five kinds of systems for which structural misfits may pose usability difficulties.
- List five kinds of systems for which any misfit-related usability difficulties are likely to be surface-level.

Discussion

- What issues have arisen today that have not yet been resolved?
- What kinds of system are *you* responsible for? And would CASSM be helpful or not for analysing them?
- What are the outstanding questions about CASSM?

Thank you!

More information on CASSM can be
found at

<http://www.cs.ucl.ac.uk/annb/OSM.html>