

Chapter 10

Coherence

Ian Sommerville, University of St Andrews

Summary

Coherence is a systematic approach to carrying out field studies of work. It suggests that observations of work should be focused around three social viewpoints namely distributed coordination, plans and procedures and awareness of work as well as a number of cross-cutting social concerns namely paperwork and computer work, skill and the use of local knowledge, spatial and temporal organisation, and organizational memory. The UML should be used, wherever appropriate, to document fieldwork.

The chapter on Requirements and Design discusses more general issues of fieldwork and requirements engineering.

Background

The work on Coherence was proposed to provide guidance and a framework for people without experience or training of ethnographic fieldwork to carry out observations of work, organise the record of these observations and document them using a notation that is familiar to software engineers. Realistically, few organisations have access to trained ethnographers and the intention of Coherence is to allow software engineers to be directly involved in fieldwork and communicate their results to colleagues in a familiar notation.

The Coherence approach is primarily intended to support the process of requirements engineering and a possible outcome of applying the approach is

a set of use-cases that can be a starting point for more detailed analysis and design.

The starting point for the design of Coherence was PreView, a method for requirements engineering that incorporated the notion of viewpoints – different perspectives on the requirements and concerns. Concerns are cross-cutting system wide notions (such as reliability) that are relevant to all viewpoints. The PreView approach is based on decomposing concerns to questions then putting these questions to stakeholders to understand their requirements and constraints. Papers on PreView are included in the Appendix to this chapter.

Social viewpoints

Ideally, merging all of the requirements from all viewpoints should give you all of the requirements for a system but, in practice, this is rarely the case. There are usually system-wide requirements which are not associated with any viewpoint and which are often constraints or ‘shall not’ requirements (the system shall not allow access to unauthorised users e.g.). As we shall see, these system wide requirements are derived from so-called concerns.

The idea of social viewpoints is that these viewpoints capture the requirements that emerge from the notion that work is a social process that is embedded in an organisation with its own culture and ways of working. The social viewpoints that we propose come from our experience of fieldwork in a range of settings, where these viewpoints have been relevant in different places. The three social viewpoints in Coherence are:

1. **Distributed coordination** – the ways in which a team of people, who may be working in distributed locations, organise their work to allow it to be coordinated.
2. **Plans and procedures** – the formal model of work and the way that this is manifested in formal objects in the workplace. Also of interest is the ways in which work as it is practised, deviates from these formal expressions of the work.
3. **Awareness of work** – the ways in which people in the workplace make others aware of what they are doing and how they themselves become aware of the actions of others.

Social concerns

Concerns are cross-cutting notions that are applicable to all viewpoints (and to the system as a whole). The simplest way to understand what a concern is to think about characteristics of the system as a whole – such as reliability, safety, etc. (which is where PreView started). Social concerns are analysis issues that affect all viewpoints:

1. **Paperwork and computer work** – how are existing paper-based and computer-based systems used?
2. **Skill and the use of local knowledge** – to what extent is the operation of the system dependent on the skills and knowledge of individuals who are part of that system.
3. **Spatial and temporal organisation** – how is work organised in physical space and time?
4. **Organizational memory** – how are plans and procedures and coping strategies for errors remembered in an organisation

The key elements of these concerns is that they are the starting point for more detailed decomposition which eventually leads to a set of questions that have to be answered either by questioning the appropriate system stakeholders, from the system documentation or from observations of people using the system.

Question-driven analysis

Viewpoints and concerns steer the analysts attention towards social issues that we know are often important in analysing organisational systems. To gather information that is relevant to these social viewpoints and concerns, we have extended the question-driven approach in PreView. Generic and specific questions are identified and the answers to these questions provide information that can be the basis for identifying system requirements and associated UML use-cases.

Each viewpoint has an associated set of focus questions, which are explained in more detail in the papers included in the Appendix to this chapter. Examples of focus questions are:

1. **Distributed coordination**: How clear are the boundaries between one person's responsibilities and another's?

2. **Plans and procedures:** What happens when formal plans and procedures fail?
3. **Awareness of work:** How does the spatial organization of the workplace facilitate interaction between workers and with the objects that they use?

The analysis of the social concerns is also question-based. In this case, you start with a concern – say Paperwork and Computer Work and decompose this into sub-concerns – say, Use of paper, Use of web, Use of local files and, if necessary sub-sub concerns. You then identify a set of questions for each sub-concern that help gather the information you need. For example, for the Use of Web sub-concern, questions might be:

1. To what extent do users routinely consult web sites for information?
2. Are there ‘trusted’ web sites that are frequently used?
3. How do users share information about trusted web sites?
4. Are internal web sites used?

How you find the answers to these questions depends on the work being studied. Sometimes you consult documents, sometimes observe what people are actually doing and sometimes you can ask them directly. If you ask questions, however, you should check by observation that what people actually do is the same as what they say they do.

UML representation

The idea underlying Coherence was to use the UML, wherever appropriate, to represent the work being studied – so the UML models that are produced were models of the system as it is, rather than models of the system that is required. The three UML diagram types that are most useful are:

1. **Use cases**, to identify specific work activities
2. **Sequence diagrams**, to show the order of activities/sub-activities
3. **Object diagrams**, to represent objects in the workspace

The problem with UML modelling (which is a generic problem rather than specific to this case) is that the UML is really not good for modelling exceptions where the ways that the exception is handled depends on when it happens, where

it happens and who is available. We advocate simply using diagram annotations to handle exceptions rather than trying to create lots of exception use cases or to use conditional sequences, etc.

When we invented Coherence, we anticipated (correctly) that the UML would become the standard modelling notation but over-estimated the impact of the UML on practical software engineering. The advent of agile methods and minimal documentation has meant that many small to medium sized development projects don't develop system models.

Therefore, if you use Coherence as a framework for helping you understand the social nature of work, you may prefer to document your fieldwork in a less formal way, which can then be used in discussions about the system requirements.

Retrospective

We believe that the original ideas behind Coherence are still relevant and that the general problem of providing help and guidance to people who need to understand how social and organisational issues affect work remains.

Coherence was developed before the widespread use of the WWW to support work and, without doubt, could and should be evolved to take this into account. To be effectively used, tool support is probably necessary and whilst prototype tools were developed in a separate project for PreView, these were never extended to take social viewpoints and concerns into account.

Coherence provides a general framework for a process of social analysis but does not, in itself, outline a process that could be used by software engineers. There is a need for the Coherence approach to be developed so that much more guidance is provided for people who are getting started with the process. In particular, the distinctions in practice between social viewpoints and social concerns have to be clarified. The approach should also be revised to take recent development in the UML into account.