The Viewpoints FAQ

Anthony Finkelstein
City University, Department of Computer Science, Northampton Square, LONDON EC1V 0HB, UK
<acwf@cs.city.ac.uk>

Ian Sommerville
Department of Computing, Lancaster University, LANCASTER LA1 4YR, UK
<is@comp.lancs.ac.uk>

The structure of this brief paper follows an emerging convention - the FAQ - Frequently Asked Questions list. FAQs have grown out of Internet newgroups where participants, tired of seeing the same questions repeated by newcomers, provide a list of canned answers to the most frequently asked questions. An FAQ also provides a covert role in defusing tiresome or unduly acrimonious debates by summarising the arguments and pre-empting outbreaks of “flame wars”. This is our attempt, somewhat tongue-in-cheek, to do the same for viewpoints. The FAQ serves as our introduction to the theme of the special issue papers which follow.

1. What are viewpoints?

The construction of a complex description or model involves many agents (aka participants or actors). These agents have different perspectives or views of the artifact or system they are trying to describe or model (the domain of discourse). These perspectives or views are partial or incomplete descriptions which arise because of different responsibilities or roles assigned to the agents. These responsibilities or roles may be organisationally defined, follow some defined structuring of the underlying artifact or system, or may reflect different modelling or descriptive capabilities. The combination of the agent and the view that the agent holds is termed a viewpoint. The study of viewpoints embraces the relations between views, between views and agents, and between agents.

2. How do viewpoints relate to software engineering?

Viewpoints are a general feature of group work but crop up with particular frequency in software engineering. Software engineering is characterised by a concern with large systems displaying complex structure and with many interlocking constraints on their construction and behaviour. Working with such systems necessitates multiple viewpoints for complexity control and separation of concerns. During the specification of such systems viewpoints naturally arise out of differences of opinion, varying goals and mistakes or errors.

Work on viewpoints has commonly been linked to work on requirements engineering. Though the problem is particularly acute during the early stages of development, particularly elicitation where many diverse client views are prevalent, it is not restricted to it. Viewpoints are encountered throughout the process of system development. For example, many design methods have implicit viewpoints in that they suggest the creation of several system models such as a data-flow model, an entity-relation model, etc.

3. What do viewpoint-oriented methods do?

Conventional system development methods do not recognise the existence of viewpoints as user-definable entities. Instead they provide rigid structuring schemes and strictly control both the diversity of viewpoints and the relations between them. By contrast viewpoint-oriented methods make viewpoints first class objects so they can be defined by method users, relationships between them can be established, etc.

Though viewpoint-oriented methods differ considerably in scope they commonly provide a means of representing and managing the viewpoints that arise during system development and a framework or techniques for viewpoint integration/resolution. The objective of all viewpoint-oriented methods is to strike a balance between, on the one hand, preservation of multiple perspectives during system development and, on the other hand, the demands for consistency and coherence arising out of group work.

4. What is viewpoint integration/resolution?

The essential problem that viewpoints present is of consistency or coherence. Given that viewpoints may overlap (that is may refer to the same phenomena in the domain of discourse) if the agents have a shared goal or their goals are potentially interfering the consistency of their views must be established. This consistency need only be partial, that is sufficient to achieve the goals. Consistency can be achieved by integrating the viewpoints that is merging them or by locally resolving inconsistencies as they arise.

5. What are the practical problems in applying viewpoints in large projects?

There are really three problems which have to be addressed:

(a) choosing the right viewpoints model
(b) identifying viewpoints
(c) managing the information from viewpoints

The different models of viewpoints which have been developed are best suited to different activities in the software process. If your problems are mostly in requirements elicitation and abstract requirements definition, you could choose an approach such as that suggested by Leite (1989) or in the paper by Kotonya and Sommerville in this issue. If you are most concerned with detailed analysis and conflict resolution, an approach such as that proposed by Finkelstein et al. (1992) which is developed in the paper by Nuseibeh and Easterbrook in this issue might be more suitable. If you are most concerned with conventional development methods, then a viewpoints approach such as is used in CORE (Mullery, 1979) where viewpoints are sources or sinks of data may be most appropriate.

Choosing viewpoints is not easy. There are no simple answers to this question. You need to have a enough viewpoints to give you a sufficient diversity of perspective but not too many, otherwise you will generate an unmanageable amount of information. As in all methods, making the choice of fundamental abstractions requires knowledge of the application domain, the specific problem and the technologies which will be used in the development process.

As viewpoints allow you to collect system information from multiple perspectives, you will inevitably collect a lot of information which must be managed. This is both a strength and a weakness of the approach. It's a strength because you are much less likely to miss information which is critical to the success of the system; it is a weakness because, somehow, you have to manage this information and eliminate redundancy from it. Although some tools have been developed as research prototypes and for specific company use, we don't know of any currently available commercial tools in this area. Perhaps the best approach is to look at how you can adapt your existing CASE tools to manage viewpoint information.

6. What research work has been done on viewpoints within software engineering?

It is difficult to trace a clear intellectual heritage for current work on viewpoints within software engineering. There appear to be at least three interlocking strands: construction of specifications, notably building formal specifications incrementally; software engineering environments, particularly support for distributed group work; requirements specification and analysis methods. Each of these strands is reflected in the papers contained in the special issue.

The original work on viewpoints is attributable to CORE, a requirements engineering method based on the partition of a system into viewpoints each with an associated client authority. This method achieved reasonably widespread use and much subsequent work on viewpoints can be traced to its influence. Most of the early lines of research are still ongoing, and are reflected in this issue, others worthy of note are Robinson (1990) and Dubois et al. (1988). This is the first specific collection of work on viewpoints however continuing work has been reported in the requirements engineering track of successive International Workshops on Software Specification and Design. Sessions devoted to viewpoints took place at the International Symposium on Requirements Engineering.

7. What is happening at the cutting edge?

A number of research groups, loosely bound together by joint projects and other links, are working on viewpoints. Particular mention should be made of the Nature project (Jarke et al. 1993). Ongoing viewpoints research addresses a diversity of issues including the combination of goal-oriented methods and viewpoints; formal support for reasoning in the presence of inconsistency; the use of viewpoints as a conceptual tool in the specification and analysis of open distributed systems; formal models of viewpoints; viewpoints for organising domain models; the application of viewpoints for other types of analysis (e.g. safety analysis, ethnographic analysis); and in managing requirements traceability.

8. Are any other disciplines working on viewpoints?

Of course viewpoints are not solely a software engineering challenge. For a generic treatment, work on view integration, multi-databases and database interoperability (Litwin, Mark & Rossopoulos, 1990) is directly relevant. Computer supported cooperative work (CSCW) addresses the problems of viewpoints though the focus is largely at the organisational level (Johansen, 1989). Viewpoints are explicitly recognised in the work on open, distributed systems. Some work on conventional engineering design, within the broad framework of concurrent engineering, tackles the viewpoints problem (Klein, 1994). Research groups working in the area of distributed artificial intelligence are showing an interest in viewpoints related issues, though as yet few results are reported. Work on knowledge representation schemes specifically work on translating between different languages and defining interchange formats has connected with viewpoints (Delugach, 1992). In a very general sense work in the social sciences, particularly anthropology, can be said to yield useful insight into viewpoints at a societal level (Sperber, 1985).

9. How can I find out more?

To get started in reading around this area, we recommend the following papers which describe three different approaches to the use of viewpoints in software engineering. These papers all include extensive references and are available in readily accessible journals.

Leite, J.C.S.P. and Freeman, P.A. "Requirements Validation through Viewpoint Resolution", IEEE Trans. on Software Eng., 1991, 17 (12), 1253-69.

Kotonya, G. and Sommerville, I. "Viewpoints for Requirements Definition", IEE/BCS Software Eng. J., 1992, 7(6), 375-87.

Nusibeh, B., Kramer, J. and Finkelstein, A. "A Framework for Expressing the Relationships between Multiple Views in Requirements Specifications", IEEE Trans. on Software Eng., 1994, 20 (10), 760-73.

The editors and the authors of the papers in this issue can be contacted by email and would be pleased to answer questions or make suggestions for further reading.

References

Delugach, H.S. "Analyzing Multiple Views Of Software Requirements" [In] Conceptual Structures: Current Research and Practice, Eklund, P., Nagle, T., Nagle, J. & Gerholz, L. [Eds.], 1992, 391-410, Ellis Horwood.

Dubois, E., Hagelstein, J. and Rifaut A. "Formal Requirements Engineering with ERAE", Philips Journal of Research, 43 (3/4) 393-414.

Finkelstein, A., Kramer, J., Nuseibeh, B., Finkelstein, L. & Goedicke, M. "Viewpoints: a framework for integrating multiple perspectives in system development", International Journal of Software Engineering and Knowledge Engineering, 2 (1), 1992, 31-57.

Jarke, M., Bubenko, J., Rolland, C. and Sutcliffe, A. "Theories Underlying Requirements Engineering: an overview of NATURE at Genesis", Proc IEEE Int. Symposium on Requirements Eng., 1993, 19-33, IEEE CS Press.

Johansen, R. "User Approaches to Computer-Supported Teams", [In] Olson, M. [Ed.], Technological Support for Work Group Collaboration, 1989, LEA.

Klein, M. [Ed.] Concurrent Engineering (Research and Applications) Journal, Special Issue on Conflict Management in Concurrent Engineering, 1994, 2(3).

Leite, J.C.S.P. "Viewpoint Analysis: a case study", Proc. 5th Int. Workshop on Software Specification and Design (IWSSD-5), 1989, 111-119, IEEE CS Press.

Litwin, W., Mark, L. and Rossopoulos, N. "Interoperability of Multiple Autonomous Databases", ACM Computing Surveys, 1990, 22(3) 267-293.

Mullery, G. "CORE - a method for controlled requirements specification", Proc. 4th Int. Conf. Software Engineering, 1979, 126-135, IEEE CS Press.

Robinson, W. "Negotiation Behaviour During Requirements Specification", Proc. 12th International Conference on Software Engineering", 1990, 268-276, IEEE CS Press.

Sperber, D. "On Anthropological Knowledge", Cambridge Studies in Social Anthopology, 1985, CUP.