

# Viewpoints 96: International Workshop on Multiple Perspectives in Software Development

(SIGSOFT '96)

## Workshop Report

George Spanoudakis, Anthony Finkelstein & Wolfgang Emmerich  
Department of Computer Science,  
City University,  
Northampton Square, London EC1V 0HB, UK  
{ gespan | acwf | emmerich @cs.city.ac.uk }

### 1 Workshop motivation

This report has been compiled by the workshop organisers. We have attempted to be fair and unbiased but probably failed! A short report cannot do justice to the discussion. You are best off reading the papers or talking to an attendee. What follows is definitely second best.

The construction of a complex description or model in software development involves many agents (aka participants or actors). These agents have different perspectives or views of the artefact or system they are trying to describe or model (i.e. the domain of discourse), and the process of constructing it. Examples of such perspectives might be performance, security, architecture, and so on. These perspectives or views are partial and incomplete descriptions that arise because of different responsibilities or roles assigned to the agents. These responsibilities or roles may be organisationally defined, or follow some defined structuring of the underlying artifact, system or activity, or may reflect different modelling and descriptive capabilities. The combination of the agent and the view the agent holds is termed viewpoint.

The Viewpoints '96 workshop may be seen as an attempt to establish a joint understanding of the critical issues in the study of viewpoints, which embraces the relations between views, between views and agents, and between agents. The workshop brought together the main international strands of relevant work in software engineering. It also attracted a few scientists from groups working on similar problems in areas outside software engineering including databases and multi-agent decision making.

The proceedings of the workshop have been published by the ACM press and are also available from <http://www.cs.city.ac.uk/homes/gespan/vptoc.html>.

### 2 Workshop Themes

The workshop was articulated around the following themes:

- nature of viewpoints – definitions
- detection of inconsistencies & conflicts between viewpoints
- resolution of inconsistencies & conflicts between viewpoints
- special requirements of applications of viewpoints
- representation of viewpoints
- infrastructure required to support viewpoints
- relations with other areas of computer science
- issues and agenda for further research

These themes were discussed in a sequence of sessions and these discussions are summarised below.

### 3 Nature of viewpoints – definitions

Viewpoints are seen as a means for separating concerns in software development in accordance with a variety of criteria, such as the interests and perspectives of agents, the development strategies and methods, the notations employed, or the services provided by the system to various parties (motivated by related work in the area of databases). This separation of concerns is useful to the extent that aids in dealing with complexity, facilitates the elicitation of information, supports learning, and has a significant role in achieving quality.

Given the benefits of viewpoints two issues become critical. The first is the choice of different viewpoints (i.e. the decomposition problem), and the second is the management of "interference" between viewpoints, including ways to integrate or reconcile them (i.e. the composition problem). The numerous ways in which viewpoints vary, for example representation, levels of abstraction and formality, suggest the need for tools, techniques, methods and environments to support their owners in dealing with them. However, it is important to keep in mind that since viewpoints are tightly tied to people and their interactions, their management has a significant social aspect which should always be taken into account.

#### **4 Detection of inconsistencies & conflicts between viewpoints**

Different forms of inconsistencies and conflicts between viewpoints were identified during the workshop. All of them were acknowledged to stem from the same phenomenon, that is the existence of "ontological overlaps" between viewpoints. Viewpoints overlap to the extent they incorporate components that refer to the same entity in a domain of discourse. Overlaps introduce the potential for inconsistencies, which may be static (i.e. inconsistencies in descriptions of static artefacts) or dynamic (i.e. inconsistencies in descriptions of behaviours).

The static inconsistencies were further distinguished into:

*syntactic* – inconsistency of a description w.r.t the syntax of the language it is expressed in

*static semantic* – violation of semantic properties associated with particular constructs of the language which are used to express viewpoint components

*logical inconsistencies* – breach of some static consistency relation

*divergences* – potentially incompatible values of features defined in overlapping viewpoints

The dynamic inconsistencies were further distinguished into:

*deviations* – discrepancies between the actual actions that are executed in performing an activity and the description of the activity in some model

*logical inconsistencies* – breach of some dynamic consistency relation

All these kinds of inconsistencies are observed at the viewpoint level and may concern viewpoints of the same or different "types". In addition, we may have meta-inconsistencies, these are inconsistencies between the rules which determine consistency. The detection of inconsistencies is complicated by viewpoints which are expressed in different languages and may be incomplete. It may be based on a variety of techniques including theorem proving, human inspection, simulation, compiler-like checks (symbol tables) and meta-modelling. Inconsistency detection may be applied in an eager or a delayed mode depending on the significance of the type of inconsistency and the willingness of the viewpoint owners to activate the checks. Regardless of the technique and the time of detection, it is critical to maintain information about inconsistencies for traceability and to support systematic evolution and maintainability.

#### **5 Resolution of inconsistencies & conflicts between viewpoints**

The resolution of inconsistency is an inherently complex activity because it involves interactions between humans, it is affected by the interplay among requirement, design, implementation and political issues, and it requires a significant amount of domain knowledge. Iterations between inconsistency detection and resolution activities are necessary since the resolution of some conflicts may introduce other conflicts. In most of the cases, the viewpoint owners are the principal actors in resolution activities. However, it is envisaged that, in certain cases, a third party resolution may be more effective, especially if the conflict concerns sensitive political issues or high level goals.

Viewpoint owners need method, automated reasoning and tool support for resolving inconsistencies. The role of such support would be to generate, and present participants with alternative resolutions to choose from. A survey of relevant work, that was presented in the workshop, reviewed value and structure oriented methods, and transformational techniques. As expected, these techniques and methods tend to be inconsistency type specific. It was also pointed out that analogical reasoning and version control have potential as support for resolution activities.

#### **6 Special requirements of viewpoint applications**

The workshop considered two areas of viewpoints applications, namely software processes and workflow.

Separations of concerns from both technical and customer standpoints are natural and frequent in these areas. It was observed that, in these areas, viewpoints emerge as independently developed models rather than agent-tailored projections over a single common model, which in reality rarely exists as a concrete artefact on its own.

The detection of inconsistency, may benefit from the assessment of similarities, or equivalently discrepancies between viewpoints. Also, conflict resolution must maintain ownership information and be well documented, that is be explicit and include decisions and trade-offs.

## **7 Representation of viewpoints**

It is very difficult, if not impossible, to select a single appropriate representation scheme for viewpoints. This is because viewpoints need to be represented in languages which are comprehensible and readily useable by their owners, if they are to achieve their role as effective means of expressing their perspectives. However, at the same time, to the extent they interfere and their interference needs to be managed, they must also be amenable to various forms of reasoning and analysis. These forms of reasoning and analysis may not be feasible unless the viewpoints are expressed in specific languages, not necessarily the ones their owners understand and use. A further complication arises from the fact that the variety of relationships between viewpoints, which emerge from the activities of interference management (e.g. overlaps, inconsistencies, conflicts, resolutions) must be represented and maintained for traceability reasons.

The workshop considered various answers to this language impediment problem. One of them was the simultaneous maintenance of external (owner-oriented) and internal (analysis oriented) representations of viewpoints. Another solution would be to have heterogeneous representations of different viewpoints and mechanisms to export information in some common communication language to support different types of analysis. The workshop did not reach consensus on which is the best way to solve this problem.

## **8 Infrastructure**

An appropriate infrastructure for viewpoints must accommodate multiple viewpoints expressed in heterogeneous representations, computer tools to manipulate them, and mechanisms to manage their interference in an effective and readily extensible manner. The various approaches to the construction of such

infrastructures could be roughly distinguished into centralized vs. distributed and homogeneous vs. heterogeneous. The former distinction relates to the physical storage and maintenance of viewpoints, whilst the latter relates to their representation. These criteria are orthogonal, although there is some confusion because of the centralized approach which – in addition to a single central repository – usually assumes the expression of viewpoints in a common canonical representation. In the centralized approach, viewpoints are generated and accessed by various tools in user-tailored presentations which need to be transformed from and into the common canonical representation. Centralized, homogeneous infrastructures were criticized for the degree of flexibility and reliability they offer. At the other end of the spectrum, infrastructures for distributed and heterogeneous computing allow for separate storage, different representations and multiple tools but present some communication and coordination problems.

## **9 Relations with other areas of computer science**

Technologies and research results from other areas of computer science may offer solutions to some of the problems associated with viewpoints.

For instance, the programming languages technology offers some consistency checking (e.g. type checking and inference, signature matching) and conflict resolution mechanisms (e.g. composition by multiple inheritance, incremental compilation and linking mechanisms). It also provides infrastructure components, such as programming support tools and environments. Further work in the viewpoints area needs to consider the recent trends of subject and aspect oriented programming, which reflect customer and designer oriented separations of concerns in programming respectively.

Other areas of computer science that may contribute results to the study of viewpoints include the area of computer supported cooperative work (e.g. human factors and negotiation models for conflict detection and resolution activities), some areas in artificial intelligence (e.g. case-based and analogical reasoning for exploration of overlaps and conflicts), and the areas of decision support systems (conflict resolution) and information retrieval (exploration of overlaps).

## **10 Issues and agenda for further research**

Reviewing the discussions that took place during the workshop, the participants identified some issues which deserve further research. These issues include:

- the systematic acquisition of viewpoints (help people identify and express their views in reference to a problem)
- the interactions among viewpoints (understanding overlaps, shared phenomena and consistency relations)
- the coordination among viewpoint owners for various purposes including conflict resolution
- the management of viewpoint information (storage, retrieval, representation, proliferation handling), and
- the development and evolution of appropriate infrastructures to support viewpoints

## **11 Conclusions**

A number of research groups in academia and, recently, in industry are concerned with how to support multiple perspectives in software development. The Viewpoints '96 workshop offered a forum for common discussions and sharing of experience among these groups. Rather than focusing on specific pieces of work, it considered the general problems of multiple perspective software development and tried to identify the critical open issues associated with these problems. As a result various promising research directions emerged. In retrospect, the workshop was organised at about the right time for a reconceptualisation of the area, revealed a significant community of interest and provided a useful basis for exchanging ideas.