Managing Interference

Anthony Finkelstein, George Spanoudakis & David Till
Department of Computer Science,
City University
Northampton Square, London EC1V 0HB, UK
email: {acwf | gespan| till@cs.city.ac.uk}

Abstract

The construction of a complex software system involves many agents with different perspectives or views of the system they are trying to describe or model. This gives rise to many partial specifications – viewpoints. These viewpoints may "interfere" with each other that is the goals of the agents may be mutually interdependent. This interference is inevitable and acceptable in system development. In this paper we examine how interference can be "managed" and the tasks that this entails. We summarise ongoing research and suggest new research directions.

1. Introduction

The construction of a complex software system involves many agents (*aka* participants or actors). These agents have different perspectives or views on the artifact or system they are trying to describe or model and different goals with respect to it. This gives rise to many partial specifications – viewpoints – reflecting those perspectives [1, 2]. These viewpoints may "interfere" that is the goals of the agents may be mutually interdependent. This is a particular feature of the requirements engineering setting. In §2 we outline the various forms of interference and describe how they manifest themselves.

We believe that interference is inevitable and acceptable in system development [3]. It is inevitable, as a consequence of multiple perspectives, and it is acceptable as a support for innovative thinking, deferment of commitments and exploration of alternatives. The consequence of this stance is that interference between specifications needs to be "managed". An outline of interference management is presented in §3.

Interference management is complicated by viewpoints which might be held by different owners; might use different languages; might be at different degrees of abstraction, granularity and formality; might deploy different terminologies; might be at different stages of development or elaboration. In §4 we describe how these factors impact interference management approaches.

The complexities alluded to above, set alongside the normal software engineering problems of scale, suggest the need for automated reasoning and method support for interference management. In §5 we provide a brief summary of our research progress in this area and outline new research directions.

2. Interference

Terminology is always dangerous territory, hence we present our working definitions of the key concepts. *Interference*, defined above, is the general term we have adopted for the phenomena that are of interest to us.

Viewpoints *overlap* if they incorporate components, which refer to common aspects of the system under development and its domain. For example, the same object in the domain of discourse might be referred to by different components of different viewpoints. Overlap is a prerequisite for the other forms of interference.

The existence of an overlap implies a *consistency relation* between the viewpoints. An *inconsistency* is a breach of this relation. This generally manifests itself as a logical inconsistency, the assertion of a fact and its negation for a pair of overlapping components. Inconsistencies may be hidden because of the necessary incompleteness of specification or because a consistency relation has not been identified. We term such hidden inconsistency, *incompatibility.*

Different viewpoints are commonly expressed using different representation schemes. A representation scheme embeds a theory of the domain of discourse, of the phenomena of interest in it and of the relationship between these phenomena. Thus the consistency relations *embedded* in the representation scheme (sometimes referred to as its static semantics) are a statement of overlaps as seen by that representation scheme. To relate different representation schemes it is necessary to understand the overlaps between them, in other words to relate the theories. Pragmatically, overlaps between components of viewpoints may be identified which are not reflected in the representation scheme, that is the theory is inadequate. The long-term solution to this is to enrich the representation scheme by adding further concepts or relating it to another scheme. In the short-term it may be necessary to add *on-the-fly* consistency relations.

3. Interference Management

Interference management is the process by which interference is handled so as to support the goals of the agents concerned. It comprises the following significant activities.

Overlap identification – Recognising those components that refer common objects or phenomena in the domain of discourse. This activity necessarily requires human intervention.

Consistency relation construction – Building the relationships between the different viewpoints to reflect the overlap. These may embedded or on-the-fly consistency relations.

Policy specification – Identifying and representing an appropriate policy for interference management. Such a policy specifies when consistency relations should be checked and the actions consequent on the checks. Policy can be specified in terms of high level strategies or goals. We loosely distinguish between *preventive policies*, involving the immediate rejection of an action whose completion would cause the occurrence of an interference and which are appropriate if further actions upon viewpoints would be affected by an unresolved interference, and *remedial policies* in which interference trigger resolution actions. Remedial policies are tempered by *toleration policies* which define the scope and extent of toleration of interference. Preventive policies parallel the traditional (or unitary) perspective on conflict in studies of organisational behaviour [4]. In this view conflict is a malfunction within an organisation or a group and as such it should be avoided. By contrast remedial policies reflect the behavioural perspective in which conflicts are the natural result of individuals and groups each pursuing their own interests and objectives within an organisation, and in which management has to achieve compromises which resolve them.

Policy application – Monitoring progress with respect to the policy and performing actions to achieve the policy goals.

Detection – Checking for consistency through the application of the consistency relations and detecting inconsistencies.

Analysis or diagnosis – Providing analytic or diagnostic feedback on the consistency of the viewpoints beyond simply the presence of an inconsistency.

Tracking – Tracing inconsistencies and preserving analytical or diagnostic information in the face of ongoing changes.

Resolution – Settling the disagreement underlying an inconsistency through an agreed *rectification* or, possibly, an *amelioration* which improves matters by providing a partial settlement that reduces the space of disagreement.

Rationale provision – Recording information gained through the processes of interference management, decisions reached or deferred and the reasoning or argumentation underpinning them.

Any or all of the activities identified above should ideally be capable of being "safely" performed in the presence of inconsistency or suspected inconsistency.

4 Complicating Factors

Many of the activities identified above are amenable to tool and method support. Some progress has been made in developing techniques that partially cover these activities. However, these techniques are critically dependent on three factors – the granularity of the interference; the specific representation schemes that are deployed by the viewpoints; and, the agents that are party to the process and their capabilities.

Granularity: Overlap, inconsistency and incompatibility may arise within a single viewpoint *(intra-viewpoint interference)* or among different viewpoints *(inter-viewpoint interference)*. Dealing with inter-viewpoint interference is more difficult than dealing with intra-viewpoint interference because different viewpoints may have been expressed using different representation schemes, may be at different levels of abstraction, elaboration and formality and may be the responsibility of different agents – *owners*..

Language specificity: Various interference management techniques can only be applied to viewpoints expressed in specific languages. For instance in [3] we describe an inconsistency detection technique which is based on theorem proving and therefore applicable only to viewpoints expressed in a first order language. Another example is the technique we present in [5] for detecting ontological overlaps, which applies to viewpoints expressed in a particular class of semantic modelling and object oriented languages and requires that components of the viewpoints are classified with respect to a specific meta-model. Although language independence is a desirable feature, since it makes techniques applicable to a wider spectrum of settings, it is hard to achieve. This is especially for techniques which are based on underlying computational reasoning mechanisms.

Involvement: Interference management techniques may be *participatory* that is requiring the cooperation of the viewpoint owners, or *autonomous,* in which the viewpoint owners play no part in the application of the technique. The selection of appropriate techniques depends upon the capabilities and responsibilities of the agents that are involved and the extent of the shared knowledge about the domain that can be relied on.

5 Research

We are interested in developing a toolkit to support many of the interference management activities described above. We would like these tools to work on a range of, primarily graphical, formalisms widely used for software specification. We hope that we will be able to generalise from techniques applied to formal representations in order to provide tools to support people working with (semi-structured) natural language texts as part of large document sets. In this context we are seeking techniques that would support the identification of overlaps, detect gross inconsistencies, track these inconsistencies, perform meaningful diagnoses and help document rationale.

Our current research has two strands. We are attempting to develop a framework which will help us understand the relations between the concepts, activities and factors briefly discussed above. The whole area of interference seems to us ripe for reconceptualisation and for a synthesis of significant related work. In this regard our aims closely align with those of an informal group of research teams spanning the work of [6,7,8,9]. We are working on some specific techniques which might form the first components of the toolkit, described below.

Reconciliation is a technique which supports the detection, verification and tracking of ontological overlaps. It has two basic stages — analysis and revision. It detects ontological overlaps using a computational model of similarity and a classification of specification components with respect to a meta-model of domain-independent, semantic modelling properties — analysis. It also supports the remodelling of specification components so that the results of similarity analysis and viewpoint owners assessment of overlaps converge — revision. The goal of this process is to ensure that the modelling of specifications is consistent with the human assessment of ontological overlaps between them and

establish a shared understanding among viewpoint owners of the potential for inconsistency. This work is described in [10]
In a software engineering setting viewpoint owners are party to a development process which defines progress and prescribes critical coordination points. We use a model of the software development process, enacted in a decentralised manner, to guide or enforce the application of the consistency relations and other interference management activities. In this approach the software process model embeds the interference management policy. This work is described in [11].

A related interest to that of viewpoints within our research programme is requirements traceability [12]. We have developed a mark-up scheme which annotates specifications with details of the contributors to that specification and their roles with respect to it. This mark-up scheme, when combined with knowledge about the relations between elements of the specification allows a detailed picture to be built up of the responsibilities and commitments of those contributors, which can be used to support requirements change and evolution. This work is described in [13]
We are currently extending the framework to annotate viewpoints and associated consistency relations with contributor and ownership information. This information can then be used to identify the participants in interference management activities such as resolution through negotiation. A richer notion of contributor and owner will allow us to map viewpoints onto the complex organisational structures that underpin software development.

We are collaborating with [14] to provide formal, logically based, schemes for reasoning and analysis of inconsistent information. We see this work as contributing to both the practical research strand, primarily through the provision of diagnostic tools, and to the framework, through a better formal understanding of inconsistency.

Moving from individual tools to an integrated toolkit implies an environment to support viewpoints and related information. We have until this point used The Viewer [15] a sketch of a proposed successor which will provide us with a better basis for exploring the critical issues can be found in [16]. Another concern is with the scaleability of approaches to interference management an area in which we have, as yet, little experience.

Acknowledgements

The work reported in this paper has been supported by the EPSRC through the Voila project. We are grateful to the British Council for their support for cooperation with Universitaet Essen and Politechnico di Milano. Thanks to Jeff Kramer for many useful discussions.

References

[1] 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), 31-58.

[2] Maiden N., Assenova P., Jarke M., Spanoudakis G. et al. Computational Mechanisms for Distributed Requirements Engineering, Proceedings of the 7th International Conference on Software Engineering & Knowledge Engineering (SEKE '95), Pitsburg, Maryland, USA, pp. 8-16,June 1995

[3] Finkelstein, A., Gabbay, D., Hunter, A., Kramer, J., & Nuseibeh, B., "Inconsistency Handling In Multi-Perspective Specifications" IEEE Transactions on Software Engineering, 20, 8, (1994), 569-578.

[4] Huczynski A., Buchanan D., Organisational Behaviour: an introductory text, Prentice Hall International Ltd, (1991), ISBN 0-13-639899-5.

[5] Spanoudakis G., Constantopoulos P., "Integrating Specifications: A Similarity Reasoning Approach" Automated Software Engineering Journal, 2, 4, (1995), 311-342.

[6] Nuseibeh, B., "Towards a Framework for Managing Inconsistency Between Multiple Views" Proceedings Viewpoints 96: International Workshop on Multi-Perspective Software Development, ACM Press, (1996), This Volume.

[7] Piwetz, C. & Goedicke, M., "A Module Concept for ViewPoints" Proceedings Viewpoints 96: International Workshop on Multi-Perspective Software Development, ACM Press, (1996), This Volume.

[8] Cugola, G., Di Nitto, E., Fuggetta, A. & Ghezzi, C., "A Framework for Formalizing Inconsistencies and Deviations in Human-centered Systems" ACM Transactions on Software Engineering and Methodology (to appear, 1996).

[9] van Lamsweerde, A., "Divergent Views in Goal-Driven Requirements Engineering" Proceedings Viewpoints 96: International Workshop on Multi-Perspective Software Development, ACM Press, (1996), This Volume.

[10] Spanoudakis, G., & Finkelstein, A., "Reconciling Requirements: a method for managing interference, inconsistency and conflict" Annals of Software Engineering (to appear, Special Issue on Software Requirements Engineering, 1996)

[11] Leonhardt, U., Finkelstein, A., Kramer, J., & Nuseibeh, B., "Decentralised Process Enactment in a Multi-Perspective Development Environment" Proceedings of 17th International Conference on Software Engineering (ICSE-17), Seattle, Washington, USA, 24-28th April 1995, IEEE CS Press.

[12] Gotel, O., & Finkelstein, A. "An Analysis of the Requirements Traceability Problem" Proceedings of International Conference on Requirements Engineering 1994, IEEE CS Press, (1994), 94-101.

[13] Gotel, O., & Finkelstein, A., "Contribution Structures" Proceedings of 2nd International Symposium on Requirements Engineering RE95, IEEE CS Press, (1995), 100-107.

[14] Hunter, A. & Nuseibeh, B., "Managing Inconsistent Specifications: reasoning, analysis and action" Department of Computing Technical Report Number 95/15, (October 1995), Imperial College, London, UK.

[15] Nuseibeh, B., & Finkelstein, A., "Viewpoints: a vehicle for method and tool integration" Proc. 5th International Workshop on CASE - CASE 92, IEEE CS Press, (1992), 50-60.

[16] Emmerich, W., "An Architecture for Viewpoint Environments based on OMG/CORBA" Proceedings Viewpoints 96: International Workshop on Multi-Perspective Software Development, ACM Press, (1996), This Volume.