Managing Process Inconsistency Using Viewpoints

Ian Sommerville, *Member, IEEE Computer Society*,
Pete Sawyer, *Member, IEEE Computer Society*, and
Stephen Viller, *Member, IEEE Computer Society*

Abstract—This paper discusses the notion of process inconsistency and suggests that inconsistencies in software processes are inevitable and sometimes desirable. We present an approach to process analysis that helps discover different perceptions of a software process and that supports the discovery of process inconsistencies and process improvements stimulated by these inconsistencies. By analogy with viewpoints for requirements engineering that allow multiple perspectives on a software system specification to be managed, we have developed the notion of process viewpoints that provide multiperspective descriptions of software processes. A process viewpoint includes a statement of focus or "world-view," a set of sources of process information, a process description and a set of organizational concerns that represent goals or constraints on the process analysis. We present a description and rationale of process viewpoints, discuss the process of applying process viewpoints for process understanding and illustrate the overall approach using part of a case study drawn from industrial processes that are part of a safety-critical system development.

Index Terms—Viewpoints, software process, process improvement.

1 Introduction

Over the past few years, there has been a growing awareness that organizations can leverage significant productivity improvements by improving their business processes. Software development processes are one class of business process that have received particular attention and, since the early '80s, a significant body of research into software processes and software process improvement has developed.

Very roughly, this research can be split into two main categories:

1. Fundamental research concerned with process modeling, enactment, and support technology. This involves developing notations for representing processes and looking at how some automated support for these processes can be provided. Good summaries of research in this area are provided by [1], [2].
2. Research concerned with process improvement, that is, the introduction of process changes to improve software productivity, quality, etc. This is related to general work on business process reengineering [3] although software process improvement programs tend to be evolutionary rather than revolutionary. The best known work in this area is that of the Software Engineering Institute and their Capability Maturity model [4], [5] but other related work on maturity models (Bootstrap and SPICE) for process improvement has also been carried out [6], [7].

To a large extent, these different aspects of process R&D have been parallel research streams. The fundamental process research has not really addressed the issues of how the research results can be applied to facilitate change and improvement. The process improvement work has taken a pragmatic approach to process description and is not dependent on structured or formal process notations. Proponents of this approach suggest that automation is not central to process improvements. Indeed, the SEI Capability Maturity model discourages automation in the early stages of improvement. Its developers believe that process enactment support is unlikely to add significant value to managers or developers until a disciplined and repeatable process has been put in place.

Process improvement frameworks such as the Capability Maturity model, the Bootstrap model, and the SPICE model are based on improving process maturity by introducing good engineering practice. They are designed to be independent of the current process and this is both their strength and their weakness. It is a strength as it allows the frameworks to be generic; it is a weakness as it does not support improvements based on good features of the existing processes that lie outside the inprovement model.

While we are generally supportive of process improvement frameworks, we believe that process improvement is more likely to be accepted if we understand existing processes and evolve these processes in ways that maintain their strengths. As discussed next, these processes are often inconsistent so to understand them we must understand this inconsistency. To help with this understanding, we have developed a framework for managing process descriptions and an associated process analysis process that reveals the process inconsistencies and uses them to identify potential process improvements. Our work, therefore, supplements

• *The authors are with the Computing Department, Lancaster University, Lancaster LA1 4YR UK. E-mail: {is, sawyer, viller}@comp.lancs.ac.uk.*

Manuscript received 15 Sept. 1997; revised 25 July. 1998.
Recommended for acceptance by C. Ghezzi and B. Nuseibeh.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 109060.

0098-5589/99$10.00/99/$10.00 © 1999 IEEE

and is complementary to process improvement frameworks and does not overlap with them.

The work that we describe in this paper was carried out in a project that was specifically aimed at discovering routes to requirements engineering process improvement [8]. It is part of a spectrum of work to assist process improvement including the development of methods for human factors analysis of processes [9], the development of good practice guidelines for requirements engineering processes [8], and the development of a practical viewpoint-oriented approach to requirements engineering [10], [11].

In the remainder of the paper, we discuss different types of process inconsistency and how an understanding of inconsistency can lead to process improvements. We then go on to describe our viewpoint-based process analysis framework and the process of applying viewpoints for process analysis. Finally, we present a small case study of using the approach for the analysis of process descriptions in a safety-critical system, illustrate process inconsistencies that were discovered, reflect on the lessons learned, and draw some conclusions about our work.

1.1 Process Inconsistency

There are two opposing sets of beliefs about processes. One belief is that processes are ways of describing how work *should be* done. A process model or process description is a representation of this ideal process. Process inconsistency is seen as a discrepancy in these process descriptions or a discrepancy between these process descriptions and the actual process. Process improvement is achieved by modifying the actual process to make it more prescriptive and compliant with this ideal process. Inconsistency is undesirable and the rationale for identifying inconsistency is to remove it from process descriptions. This belief is widespread in the software engineering community and is exemplified by research concerned with process programing and process enactment support tools [12], [13], [14] and with process changes through the introduction of good practice [15].

An alternative belief, which is the norm among those interested in the sociology of work, is that processes are ways of describing how work *is* done. A process description or model is not a description of an ideal process with some gap between that process model and reality but is simply a description of the work as it is actually carried out by the people involved in the process. There is, therefore, no distinction between a process and a process description. Process inconsistency, in this view, means that different actors reach the same goal in different ways. Process improvement is achieved through identifying good practice, relating it to the existing process and convincing actors in the process to change the way that they work and adopt that practice. From this social perspective, process models may have an educational role but they should not be prescriptive. Prescribing processes to remove inconsistencies can lead to reduced process efficiency because the people involved feel that their professionalism is being undermined [16].

We have been influenced by both of the above opinions about processes. A standardized process model is helpful to facilitate communications about processes and to get started on activities. Such a model must define key points where the documented and actual processes must be in line with each other. The number and extent of these varies depending on the type of application being developed and the size of the development team. In a research environment, there need only be a few areas of correspondence between defined and actual processes; in the development of a large, critical system there should be much less divergence between the published and the actual process. However, any practical process must allow scope for professional judgment and different working practices.

The above perspectives highlight the problems of arriving at a simple definition of process inconsistency. From a formal perspective, inconsistency can be considered as some kind of logical contradiction so, with formally defined process models, inconsistencies can be detected by mathematical analysis. Some specific work on process inconsistencies, based on formal process models, has been reported by Cugola et al. [17]. In this work, they define formal models of both human-centered processes and process-support systems. The models that they propose are state-based and they introduce the notion of a deviation (an event that leads to an inconsistent state) and an inconsistent state which is a state that is not a member of the inconsistent states defined by their model. There can be inconsistencies in the model of the human-centered system (which they suggest may be tolerable) and between the model of the human-centered system and the process support system.

However, mathematical models of processes rarely reflect the reality of everyday work. If processes are considered from a social and organizational perspective, it is much more difficult to define process inconsistency in a precise and unambiguous way. Inconsistencies are differences of various kinds in the ways in that different people perceive or execute a process. For example:

1. *Inconsistencies in process execution.* Many tasks require software engineers to exercise professional judgment and experience. Different people may do the same thing in different ways, all of which may be effective. For example, one requirements engineer may elicit information about requirements by defining structured system models and discussing these models with end-users. Another engineer may prefer to develop a prototype system and work through this with end-users. Each engineer uses their professional judgment to decide which is the best approach in a particular context.
2. *Inconsistencies in process organization.* Different processes with the same objectives may have developed in different parts of the organization as a consequence of local factors such as tool and equipment availability, management style, etc. For example, configuration management processes in the same company may differ because some engineers use Unix workstations and some use PC-based systems for software development. Here the inconsistency is not a matter of professional judgment but of managerial decisions that may be historical or that may take into account local organizational factors

such as the experience of the workforce or budgets for tool procurement.

3. *Inconsistencies of process perception.* Depending on their particular interests, different people may focus on different aspects of a process. For example, a software engineer may be mostly concerned with process activities that are involved in the production of the system whereas a project manager is concerned with the scheduling and resources used by activities identified in the project plan. There may be a mismatch between the project plan activities (the manager's perspective) and the actual activities involved in developing the software (the engineer's perspective).

The above inconsistencies are not mutually exclusive and may coexist. Furthermore, the meaning of these inconsistencies depends on the particular process perspective that is adopted. For example, from an engineering perspective, inconsistencies in process execution reflect either an inadequate underlying process that does not include descriptions of possible alternative approaches or inconsistencies in the process description. From a sociological perspective, however, inconsistencies in process execution are fundamental to the notion of professionalism and reflect a process strength rather than a process weakness. From an engineering perspective, inconsistencies in process organization reflect the failure of an organization to standardize its processes; from a sociological perspective, they reflect the ability of the organization to adapt its processes to local circumstances.

A fundamental assumption underlying our work is that it is important to understand process inconsistencies in actual processes because these inconsistencies may point to potential process improvements:

- Areas of inconsistency may be accidental in that there are no good reasons for them; reconciling these inconsistencies may lead to more efficient process enactment.
- Areas of inconsistency may suggest process redundancy where activities are unnecessary. Fully or partially removing this redundancy may lead to process improvements.
- Areas of inconsistency can highlight particularly good practice. Where different processes for accomplishing some goal have evolved, the most effective of these processes may be selected for widespread dissemination across the organization.

We consider process inconsistency management to be the activities involved in eliciting process details, analyzing and identifying process inconsistencies, and "resolving" these inconsistencies. Resolving inconsistencies does not mean that these inconsistencies are necessarily removed from the different process perspectives or descriptions of the process. Where appropriate, there may be process modifications to remove the inconsistency but, in other cases, all that may be necessary is to flag the inconsistencies and ensure that the reason for their existence is understood by the process participants.

2 Process Viewpoints

Most research on process modeling notations [18] has been based on the assumption that a single model of the process can be elicited and agreed upon by all process participants. As we have already discussed, we do not believe that this deterministic perspective reflects the reality of software processes. Nor do we believe that it necessarily a goal that will lead to more effective software processes. Our experience from working with industry on process studies suggested that:

- Different people involved in the "same" process think of and describe that process in completely different ways. These differences are so significant that there is no practical way of combining the different descriptions into a single description or model except at a very abstract level. Attempting to integrate detailed process descriptions to produce a shared model results in a model of such complexity that it is unrecognizable to the process participants and is effectively useless for analysis.
- There is no single notation that all process participants and stakeholders are familiar and comfortable with. People want to describe "their" process in their own way and imposing some, perhaps formal, notation on this is counterproductive. We believe that there is a lot to be said for Checkland's "rich pictures" [19] where processes are described using representations that are meaningful to the people involved in the process.

Viewpoints, first proposed by Mullery for requirements engineering [20] are a technique for organizing and structuring partial descriptions of a system. Our previous experience of requirements engineering viewpoints [21], [22] suggested that they might be an appropriate framework for encapsulating process perspectives. We decided to adapt the approach to requirements viewpoints that we had developed [10] for process representation. This approach to requirements engineering was explicitly designed for application in a range of industrial applications [11]. It allows for different types of viewpoint (end-user, stakeholder, domain) to be accommodated within a single generic framework and provides a mechanism whereby business goals and constraints drive the requirements engineering process.

Process viewpoints are an approach to inconsistency management that support the activities of process elicitation, analysis, and understanding. They encapsulate process information elicited from different sources and allow the process to be described in notations that are familiar to the sources of the viewpoint information.

Our initial process viewpoint model was based on our requirements viewpoint model [23] but found that this was not entirely workable in practice. We, therefore, derived a simpler model of a process viewpoints with five components as follows:

$$PV = < \text{name, concerns, focus, sources, process description} >$$

Briefly, the *name* identifies the viewpoint, the *concerns* reflect business goals and constraints, the *focus* sets out the particular viewpoint perspective, the *sources* document the source of process information and the *process description* is a description of the process from the perspective defined in the viewpoint focus.

A very simple example of a process viewpoint is shown in Fig. 1. This is a viewpoint on a requirements review process. For simplicity, we have not actually included the process description here.

We must emphasize that process viewpoints are not projections of some underlying, definitive process and that merging the process descriptions in the different process viewpoints does not lead to a global model of the process. As discussed, a fundamental assumption underlying our work is that it is practically impossible to produce a single, detailed process model that accurately reflects the real, complex processes in which participants are involved.

2.1 Viewpoint Naming

The name of the process viewpoint is a meaningful identifier that should reflect the process perspective documented in that viewpoint. For example:

- The name of a role or department in an organization such as "configuration management," "quality assurance," "customer," etc. This implies that the process description will focus on the process activities, inputs and outputs that are most important to that department or role.
- The name of a process characteristic that is of particular interest. This can either be a functional characteristic such as "process activities," "roles and actions," or can be a nonfunctional process attribute such as "repeatability," "performance," etc. The particular process characteristic may be reflected in the notation used for the process description. For example, in a "process activities" viewpoint, a Petri net based notation may be used; in a "performance" viewpoint, the process may be represented using bar charts.

2.2 Concerns and Concern Decomposition

Process improvement should always be driven by the needs of the organization enacting the process. To allow for this, we have introduced the notion of viewpoint concerns that reflect organizational goals, needs, priorities, constraints, etc. Concerns are not just another type of viewpoint. A viewpoint is an encapsulated process description; a concern relates the process description to the business needs of the organization enacting the process. Concerns cut across viewpoints and drive the process analysis so that proposed process changes and improvements contribute to the real needs of the business.

We have identified several different classes of concern:

- *Understanding concerns.* These reflect the organization's objectives for process understanding. The organization may wish to understand a process to discover its relationships with other organizational processes, to define the process in a quality plan, to analyze the process for improvements, etc. Where an organization has immature processes, understanding these processes is the first step toward process definition and, ultimately, improvement.
- *Improvement concerns.* These reflect the objectives of the organization as far as process improvement is concerned. At a very abstract level, these may be reduced time to process completion, reduced process costs, etc. However, as we discuss below, these have to be decomposed into realistically achievable goals.
- *Constraint concerns.* These are organizational constraints placed on the process or on the process improvement activity. They may limit the analysis or possible process improvements.

Concerns are similar to goals in Basili and Rombach's GQM approach to process measurement [24]. In practice, therefore, concerns are decomposed into a set of questions that are put to process sources. Therefore, if an understanding concern is *process definition*, then this may be translated into an abstract question:

"What are our requirements engineering process activities."

If an improvement concern is process cost reduction, the most abstract question becomes:

"How can the costs of the process be reduced."

However, these questions are so general and abstract that they are not particularly useful for eliciting process details. Therefore, concerns are decomposed into subconcerns and, ultimately, into a specific question list which may be put to viewpoint sources.

Name	Quality management
Concerns	Time to market, product defects
Focus	The requirements review process and how overall system quality may be influenced by that process. Product defects are being introduced as a result of requirements errors. Product development schedules are longer than they should be because of the need to detect and remove these errors.
Sources	project managers, quality managers, company standards,
Process description	*A description of the review process including inputs, outputs, activities, process participants and commentary on the process and its influence on system quality.*

Fig. 1. A process viewpoint.

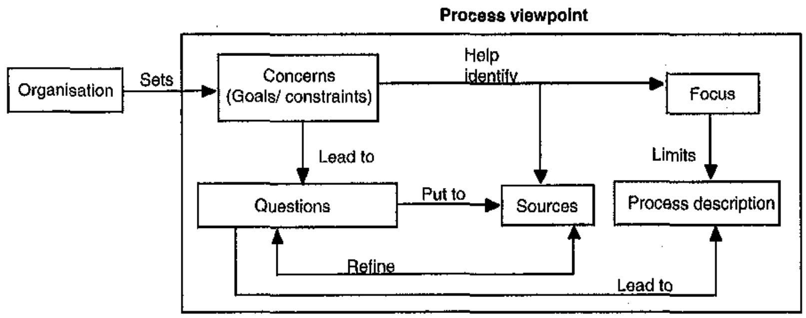

Fig. 2. Decomposition of a cost reduction concern.

As an illustration of this, Fig. 2 shows how the improvement concern to reduce costs might be decomposed. Notice that improvement concerns will almost always decompose to a mixture of understanding subconcerns (what are we doing now?) and improvement subconcerns (how can we do it better?). It is artificial to try to separate these as knowledge that can contribute to improvement emerges naturally as understanding of the process is developed.

Specific questions may be associated with nodes in this decomposition. In this example, we may have:

Q1. What are the types of resource used in each activity?

Q2. How much of each resource is used in each activity?

Q3. Does the output from the activity justify the resource input?

Q4. Are there comparable activities that use disproportionate amounts of some resource?

The formulation of questions helps identify the level of detail that should be included in a process description. In essence, the detail in the description must allow these questions to be answered. If the process description is too abstract, the questions cannot be answered and if it is too detailed, the questions are lost in that detail. Therefore, several attempts at the process model are likely to be required with the questions used to test the detail of the model.

For the above set of questions associated with the cost reduction concern, the process description must identify the activities for which resource utilization information is available. The costs of these activities can then be computed before and after the process improvement has been implemented. If this information is not available, process improvements cannot be evaluated. In such a situation, attempts at process improvement do not make sense until process data collection has been implemented.

As well as understanding and improvement concerns, organizations also place constraints on both the process improvement process and the possible improvement proposals. These constraints may also be expressed as concerns and decomposed into questions that must be addressed during the stages of process analysis and improvement suggestion. Examples of concerns that are constraints might be:

1. *Budget.* The budget available to the process improvement team is $XXXX.
2. *Existing tools and standards.* Existing design notations such as SADT must be used to describe the system requirements.
3. *Training.* Proposed improvements should require no more than Y days of additional training time per team member (or alternately Z days across the whole team).

These concerns are process requirements and process improvements that are proposed must be validated against them.

2.3 Viewpoint Focus

A viewpoint's focus is a succinct description of the viewpoint's perspective on the process. It, therefore, adds to the information implied by the viewpoint name. The focus description should normally identify the subprocesses that are of interest to that viewpoint. It may also include a statement of the organizational functions that are significant in the analysis of a process, a statement of the role of viewpoint sources or a statement of the particular type of model that will be presented.

Examples of focus descriptions might, therefore, be:

"Configuration management in the requirements engineering process,"

In this case, the subprocess is the requirements engineering process and this process is of interest from a configuration management perspective.

"A system architect's view of the requirements engineering process"

Again, the subprocess is the requirements engineering process but the perspective is that of the system architect. He or she will not necessarily be interested in the same activities as the configuration manager.

"An entity-relationship model of the documents used in the quality management process"

Here the subprocess is the quality management process and the perspective is stated in terms of the models that should be included in the viewpoint.

We have found that explicitly defining the focus of a viewpoint is valuable for three reasons:

1. It helps to identify sources of process information.
2. It can be used in the development of organizational concerns.

3. It can be used to help discover overlapping viewpoints (where inconsistencies in process perception are most likely) and gaps in the viewpoint coverage of the process.

The viewpoint focus may also have an associated rationale that is comparable to the notion of *Weltanschauung* or "world view" in Soft Systems Methodology [19], [25]. Examples of rationale which could be associated with the above focus descriptions are:

"Our current configuration management process is not integrated with our requirements engineering process"

"System architects are normally consulted after the requirements have been defined and this can cause serious design problems"

"We need a formal description of the process entities to support improved configuration management"

This rationale presents assumptions on which the viewpoint is based and helps the reader understand why the viewpoint has been identified for process analysis.

2.4 Viewpoint Sources

Viewpoint sources are an explicit record of where the information about the process has been collected. The most important sources of process information are usually:

1. the participants in the process
2. management in the organization where the process is being enacted.
3. organizational process charts, responsibility charts, quality manuals, etc.

The list of sources connected with a viewpoint is useful because it provides an explicit trace to where the process information was derived. This allows the original sources to be consulted for possible problems when process improvements and process changes are proposed. Source information may be represented as names, associated roles, and contact information if the sources are people, document identifiers, page references, and Web URLs, etc.

2.5 Process Description

We do not mandate any particular notation for process description. Our experience showed that most engineers in industry prefer informal process descriptions made up of diagrams and explanatory text. While these are more subject to misinterpretation than formal descriptions, we believe that this is more than compensated for by their understandability and flexibility in describing processes where exceptions are common. Of course, for some viewpoints that are concerned with particular types of process model, such as an entity-relationship model, an appropriate formal or structured notation may be used.

Because of individual differences in process enactment, there may be alternative perceptions of a process presented by different sources in the same viewpoint. This is particularly likely where one of the sources is process documentation that defines the organizational perception of a process (or what a process ought to be) and another source is a process participant who can explain what really happens. If these differ very radically, they should really be separate viewpoints but where the differences are in the detail of the enactment, they can be accommodated within a single process description.

These differences are accommodated by including a stable part and a variable part in the process description:

1. The stable part of a process description is the part of the description that is shared and accepted by all of the sources contributing to the process viewpoint.
2. The variable part of the process description highlights those parts of the process that exhibit variability and documents the different ways in which this variability occurs. In many cases, the variability manifests itself in the exception handling—different people cope with problems in different ways.

Describing processes using stable and variable parts is one way of tolerating inconsistency in process descriptions. As we discuss in Section 3, that describes the process of acquiring process descriptions, we try to reconcile inconsistencies as soon as they emerge but, if this is impossible, we simply leave them in the description. The inconsistency analysis that we also discuss later is then applied within the viewpoint as well as across process viewpoints.

The process description may be a hierarchical description with the process described at different levels of abstraction. At the top level, we recommend that the process description should fit onto a single page so that it may be understood as a whole. All, or some parts of, the process may then be described in more detail as necessary.

3 Using Process Viewpoints in Process Analysis

The process viewpoint model that we have described is intended to help elicit and analyze information about processes with a view to subsequent process improvement. To support the use of the model, we have developed a process for process modeling and improvement that is shown in outline in Fig. 3.

In the notation that we use here and in later figures, process activities are denoted in boxes. Dashed arrows linking boxes mean that there is a temporal relationship between these activities. It is not possible for an activity at the destination of an arrow to be completed until the activity at the source of the arrow has completed. However, the destination activity may start before the source activity has completed and the activities may be interleaved or may run concurrently. In Fig. 3, the arrow linking the last box with the first box means that the process is cyclic and can be reentered after a set of improvements have been proposed.

The overall process has four main phases:

1. *Concern definition.* During this phase, the main business goals and constraints are identified. The people involved are the process improvement team, senior managers in the organization, and project managers. Concerns are decomposed into a set of questions as discussed above.
2. *Viewpoint and source identification.* Possible viewpoints and associated sources of process information are identified in a review involving process participants and the process improvement team. We discuss this stage in more detail below.

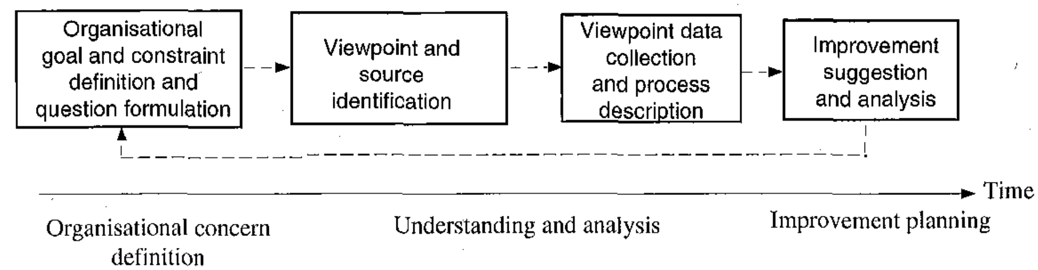

Fig. 3. Process improvement with viewpoints.

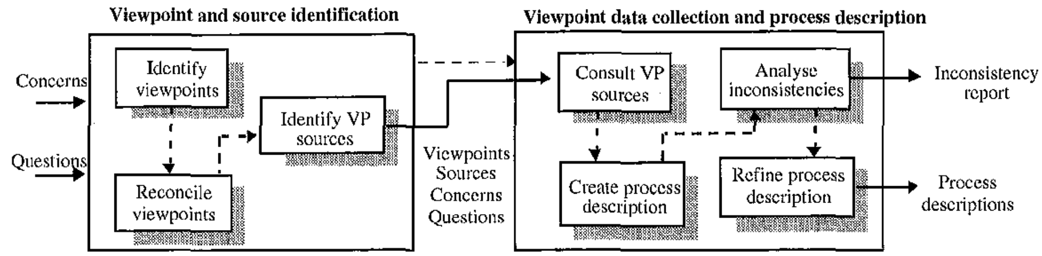

Fig. 4. Viewpoint identification and process elicitation.

3. *Data collection and process description.* Information about the process is collected and the process is documented. Process inconsistencies are identified during this stage. Again, we discuss this in more detail below.
4. *Improvement suggestion and analysis.* The processes as described in each viewpoint are compared and analyzed. Process inconsistencies are the focus for improvement and may point to potential process modifications to select best practice or to reconcile these inconsistencies. Improvements are analyzed against the concerns to ensure that they are consistent with business goals.

As the theme of this paper is process viewpoints and the support they provide for managing inconsistency, we will concentrate on the middle two stages, where viewpoints are applied and process inconsistencies are identified. Fig. 4 is a more detailed description of these two stages. In this figure, solid arrows between activities indicate data flow.

The process is iterative and we recommend that it should begin as soon as some viewpoints have been identified. Once a viewpoint has been identified, some information about the process can be collected and this may then be used to help with the identification of further viewpoints.

3.1 Viewpoint and Source Identification

This stage of the process is concerned with identifying relevant viewpoints and the information sources associated with these viewpoints. Viewpoints and their sources are identified in an iterative way so that these activities are interleaved. The inputs to this activity are concerns and associated questions. These questions may either elicit process details, discover rationale for process activities, discover information about the timing, duration, and interdependencies of activities or the support for the process that is available.

The subactivities involved in this stage are:

1. *Identify viewpoints.* This is concerned with identifying the most appropriate process perspectives that can contribute useful information about the process and representing these as viewpoints. As a starting point, viewpoints covering organizational standards, process participants, management and, where appropriate customers should be identified. There should be no restrictions on the numbers of viewpoints identified at this stage.
2. *Reconcile viewpoints.* The focus descriptions of the different viewpoints are used to identify viewpoints that may be merged. Merging viewpoints is possible when their foci overlap or are closely related. For example, a viewpoint whose focus is software testing may be merged with a viewpoint whose focus is static analysis of the system to create a general verification and validation viewpoint. Our experience of applying a viewpoint-oriented approach to requirements engineering [11] suggests that for practical reasons, the number of viewpoints should be limited. If there are a large number of viewpoints, the costs of information collection and analysis become impractically high.
3. *Identify viewpoint sources.* Viewpoint sources are information sources that can adopt the viewpoint focus. They may be people, documents, standards, domain knowledge, etc. These are identified by consultation with managers and engineers involved in the process.

The process of viewpoint identification may bring concerns to light that have not already been considered. Therefore, it is sometimes necessary to revisit the previous stage to refine these concerns before moving on to elicit information about the process description.

3.2 Viewpoint Data Collection and Process Description

This stage of the process improvement process is concerned with understanding, analyzing and describing the current process that is used. We recommend an incremental application of the steps described below for each identified viewpoint.

The questions and process descriptions should be refined as process information from viewpoint sources is elicited. That is, an initial set of questions to elicit process information is derived, viewpoint sources are consulted and a process description is proposed. This inevitably reveals omissions and problems with the initial set of questions. The analyst then refines the questions and repeats this consultation and refinement process until all viewpoints have been covered.

The stages in this process are:

1. *Consult process viewpoint sources.* The analyst puts the questions derived from concerns to the viewpoint sources to discover process information. These questions may need to be refined for the specific viewpoint (e.g., by changing the terms used) depending on the background of the source. As well as the questions, of course, sources should be asked to comment on their processes. We have found that the best way to elicit information is to ask them to critique an existing process description (however inadequate this may be) that may be derived when concerns are established. The process description focuses the elicitation as stakeholders can point out where it is incomplete and differs from their actual process. Once a more detailed process description has been elicited from one viewpoint, it may then serve as a basis for discussion about the process in other viewpoints.
2. *Create process descriptions.* A process description is created taking into account the differences as seen by different viewpoint sources. Any notation may be used here to describe the process so long as it is understandable by all people who are viewpoint sources.
3. *Analyze inconsistencies.* This activity is concerned with analyzing the process descriptions to discover redundancy and inconsistencies. We discuss this in more detail in Section 4.3.
4. *Refine process descriptions.* The results of the inconsistency analysis are fed back to the process sources and, where appropriate, the process descriptions are modified. For example, where different people use different names for the same process, a single term may be agreed. Where inconsistencies cannot be reconciled, they are documented in an inconsistency report that is an input to the next phase of the process concerned with process improvements.

This activity is also likely to reveal problems with the identified concerns and questions. Some iteration may be required to redefine the concerns and the associated questions.

3.3 Inconsistency Analysis

During the inconsistency analysis, the process descriptions that are encapsulated in each viewpoint are compared and reviewed by a team including process participants and members of the organizational process improvement group. Process inconsistencies are highlighted but no decisions about process modifications are made at this stage. This happens in a subsequent process where the concerns are used to design process changes that contribute to the business goals.

Inconsistency analysis has been shown as a separate process stage in Fig. 4 but, in fact, much of the work actually takes place during the elicitation of process information. Once a process description is available from one viewpoint, it may be used as an input to the next elicitation activity. During that activity, process stakeholders identify inconsistencies by pointing out how their view of the process differs from the view that is presented to them. In some cases, where inconsistencies are a result of misunderstandings (for example, where different terminology is used) it may be possible for the people involved to see immediately how to resolve the problem and the process descriptions are changed during elicitation to remove the inconsistency. In other cases, however, the inconsistency reflects a genuine difference and it is documented for subsequent analysis.

Inconsistencies may be identified by individuals when they examine the different process descriptions or in a process review meeting. In this review, a process or process fragment is examined and discussed by the meeting participants. Different views of the same process are considered and inconsistencies across these views are identified. Each of the identified inconsistencies is then classified using the categories shown in Fig. 5. The outcome of the review is an inconsistency report that is passed on to the next stage of the process improvement process.

Classifying process inconsistencies in this way is helpful for making judgments about possible process improvements. These different types of inconsistency may be used in different ways when considering process improvements:

1. *Tolerable inconsistencies* often reflect different professional judgment and we believe that good processes allow skilled professionals to exercise that judgment. Removing tolerable inconsistencies is likely to be demotivating for skilled professional staff. Therefore, improvement proposals should be assessed to check that they are not too prescriptive in defining how professional tasks are executed.
2. *Constraining inconsistencies* are inconsistencies that must be maintained in any process improvement proposals because they reflect local factors (such as the use of standards) that are outside the control of the process improvement team. Therefore, improvement proposals should be assessed to check that they do not inadvertently remove this type of inconsistency.
3. *Accidental inconsistencies* should be resolved as part of the process improvement.
4. *Improveable inconsistencies* are pointers to differing practices within an organization. The process improvement team should look for the best practice and should propose improvements that disseminate

Class of inconsistency	Explanation
Tolerable	There is an inconsistency in processes as seen from different viewpoints that does not materially affect the outcome of these processes. This type of inconsistency often arises from different judgements made by different process participants. An example of this might be where different techniques are used for object identification and no one technique is clearly superior.
Constraining	The inconsistency in processes must be maintained because it has arisen as a consequence of some external factors that are outside the influence of the process improvement team. For example, teams in the same organisation working in different countries may operate difference, inconsistent processes. Process inconsistencies arise as a consequence of local laws, such as health and safety regulations, that must be followed. The inconsistency therefore constrains process evolution.
Accidental	These are process inconsistencies that have been introduced by accident and that are a consequence of human mistakes and omissions. An example of this type of inconsistency is where an activity appears in one view of a process but where that activity has been omitted from another view where it should have been included
Improveable	These are inconsistencies where different sub-processes are used by different people or in different parts of an organisation. Some of these sub-processes may be clearly better than others. For example, one part of an organisation may support their testing process using CASE tools whereas, in other testing processes, no automated support is used.

Fig. 5. Classes of process inconsistency.

this good practice across the organization. They are distinct from tolerable inconsistencies in that the choice of practice is not a question of professional judgment but is a consequence of lack of information or arbitrary historical decisions.

4 A Simple Case Study

In this section, we present part of a process analysis case study where we applied the process viewpoint approach. The processes concerned were development and management processes for a safety-related expert system that was being developed by a relatively small company for a large client. The system involved the development of a "safe" expert system shell and the instantiation of this shell with specific domain data. The system was to be formally specified and the implementation validated by correctness arguments (not a complete proof) against the specification. The company developing the system is a specialist in critical computing systems, is technically mature and is strongly committed to quality and quality improvement.

Our aim in carrying out this case study was to evaluate our approach in a formative way. That is, we wanted to try it out on a real example and learn from the problems that we encountered without prejudgment of what these problems might be. We do not believe that summative evaluation where a hypothesis is proposed and then confirmed or denied is realistic in an engineering discipline. We approached the case study with an open mind about what we would learn and how that might affect the process viewpoints approach. We summarize some of these lessons and some of the difficulties we faced later in this section.

4.1 Concerns and Questions

The first stage in the process analysis is to identify the organizational concerns that contribute to the process analysis. As this is a safety-related system, the most important of these concerns was *safety*. A critical business goal of the organization was to ensure that the systems it developed were safe and processes always had to take this into account. A further business concern in this particular case was *customer relationships* as the customer in this case was a large organization and a likely source of future contracts. The final concern identified was *skill utilization*. The organization developing the system is very highly skilled in a number of areas and processes had to be designed to utilize these skills.

Note that these concerns don't fall neatly into our earlier classification of understanding, improvement, or constraint concerns. For example, safety is both an improvement concern and a constraint. The organization wishes to develop a safer process where fewer product defects are introduced but also constrains any process improvements to be such that the safety of the process is not impaired.

To illustrate question derivation, let us consider the *safety* concern. Some relevant process questions associated with this concern are:

1. Is there an independent validation activity for each of the outputs of the processes?
2. Have adequate resources been assigned to each validation activity?
3. How are identified hazards tracked at each stage in the process?

4. Are the processes conformant with the safety plan agreed with the customer?
5. Are descriptions of the system in different notations checked for consistency?

By applying these questions to process sources, we built up a model of the process as seen by different process viewpoints.

4.2 Viewpoint Identification

The next stage of the process of analysis is to identify relevant viewpoints. The process we recommend involves identifying as many viewpoints as possible then merging these to reduce the number of viewpoints for analysis. In this case, we will skip the first stage and simply discuss the final viewpoints used in the analysis. Three viewpoints were identified as significant in this case:

- A *project management* viewpoint
- A *quality management* viewpoint
- An *engineering* viewpoint

The scope of each of these viewpoints is defined by describing their focus. All viewpoints are concerned with the whole of the development process for the system so this need not be explicitly set out in the viewpoint focus descriptions:

1. *Project management.* The project management viewpoint is concerned with the process as defined by those activities that are identified in the project plan and that are assigned specific resources and a schedule for completion.
2. *Quality management.* The quality management viewpoint is concerned with those aspects of the process where the customer requires explicit evidence of validation activities and the conformance of process deliverables to requirements and standards.
3. *Engineering.* The engineering viewpoint is concerned with the process that is actually enacted by the engineers involved in the system development.

In conjunction with the process of viewpoint identification, sources of information associated with the viewpoint should also be identified. The focus description may be used as a starting point for identifying information sources. In this case, once one source was identified, he helped in the identification of other possible sources.

For brevity, we will focus the remainder of this discussion on the first two viewpoints above namely *quality management* and *project management.* Sources of information associated with these viewpoints are:

- *project management:* project plan, project manager, customer project manager, software developers
- *quality management:* project quality plan, organizational quality manual, quality manager, customer project manager

In practice, it is not always possible to get access to all sources. In this particular case, we had close links with the development organization but no access to their customer. We were therefore unable to get any information from customer-based sources. This illustrates a general problem with industrial case studies. External factors mean that these are not always ideal; they may not allow complete coverage of a method and they may have to be carried out to a very tight schedule. However, we argue that they are inevitably useful in some respects and are more valuable than academic examples that don't capture the messy reality of industrial software development.

4.3 Process Description

Once the sources have been identified, they are consulted and a process description is developed. The principal source for the project management and quality management viewpoints was the very comprehensive process documentation that had been produced to satisfy the quality requirements of the customer. The questions which were derived from the concerns were used to elicit process details from this documentation and from the people involved in the process.

From the project management viewpoint, resources and schedules had been drawn up for 21 activities; from the quality management viewpoint, 36 explicit validation activities were identified. In each of these cases, there were logical groupings of activities so it was relatively straightforward to produce a more abstract model of the process.

This high-level project management model is shown in Fig. 6. Again, the dotted arrows mean temporal sequence where a destination activity may start but may not finish before the source activity has finished. Where activities are vertically aligned, this means that they may (but need not) be carried out in parallel.

From the quality management perspective, the high-level model focused on validation activities. This model is illustrated in Fig. 7.

Because of the safety-related nature of this project, auditable validation of all process deliverables is essential. Therefore, for model consistency, we should look for a 1:1 relationship between project management activities and validation activities. If there are validation activities that do not appear in the project management model, this implies that these have not been allocated project resources.

Bearing this in mind, a comparison of Figs. 6 and 7 reveals two inconsistencies:

1. The project management model has an activity "acceptance testing" that does not appear in the quality management process. This is a *tolerable inconsistency* as the responsibility for validating the acceptance test lies with the customer for the software. The developer, therefore, need not assign resources to this activity.
2. The quality management model includes an activity "safety plan review" that is not included in the project management model. This is an *accidental inconsistency* that highlights an error in the project management view of the process. In practice, the inconsistency could be resolved by adding a subactivity to "technology assessment" that was concerned with preparing inputs for use in the project.

Further inconsistency analysis requires a more detailed look at the process. Let us look at the processes, inputs and outputs for the shell specification activity shown in Fig. 6. This more detailed model is shown in Fig. 8. Solid arrows

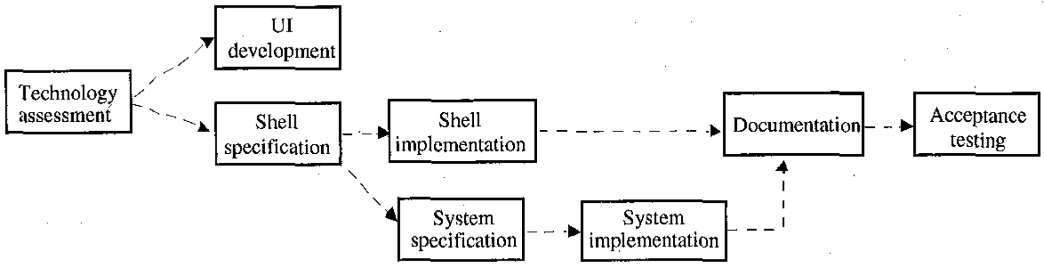

Fig. 6. Project management model.

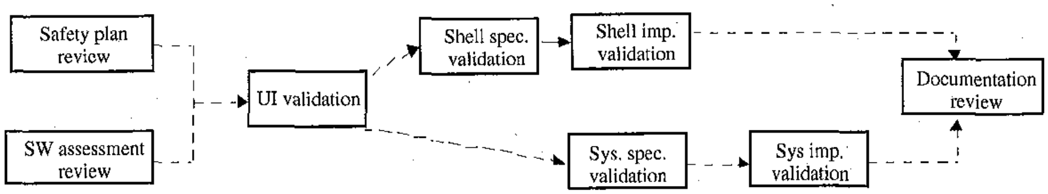

Fig. 7. Quality management model.

linking boxes indicate data-flows. The System SRS is a software requirements statement, written in English, that was produced by the customer for the system.

The comparable quality management model that identifies validation activities, inputs and outputs is shown in Fig. 9. In this process, each validation activity produces a report and has a set of associated success criteria. These are not really inputs to the next stage (the inputs to validation are outputs from development activities) so they are not included as data-flows on this diagram.

4.4 Inconsistency Analysis

In analyzing these more detailed models for inconsistency, we put the process descriptions from the quality management and the project management viewpoints side-by-side and ask a number of questions motivated by the concerns and their associated questions. Examples of these questions are:

1. What is the correspondence between tasks for which resources have been allocated and identified validation activities? That is, is it possible to equate, perhaps by name similarity, the activities in the different viewpoints.
2. Are there any mismatches in the process descriptions such as development activities with no corresponding validation activity or validation activities with no allocated resources?
3. Does the process description in the project management viewpoint clearly identify the deliverables that must be validated to ensure compliance with the project safety plan?
4. Is the identification of outputs in the project management viewpoint consistent with the identification of process inputs in the quality management viewpoint?

When we examined the process descriptions in the project management and process management viewpoint and put these questions to the models, we found inconsistencies in the process. From the process fragments shown in Figs. 8 and 9 we can see that, in the project management view, there are three validation activities in the process (activities 2, 3, and 5). Somehow, these must map onto the six activities identified in the quality management view of the process. When we looked at the processes in detail using the above questions, we found three inconsistencies:

1. The quality management view requires a static analysis of the formal specification (activity 1) that is not explicit in the project management view. This is probably an accidental inconsistency resulting from a failure in communication between the project manager and the quality manager. However, it could be an improveable inconsistency. If the project manager and the quality manager are from different backgrounds, then the project manager may not be aware of the benefits of static analysis.
2. There is no activity in the project management model where test planning is explicit and the test plan is not specified as an output of any activity. It is, however, an input to validation activities 4 and 5 that review the plan. As there are two test plan reviews, it is clear that, from a quality management viewpoint, significant effort should be devoted to test planning. This is probably an accidental inconsistency that suggests that the project management view of the process should be modified to include a test planning activity.
3. The quality management viewpoint identifies inputs as "revised formal spec," but it is not clear which version of these inputs should be used. That is, the project management model assumes a number of revisions of this specification (identified in the process as V1, V2, V3, and V4) but the quality management viewpoint does not explicitly state which of these are inputs to which validation activity. This is an accidental inconsistency that suggests that the quality management process description should be modified to include explicit version identification.

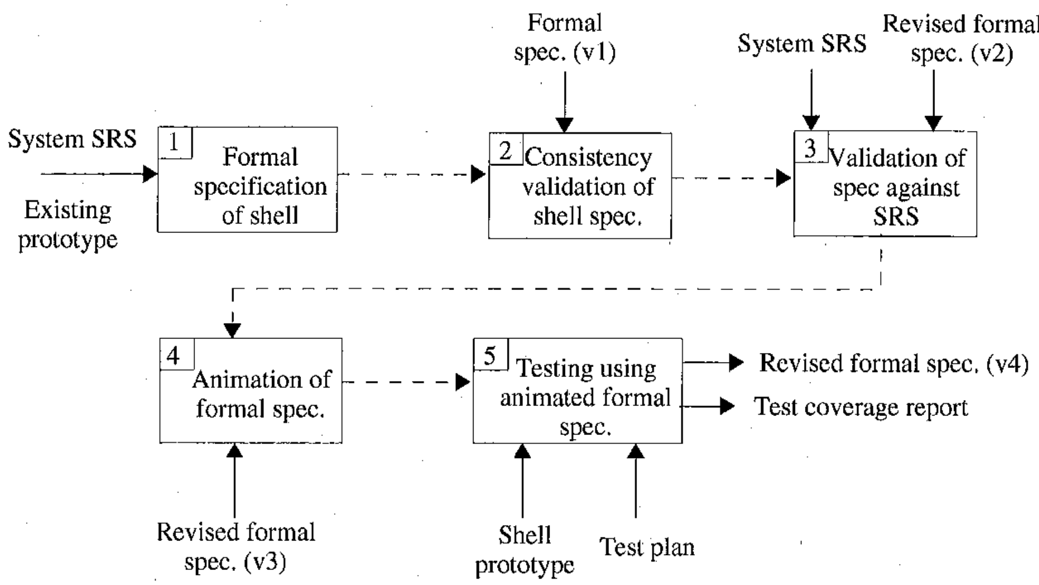

Fig. 8. Project management model of shell specification.

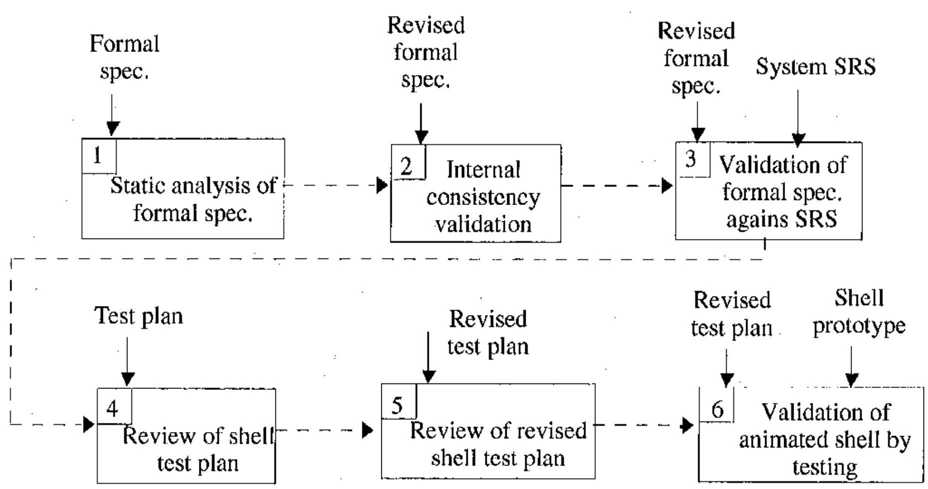

Fig. 9. Quality management view of shell specification validation.

The process of process analysis is continued for the other process fragments associated with the activities identified in Fig. 6 and Fig. 7. We will not show this here as it simply involves applying the same process as we have described for the shell specification. Overall, we identified a total of 14 inconsistencies between the processes as seen from the project management and quality management perspective. Of these, two were tolerable inconsistencies and the remainder were either accidental or improveable inconsistencies. We did not find any constraining inconsistencies. This was not surprising as we expect this type of inconsistency to arise when comparable processes in different parts of an organization are compared.

4.5 Lessons Learned

Applying the approach to a real industrial case study was very helpful. The nature of this case study was such that it did not allow all aspects of the process viewpoint approach to be evaluated but we gained very useful feedback particularly on the application of the approach to process analysis. Some lessons that we learned were:

1. Even in the relatively simple processes studied here, there were a significant number of process inconsistencies and structuring the process analysis using viewpoints was a helpful way of discovering these inconsistencies.
2. The distinction between the types of inconsistency that we proposed in Fig. 5 is not as clear cut as we anticipated. It was helpful to identify inconsistencies but, in this particular study, it was sometimes unclear whether these inconsistencies were accidental i.e., a result of process errors or misunderstandings or improveable i.e., a result of one view being unaware of better practice that is part of a different view. It is perhaps the case that making a distinction between

types of inconsistency is irrelevant. Further work is clearly needed to determine if firstly, inconsistency classification is useful and, secondly, if the classifications discussed here are the right ones.

3. Our original notion that processes may be described, within a viewpoint, using any notation was a valid one. For example, from a project management viewpoint, processes were described using bar charts and from a quality management viewpoint they were described using narrative text. However, we quickly realized that we needed some common notation to make effective comparisons of processes and we redescribed the processes using the simple graphical notation that we have used here.
4. The tool support requirements for the approach are relatively simple. Tool support would have been helpful in translating the elicited process descriptions into a common format and comparing these descriptions. We used a graphical editor for this but specialized tool support that would have allowed us to explicitly link processes in different viewpoints that had different names would have been helpful.
5. The identification of concerns was a very useful step in the process as it helped us understand the priorities of the process "owners." The questions derived from concerns were useful because they helped us to get started with the process analysis. However, it wasn't sensible to stick with our pre-prepared list of questions as it quickly became clear that some of them were irrelevant to some of the sources of process information. However, the fact that we did not see proposed process improvements meant that we could not validate our notion that concerns could be used to check that proposed improvements were conformant with business goals.
6. Inconsistency analysis often raised further questions about the process. If we consider the example of the static analysis inconsistency discussed above, further issues came to light. It is not clear whether or not the specification should "pass" the static analysis with no significant errors or how the results of the static analysis are used in the internal consistency check of the specification. Nor is it clear if the static analysis report is delivered to the customer or not.

The essential lesson that we learned from the case study was the need for flexibility. The fundamental framework that we have described here was helpful but when we tried to realize it in a practical context, we had to be opportunistic about what information could be collected and when it was collected. It wasn't really possible to separate the activities of viewpoint and source identification and sources sometimes pointed out process inconsistencies before actually describing the process used.

5 Related Work

The earliest work that we are aware of that has taken a multiperspective approach to process modeling was carried out by Rombach in the late '80s. He developed MVP-L [26], a process modeling language that supported multiple views. This work has continued with a newer version of the language MVP-L 2 now available [27]. MVP-L has a fixed set of views namely a product view, a process view and a resource view. It therefore provides an integrated mechanism to define different representations of a process and its supporting environment MVP-E [28] supports some consistency checking across these representations. Our notion of viewpoints is really quite different from that in MVP-L and reflects the process as seen by different stakeholders. We are not concerned with cross-checking different representations of the process that are derived from the same source.

Verlage [29], in work related to MVP-L, confirms the need for a multiple perspective approach to process modeling and presents a set of requirements for this approach. He suggests the following key requirements:

1. different perspectives on the same process element must be offered
2. tailorable (user-defined) perspectives
3. structuring of views
4. independent modeling of views
5. detecting similarities between views
6. detecting inconsistencies between views.
7. dynamic change of perspectives
8. concurrent views—different views for different people

We believe that this is a helpful list of requirements and we return to it in the conclusions of the paper where we assess our process viewpoints approach against it.

Bandinelli et al. [30], [31] discuss an experiment where they used a formal process modeling language to help discover process inconsistencies across different views of a process. This paper is an interesting illustration of the belief of processes as descriptions of how work should be done as discussed in Section 1.1. The authors identify five different views of a software process:

1. the *desired* view, which is the process that the organization really wants
2. the *official* view, which is the process that has been documented
3. the *perceived* view, which is the process that different actors think has been documented
4. the *observed* view, which is what observers examining the process see happening
5. the *actual* view, which is the process actually followed by process actors

The objective of the process analysis was to increase the conformity of the official and desired process, improve the specification of the official process, and move the actual process closer to the officially defined process. This work discovered a number of process inconsistencies and, in particular, identified a number of problems in the description of the "official" process. Clearly, the use of different views was helpful in focusing the inconsistency analysis and the authors were committed to the use of formal process modeling to support the analysis.

The paper shows that formal modeling of processes can be used to discover inconsistencies but it neither claims nor demonstrates that this is a cost-effective approach to

Requirement	Assessment
Different perspectives on the same process element must be offered	A fundamental feature of our approach
Tailorable (user-defined) perspectives	A fundamental feature of our approach
Structuring of views	The requirement suggests that process elements should be explicitly grouped to form a view. Our notion of focus would appear to support this although it is not entirely clear to us what the requirement means.
Independent modelling of views	As suggested, correspondence of views is established after their description
Detecting similarities between views	No automated support. This requirement and the requirements for inconsistency detection is partially supported through a suggested process for using process viewpoints
Detecting inconsistencies between views	No automated support.
Dynamic change of perspectives	This suggests that it must be possible for the same actor to view a process differently in different situations. This is supported by defining different viewpoints but with common sources of viewpoint information.
Concurrent views - different views for different people.	This is supported by allowing flexibility in the notations used to describe processes.

Fig. 10. Assessment of process viewpoints.

inconsistency analysis. The paper does not discuss whether the formal models were used directly with the process owner and process agents or if these were paraphrased for discussions with them about the process. Their informal process views are sensible but we believe that there are likely to be many actual processes (corresponding to different process actors) and, perhaps, several official processes (documented in different places) and that this cannot be accommodated in a predefined set of viewpoints.

Turgeon and Madhavji have also developed a multiview approach to elicitation with some automated consistency checking. This is based on their work on process elicitation and the methodological and tool support that has been developed for this. Their original Elicit approach [32], [33] supported the notion of multiple views to manage different representations of processes such as an activity view, a resource view, etc. More recently, they discuss [34] the need for role-driven views for process modeling. This work is still under development but they are also taking a language-based approach to inconsistency detection and offer some support for automated detection of inconsistencies in a formal process model.

Of all other research in this field, this is the closest to the work we have reported here. We share the notion that views can be associated with roles and that there may be specific models (Turgeon and Madhavji use the term "aspect" here) associated with a view. Our notion of a viewpoint is a more general one. We do not constrain viewpoints to be associated with roles (although this is certainly the most common type of process viewpoint) and the use of concerns and associated questions provides a starting point for the elicitation process. We are not concerned with automated inconsistency checking so work with a more flexible notion of inconsistency that can encompass inconsistencies derived from human and organizational factors.

Our work has also been influenced by a body of work that has been concerned with inconsistency management across viewpoints in requirements engineering [35], [36]. The goal of this work is to provide a framework for allowing potentially inconsistent requirements to coexist and to help discover those inconsistencies. This relies on automatic or semiautomatic analysis of formal process descriptions to discover inconsistencies across viewpoints [37]. This work has been successful for small-scale requirements analysis but the difficulties of creating formal descriptions of requirements and in reconciling functional and nonfunctional requirements inconsistencies mean that it is a major challenge to scale the approach for industrial systems.

6 Conclusions

This paper has described an approach to process analysis and improvement based on viewpoints where each viewpoint manages a distinct process perspective. It allows inconsistent models of processes to be managed and provides a framework for analyzing the inconsistencies with a view to subsequent process improvement. Fig. 10 is an assessment of our work against the requirements for multiview modeling of software processes identified by Verlage [29]. This demonstrates that there is a substantial correspondence between our process viewpoints and the identified requirements.

All our work was strongly influenced by practical industrial requirements. Consequently, we have designed process viewpoints to be a flexible approach to process analysis that does not require users to take the risk of committing themselves to new, specialized notations. The case study described here demonstrated that it was realistic to apply our approach to real industrial processes, although we must admit that this was a relatively small project with a limited number of stakeholders. It did not exercise all aspects of our approach to process analysis, but the results suggested that there was some merit at least in using viewpoints when trying to understand and improve processes. We do not yet know how the approach will scale up to the processes used in large software engineering projects.

Acknowledgments

The work described in this paper was partially supported by the European Commission's ESPRIT programme under project REAIMS (8649). Particular thanks are due to Robin Bloomfield at Adelard for case study information.

References

[1] A. Finkelstein, J. Kramer, and B. Nuseibeh, eds., *Software Process Modelling and Technology*, Taunton, Somerset, England: Research Studies Press Ltd., 1994.

[2] A. Fuggetta and A. Wolf, eds., *Trends in Software: Software Process*. Chichester, U.K.: John Wiley & Sons, 1996.

[3] M. Hammer, "Reengineering Work: Don't Automate, Obliterate," *Harvard Business Review*, pp. 104-112, July/Aug. 1990.

[4] W.S. Humphrey, "Characterizing the Software Process," *IEEE Software*, vol. 5, no. 2, pp. 73-79, 1988.

[5] M.C. Paulk B. Curtis, M.B. Chrissis, and C.V. Weber, "Capability Maturity Model, version 1.1," *IEEE Software*, vol. 10, no. 4, pp. 18-27, 1993.

[6] V. Haase, R. Messnarz, G. Koch, H.J. Kugler, and P. Decrinis, "Bootstrap: Fine Tuning Process Assessment," *IEEE Software*, vol. 11, no. 4, pp. 25-35, 1994.

[7] K. El Emam, J. Drouin, and M. Welo, *SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*. Los Alamitos, Calif.: IEEE CS Press, 1997.

[8] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide*. Chichester U.K.: John Wiley & Sons, 1997.

[9] L. Emmet, R. Bloomfield, J. Bowers, and S. Viller, "PERE: Evaluation and Improvement of Dependable Processes," *Proc. Safecomp'96*, 1996.

[10] I. Sommerville and P. Sawyer, "Viewpoints: Principles, Problems and a Practical Approach to Requirements Engineering," *Annual Software Eng.*, vol. 3, 1997.

[11] I. Sommerville, P. Sawyer, and S. Viller, "Viewpoints for Requirements Elicitation: A Practical Approach," *Proc. Int'l Conf. Requirements Eng.*, 1998.

[12] L. Osterweil, "Software Processes are Software Too," *Proc. Ninth Int'l Conf. Software Eng.*, pp. 2-12, 1987.

[13] C. Fernström, "Process Weaver: Adding Process Support to Unix," *Proc. Second Int'l Conf. Software Process*, 1993.

[14] S.M. Sutton and L.J. Osterweil, "The Design of a Next-Generation Process Language," *ESEC/FSE'97*, pp. 142-58, Springer-Verlag, 1997.

[15] M.C. Paulk, C.V. Weber, B. Curtis, and M.B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, Mass.: Addison-Wesley, 1995.

[16] I. Sommerville and T. Rodden, "Human, Social and Organizational Influences on the Software Process," *Trends in Software*, A. Fuggetta and A. Wolf, eds., pp. 89-110, New York: John Wiley & Sons, 1996.

[17] G. Cugola, E. Di Nitto, A. Fuggetta, and C. Ghezzi, "A Framework for Formalizing Inconsistencies and Deviations in Human-Centered Systems," *ACM Trans. Software Eng. and Methodology*, vol. 5, no. 3, pp. 191-230, 1996.

[18] R. Conradi and C. Liu, "Process Modelling Languages: One or Many," *Proc. Fourth European Workshop Software Process Technology*, Springer-Verlag, 1995.

[19] P. Checkland, *Systems Thinking, Systems Practice*. Chichester U.K.: John Wiley & Sons, 1981.

[20] G. Mullery, "CORE—A Method for Controlled Requirements Specification," *Proc. Fourth Int'l Conf. Software Eng.*, pp. 126–35, IEEE CS Press, 1979.

[21] G. Kotonya and I. Sommerville, "Viewpoints for Requirements Definition," *BCS/IEE Software Eng. J.*, vol. 7, no. 6, pp. 375-87, 1992.

[22] G. Kotonya and I. Sommerville, "Requirements Engineering with Viewpoints," *BCS/IEE Software Eng. J.*, vol. 11, no.1, pp. 5-18, 1996.

[23] I. Sommerville, P. Sawyer G. Kotonya, and S. Viller, "Process Viewpoints," *Proc. Fifth European Workshop Software Process Technology*, Springer-Verlag, 1995.

[24] V.R. Basili and H.D. Rombach, "The TAME Project: Towards Improvement-Oriented Software Environments," *IEEE Trans. Software Eng.*, vol. 14, no. 6, pp. 758-773, June 1988.

[25] P. Checkland and J. Scholes, *Soft Systems Methodology in Action*. Chichester U.K.: John Wiley & Sons, 1990.

[26] H.D. Rombach, "MVP-L, A Language for Process Modeling In-the-Large," Inst. for Advanced Computer Studies, Univ. of Maryland, 1991.

[27] A. Bröckers, C.M. Lott, H.D. Rombach, and M. Verlage, "MVP-L Language Report, version 2," Dept. of Computer Science, Univ. of Kaiserslautern, Germany, 1995.

[28] U. Becker, D. Hamann, J. Münch, and M. Verlage, "MVP-E: A Process Modeling Environment" *IEEE Trans. Computer Software Eng. Software Process Newsletter*, vol. 10, pp. 10-15, 1997.

[29] M. Verlage, "Multi-View Modeling of Software Processes," *Proc. Third European Workshop Software Process Technology*, Springer-Verlag, pp. 123-127, 1994.

[30] S. Bandinelli, A. Fuggetta L. Lavazza, M. Loi, and G.P. Picco, "Modeling and Improving an Industrial Software Process," *IEEE Trans. Software Eng.*, vol. 21, no. 5, pp. 440-454, May 1995.

[31] S. Bandinelli, A. Fuggetta , and C. Ghezzi, "Software Process Evolution in the SPADE Environment," *IEEE Trans. Software Eng.*, vol. 19, no. 12, pp. 1,128-1,144, 1993.

[32] N.H. Madhavji, D. Holtje, H. WonKook, and T. Bruckhaus "Elicit: A Method for Eliciting Process Models," *Proc. Third Int'l Conf. The Software Process*, IEEE CS Press, 1994.

[33] G.T. Heineman et al., "Emerging Technologies that Support a Software Process Life Cycle," *IBM Systems J.*, vol. 33, no. 3, pp. 501-529, 1994.

[34] J. Turgeon and N.H. Madhavji, "A Systematic View-Based Approach to Eliciting Process Models," *Proc. Fifth Euopean Workshop Software Process Technology*, Springer-Verlag, 1996.

[35] A. Finkelstein, J. Kramer, B. Nuseibeh, and M. Goedicke, "Viewpoints: A Framework for Integrating Multiple Perspectives in System Development," *Int'l J. Software Eng. and Knowledge Eng.*, vol. 2, no. 1, pp. 31-58, 1992.

[36] B. Nuseibeh, J. Kramer , and A. Finkelstein, "A Framework for Expressing the Relationships between Multiple Views in Requirements Specifications," *IEEE Trans. Software Eng.*, vol. 20, no. 10, pp. 760-773, Oct. 1994.

[37] S. Easterbrook and B. Nuseibeh, "Using ViewPoints for Inconsistency Management," *BCS/IEE Software Eng. J.*, vol. 11, no. 1, pp. 31-43, 1996.

Ian Sommerville received a BSc degree in physics from Strathclyde University and the MSc and PhD degrees from St. Andrews University, Scotland. He is a professor of computer science and chair of the Computing Department at Lancaster University. He has been involved in software engineering research and teaching for the past 20 years and his current research interests are in requirements engineering, system evolution, and human and social factors in systems design. He is a member of the ACM, the IEEE Computer Society, and the British Computer Society, and is a fellow of the Institute of Electrical Engineers.

Pete Sawyer holds BSc and PhD degrees in computer science from Lancaster University. He is a senior lecturer in the Computing Department at Lancaster University, where he has held a variety of positions since 1986. His principal research interests are in requirements engineering, dependable systems, and software process improvement. He is a member of the IEEE Computer Society and the ACM.

Stephen Viller has a BSc degree in computation from the University of Manchester Institute of Science and Technology, an MSc degree in cognitive science from Manchester University, and a PhD in computer science from Lancaster University. He is currently a research fellow in the Computing Department at Lancaster University. His research interests are broadly within the fields of requirements engineering and computer supported cooperative work, focusing on improving support for human activity in the design of cooperative systems. He is a member of the IEEE Computer Society and the ACM.