A Group Collaboration Tool for Software Engineering Projects

Martin Purvis, Maryam Purvis, and Paul Jones

Computer and Information Science
University of Otago
Dunedin, New Zealand

Abstract

A CASE tool for the support of collaborative work in software engineering processes is described. The tool supports multiple, concurrent users who may be distributed over a network and who are engaged in group design deliberation on software engineering projects. The approach underlying the tool is based on Rittel's issue-based information systems, but new features have been added to support more detailed evaluation of why particular positions have been chosen by a collaborative group to resolve a given issue. A discussion of the new approaches and how they are incorporated into the tool is provided.

1 Introduction

Software systems are increasingly intertwined with the day-to-day processes of businesses and organisation, either as reactive information systems or as real-time process control engineering systems. In such cases their design, construction, and operation requires the participation of or consultation with a variety of stakeholders with various perspectives on how the system should function. Consequently the processes of software engineering are more and more becoming social activities, where key decisions are made in the context of collaborative groups. This is consistent with the general trends in business operations as a whole, which have been observed to be increasingly team-oriented [1]. In view of these trends and of the fact that the collaborative groups of software engineering projects are likely to change in number and make-up over the courses of the projects, it is desirable to have mechanisms that can assist the group decision-making process and help record how the decisions are made. This latter activity is sometimes referred to as capturing design rationale [2,3,4,5]. In this paper we describe a tool under development, wIBIS, that is intended to assist the decision-making process for distributed collaborative groups and for recording the rationales behind their collective decisions.

We note that despite the term "design rationale", the activity of group decision making takes place throughout the software engineering process and has been studied with reference to planning, design, and maintenance. Certainly the early stages of a software development project are characterised by a high degree of group coordination and negotiation, and it is during this period when the key decisions are made that will give shape to the project. However, software development practice is moving beyond the strictly phased software development lifecycle approach and is more often likely to go back and review requirements and design decisions during the course of development. When Curtis et al. reviewed large U.S. software development projects, they found that most projects experience changes in the requirements during the course of development [6]. Thus it is sometimes useful to view the processes of requirements specification, design, implementation, and maintenance as concurrent activities that extend over most of the development period and can experience changes to plans at any time. Each of these activities may at any time need to refer to the decisions that have been made in connection with the other process activities of development.

Mechanisms that are used to keep track of these decisions should also enable reviewers to examine the nature of the deliberations that went into decisions, in case the original decisions need to be amended in light of new development associated with the other, concurrent activities. Moreover if the original decisions are not recorded in a satisfactory manner, particularly with respect to why the decisions was made, the essential reasoning behind the original decision can be lost -- what is left is merely the text of the decision. This text of the decision can be subject to different interpretations by different stakeholders, who have different perspectives concerning the problem at hand. Such differing interpretations can give the appearance that the specification is changing, even when it is not. This phenomenon was observed by Curtis, who found that the most inexperienced and unstructured software development teams encountered apparent changes to the specification, even when it was not changing at all [6].

The manner in which group collaborative decisions are captured has been the subject of considerable discussion in the research literature [4,5,7,8]. A common, straightforward approach is simply to record the minutes of groups meetings. For these purposes it is necessary to record, however, not only
the final group decision, but also the reasons why the decision was made. Because a number of different ideas may have been discussed before coming to a decision, it is often difficult for the minutes-taker to decide what should and should not be recorded. Minutes-takers may vary considerably regarding the degree of detail that they record, and some minutes-takers who are also discussion participants find that the overhead of taking minutes interferes with their ability to participate in discussion, with the consequence that they sometimes neglect their minutes-taking chores. Frequently when team members examine minutes of a meeting at a later date, they feel that key items have been left out of the record. They also frequently find that much recorded information is trivial and makes navigation through the recorded minutes more tedious. In an effort to address these difficulties, approaches have been suggested that are semi-structured: the structure is designed to give guidelines concerning what should and shouldn't be recorded, while at the same time not being so schematic and inflexible as to filter out complex ideas.

2 Capture of Design Rationale based on Speech-Acts

Several of the proposed semi-structured approaches for the capture of design rationale are based on the notion of speech-acts [9,10,11]. These, in turn, can be associated with the notion of "wicked" problems [12]. The basic idea is that human beings fundamentally represent action (and the need for action) linguistically. Our conversations characteristically take place in the context of describing, explaining, or deciding upon a course of action. From this perspective, language is not seen so much as a system for representing the world or for conveying information; instead language specifies our ontology: the distinctions that enable us to act together in a common world. For example a linguist might regard the utterance, "the car is outside with the motor running," as merely a statement of fact. But when we consider that statement in a plausible context, it might suggest that an action is requested. Perhaps someone is here to pick me up and take me somewhere; or perhaps I am being told that I forgot to turn the engine off, and I should go out and attend to it. Thus the utterance may be a request or a response to a request or something else (offer, promise, etc.), but we still invariably consider the statement in a world of potential actions.

Continuing with this theme a little more, we observe that when things are going smoothly and the immediate items in our world are understood, no action is called for, and we can consider the world from an "objective" viewpoint. But at some point, a "breakdown" occurs and some action is desired (a breakdown might be something negative, such as a malfunction, or positive, such as a new opportunity). At this point we engage in conversations in order to induce some actions or acquire information. Thus our conversations represent our efforts to figure out what to do next. This "what to do next" could be something relatively trivial, or it could represent the solution to an exceedingly difficult problem. Some of the most challenging of these problems were characterised by Rittel as "wicked" problems [12]. Wicked problems are doubly difficult, because not only do we have no known method or algorithm for solving the problem, we have considerable difficulty defining the nature of the problem in precise terms. For example the problem of designing an algorithm for winning every possible match of chess would be exceedingly difficult, but it is not a wicked problem. Wicked problems appear in human contexts, such as how to give a child moral training, or how to win the trust of those whom we favour. Wicked problems often have no stopping rules and usually have only qualitative solutions. Since computer information systems are embedded in human social contexts, the design of such systems is also a wicked problem.

Rittel devised a semi-structured mechanism for carrying out the conversations involved in attempting to solve wicked problems and building up an information base that describes possible approaches to their solution, which he called an "issue-based information system" (IBIS) [12]. His method has served as the inspiration for a number of conversation-based tools for capturing design rationale. In Rittel's IBIS system, there are only three basic elements to a wicked-problem-solving conversation: issues, positions, and arguments.

- An issue is essentially a problem, or in its simplest form, a question for which a decision can be made that will advance the project group. In an IBIS record, only issues are recorded. Items on which there is initially basic group agreement are left out of the IBIS record. Example issue -- "What operating system should we use?"

- A position is a response to an issue that represents an attempt to resolve the issue. For each issue there can be many positions taken with respect to the issue. Example position (response to above issue) -- "UNIX"

- An argument represents a statement either in support of or in opposition to a particular position. Example argument (in support of above position) -- "Our staff has experience with it."

Issues, positions, and arguments are expressed in standard English text and are of no set length, although brevity is clearly desirable. The advantage of IBIS is its simplicity: there are only three types of rhetorical utterances. When a meeting is taking place, the IBIS recorder need only record the issues, positions, and arguments that are given at the meeting. Every

---

1 Issues, positions, and arguments used in Rittel's technical sense will be written in italics.
position and argument must be properly related to its parent element.

The simplicity of IBIS leads naturally to a plain text representation, itIBIS ("indented text IBIS"), which can be used manually to record IBIS elements with pencil and paper at group meetings [5,13]. With itIBIS, issues are labelled with "I", positions with "P", and arguments with "A". The relationships between the elements are indicated by appropriate text indentations. Additional annotations are added to the labels according to the following scheme:

Issues: *I --> an issue has been resolved.
Positions: *P --> this is a winning position.
          -P --> this position has been rejected.
          ?P --> no decision has yet been made concerning this position
Arguments: +A --> a supporting argument
            -A --> an opposing argument

The following is an itIBIS example of a group meeting to discuss a proposed computer platform:

*I: Which computing platform?
  ?P: a UNIX workstation
      +A: fast execution
      -A: expensive
  *P: a PC (Intel-based)
      +A: preferred by customers
      +A: open architecture
      -A: closed architecture
      -A: slow execution
      +A: good graphics

There have been several attempts to extend the IBIS scheme to cover more semantic issues, although some have strayed from the original speech-act-based inspiration [4,14,15]. The advantage that Rittel's original formulation has over most of the extensions is its inherent simplicity. Stakeholders that come to a meeting grasp the IBIS format very quickly, and the semi-structured format provides for a uniform record keeping approach and that doesn't impose undue overhead on the IBIS recorder.

Previous efforts to develop hypertext tools to facilitate the representation of IBIS records [7,14] have represented issues, positions, and arguments as graphical nodes, with arcs representing their relationships. This has enabled the representation of secondary links between nodes. Thus an argument could have a primary link in support of or opposition to a particular position, but could also have a secondary link (in support of or opposition to) another position that is in response to an entirely different issue. These additional relationships are not really suitable for itIBIS but can be represented graphically with a hypertext tool. The legal IBIS relationships can then be represented by the diagram shown in Figure 1.

![Figure 1. Legal IBIS Relationships.](image)

Thus new issues can be generated at any point in connection with an existing issue, position, or argument, and the relationship of this new issue with respect to the existing node is represented by the is-suggested-by arcs shown in Figure 1. The wIBIS tool discussed below belongs to this family, but contains some features that differ from other IBIS-type tools. A sample representation of an elementary IBIS record in such a hypertext tool is shown in Figure 2. When the individual nodes are double-clicked with the mouse, a dialogue-box will appear that shows editable information about that node.

3 The wIBIS collaboration tool

The wIBIS tool under development employs a client-server architecture so that multiple participants may participate in group decision-making by using the wIBIS tool concurrently. Information concerning the IBIS record is stored in a SQL-based relational database server that employs the Microsoft Open Database Connectivity (ODBC) [16] protocols for client-server connections from individual wIBIS clients connected to the network. wIBIS clients run on PCs operating under the Microsoft Windows NT environment [16].

364
The hypertext environment of wIBIS (Figure 2) is somewhat similar to that of the gIBIS tool developed by Conklin and Begeman for UNIX systems [14], but offers a multiple document interface to the user so that more than one IBIS database can be open by the user at one time. If several separated wIBIS users are editing an IBIS database at the same time on the network, then when a new node is created by one user, it will immediately appear on the screens of the other users. However, only the original author (plus a group leader with "system" privileges) can edit the contents of an IBIS node or delete the node. In order to maintain order in the IBIS database, database locking is employed so that while an issue, position, subposition, or argument is being edited (the given dialogue box is open for that node), all the neighbouring nodes up to the nearest parent issue node are locked by that user and cannot be edited by anyone else.

Each IBIS database also contains information for each individual contributor permitted to participate in the discussion. In particular, each participant can configure the layout of the IBIS database as he or she sees fit, and the individual positions of all the nodes will be stored under his or her personal record in the database.

In order to associate the wIBIS tool with itIBIS usage, the tool will be able to read ASCII text versions of itIBIS and display the information as an IBIS network, as well as be able to generate an itIBIS text version of an IBIS network that was generated using wIBIS.

4 wIBIS syntax

The wIBIS tool that we present here has the basic IBIS nodes discussed above, plus additional notational considerations. One of these has to do with the idea of a subposition. Besides the basic node types of issues, positions, and arguments, there can also be a subposition node type in wIBIS: whenever a position can be viewed as essentially a composition of smaller elements, we represent the more elemental positions as subpositions. Thus wIBIS has four basic node types -- issues, positions, subpositions, and arguments. The relationships among these node types is indicated in Figure 3.
As with the conventional IBIS, it is possible in wIBIS to raise a new issue from any of the other node-types, including a subposition. The motivation for introducing a subposition node into wIBIS is so that positions that might have a composite structure can have arguments directed in support of or in opposition to individual elements (subpositions) of the position. Note that for simple positions there need not be any subpositions, and the original IBIS notation is sufficient. It is possible in principle, however, for subpositions to be composed of smaller subpositions, and so on. This is rarely necessary; we find that a single level of subpositions is usually sufficient.

A second addition to the original IBIS notation is the idea that sibling subposition or argument nodes may share an AND-relation or an OR-relation. Two subpositions that point to the same parent position may be related by an AND-relation or an OR-relation. The AND-relation between two (or more) subpositions indicates that both subpositions must be taken together for the parent position to be valid. For example a given position (P1) might be

P1: Build our system using Brand A computer.

This position might be based on two subpositions (SP1.1 and SP1.2)

SP1.1: Our principle customer should be the government

SP1.2: The government will be happier with systems built on Brand A computers.

Both SP1.1 and SP1.2 must be valid for P1 to be valid, so they are connected by an AND-relation. On the other hand when a position can be broken down into two independent subpositions, the subpositions are related by an OR-relation. An example might be

P2: Don't build on Brand A computers

SP2.1: Instead, build on Brand X computers

SP2.2: Instead, build on Brand Y computers.

Similarly two (or more) arguments may also be related by an AND-relation or an OR-relation. In this case, the two related arguments must be either all in support of or all in opposition to the parent position or subposition. For example the parent position might be

P1: We should hire outside computer programmers to build new system X5.

and two AND-related arguments for P1 might be

+A1: Our own staff programmers must finish project X2 before they start any new projects.

+A2: Waiting for the completion of project X2 will inordinately delay the start of project X5.

OR-related arguments, on the other hand, are independent of each other and characterise the relations between arguments of the original IBIS semantics.

We note that the distinction between subpositions and arguments can sometimes be hazy. When a complex position is broken down into its elements, the statement of subpositions can sometimes appear to be similar to arguments. The point to remember is that positions and subpositions are essentially assertions, whereas arguments are complex and suggest some implicit reasoning behind them. Since the wIBIS approach is designed to maintain a record of group decision-making that is intelligible to the group participants, the choice between using subpositions or arguments should generally be made on the basis of whatever appears to be more natural to the majority of group members.

The graphical representation of wIBIS AND- and OR-relations is shown in Figure 4. OR-relations are taken to be the default relations between sibling subpositions or arguments and are not represented graphically. AND-relations are represented by a connecting arc (with nodules at each end) between the two link-lines connecting individual subposition or argument nodes to their parent nodes.
5 Evaluations of wIBIS positions

Another feature that has been added to wIBIS concerns the recording of a measure of the perceived strengths of the arguments supporting or opposing particular positions. We believe that beyond recording the text of the issues, positions, and arguments of a discussion and which positions were ultimately selected, it can also be useful to keep track of the relative strengths of the arguments and how these have affected the perceived strengths of the positions or subpositions they supported or opposed. Consequently the argument nodes of a wIBIS record also have a field for recording the qualitative strength of the argument. Users may select one of nine qualitative values for assessing the degree to which a given argument is convincing. Behind the scenes, wIBIS stores a numerical value, between 0 and 1, that is associated with these qualitative assessment values (Table 1).

<table>
<thead>
<tr>
<th>Argument strength</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>conclusive</td>
<td>1.00</td>
</tr>
<tr>
<td>very strong</td>
<td>0.88</td>
</tr>
<tr>
<td>strong</td>
<td>0.75</td>
</tr>
<tr>
<td>moderately strong</td>
<td>0.63</td>
</tr>
<tr>
<td>medium strength</td>
<td>0.50</td>
</tr>
<tr>
<td>moderately weak</td>
<td>0.38</td>
</tr>
<tr>
<td>weak</td>
<td>0.25</td>
</tr>
<tr>
<td>very weak</td>
<td>0.13</td>
</tr>
<tr>
<td>fallacious</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 1. Argument strengths.

Although only the author of a node may edit the text of the node, all discussion participants may offer their opinion concerning the strength of a given argument. The IBIS database server maintains a mean value of all the votes cast by the users for a particular argument, and this average is used in subsequent evaluations of the argument's strength.

In addition to the strength of an argument, users also select a qualitative value for the degree to which the argument is relevant to the particular position addressed. These values are associated with the arc-link that connects the argument to the subposition or position. Note that an argument may be connected to more than one subposition or position, and so the relevance of that argument to each of the subpositions or positions to which it is connected will be stored on the corresponding arc-links. Seven values are available for selection (absolutely relevant, very relevant, moderately relevant, medium relevance, somewhat relevant, slightly relevant, absolutely irrelevant), and there are seven associated numerical values (1.0, .83, .67, .50, .33, .17, 0.0) that are stored in the IBIS database.

To determine the value or score for a particular subposition or position, an approach inspired by fuzzy logic is used. For a given subposition, each argument's mean strength value is multiplied by its mean relevance value to determine the argument's overall strength for the given arc-link. For all the supporting arguments, a combined (positive) numerical value is obtained by employing one of two fuzzy computations [17] (see Table 2).

<table>
<thead>
<tr>
<th>Method #1</th>
<th>X</th>
<th>Y</th>
<th>X AND Y</th>
<th>X OR Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>a</td>
<td>b</td>
<td>min(a,b)</td>
<td>max(a,b)</td>
</tr>
<tr>
<td>#2</td>
<td>a</td>
<td>b</td>
<td>a * b</td>
<td>a + b</td>
</tr>
</tbody>
</table>

Table 2. Fuzzy combinations.

For a given IBIS network only one of the above two computational methods is used throughout. Using Method #1 (the default), arguments whose arc-links are associated by an AND-relation are combined by taking the minimum of the values. The results of these evaluations plus all the argument values whose arc-links are OR-related are combined by taking the maximum of the values. This gives a combined value for all the supporting arguments, which is taken to be a positive number. For all the opposing arguments, a similar calculation is performed, but the resulting value is then made a negative number. This number for the opposing arguments is then added to the result for the supporting arguments to determine the overall score for the subposition.

For positions that do not have subpositions, the evaluation is done exactly as described above for evaluating subpositions. For positions that do have subpositions, the
contributions from the subpositions must also be evaluated. This is done by taking the subposition scores and combining them, using either method #1 or method #2 from Table 2, appropriately with respect to AND and OR-relations that may exist between the subpositions. Unlike arguments, the relevance of subposition arc-links is not considered, since the relevance is always assumed to be high (1.0) for subpositions.

Using this approach, which is based on all the votes cast among the discussion group participants concerning the relative strengths of the arguments that have been put forward, the wIBIS tool will determine a numerical value for each position in the IBIS database. The position that has the highest value is assumed to be the winning position.

One further complication that must be considered is when new issues are generated during a wIBIS discussion. A new issue may be spawned during a wIBIS debate and be either a standalone issue or be connected to an existing position, subposition, or argument. If it is connected to an existing node, then its connecting arc-link may be designated "is-suggested-by" or "questions". If the arc-link is "questions", then the new issue is calling into question the existing position, subposition, or argument. In that case (and only that case) the score or strength of the corresponding position, subposition, or argument is held in abeyance until the questioning issue is resolved. In other words, the position, subposition, or argument whose very existence is questioned by an issue does not have its current score contribute to any other evaluations until the questioning issue is resolved. When the questioning issue is finally resolved, it may be necessary to go back and re-evaluate the assessed strengths associated with nodes that are connected to the just questioned node.

When the evaluations are finally made, those positions with the highest scores should, in the best of all possible worlds, correspond to those positions that the group actually prefers. Of course upon reflection, the consensus of the group may well be that another position, whose value is not the highest, is actually the preferred option. This can come about because the valuation mechanism in wIBIS is not a perfect reflection of the judgement procedures of the group and because of inherent uncertainties in individual assessments of strengths along the way. However, we believe that this semi-structured way of recording the qualitative assessments of the strengths and relevances of arguments can be a useful option with respect to group decision-making. It can be particularly valuable when a group, perhaps because of newly acquired information, goes back to review the manner in which a previous decision was made. Non-selected positions that were near-misses, can have their relative assessments recorded in the IBIS record. As new information comes to light, near-miss positions can be identified and re-evaluated for possible renewed selection. Moreover, new group members who join a group in midstream may benefit from being able to examine assessments of positions and arguments with respect to issues previously considered by the group.

6 Conclusions

The wIBIS group collaboration tool under development supports a semi-structured approach for facilitating and recording group decision-making. It goes beyond previous approaches based on the IBIS method by introducing new structural features and mechanisms for recording the impact of perceived strengths and relevances of arguments. Prototypes of the tool exist, but more robust versions of the software are under construction.

The intention is to test the software and develop associated procedures in the context of software engineering projects at the University of Otago. Experience at Otago with the itIBIS approach [13] suggests that the wIBIS tool will contribute to improved software engineering processes, particularly with respect to software maintenance activities in those projects. In addition, it is our intention to use wIBIS in connection with current research projects that involve remote collaborators at other university sites in New Zealand.

References


[16] Microsoft Corporation, Redmond, WA.