5 Hypermedia Support for Argumentation-Based Rationale: 15 Years on from gIBIS and QOC

S.J. Buckingham Shum, A.M. Selvin, M. Sierhuis, J. Conklin, C.B. Haley, B. Nuseibeh

Abstract: Having developed, used and evaluated some of the early IBIS-based approaches to design rationale (DR) such as gIBIS and QOC in the late 1980s/mid-1990s, we describe the subsequent evolution of the argumentation-based paradigm through software support, and perspectives drawn from modeling and meeting facilitation. Particular attention is given to the challenge of negotiating the overheads of capturing this form of rationale. Our approach has maintained a strong emphasis on keeping the representational scheme as simple as possible to enable real time meeting mediation and capture, attending explicitly to the skills required to use the approach well, particularly for the sort of participatory, multistakeholder requirements analysis demanded by many design problems. However, we can then specialize the notation and the way in which the tool is used in the service of specific methodologies, supported by a customizable hypermedia environment, and interoperable with other software tools. After presenting this approach, called *Compendium*, we present examples to illustrate the capabilities for support security argumentation in requirements engineering, template driven modeling for document generation, and IBIS-based indexing of and navigation around video records of meetings.

Keywords: rationale capture, cognitive overhead, hypermedia, argumentation, compendium, IBIS, QOC

5.1 Introduction and Overview

Few would disagree with this book's opening chapter that the systematic management of design rationale (DR) is not yet common software engineering practice. By extension this applies to the particular flavor of DR with which we work, namely the IBIS/QOC approaches to creating graphical argumentation maps of design deliberation (reviewed in Chap. 1 and classed as "prescriptive, intrusive" in nature). It is the "intrusive" nature of such notations that represent an obstacle to adoption (we will unpack in more nuanced terms what this means), and which has led many to the conclusion that DR based around explicit, graphical argument maps is yet another failure of exciting research ideas to overcome the harsh realities of actual day-to-day practice.

This chapter argues that the story is more complicated but more hopeful. Since the late 1980s, through business and industrial case studies, detailed lab analysis, and continual design refinement, we have been reflecting on the set of interacting factors which together can "make or break" them in the heat of collaborative analysis, modeling and design. The *Compendium* technique and tool has matured to the point where a steering group (a subset of the authors) is coordinating the development of an open source Java hypermedia IBIS mapping tool, with an international user community spanning government, NGOs, education and business, documented case studies, and training courses and online resources. Clearly, there are no silver bullets, but progress has been made since the intense activity that led up to the first DR book in 1996, and the subsequent decline in activity as the challenges of truly embedding argumentation-based DR in work practices sank in. In particular, although quality software support is required, it turned out to be the human factors that required closer attention.

The objective of this chapter is to update the software engineering community on how and why the QOC [20, 21] and gIBIS approaches [10, 11] we helped to create originally, have subsequently evolved into the current Compendium approach and tool.

5.2 The Vision

Chapter 1 has already provided a broad summary of the rationale behind Horst Rittel's IBIS, and the ways on which software engineering DR researchers have appropriated and extended it, so we will not duplicate that review. What we can add by way of introduction is an amplification of the rationale behind "prescriptive, intrusive" approaches, whose goal is to support and improve design reasoning. A converging strand of research in the history of computing to augment intellectual work, Rittel's work converged with that of computing pioneers such as Vannevar Bush, Douglas Engelbart and John Seely Brown to forge an exciting vision of the power of cognitive, collaborative tools to both capture and augment design reasoning. The research community envisioned that hypertext groupware would make it easy to capture and structure the spectrum of informal and formal knowledge that goes into DR. Designers could capture their deliberations on the fly during design sessions. Visual networks of icons would be intuitive enough to realize the vision of participatory analysis amongst diverse stakeholders, who would not need to learn cryptic formal schemes in order to contribute tangibly to system requirements. Captured DR's might be reusable, or at least would contribute greatly to the process of maintaining and evolving that system over time by providing a skeletal group memory to help reconstruct what led to a decision.

We are simplifying a little for brevity (we review the roots to the field in more depth in [3]), but something close to the above vision was very much the driving energy in the decade from about 1986 in many leading computer science and HCI research groups. As will become clear, we consider many aspects of this exciting vision to merit continued pursuit, since providing traces of complex intellectual work has enormous potential. However, as we will elaborate in Sect. 5.3, great attention needs to be paid to the socio-technical skills required to successfully use such an approach, and there was naivety in some of the early assumptions. In particular, we had to solve "the DR capture problem."

5.3 The Design Rationale Capture Problem

The capture problem is the specter haunting all DR efforts (indeed, all knowledge management efforts attempting to meaningfully capture elements of human reasoning and discourse). How does one acquire quality input to a rationale management system, without disrupting the very process it is designed to support, or without having to employ dedicated scribes who do nothing but maintain rationale libraries?

The cost–benefit tradeoff is a slippery tightrope to walk, and has focused our energies on a "value now, value later" imperative. As Grudin [13] has pointed out, there cannot be a disparity between who invests effort in a groupware system, and who benefits. No designer can be expected to altruistically enter quality DR *solely* for the *possible benefit* of a *possibly unknown person* at an *unknown point in the future* for an *unknown task*. There must be immediate value. The difficulty, of course, is that it is not merely a "capture" problem, but "useful capture". One could minimize the capture effort and simply video record every design meeting, but this would not render a useful archive. Computationally tractable structure must be added by some means. Extracting useful content automatically from multimedia meeting records is an active research area, but very challenging. Later, we will report on the synergy of combining the richness of video-based DR with argumentation-oriented approaches, but let us first focus on the specific capture problem associated with the latter.

Very soon after "idea processing" visual hypertext systems such as NoteCards [14] and gIBIS [10] began to be used for structuring ideas, reports began to emerge of "cognitive overhead". A 1994 survey [3] found comparatively weak evidence regarding usability and utility compared to what might have been expected given the scale of system development efforts. A later survey echoed this, highlighting the pattern of failure in many kinds of interactive systems that assume the willingness of users to structure information [30]. The ray of hope that somehow we might find just the right balance of intuitive user interface, natural representation scheme, and fast computers began to dim, and many researchers moved on to other challenges.

Nonetheless, encouraged by the limited success of the gIBIS prototype in an industrial case study [11] that the problems stated earlier were surmountable, the early 1990s saw the launch by Conklin and colleagues of a commercial software tool that combined graphical hypertext, IBIS and groupware capabilities. The *QuestMap* Windows single user and groupware product made a mark in the hypertext and groupware communities, and even resulted in a few isolated cases of extended industrial-strength use [8]. However, this product ultimately succumbed to market pressures, and is no longer available. Much was learnt from this episode, in particular an appreciation of the value that can be added in design meetings *once people have learnt the meta-cognitive skills of using IBIS*, some of whom may then appreciate quality software support to overcome the limits of mapping on paper, whiteboards, or a generic drawing tool. Let us consider the nature of this skill in more detail.

5.4 Understanding Cognitive Overhead

We have studied the issue of "intrusiveness" (see Chap. 1) in depth via detailed, video-based analyses. Moreover, we are interested in characterizing not just the initial learning curve (which is what most people have focused on) but also the nature of highly skilled practice.

One study of beginners focused on software designers learning to use QOC (on paper), and provided a detailed account of how designers must learn to manage four interleaving cognitive tasks [2]: *unbundling* (identifying and separating constituent elements of ideas which have been 'bundled together' when they were initially expressed, but which from an argumentation perspective need to be teased apart), *classification* (deciding whether a contribution is a Question, Option, or Criterion), *naming* (labeling the new contribution succinctly but meaningfully), and *structuring* (linking in a new element to other ideas).

Should we be surprised that this feels like extra work? In introducing subsequent video analyses of these designers, Buckingham Shum et al. [4] argued that "On reflection, reports of cognitive overhead should not be surprising. The basis on which [concept mapping tools] work is that deeper understanding of a domain comes through the *discipline* of expressing knowledge within a structural framework, working to articulate important distinctions and relationships."

At this point, however, although Buckingham Shum had a lab-based account of when QOC seemed useful or obstructive, he had a poorly developed conception of how to turn that effort to the group's advantage. This was a "missing piece of the jigsaw" that some of the other authors of this chapter provided: Conklin from a facilitation perspective developed during the QuestMap/IBIS consulting period, and later, Selvin and Sierhuis from a collaborative modeling perspective [25]. Section 5.5 describes how these insights combine in our current understanding.

Beyond the initial learning curve for novices, we have recently begun to characterize the learning curve as one gains proficiency. What does it mean to become an expert in mapping IBIS structures to support problem solving and design cognition? Selvin [26] has characterized the kinds of skills that such a practitioner needs to possess, and more recently has begun to articulate, based on video analysis of Compendium in use in web-mediated meetings, the kinds of 'moves' that a mapper can make to assist the team in the problem solving, and the associated skills [27].

To summarize, DR that yields insight into the complex ideas and arguments that may lie behind a decision does not come "for free": effort must be invested at some point in the rationale management lifecycle.

5.5 Compendium

Compendium represents our current effort to take the raw conception of IBIS, and deliver it in a form where it can smoothly integrate in the 'matrix' of everyday tools and practices. Our technical objective is to provide a robust, open environment in the IBIS/argumentation-based DR paradigm, which can then be integrated with other DR paradigms and tools, such that services can be implemented over the extended-IBIS representational substrate.

Our approach to the capture problem is to invest rationale structuring effort primarily at the point of capture, validating it with the key stakeholders. This capturing process serves the stakeholders' needs to understand each other and know that their viewpoint has been heard. This co-evolves a shared picture of the problem, possible ways forward, and the rationale for deciding how to proceed. This is supported by a software tool which can further lower the data entry overhead: data already entered in other key tools can be imported, and data entered in the rationale tool can automatically populate other tools, or generate documentation.

There are three dimensions to understanding Compendium: (1) its functionality as a hypermedia concept mapping environment, (2) how it uses IBIS to support collaborative modeling of a problem using any conceptual framework, and (3) in the context of mapping ideas in real time during a meeting, the role of the person doing the mapping to facilitate the task at hand.

5.5.1 Hypermedia Concept Mapping

Compendium comes "preloaded" with node and link types for IBIS, derived from QuestMap's interpretation of the notation, for connecting key issues, possible responses to these, and relevant arguments. Figure 5.1 shows the default node types, which include additional nodes beyond IBIS for *Lists and Maps* (containers for nodes), *Decisions, Notes,* and *References* that can hyperlink to open a web page or other document**.**

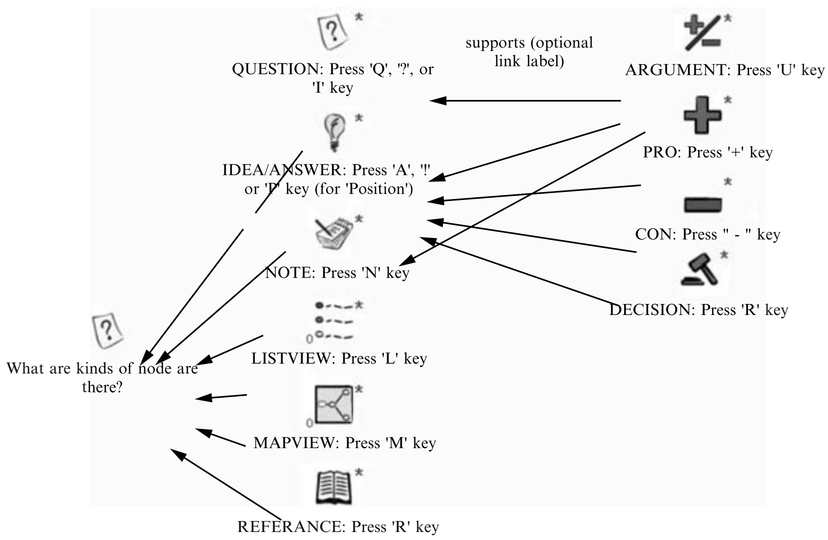

Fig. 5.1. IBIS plus additional node types rendered in Compendium. Any application document or website can be dropped in to create a hyperlink. Nodes can contain text content, and links can be labeled if desired

Figure 5.2 shows a DR extract from a project meeting, in which an issue is raised, two options explored, and one justified. Figure 5.3 shows the use of Compendium simply to record decisions (about metadata). While these might simply have been recorded in a word processor or slide tool, such tools do not support (i) the possibility of capturing important discussion/rationale if it arises, or (ii) the reuse of a decision in subsequent other contexts – see the links on the bottom node to its other appearances in the database. Users can also define their own custom modeling language, by building their own palettes of icons (called Stencils) and relational types (Linksets). This is not currently a full meta-modeling tool, however, in that constraints cannot be specified between nodes and links: any two nodes can be linked using any linktype.

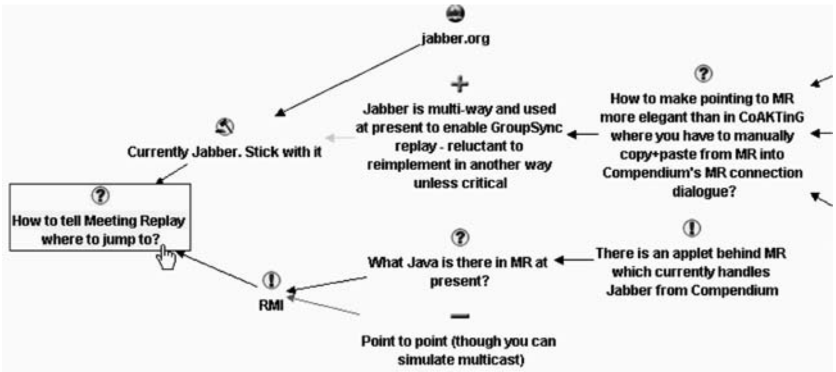

Fig. 5.2. Extract from a software design meeting, in which Compendium is used to map issues, options, arguments, the decision, and a relevant website. (This meeting was an Internet video conference, with Compendium viewed by participants via a desktop sharing application)

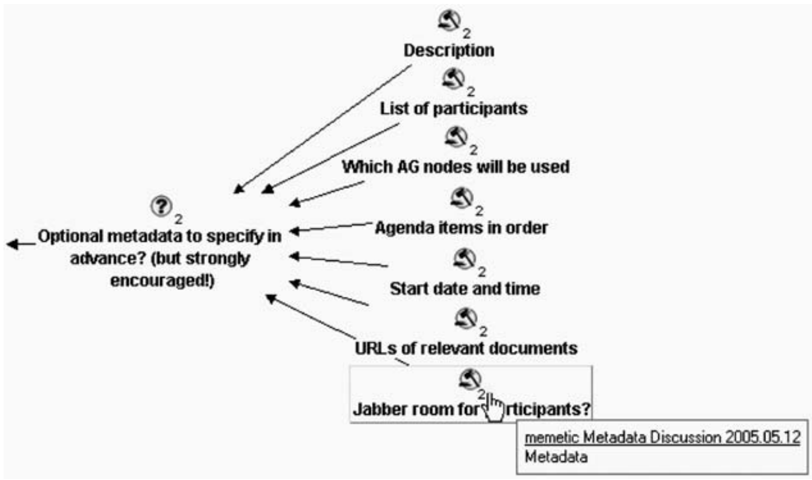

Fig. 5.3. Recording decisions (in this case without any significant rationale) in Compendium. Rolling the mouse over the digit on a node displays a link menu to other maps which contain the node

Compendium maps are not 'flat' drawings, but views onto a relational database that can be rendered in multiple formats. A given node (e.g., representing an idea, argument, entity, or document) can appear and be updated in multiple views. Since any application document or URL can be dragged and dropped into a map as a *Reference* node, so an external document can be linked into one or more discussions and tracked – that is, given one or more meaningful contexts where it plays a role. Corrections or updates to a node are immediately updated in every context in which it appears. This provides precisely the representational capability needed to build semi-structured models in which a particular object is systematically reused (e.g. an idea, plan, person, system, location).

Compendium is implemented as a Java application that can swap between either the MySQL[2] or Apache Derby[3] relational databases. XML export/import enables data between clients using a Document Type Definition (DTD), and in research projects, interoperability has been extended to the semantic web's RDF. An Applications Programming Interface (API) enables other systems to read and write to the database directly, so concept maps can be generated from another data source or interpreted for processing by another system. Full groupware capabilities are not yet implemented, although demand for this is growing. A shared database can be maintained either by using an MySQL server, or in experimental versions, through mirroring databases synchronously between two clients over the Internet, using the Jabber XML messaging protocol (which also enables Compendium to send and receive nodes from Jabber instant messaging clients on any device[4]). The most common means of sharing data is via XML. All maps can be published to the Web as interactive image maps or linearised as HTML outline documents.

5.5.2 Overlaying Conceptual Frameworks

Compendium extends the use of IBIS from modeling a discussion, to more systematic modeling of a problem. A modeling approach focuses attention on a specific subset of issues and information, it may constrain the kinds of options one considers, and it may also focus attention on how one assesses them. In Compendium, a modeling approach is translated into an *issue template*, which can also be created simply to deal with any well understood situation where there is a recommended approach to proceed, for

[2] See http://www.mysql.com.
[3] See http://db.apache.org/derby.
[4] See the CoAKTinG Project: www.aktors.org/coakting

instance, from best practice or a standard operating procedure. Figure 5.4 shows a template for modeling a business process, prior to its instantiation.

Templates were created to support structured modeling within the IBIS framework, which by definition moves the tool into the space of reasonably well-structured problems. These are much easier contexts in which a beginner can use Compendium, since they are provided with a representational scaffold for working through a set of predefined issues. Assuming the meeting has faith in the template, when its questions have been answered, the meeting can be confident that they have made some progress. A hallmark of the approach is, however, the ability to break from formal and prescribed representations into informal, ad hoc communication, incorporating both in the same view if that is helpful to the participants (e.g. "in this context we should really ask a different question…"). Hypertext nodes and links can thus be added either in accordance with templates or in an opportunistic fashion.

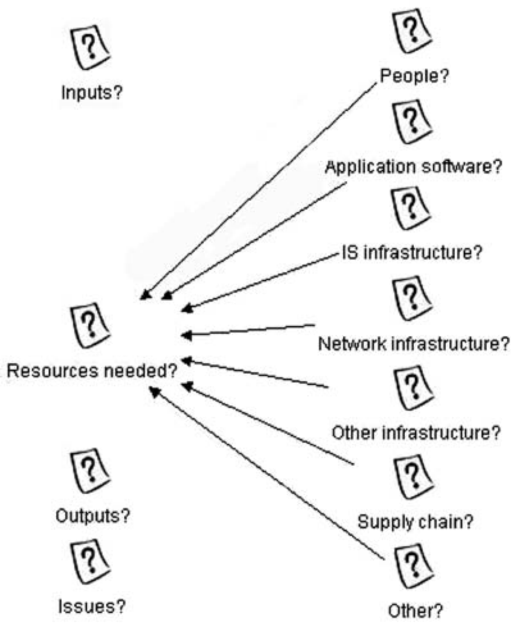

Fig. 5.4. An issue template that can be imported when required, linked to other views, and tagged with metadata. The issues raised are now stepped through, linking in answers and arguments as appropriate, and breaking out of the template if necessary to capture unexpected material, ideas or argumentation

A complement to issue templates are *tags* (metadata keywords) assigned to any concept (node) in the database to show connections through membership in a common category. Tags serve to specialize a node type with as many attributes as required for it to play multiple roles in different contexts. At the end of the session all of the nodes so marked can be harvested. In modeling, nodes sharing a tag are often tracked as a 'catalogue' of nodes stored for future reuse. Tags may reflect generic meeting processes (e.g. *Action-Jane*), or may be driven by an underlying methodology that Compendium is being used to support (e.g. *Data-Provider*). Alternatively, ad hoc tags can be created on the fly, to reflect the emergence of a new theme.

As reviewed in Chap. 1, it has long been recognized that DR cannot exist in a vacuum but must be connected to relevant design artifacts and views. This can be done by dropping an application document or Web URL into Compendium to create a hyperlinked Reference node, but tags provide a mechanism for deeper level connections. Since nodes may originate from other systems (written directly via the MySQL API or manually imported as XML) it is possible to use tags to mirror attributes of the domains which these external systems model. The world of IBIS is thus connected via the simple mechanisms of templates, tags and hyperlinks to any other relevant domain, from end-user scenarios and organizational processes, to software architecture and project management.

5.5.3 Meeting Facilitation Through Dialogue Mapping

Turning to the third element of the approach, *facilitation*, Dialogue Mapping[5] is a set of skills for mapping ideas as IBIS structures in order to support the analysis of wicked problems, as defined by Rittel.[6] It has turned out to be a critical development in argumentation-based DR, since it provides a way to negotiate the capture bottleneck: the structure required to construct useful DR is added in real time during the meeting, adding immediate value to the participants, but also creating a record. Mapping ideas in IBIS during a meeting is unquestionably an acquired ability, but equally, one that can be learnt (there is an international Compendium user community). This was the key oversight in early argumentation-based DR research, which experimented with small-scale demonstration examples,

[5] For an introductory account of how Dialogue Mapping is used during a meeting, see the fictional scenario at www.cognexus.org/dmepaper.htm.

[6] Churchman [7] appears to be the first person to have published the term 'wicked problem', in 1967, but in this brief editorial, he credits Rittel with the term.

and did not invest enough in what we now think of as hypermedia/IBIS "literacy". See Conklin [8] for a longer introduction to the craft skill involved in choreographing meetings and representational activities that we introduce later, and [9] for an extended resource.

The facilitation perspective places the Dialogue Mapper in a potentially very powerful role, quite the opposite of the lowly "DR scribe" whose role runs the risk of relegation to minute-taker or documenter. The mapper actively crafts structures on a shared display screen that both capture the meanings and ideas of the group and reflect back to it the larger implications of their thinking. There is a spectrum of how strongly discourse is mediated via this display (described in the DR continuum [3]). It may be used to periodically summarize and review "normal discussion" (e.g., at decision time), screens can be shown to reflect on progress, or the discussion and the map can "dance" – each shaping the other. It is hard to convey this in writing, but we contend that it exemplifies the kind of synergy between tools and sensemaking that was envisioned by the developers of early "idea processing"/DR hypertext systems.

To borrow a musical metaphor, there are several shifts in the "rhythm" or "timbre" of a meeting when Compendium is used well:

– *Beneficial slowing down*. A complaint sometimes heard when argumentation-based DR is first introduced to meetings, is that it disrupts the flow of the meeting [2,12]. When done appropriately, however, we find that it can be extremely beneficial to "disrupt" dysfunctional dynamics by focusing attention on a feature of the hypertext map. After a period of use, people become noticeably unhappy when their contributions are not mapped, because once captured on screen, they know that their view has been heard, correctly recorded, and will be harder to ignore when the map is assessed at decision time.
– *Depersonalization of conflict*. When ideas and concerns are mediated via a shared display, challenges to positions assume a more neutral, less personal tone. In situations where there are competing agendas, it helps participants clarify the nature of their disagreement (e.g., the definition of 'the problem'; understanding different criteria of "success"). We have seen Compendium defuse meetings which otherwise looked to be polarized, for instance, by surfacing the different connotations of a particular question. Recent work with Compendium has deployed specifically in conflict resolution and mediation [24].
– *Flexible rhythmic review*. To a surprising degree, collaborative knowledge work can be characterized as "group list processing." Whether the list is a set of requirements, budget items, or action items, a common activity is group review of a list of potentially complex elements. While

some items draw little comment, others can lead into deep discussions and even debate. A good mapper can establish a "call and response" rhythm with the group, creating a sense of shared purpose and momentum. When occasional elements lead to intense discussions about meaning, or spark disagreement among group members, the Compendium practitioner can open a new map and keep mapping or modeling the new conversation. With the new issues captured in the shared display, the group can return to the previous review task without losing momentum.

5.6 Reasoning Services and Verification

Referring back to Chap. 1's lists of requirements for future software engineering DR environments, in principle, Compendium's functionality could contribute to any software engineering activity and phase where issue-based deliberation or modeling is required. But our interest in collective sensemaking clearly has a particular orientation to the tasks listed under "supporting collaboration".

The evolution of Compendium from QuestMap and gIBIS has, however, opened the door technically and conceptually for integration with other software engineering tools and DR tools. Compendium does not come with any preprogrammed verification services that can perform structural checking (which could for instance be used to provide a DR service such as dependency management). Given the breadth of our user community, which goes beyond just software engineering and DR, our strategy has been twofold: (1) to create an open architecture (unlike QuestMap's) with a standard SQL database, XML DTD, and Java source code to enable other groups to access all levels of the system functionality and data; (2) to provide a visual user interface and generic issue-oriented representational substrate as described earlier (extended IBIS, a customizable visual language, tags, templates, node reuse, graphs, and lists) which can be appropriated to express many different kinds of design knowledge.

We have already shown (in the mission planning domain) that Compendium can be integrated with a tool that uses a more formal issue ontology and planning engine to reason about available options and constraints on issues [31]. We are now beginning to explore the requirements for a new layer over the generic environment, which would extend Compendium with services to support argumentation around the security of requirements specifications, a domain which provides the worked example described shortly.

5.7 Revisiting 'Intrusiveness'

After over 15 years' deployment in the field (gIBIS, QuestMap, Compendium), there is now a response to those who have argued that the need to be skilled in the use of IBIS is a fundamental weakness of the approach.

First, this has now been shown to be an effective strategy to negotiate the cost/benefit tradeoff associated with IBIS and its descendants: people *can* learn to do this, and can construct representations which their peers value both in the meeting, and afterwards. All of this evidence is from the field, often anecdotal from practitioners who are not interested in writing research papers, but experiences are beginning to be documented [6, 8, 23, 25, 27, 29]. Second, like any other complex artifact (whether a software tool, a physical tool, or a musical instrument), Compendium yields greater benefits with practice.

That being said, a DR approach is of no use if people cannot learn it in a reasonable period of time. The "facilitation" perspective has proven to be an important step forward in providing us with a language and orientation to describe to new users how personal and collective deliberation, a subset of which will be DR, can be captured. Two-day Dialogue Mapping training courses and on-line tutorials are available.[7] Experience to date suggests that novices can gain value from the tool as a personal concept mapping aid within days, while confident, effective use in meetings takes longer, although we have seen people use it effectively in meetings with minimal practice. Expert Compendium practitioners may be needed in contentious, unstructured contexts, but less experienced users can use the approach in more stable contexts by completing templates.

It is by no means the case that everyone who attends the two-day training course goes on to use the approach at work, but we are now supporting a sizeable online user community, with over 5,000 downloads of the application to date. Several consulting companies currently use Compendium to support clients in clarifying and integrating multistakeholder requirements in wicked problem contexts, and the approach is also in internal use within both commercial and nonprofit organizations.

[7] Compendium training: www.CompendiumInstitute.org/training/training.htm.

5.8 Examples of Compendium in Use

Compendium has been used on over 100 projects during the last 10 years[8] some of which are concerned with software and broader socio-technical systems design, though by no means all of them. Readers seeking empirical evidence of the approach's learnability and effectiveness from analyses of real world cases in the field can review [6, 8, 23, 25, 28, 29], while close video analysis is found in [27]. Pre-Compendium, video analysis of the QOC approach can be found in [2, 3, 4]; Conklin has reported on a large deployment of gIBIS [11] and a decade long deployment of QuestMap [8], Carr [5] has used QuestMap to teach legal argumentation, while Isenmann and Reuter have reflected on HyperIBIS [17] and Fischeret al. [12] on IBIS and PHI.

In this section, we present a small software engineering worked example that illustrates Compendium support for a particular form of argumentation in software engineering. We then extend this with two different examples to show first, the use of templates to drive organizational modeling and generate documentation, and secondly, the use of Compendium maps to index, navigate and query videos of meetings.

5.8.1 Security Satisfaction Arguments in Compendium

Satisfaction arguments [16] need to be constructed when analyzing the security needs of a system. One begins by representing the system using Jackson's problem frames [18], adds security requirements in the form of constraints [22], and then attempts to argue that the system satisfies the security requirements. These arguments are the satisfaction arguments.

In most cases, an initial argument will not be sufficiently convincing for one or more reasons:

1. The argument depends on properties of the system that are not currently known
2. The behavior of *domains* (the actors/components in the system) is not sufficiently understood
3. Domains required to satisfy the security requirements are not included in the system

To address the first two cases, the analyst might choose to go deeper into the system with the goal of better understanding the behavior and properties of the domains in the system. Unfortunately, this process can go

[8] Compendium case studies: www.CompendiumInstitute.org/library/library.htm.

on for a long time and, in the end, be inconclusive. At some point the analyst will decide to *trust* that the stated behavior and properties are as described. These decisions are called *trust assumptions* [15], and become an integral part of the satisfaction argument.

To support this kind of modeling, a new Compendium *Stencil* was created to provide a palette of Problem Frame modeling icons, specializations of the generic *Reference* node. If desired, a specific relational vocabulary (*Linkset*) can also be defined to provide labeled edges.

Consider a simple human resources personnel information display system. The proposed system has one requirement: *provide the HR data requested by a user*. Security goal analysis [1, 19, 24] results in one security requirement: *only to HR staff*. A problem diagram is constructed.

The attempt to construct a satisfaction argument that data is indeed provided only to HR staff shows that the analyst does not have sufficient information. One cannot answer the question *How do we know that "Users" consists of HR staff?* The problem information is not complete, and therefore the problem diagram must be changed. The choice made is to add authentication and authorization to the problem. The resulting problem diagram is shown in Fig. 5.5, and Fig. 5.6 the revised satisfaction argument.

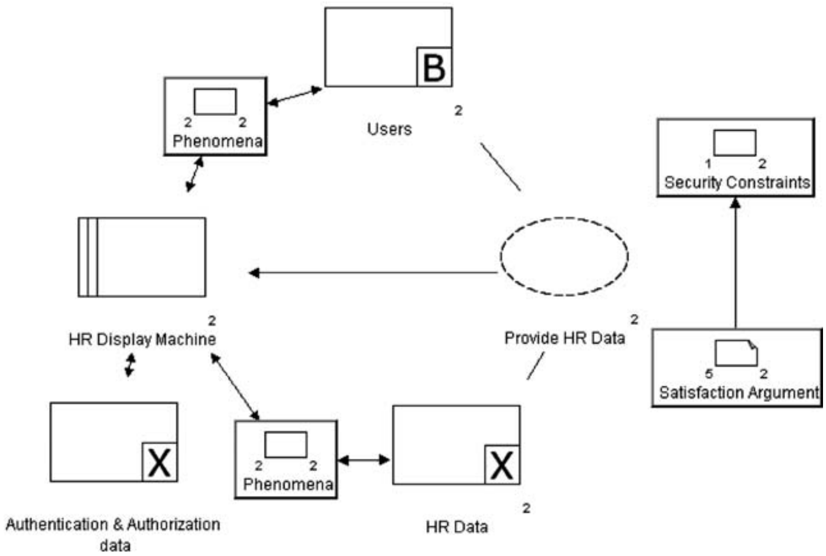

Fig. 5.5. Problem diagram with authentication

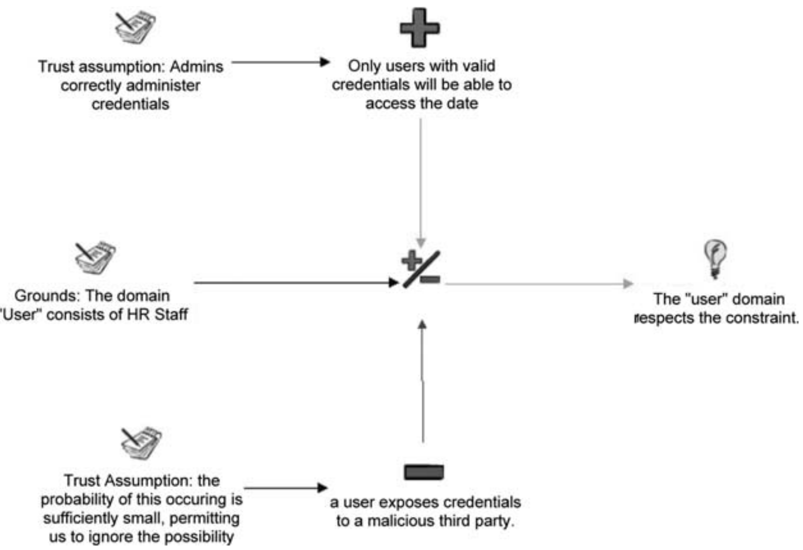

Fig. 5.6. Map of the satisfaction argument

The process by which the trust assumptions were agreed on is not shown in Figs. 5.5 and 5.6, but this could of course have been supported by Dialogue Mapping, possibly driven by a template (see the next example). Furthermore, if the design meeting was recorded on video, then the maps could become indices back into the video (third example).

5.8.2 From Template-driven Modeling To Documentation

Another case study [29] documented Compendium's use in a time-pressured initiative to conduct an enterprise-wide risk assessment for a Year 2000 Contingency Plan. In this project, as in many others, one of the most common purposes of meetings was to advance a project deliverable of some sort, in this case to generate organizational documents. Figure 5.7 illustrates how an IBIS map served first as the participatory user interface to elicit information from domain experts, after which it was then exported to a data flow diagram, and a requirements specification text.

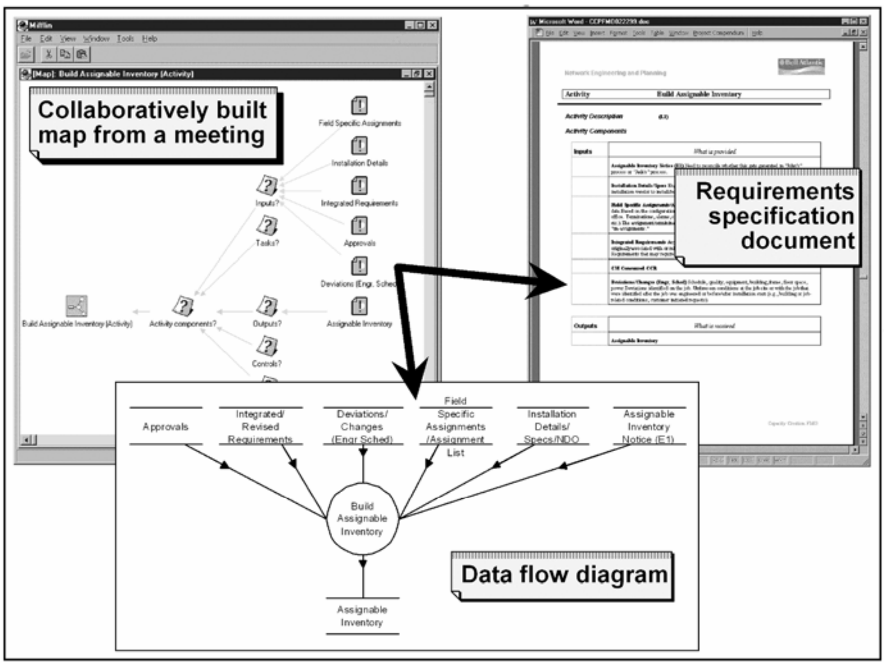

Fig. 5.7. Generating two alternative forms of documentation from a Compendium issue template

5.8.3 Rationale Management Via IBIS-Indexed Video

Our second extension to the worked example illustrates a recent dimension to meeting and rationale capture: Compendium integration with meeting videos. In the context of NASA mission planning [6], a multimedia Meeting Replay extension to Compendium was developed to assist the indexing and navigation of the meeting videos to assist one team's understanding of another's meetings, decisions, and rationale (Fig. 5.8).[9]

[9] Developed by the University of Southampton and the Open University as part of the *CoAKTinG* project: www.aktors.org/coakting.

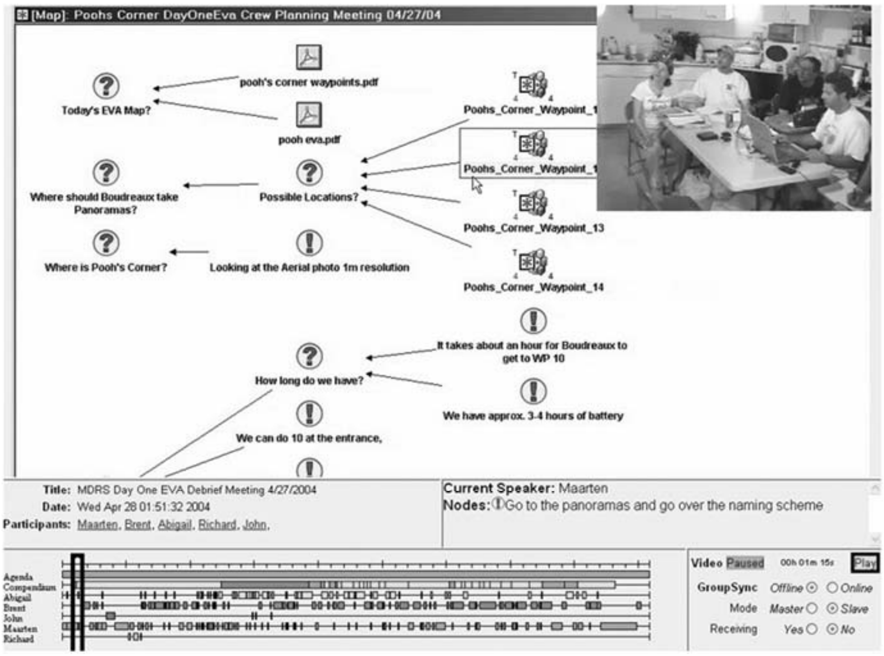

Fig. 5.8. Compendium-based Meeting Replay tool to help the science team on Earth recover the rationale behind the Mars crew's analysis and decisions

The upper region of Fig. 5.8 shows the video of the crew's meeting inset into the Compendium map they are building. The lower region contains summary information about the meeting: who was there, who was speaking, the agenda, and an overview of the current topic (derived from the Compendium map). Some of this information is presented as a timeline, providing a visual index for an RST member to navigate the video, jumping to relevant or interesting parts of the discussion by clicking on the timeline or moving the slider. As well as being able to navigate using the event streams at the footer, Compendium was extended to support conceptual navigation: thus, to see discussion prior to the recording of a particular *argument*, one can click on this node in the Compendium client and the replay jumps to the point in the meeting shortly before that node was created. Work is now under way to develop this infrastructure for wider use.[10]

---

10 The *Memetic* project: www.memetic-vre.net.

5.9 Lessons Learnt and Conclusions

In one sense, the whole of this chapter is an extended account of 'lessons learnt about the human factors of IBIS tools.' The vision of computational aids for design deliberation in the face of ill-structured, 'wicked' problems is an exciting one, but 'cool tools' alone cannot deliver this vision. The technologies of hypertext, digital video, and open standards for interoperability provide a powerful infrastructure, but to move from designers' fluid discussions to structured rationale representations, designers must become skilled with DR tools. Reluctance to persist long enough to gain some fluency with these new tools and their languages will result inevitably in the familiar complaints of intrusiveness. We have sought to show that the art and craft of DR – at least DR of this particular sort – is to know how to use the tools well enough that they are constructively disruptive, delivering immediate value to those using it, as well as supporting longer-term memory.

We recognize of course that there are representational limits to this particular paradigm, and organizational obstacles to the very idea of DR capture, as reviewed in Chap. 1. We have thus sought to assist in technical integration with other forms of rationale management tool. At this point, however, we do not yet have any examples to report, and welcome approaches from groups interested in collaboration.

In conclusion, as one would expect from the broad conception of "wicked problem," and the generic nature of IBIS as a representational scheme, Compendium is now finding application in many domains other than software engineering, but this is a virtuous circle: as the approach and infrastructure evolve to meet the challenges of new domains, they in turn provide new methodological insights (e.g., the nature of practitioner expertise; the disciplined use of templates) and practical functionality (e.g. data interoperability; modeling stencils; improved usability; document generation). Together these should assist the integration of argumentation-based rationale management with other forms of rationale, and the other tools of software engineering.

Acknowledgments. We are grateful to the book's editors and reviewers for feedback that has improved this chapter. The work reported has been funded by Verizon, NASA, The Open University, and public research grants from the UK Engineering and Physical Sciences Research Council (CoAKTinG Project) and Joint Information Systems Committee (Memetic Project), to whom we are indebted.

References

[1] Antón AI, Potts C (1998) The use of goals to surface requirements for evolving systems. In: Proceedings of the 20th International Conference on Software Engineering. Kyoto Japan, 19–25 April, pp. 157–166

[2] Buckingham Shum S (1996) Analyzing the usability of a design rationale notation. In: Moran TP, Carroll JM (eds.) Design Rationale: Concepts, Techniques, and Use. Lawrence Erlbaum Associates, Hillsdale, pp. 185–215

[3] Buckingham Shum S, Hammond N (1994) Argumentation-based design rationale: what use at what cost? Int. J. Hum.–Comput. Stud., 40(4), 603–652

[4] Buckingham Shum S, MacLean A, Bellotti V, Hammond N (1997) Graphical argumentation and design cognition. Hum.–Comput. Interact. 12(3), 267–300

[5] Carr C (2003) Using computer supported argument visualization to teach legal argumentation. In: Kirschner P, Buckingham Shum S, Carr C (eds.) Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-making. Springer, London Berlin, Heidelberg New York

[6] Clancey WJ, Sierhuis M, Alena R, Berrios D, Dowding J, Graham JS, Tyree KS, Hirsh RL, Garry WB, Semple A, Buckingham Shum S, Shadbolt N, Rupert S (2005) Automating CapCom using mobile agents and robotic assistants. American Institute of Aeronautics and Astronautics 1st Space Exploration Conference, 31 January–1 February, Orlando, FL. Available from: AIAA Meeting Papers on CD-ROM

[7] Churchman, C (1967) Wicked problems (Guest Editorial), Manage. Sci., 4(14), 141–142

[8] Conklin J (2003) Dialog mapping: Reflections on an industrial strength case study. In: Kirschner P, Buckingham Shum S, Carr C (eds.) Visualizing Argumentation. Springer, London Berlin Heidelberg New York

[9] Conklin J (2005) Dialogue Mapping: Building Shared Understanding of Wicked Problems. Wiley, Chichester

[10] Conklin J, Begeman ML (1988) gIBIS: A hypertext tool for exploratory policy discussion. ACM Trans. Off. Inform. Syst., 4(6), 303–331

[11] Conklin J, Burgess Yakemovic KC (1991) A process-oriented approach to design rationale. Hum.–Comput. Interact., 6(3&4), 357–391

[12] Fischer G, Lemke AC, McCall R, Morch AI (1991) Making argumentation serve design. Hum.–Comput. Interact., 6(3&4), 393–419

[13] Grudin J (1996) Evaluating opportunities for design rationale capture. In: Moran TP, Carroll JM (eds.) Design Rationale: Concepts, Techniques, and use. Lawrence Erlbaum Associates, Hillsdale

[14] Halasz FG (1988) Reflections on Notecards: Seven issues for the next generation of hypermedia systems. Commun. ACM, 31, 836–852

[15] Haley CB, Laney RC, Moffett JD, Nuseibeh B (2004) The effect of trust assumptions on the elaboration of security requirements. In: Proceedings of the 12th International Requirements Engineering Conference, Kyoto, 6–10 September. IEEE Computer Society Press, pp. 102–111

[16] Haley CB, Laney RC, Nuseibeh B. (2005). Arguing security: Validating security requirements using structured argumentation, Department of Computing, The Open University, Milton Keynes, UK, Technical Report 2005/04
[17] Isenmann S, Reuter W (1997) IBIS – a convincing concept…but a lousy instrument? In: Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques. Amsterdam, pp. 163–172
[18] Jackson M (2001). Problem Frames. Addison Wesley, London
[19] van Lamsweerde A (2004) Elaborating security requirements by construction of intentional anti-models. In: Proceedings of the 26th International Conference on Software Engineering. Edinburgh, 26–28 May, pp. 148–157
[20] MacLean A, Young R, Bellotti V, Moran T (1991) Questions, options and criteria: Elements of design space analysis. Hum.–Comput. Interact., 6, 201–250
[21] MacLean A, Bellotti V, Shum S (1993), Developing the design space with design space analysis. In: Byerley PF, Barnard PJ, May J (eds.) Computers, communication and usability: design issues, research and methods for integrated services. Elsevier, Amsterdam, pp. 197–219
[22] Moffett JD, Haley CB, Nuseibeh B (2004) Core security requirements artifacts, Department of Computing, The Open University, Milton Keynes, UK, Technical Report 2004/23.
[23] Palus CJ, Horth DM, Pully ML, Selvin A (2003) Exploration for development: developing leadership by making shared sense of complex challenges. Consult. Psychol. J., 55 (1), 26–40
[24] Papadopolous N (2004) Conflict cartography: a methodology designed to support the efficient and effective resolution of complex, multi-stakeholder conflicts. ViewCraft LLC. Available from http://www.viewcraft.com/pdfs/ViewCraft_ConflictCartographyMarch04.pdf
[25] Selvin A (1999) Supporting collaborative analysis and design with hypertext functionality. J. Digit. Inform., 1(4). Available from http://jodi.ecs.soton.ac.uk/Articles/v01/i04/Selvin/
[26] Selvin A (2003) Fostering collective intelligence: Helping groups use visualized argumentation. In Kirschner P, Buckingham Shum S, Carr C (eds.) Visualizing argumentation: Software tools for collaborative and educational sense-making. Springer, London Berlin Heidelberg New York
[27] Selvin A (2004) Building collaborative knowledge representations in real time: An analysis of facilitative micro-actions. Departmental colloquium, Knowledge Media Institute, The Open University, UK. Available from http://stadium.open.ac.uk/stadia/preview.php?s=29&whichevent=494

[28] Selvin A, Sierhuis M (1999) Case studies of Project Compendium in different organizations. In: Workshop on Computer-Supported Collaborative Argumentation, Conference on Computer-Supported Collaborative Learning, Stanford, CA (12–15 Decemeber). Available from http://kmi.open.ac.uk/sbs/csca/cscl99
[29] Selvin A, Buckingham Shum S (2002) Rapid knowledge construction: a case study in corporate contingency planning using collaborative hypermedia. Knowl. Process Manage. 9(2), 119–128
[30] Shipman FM, Marshall CC (1999) Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. Comput. Support. Cooperat. Work, 8(4), 333–352
[31] Tate A, Dalton J, Buckingham Shum S, Mancini C, Selvin A (2004) Co-OPR project experiment B report, Artificial Intelligence Applications Institute, Edinburgh University, UK. Available from: www.aiai.ed.ac.uk/project/coopr