TOWARDS A SYSTEMATIC REPOSITORY OF KNOWLEDGE ABOUT MANAGING COLLABORATIVE DESIGN CONFLICTS[1]

MARK KLEIN, PHD
Center for Coordination Science
Massachusetts Institute of Technology
m_*klein@mit.edu*
http://ccs.mit.edu/klein/

Abstract. Increasingly, complex artifacts such as cars, planes and even software are designed using large-scale and often highly distributed collaborative processes. A key factor in the effectiveness of these processes concerns how well conflicts are managed. Better approaches need to be developed and adopted, but the lack of systematization and dissemination of the knowledge in this field has been a big barrier to the cumulativeness of research in this area as well as to incorporating these ideas into design practice. This paper describes a growing repository of conflict management expertise, built as an augmentation of the MIT Process Handbook, that is designed to address these challenges.

1. The Challenge

Increasingly, complex artifacts such as cars, planes and even software are designed using large-scale and often highly distributed collaborative processes. Conflict (i.e. incompatibilities between design decisions and/or goals) is common in such highly interdependent activities. In one study, for example, half of all interactions between collaborating architectural designers were found to involve detecting and resolving conflicts (Klein and Lu 1991).

Better conflict management practices are needed. Current, mainly manual practices are being overwhelmed by the sheer scale and complexity of modern design artifacts. Consider the Boeing 767-F design project. This project involved the integrated contributions of hundreds of individuals in tens of disciplines and hundreds of teams spread over several continents and a span of years. The design includes millions of components and underwent thousands of changes. Design conflicts were often not detected until long (days to months) after they had occurred, resulting in wasted design time, design rework, and even scrapped tools and parts. Design rework rates of 25-30% were typical. Since maintaining scheduled commitments was a priority, design rework often had to be done on a short flow-time basis that typically cost much more (estimates ranged as high as 50 times more) and could reduce product quality. Conflict cascades that

[1] Submitted to the 2000 International Conference on Artificial Intelligence in Design.

required as many as 15 iterations to finally produce a consistent design were not uncommon. To give another example, roughly half of the labor budget for the Boeing 777 program (which is measured in the hundreds of millions of dollars) was estimated to be due to changes, errors and rework, often due to design conflicts. All of this occurred in the context of Boeing's industry-leading adoption of concurrent engineering practices such as multi-disciplinary design teams (Klein 1994).

A key barrier to the development and utilization of improved design conflict management practices has been the lack of dissemination of this knowledge in a systematized form. Conflict management is fundamentally a multi-disciplinary topic, and information in this area is scattered as a result across multiple disparate communities including computer science, industrial engineering, and management science, to mention just a few. Previous efforts to develop taxonomies of conflict knowledge (Matta 1996) (Castelfranchi 1996) (Ramesh and Sengupta 1994) (Feldman 1985) have been small in scope and have left out important classes of information, particularly *meta*-process information, which will be described below. The result is that good ideas developed within one discipline, or even within one industry, do not readily propagate to researchers and practitioners in other settings, and opportunities are lost to carry on a more systematic and cumulative exploration of the range of potentially useful conflict management techniques.

The work described in this paper addresses these challenges directly by developing a semi-formal Web-accessible repository of multi-disciplinary collaborative design conflict management expertise organized so as to facilitate key uses including:

- Pedagogy: helping students, researchers and practitioners learn about the state of the art in design conflict management
- Business process re-design: helping practitioners finding alternative ways of designing their collaborative design processes
- Research: helping researchers identify gaps in conflict management technology, identify common abstractions, facilitate discussion, and foster development of new ideas

The remainder of this paper will describe the key ideas and tools making up the conflict repository, evaluate its efficacy with respect to the goals listed above, and describe potential directions for future work.

2. Our Approach

Our approach is to capture design conflict management knowledge using a substantively extended version of the tools and techniques developed as part of the MIT Process Handbook project. The Handbook is a process knowledge repository which has been under development at the Center for Coordination Science (CCS) for the past six years (Malone and Crowston 1994) (Malone, Crowston et al. 1998). The growing Handbook database currently includes over 5000 process descriptions ranging from specific (e.g. for a university purchasing department) to generic (e.g. for resource allocation and multi-criteria decision making). The CCS has developed a Windows-based tool for editing the Handbook repository contents, as well as a Web-based tool for read-only access. The Handbook is under active use and development by a highly distributed group of more than 40 scientists, teachers, students and sponsors for such diverse purposes as adding new process descriptions, teaching classes, and business process re-design.

In the following sections we will present the core concepts underlying the Handbook, describe how these concepts and associated tools were extended to capture conflict management expertise, and give examples of how this can be used to support a range of useful capabilities.

2.1. UNDERLYING PROCESS HANDBOOK CONCEPTS

The Handbook takes advantage of four simple but powerful concepts to capture and organize process knowledge: *attributes, decomposition, dependencies,* and *specialization.*

Process Attributes: Like most process modeling techniques, the Handbook allows processes to be annotated with attributes that capture such information as a textual description, typical performance values (e.g. how long a process takes to execute), as well as applicability conditions (i.e. constraints on the contexts where the process can be used).

Decomposition*:* Also like most process modeling techniques, the Handbook uses the notion of *decomposition*: a process is modeled as a collection of activities that can in turn be broken down ("decomposed") into subactivities. A common conflict detection process in industry, for example, is the change memo, wherein a designer that makes a design change describes it in a memo and distributes it to potentially affected designers for their review and comment. The decomposition for this process is thus the following (Figure 1):

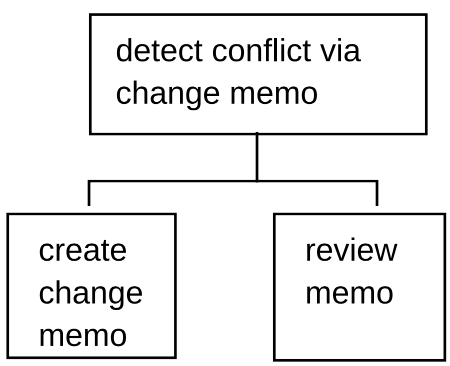

Figure 1: Decomposition for the change memo process.

Dependencies*:* Another key concept we use is that coordination can be viewed as the management of *dependencies* between activities (Malone and Crowston 1994). Every dependency can include an associated *coordination mechanism*, which is simply the process that manages the resource flow and thereby coordinates the activities connected by the dependency. The dependency graph for the change memo process, for example, is the following (Figure 2):

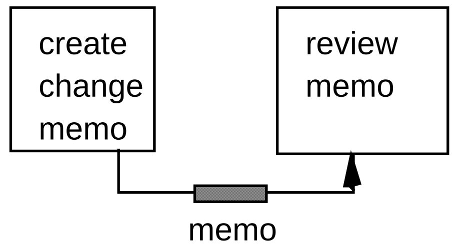

Figure 2: Dependencies for the change memo process.

Here the key dependency involves getting the change memo (i.e. the resource created by the originating designer) to the interested parties. In typical industry practice, the memos are hand-written and the coordination mechanism consists of distributing the memos via office mail to all the engineers the originating engineer thought were relevant, as the originating engineer generates them.

The key advantage of representing processes using dependencies and coordination mechanisms is that they allow us to abstract away details about how ‘core’ activities coordinate with each other, and thereby making it easier to explore different ways of doing so. We will see examples of this below.

<u>Specialization</u>*:* The final key concept is that processes can be arranged into a *taxonomy*, with very generic processes at one extreme and increasingly *specialized* processes at the other. Processes are organized based on their function, so that processes with similar purposes appear close to each other. This facilitates finding and comparing alternative ways for performing functions of interest, thereby fostering easy transfer of ideas. Sibling processes that vary along some interesting design dimension can be grouped into “bundles” with tradeoff tables that capture the relative pros and cons of these alternatives. Consider, for example, the following taxonomy fragment for conflict detection processes (Figure 3):

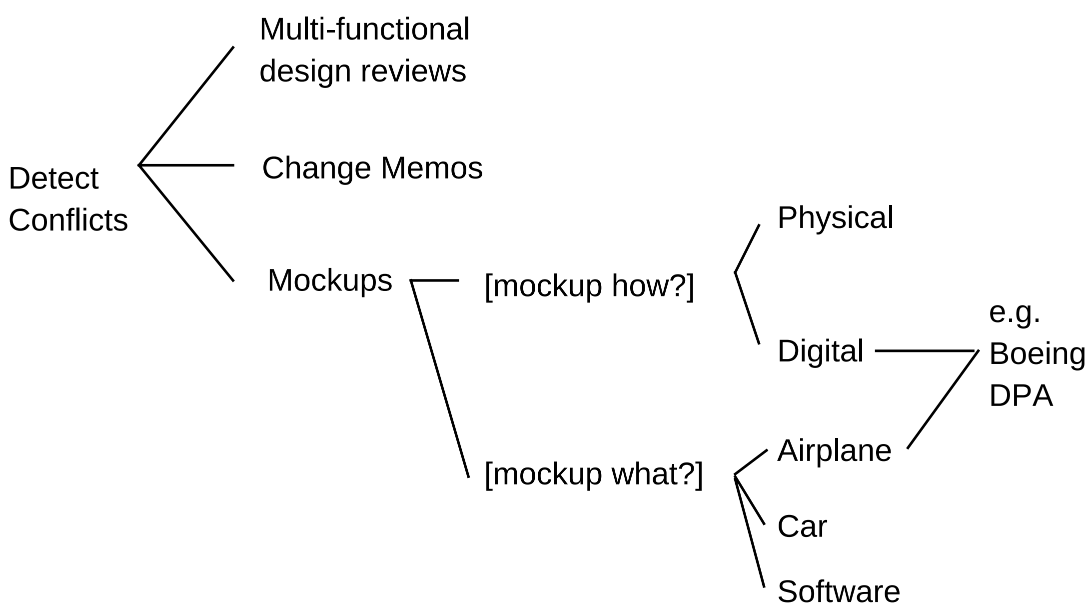

Figure 3. A fragment of the process taxonomy for conflict detection.

The taxonomy shows that there are at least three generic techniques for detecting conflicts (design reviews, change memos and mockups) and also that mockups can in turn be distinguished into physical and digital versions thereof (a physical mockup involves building a physical scale model of the artifact; a digital mockup utilizes a digital model of the artifact instead). Two bundles distinguish between different kinds of mockup-based conflict detection processes. The [mockup how?] mockup collects the different ways of doing mockups, and includes a tradeoff table capturing their relative pros and cons (Table 1):

Alternative	**Detection Speed**	**Up-front cost**	**Cost of changes**
physical	slow	medium	high
digital	fast	high	low

TABLE 1. A tradeoff table for the [mockup how?] bundle.

The table shows that physical mockups have lower up-front cost but detect conflicts relatively slowly, and are expensive to modify as the design changes. Digital mockups have greater up-front costs but are superior on the other counts.

2.2. EXTENDING THE HANDBOOK TO CAPTURE CONFLICT KNOWLEDGE

While the Handbook as described above is well-suited for describing conflict management processes by themselves, it does not capture crucial information concerning what *types* of conflicts exist, in what *contexts* (i.e. design processes) they can appear, what *impact* they have, or what conflict management processes are suitable for *handling* them. *The novel contribution of the work described herein involved extending the Handbook so it can capture this information*. This required two additional elements: the *conflict taxonomy*, and the *conflict management meta-process*. These are described below.

Conflict Taxonomy: The conflict taxonomy is a hierarchy of conflict types, ranging from general conflict types like ‘belief conflict’ to more specific ones like ‘resource budget exceeded’ (Figure 5):

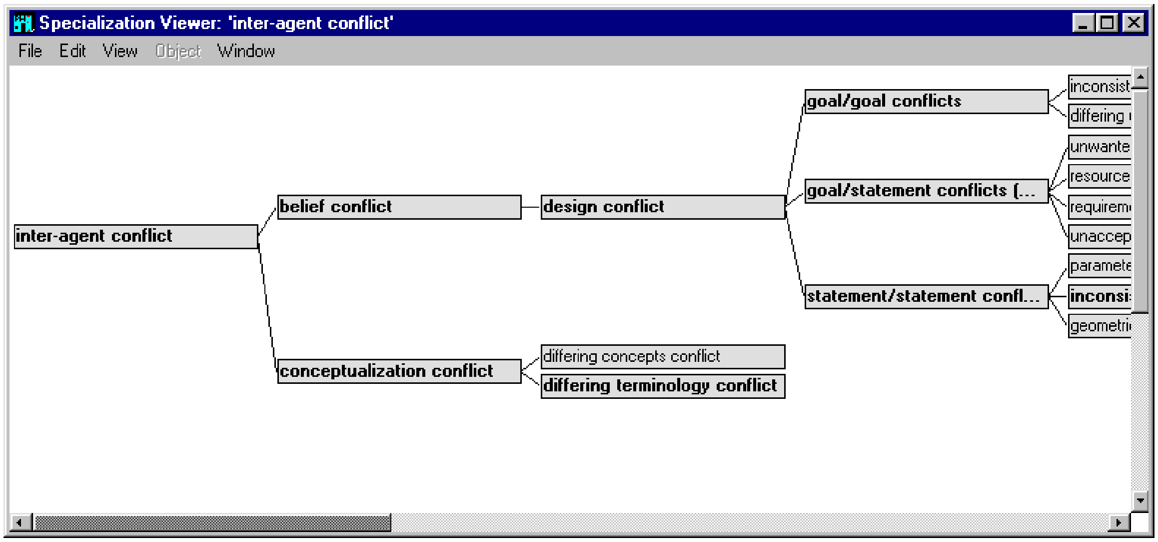

Figure 5. A fragment of the conflicts type taxonomy.

There are many types of conflict. A major dividing point in the taxonomy, for example, concerns whether the conflict involves the way the designers represent the design (conceptualization conflict) or the content of the design itself (belief conflict).

Different kinds of collaborative design processes have different characteristic conflict types. This is captured by building on a taxonomy of collaborative design processes (Figure 6).

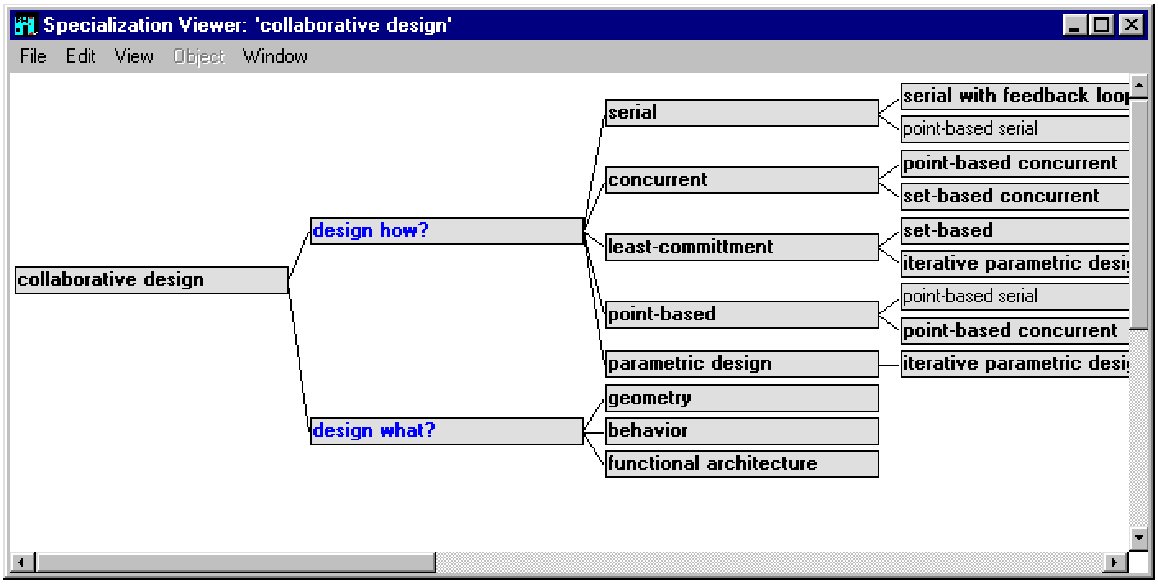

Figure 6. A fragment of the collaborative design process hierarchy.

Every collaborative design process is linked to the conflict types that characterize it. A processes' characteristic conflicts are inherited by its specializations unless explicitly over-ridden. Every conflict is annotated with its typical impact on the associated design process. All collaborative design processes, for example, are subject to the generic 'design conflict', but the severity varies. Concurrent design, for example, generally experiences fewer delays and other costs from design conflicts than does serial design.

Conflict types are linked, in turn, to the one or more processes suitable for handling them; these processes are themselves arranged into a taxonomy, producing the following overall structure (Figure 7):

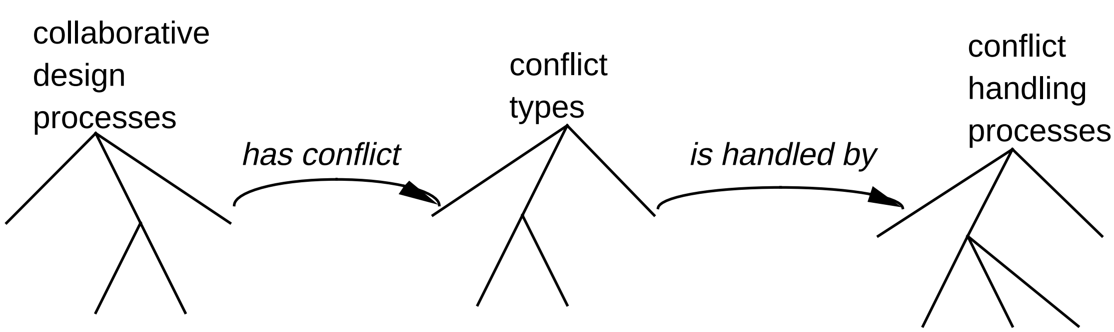

Figure 7. Linkages to/from the conflict taxonomy

The conflict handing process taxonomy (see Figure 8) is where the bulk of the repository content resides[2]:

[2] The repository uses the term 'exception' because the Process Handbook is currently being applied to capturing knowledge about coordination failures ('exceptions') in general, of which conflict is a sybtype. See (Klein and Dellarocas 2000) for more detail on this aspect of our work.

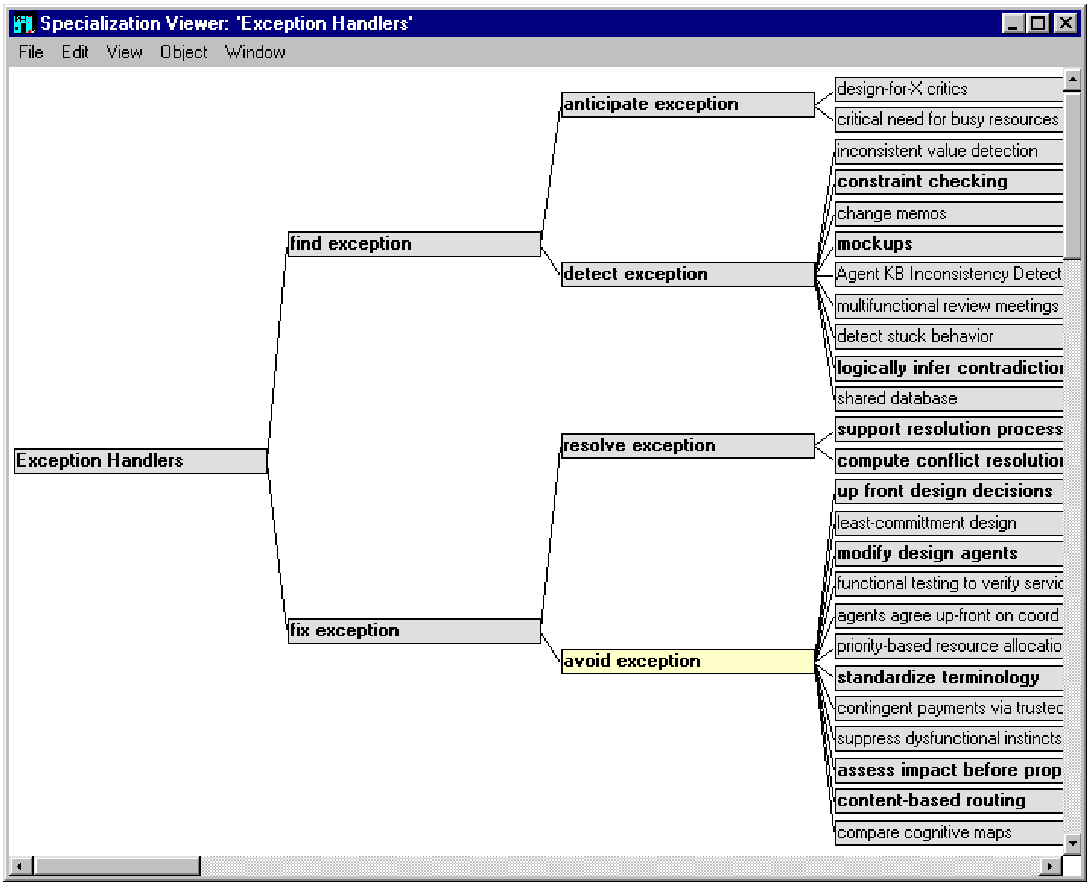

Figure 8. A subset of the conflict handling process taxonomy.

There are four main classes of conflict handling processes, divided into two pairs. If a conflict has not yet occurred, we can use:

- Conflict *anticipation* processes, which uncover situations where a given class of conflict is likely to occur. An example of such a process is one which looks for design changes that increase the use of a highly limited resource – one can anticipate that the design change may cause a conflict even without calculating the actual resource usage impact.
- Conflict *avoidance* processes, which reduce or eliminate the likelihood of a given class of conflict. Terminological conflicts, for example, can be avoided by leading the designers to standardize their terminology before starting the design.

If the conflict has already occurred, we instead can use:

- Conflict *detection* processes, which detect when a conflict has actually occurred. Change memos, design mockups, and multifunctional meetings are all, as we have seen, examples of processes used to detect conflict.

- Conflict *resolution* processes, which resolve a conflict once it has happened. Such processes can include those that structure the conflict resolution interaction between designers (e.g. facilitated negotiation) as well as those that compute a resolution to the conflict outright (e.g. multi-criteria optimization)

We have found that the applicability conditions for conflict handler processes fall into three main categories:

- Constraints on the *design process:* These describe which class of collaborative design process the conflict handler is suited for.
- Constraints on the *design agent:* These describe capabilities design agents must have in order for the conflict handler to be applicable.

Imagine a conflict resolution process like multi-criteria optimization, for example, that involves optimizing a single utility function formed by aggregating the functions of the contending design agents. The applicability conditions for such a procedure would be something like the following (Table 2):

Process	• Design proceeds by creating new entities and manipulating the parameters associated with these entities. There is a finite known set of entities and parameters.
Agent	• Agents can describe their utilities as functions that take the design parameter values as input and produce values expressed in terms of a single mutually understood goodness metric

TABLE 2. Example of conflict handler applicability conditions.

This information is useful when trying to determine if a given conflict handler is appropriate for the design context one is currently concerned with.

<u>The Conflict Management Meta-Process:</u> The conflict taxonomy and associated links described above capture the range of *possible* conflicts and associated conflict handling processes, but do not specify *which* handlers should be used *when* for *what* exceptions. This latter information is captured in the augmented Handbook as specializations of the generic *conflict management meta-process* (Figure 9):

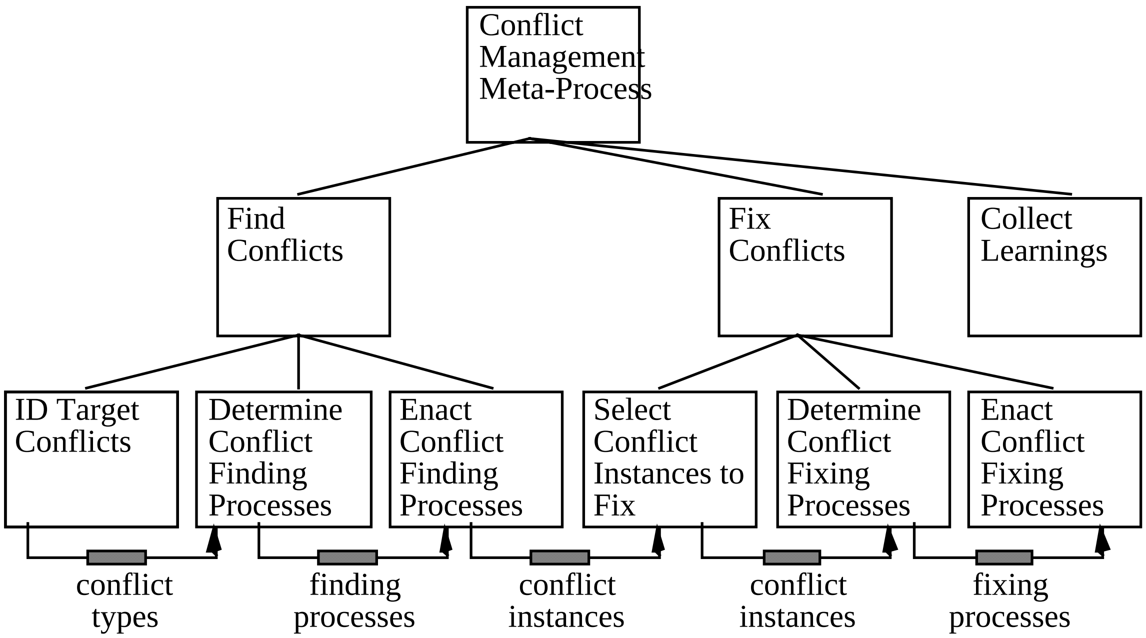

Figure 9. The decomposition of the generic conflict management meta-process.

The conflict management meta-process consists of the following subtasks:

- *Identify target conflicts,* which decides which classes of conflicts the process is going to handle, potentially in a time-varying context-sensitive way.
- *Determine conflict finding processes,* which determines which conflict finding (i.e. anticipation or detection) handlers will be used to find the conflicts of these types
- *Enact conflict finding processes,* which enacts the conflict finding processes identified in the previous step, producing one or more conflict instances
- *Select conflict instances to fix,* which sorts and prunes the list of conflict instances so uncovered
- *Determine conflict fixing processes,* which determines which conflict fixing (avoidance or resolution) processes will be used to handle these conflict instances
- *Enact conflict fixing processes,* which enacts the conflict fixing processes to actually (hopefully) complete the handling of the conflict(s) detected by the system
- *Collect learnings,* which collects information produced by any of the other steps as input to any learning capability that the conflict management system may have, presumably changing the operation of the other meta-process steps in the future.

This is a *meta*-process because the inputs and outputs of some of the steps are other (conflict handler) processes. This decomposition, patterned originally on that used in diagnostic expert systems (Clancey 1984), has been found adequate to capture all the important classes of meta-process information encountered in the conflict management literature our team has reviewed so far.

In order to make this more concrete, let us consider two specializations from the conflict management meta-process taxonomy (Figure 10):

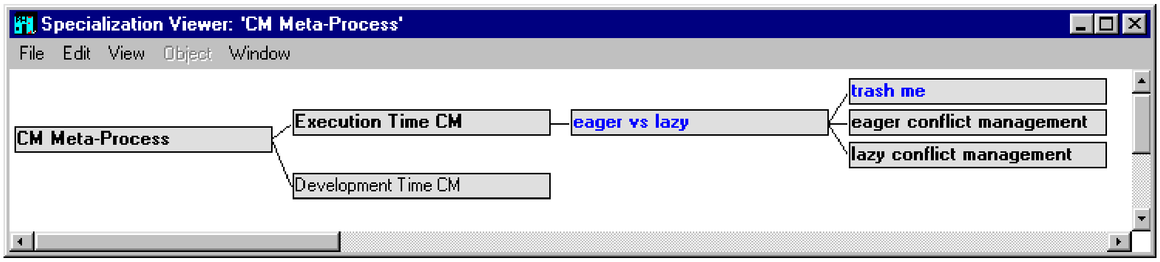

Figure 10. A subset of the conflict management meta-process taxonomy.

One major distinction in this taxonomy is whether conflict management is done at system development time, or at system execution time. Development-time conflict management has been applied extensively in the creation of expert systems whose rules are derived from human experts representing different, often conflicting, areas of expertise. This approach involves finding and resolving all possible conflicts among the knowledge base entries *before* the system is used, typically using some kind of semantic analysis of the knowledge base contents (Bezem 1987) (Trice and Davis 1989). Such a conflict management process would have the following subtasks when modeled as a specialization of the generic conflict management meta-process (Table 3):

Subtask	**How Implemented**
Identify target conflicts	The target conflicts are inconsistencies among the potential conclusions of any of the rules in the knowledge base.
Determine conflict finding processes	Use hardwired rule consistency checking code
Enact conflict finding processes	The consistency checking code is enacted by the knowledge base developers as desired when the knowledge base is being developed.
Select conflict instances to fix	All conflicts are fixed, typically in the order in which they are found.
Determine conflict fixing processes	All conflict instances are fixed by the process 'Consult human knowledge base developers'
Enact conflict fixing processes	The process 'Consult human knowledge base developers' is enacted at development time as desired.
Collect learnings	N/A

TABLE 3. Conflict management meta-process for development-time conflict management.

Execution-time conflict management, by contrast, involves detecting and resolving conflicts during the actual design process. The conflict management meta-process for one example of this approach (Klein 1997) is given below (Table 4):

Subtask	How Implemented
Identify target conflicts	A human designer selects, at any point during the design process, the conflicts he/she is interested in by selecting from a predefined conflict taxonomy.
Determine conflict finding processes	Every conflict type has a single predefined (hardwired) conflict detection process.
Enact conflict finding processes	The detection processes for the selected conflicts are enacted on-demand - when the human designer requests it.
Select conflict instances to fix	The human designer selects which conflicts to fix from the list presented by the system.
Determine conflict fixing processes	The system uses a diagnostic procedure and a knowledge base of generic conflict handling strategies to generate a sorted list of proposed specific conflict resolutions. The human designer then selects which resolution to use, or may choose to define his/her own resolution.
Enact conflict fixing processes	The system enacts the selected resolution, if any, on demand.
Collect learnings	Completed conflict resolution instances are stored as cases in a database for later use as data to help add to and refine the conflict knowledge base contents.

TABLE 4. Conflict management meta-process for execution-time conflict management.

2.3. USING THE CONFLICT REPOSITORY

As noted above, we have identified three key uses for process repositories:

- Pedagogy: helping students, researchers and practitioners learn about the state of the art in design conflict management
- Business process re-design: helping practitioners [re-] design the conflict management aspects of their collaborative design processes
- Research: helping researchers identify gaps in conflict management technology, identify common abstractions, facilitate discussion, and develop new ideas

We will now consider how the conflict repository can be used for these purposes.

Pedagogy: The original Process Handbook allows users to browse through the specialization taxonomy for processes in the domain of interest, inspecting their attributes, decompositions and dependencies, and comparing their relative merits using the tradeoff tables in bundles. The conflict repository built on the Handbook augments this by providing a richer set of links, as described above. The Web version of the Handbook, designed for pedagogical use, is shown below (Figure 11):

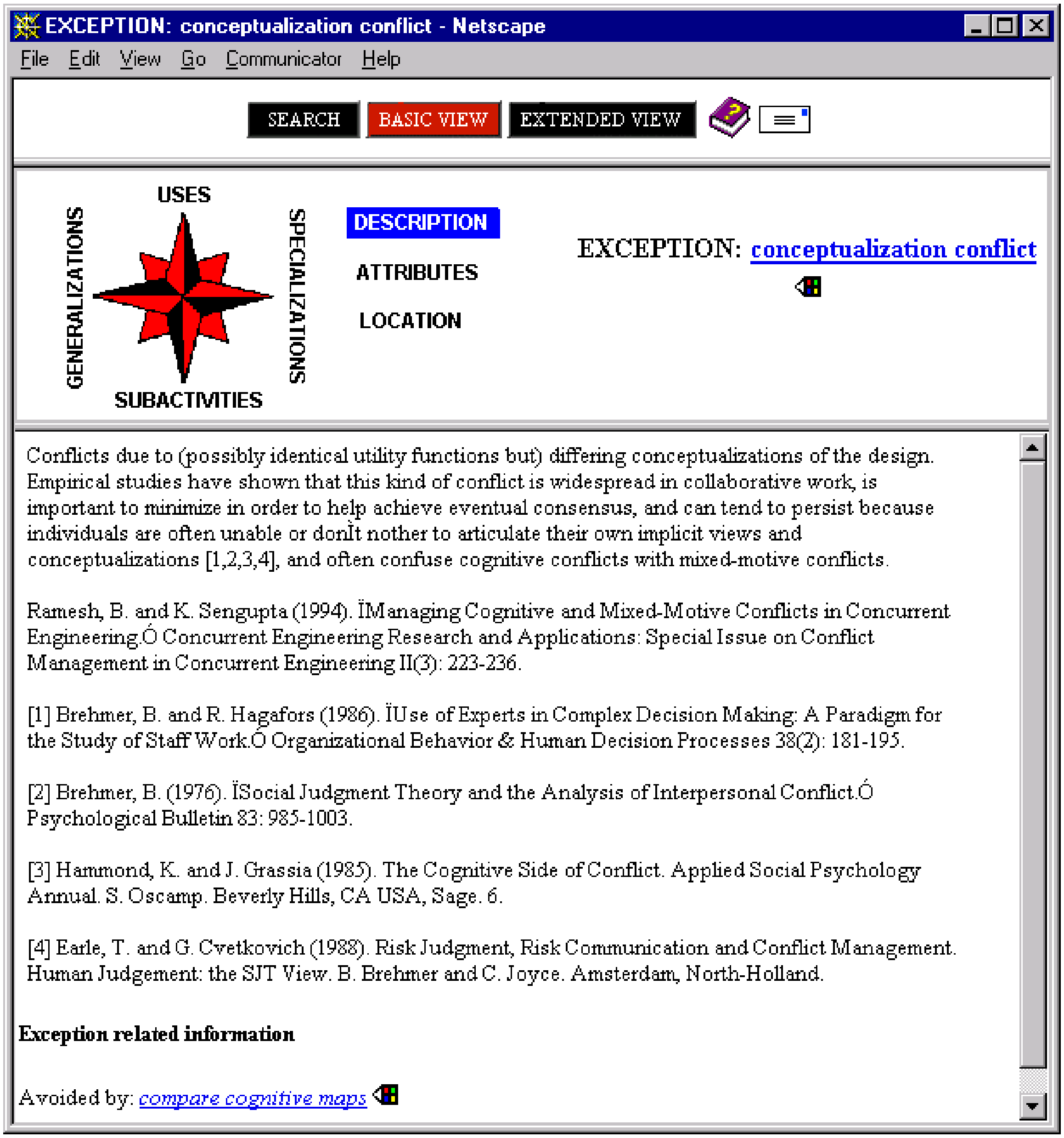

Figure 11: Screen snapshot of the Web-accessible version of the conflict repository.

One can traverse the current taxonomy 'up' or 'down' by clicking on the 'generalization' or 'specialization' buttons, or follow cross-links (in this example, links from the conflict to a conflict handler) by clicking on hotlinked item.

The specialization taxonomies underlying the conflict repository facilitate cross-disciplinary knowledge transfer by revealing commonalities in the goals and approaches of techniques from different domains. They do so by (1) highlighting the extensive overlap in conflict types across different domains, and (2) colocating conflict handling processes with similar purposes,, regardless of their origin. Should an automobile designer follow the 'is detected by' links from the 'geometric overlap' conflict, for example, he/she will immediately encounter such ideas as 'digital preassembly' (used in the airplane industry) and 'daily mockups' (used in the software industry). Similarly, an airplane designer looking at the conflict avoidance processes branch will find such ideas as 'set-based design' (used in the automobile industy).

Business Process Redesign: The conflict repository supports a simple but powerful methodology for [re-] designing the conflict management procedures used in one's design processes. It involves applying the Handbook's process re-design methodology (Herman, Klein et al. 1998) to the conflict management meta-process one is using/starting from. All of the subtasks in this process, as we have seen, have multiple alternative specializations (i.e. ways of realizing that subtask). We can therefore explore many different variations of the process by systematically varying the alternatives we select for each subtask. We can vary, for example, whether 'enact conflict detection processes' is done immediately after every design change ('eager' conflict detection), on a scheduled basis (as in the 'daily build' process used by Microsoft) or as desired by the designers or design managers ('lazy' conflict detection). We can decide whether 'determine conflict fixing processes' is done using computer tools to suggest resolutions, by providing designers access to the conflict repository, by leaving them on their own, and so on. A tool known as the 'Process Recombinator' (Bernstein, Klein et al. 1999), available under the Windows version of the Process Handbook, has been developed to support this systematic exploration of different subtask combinations (Figure 12).

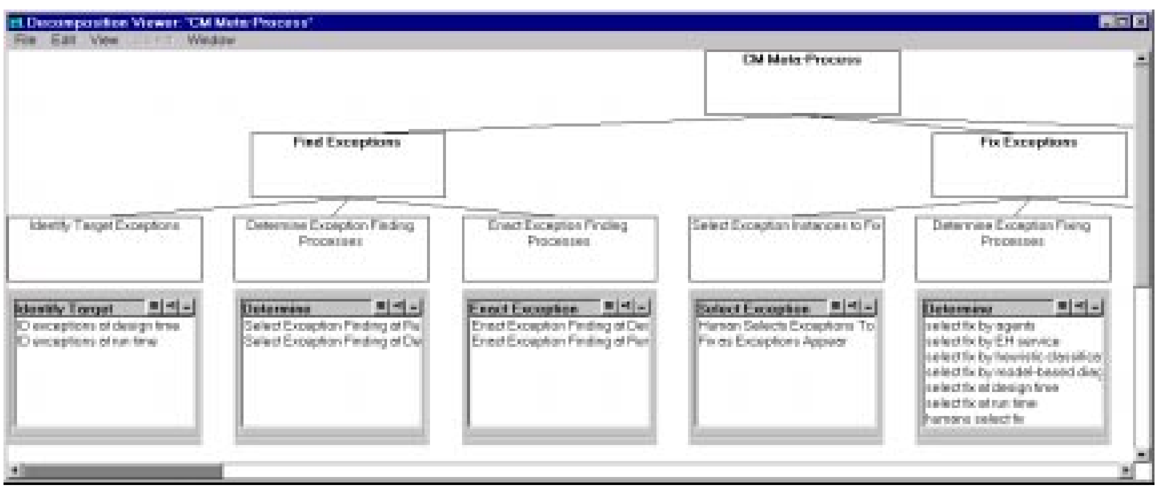

Figure 12. Snapshot of the process recombinator.

Facilitating Research: A conflict repository can serve as a valuable resource for fostering more effective, accumulative and cross-disciplinary research on conflict management, in several important ways. The taxonomic structure of the repository facilitates *finding gaps* in the conflict management knowledge. One can, for example, look for conflict types with no associated resolution strategies, or for sparsely populated regions of the conflict resolution strategy space (e.g. where a tradeoff table has no alternatives identified for common values of a key design characteristic). The conflict repository structure can enable *structured discussions* by organizing them around focus topics such as filling in a particular branch of a taxonomy, adding to a tradeoff table, or detailing a particular process description. It is our experience that such foci can be more effective than unstructured discussions for capturing process knowledge. The process re-design methodology described above can, finally, be used to help *invent new conflict management techniques*. Imagine, for example, that one wishes to explore variations to the "change memo" conflict detection process described above. One possibility is to consider different processes for managing the dependency between the "create change memo" and "review memo" steps. This quickly reveals such interesting alternatives as "making" change memos to order (i.e. when the receiving engineers are ready for them), collocating engineers to minimize change memo distribution time, and using content-based routing or filter agents to ensure that engineers get only relevant memos. This can be taken one step further by looking at 'distant analogies' (processes that address different but functionally similar challenges) as a way of suggesting creative alternatives (Herman, Klein et al. 1998). Consider, for example, the development time conflict management technique mentioned above, wherein rule bases are modified before being merged, based on the results of automated semantic analysis, to prevent them from asserting conflicting conclusions. Pursuing this distant analogy suggests the idea of using semantic conflict analysis to design specialized training curricula for designers involved in large projects, helping them avoid needless conflicts. Not all distant analogies will lead, of course, to useful ideas.

3. Evaluation of the Contributions of This Work

This conflict repository described in this paper makes substantive contributions to previous work in this area. These include greater expressiveness and content coverage, which in turn help make the repository potentially more effective in supporting prototypical uses.

Expressiveness: Previous efforts to create conflict knowledge repositories (Matta 1996) (Castelfranchi 1996) (Ramesh and Sengupta 1994) (Feldman 1985) all include either a conflict type taxonomy, a conflict handler taxonomy, or both, with links between conflict types and the potentially applicable conflict handlers. None of these efforts, however, capture the linkage between collaborative design processes and their characteristic conflict types, nor do they capture the important information encoded by the conflict management meta-process described in this paper. Finally, they don't take advantage of process abstraction and bundle/tradeoff concepts to enable quick discovery and comparison of alternative processes for similar needs. It is our preliminary judgement that the schema presented above captures all the significant aspects of the conflict management information we have encountered in the literature we have reviewed to date.

Coverage: Previous efforts in this area have produced repositories that are quite small in scale. The taxonomy described in (Matta, Ros et al. 1998) (Feldman 1985) (Ramesh and Sengupta 1994) (Castelfranchi 1996) each include no more than about 30 conflict types and handler processes. These efforts, in addition, focus on individual disciplines. Matta's work for example focuses on the concurrent engineering literature, Feldman on the sociological literature, and Castelfranchi on multi-agent systems. While one can argue that they provide complete coverage at an abstract level, they necessarily leave out descriptions of a large number of specific, potentially useful conflict management techniques.

The repository described in this paper is significantly larger in scope. It includes roughly the same number of conflict types as those described above but a significantly larger number of conflict management processes (about 200 at the time of writing). The contents of the MIT repository have been drawn from several disciplines including distributed artificial intelligence, sociology and industrial engineering, as represented by roughly 50 publications from such venues as the *Journal of Concurrent Engineering Research and Applications*, the *Journal of Artificial Intelligence in Engineering Design Analysis and Manufacturing*, the *Sloan Management Review*, the *International Conference on Artificial Intelligence in Design*, the *National Conference on Artificial Intelligence*, and so on. Our repository continues to grow, with the support of a continuing 3 year grant from the National Science Foundation (grant IIS-9803251).

Better Support for Prototypical Uses: The MIT conflict repository has been evaluated only on a limited internal basis to date, so it is premature to draw definitive conclusions about its utility for students, researchers and practitioners. It is clear, however, that the Process Handbook provides a level of enabling technology that has not been exploited in previous conflict repository efforts. Previous work has resulted mainly in textual documents (with the notable conflict of Matta et al. who made the repository available over the Web), and does not include the kind of search, navigation, business process re-design and structured discussion tools available as part of the Handbook. Previous experience with these tools suggests that they can be powerful enablers. The Handbook has been successfully used, for example, to teach classes at the Sloan School of Management as well as Babson College. The Handbook process redesign methodology has been applied in several domains, most recently (in cooperation with the consulting firm AT Kearney) to re-design the hiring processes in a major financial services firm. The participants in this study felt that the approach was effective in generating a much wider range of novel and promising process alternatives than would have been uncovered by traditional methods (Herman, Klein et al. 1998).

4. Future Work

The MIT conflict repository is a work in progress. We plan to continue to add to and better structure the repository content, drawing from multiple disciplines. We will explore the use of additional repository structuring schemes and tools, such as the notion of a “guided tour’ that provides a suggested sequence for traversing the specialization taxonomies for specific pedagogical purposes. The repository will be submitted to a series of evaluations by different classes of users in order to assess and help improve its utility. The biggest challenge, however, will be evolving the conflict repository into a living self-sustaining community resource. This will require addressing technological issues (for example, developing a Web-based authoring tool) as well as sociological issues concerning incentives for adding content.

For additional information about this and related work, including access to the MIT conflict repository itself, please see http://ccs.mit.edu/klein/.

Acknowledgements

This work was supported by NSF grant IIS-9803251 (Computation and Social Systems Program) and by DARPA grant F30602-98-2-0099 (Control of Agent Based Systems Program). I am grateful for many helpful comments from Prof. Chris Dellarocas (MIT Sloan School of Management), Prof. Feniosky Pena-Mora (MIT Department of Civil Engineering), and the members of the MIT Center for Coordination Science: Leon Grekin, John Quimby, George Herman, George Wyner, Abraham Bernstein, and Prof. Thomas Malone.

References

Bernstein, A., M. Klein, et al. (1999). The Process Recombinator: A Tool for Generating New Business Process Ideas. Proceedings of the International Conference on Information Systems, Charlotte, North Carolina USA.

Bezem, M. (1987). Consistency Of Rule-Based Expert Systems, Centre for Mathematics and Computer Science.

Castelfranchi, C. (1996). Conflict Ontology. Proceedings of the Workshop on Conflict Management, European Conference on Artificial Intelligence (ECAI).

Feldman, D. C. (1985). "A Taxonomy Of Intergroup Conflict Resolution Strategies." The 1985 Annual Conference on Developing Human Resources.

Herman, G., M. Klein, et al. (1998). A Template-Based Process Redesign Methodology Based on the Process Handbook. Unpublished discussion paper. Cambridge MA, Center for Coordination Science, Sloan School of Management, Massachusetts Institute of Technology.

Klein, M. (1994). "Computer-Supported Conflict Management in Concurrent Engineering: Introduction to Special Issue." Concurrent Engineering Research and Applications **2**(3).

Klein, M. (1997). "An Conflict Handling Approach to Enhancing Consistency, Completeness and Correctness in Collaborative Requirements Capture." *Concurrent Engineering Research and Applications*(March).

Klein, M. and C. Dellarocas (2000). "A Knowledge-Based Approach to Handling Conflicts in Workflow Systems." Journal of Computer-Supported Collaborative Work. Special Issue on Adaptive Workflow Systems.(January).

Klein, M. and S. C.-Y. Lu (1991). "Detecting and Resolving Conflicts Among Cooperating Human and Machine-Based Design Agents." The International Journal For Artificial Intelligence in Engineering.

Malone, T. W., K. Crowston, et al. (1998). "Tools for inventing organizations: Toward a handbook of organizational processes." Management Science **45**(3): 425-443.

Malone, T. W. and K. G. Crowston (1994). "The interdisciplinary study of Coordination." *ACM Computing Surveys* **26**(1): 87-119.

Matta, N. (1996). Conflict Management in Concurrent Engineering: Modeling Guides. ECAI Workshop on Conflict Management.

Matta, N., C. Ros, et al. (1998). A Generic Library to Guide Decision Making in Concurrent Engineering. Proceedings of TMCE-98, Manchester UK.

Ramesh, B. and K. Sengupta (1994). "Managing Cognitive and Mixed-Motive Conflicts in Concurrent Engineering." Concurrent Engineering Research and Applications: Special Issue on Conflict Management in Concurrent Engineering **II**(3): 223-236.

Trice, A. and R. Davis (1989). Consensus Knowledge Acquisition, Information Technologies Group, MIT School of Management.