A Knowledge Based Approach to Assisting Collaborative Relationships

by

Winston Dali Chang

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology
May 23, 2003

Copyright 2003 Winston Dali Chang. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.

Author__________________________Winston Dali Chang
Department of Electrical Engineering and Computer Science
May 23, 2003

Certified by______________________Dr. Amar Gupta
Co-Director, Productivity From Information Technology Initiative
Sloan School of Management
Thesis Supervisor

Accepted by_______________________Arthur C. Smith
Chairman, Department Committee on Graduate Theses

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
LIBRARIES
A Knowledge Based Approach to Assisting Collaborative Relationships

by

Winston Dali Chang

Submitted to the
Department of Electrical Engineering and Computer Science

May 23, 2003

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

As engineering projects become more complex and involve an increasing number of stakeholders that are becoming more geographically displaced, the necessity arises for tools that can facilitate close interaction between the participants and surmount geographic and temporal barriers. SSPARCy and Multi-attribute Interview Software Tool (MIST), which were developed by the Productivity from Information Technology (PROFIT) initiative at the Sloan School of Management at the Massachusetts Institute of Technology (MIT), offer new theoretical constructs that incorporate protocols, knowledge representations, and design methodologies to better enable stakeholders of a system to interact more closely, partially by having continuous access to the evolving knowledge of the engineering design. The combination of the SSPARCy and MIST tools results in a very comprehensive collaborative design tool; this is attained by establishing pathways of knowledge flow for the combined systems. The tool employs new methodologies to elicit knowledge from the collaborative environment in approaches and techniques that have not been currently utilized by other collaborative tools. This paper describes the broad functionality of SSPARCy and MIST and details the significant value added by such tools in a collaborative engineering environment as well as in other domains.

Thesis Supervisor: Dr. Amar Gupta
Title: Co-Director, Productivity From Information Technology Initiative,
Sloan School of Management

2
Contents

1 Introduction 6

2 Background: Evolution of Collaborative Engineering 9
   2.1 Origins of Computer-Supported Cooperative Work 9
   2.2 Effect of Internet on Collaborative Engineering 10
   2.3 Development of Collaborative Engineering Tools 12
   2.4 Current Tools in Collaborative Engineering 13
      2.4.1 Collaborative Information Space 13
      2.4.2 TeamSCOPE System 15
      2.4.3 MADE Program 17
      2.4.4 ICEMaker 18
      2.4.5 Rule-Based Algorithms from Chung-Hua University 21
      2.4.6 Collaborative Dynamic Project Management 21
      2.4.7 DICE 22
   2.5 Collaborative Tool Comparisons 24

3 Core Architecture of SSPARCy/MIST 27
   3.1 SSPARCy 27
   3.2 MIST 29
      3.2.1 Relationship Analysis Tool 30
   3.3 Four-Faceted Knowledge-Based Approach 32
      3.3.1 Knowledge Acquisition 33
      3.3.2 Knowledge Management 34
      3.3.3 Knowledge Discovery 36
      3.3.4 Knowledge Dissemination 37

4 Pathways of Knowledge Flow in SSPARCy to MIST 38
   4.1 Relating System Specifications to Design Properties 39
   4.2 Relating Attributes to Properties 40
   4.3 Knowledge Through Interviews 42
      4.3.1 Bracketing Methodology 44
      4.3.2 Single Interview 45
      4.3.3 Corner Point Interview 47
   4.4 Knowledge Discovery in MIST 48
      4.4.1 Calculating with Multi-Attribute Utility Analysis 51
      4.4.2 Alternatives to Multi-Attribute Utility Analysis 53
   4.5 Additional Satellite Design Example 54

5 Value of SSPARCy/MIST Approach 56
   5.1 Knowledge Management: Retention and Transfer 56
   5.2 Additional Domains of Application 57
      5.2.1 Commercial Engineering Application 58
      5.2.2 Professional Work Place Application 59
   5.3 Collaborative Tools in Use in Aerospace Industry 61
      5.3.1 Collaboration at Avidyne Corporation 62
**List of Figures**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Taxonomy of Collaborative Engineering Tools</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Multidimensional Navigation of CIS [8]</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>TeamSCOPE File Manager and Stakeholder Summary [9]</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>ICEMaker Project Status [10]</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>Comparative Matrix of Collaborative Tools and Their Respective Features</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>SSPARCy Data Model [15]</td>
<td>28</td>
</tr>
<tr>
<td>7</td>
<td>Main Form of MIST Tool</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>Four-Faceted Knowledge-Based Approach</td>
<td>33</td>
</tr>
<tr>
<td>9</td>
<td>MIST Knowledge Flow Using Four-Faceted Knowledge-Based Approach</td>
<td>34</td>
</tr>
<tr>
<td>10</td>
<td>Pathways of Knowledge Flow in SSPARCy and MIST</td>
<td>38</td>
</tr>
<tr>
<td>11</td>
<td>Example Design Properties and Rationale [19]</td>
<td>40</td>
</tr>
<tr>
<td>12</td>
<td>Example Design Attributes and Definition [19]</td>
<td>42</td>
</tr>
<tr>
<td>13</td>
<td>Example Design Properties and Design Attributes Matrix [19]</td>
<td>43</td>
</tr>
<tr>
<td>14</td>
<td>Full Interview Bracketing Tree.</td>
<td>45</td>
</tr>
<tr>
<td>15</td>
<td>Example Utility Function for Accuracy Attribute from a Single Interview [19]</td>
<td>46</td>
</tr>
<tr>
<td>16</td>
<td>Example Single and Corner Point Interview Results [19]</td>
<td>48</td>
</tr>
<tr>
<td>17</td>
<td>Main Form of Relationship Analysis Tool</td>
<td>50</td>
</tr>
<tr>
<td>18</td>
<td>Example Discovery of Utility Values for Multiple Attribute Sets [19]</td>
<td>51</td>
</tr>
<tr>
<td>19</td>
<td>Total Separation Rates for Various Industries in 2002</td>
<td>60</td>
</tr>
<tr>
<td>20</td>
<td>Requirements for a System Attribute Displayed in DOORS</td>
<td>63</td>
</tr>
<tr>
<td>21</td>
<td>Collaborative Tool Contributions and Possible Improvements</td>
<td>67</td>
</tr>
</tbody>
</table>
1 Introduction

In today’s society, collaboration has become a staple of most undertakings in the professional work environment and in social settings. Most enterprises and endeavors rely on the participation and contributions of many diverse stakeholders. In the professional work environment, group projects and design teams are often put together to design products and to develop business solutions. Academic research increasingly involves active participation of research groups that are geographically displaced across the globe. As projects and engineering designs become more complicated and employ more resources, a more advanced methodology needs to be constructed to help aid the stakeholders of a collaborative environment.

The major challenge in collaborative environments is the efficient coordination and maintenance of communication amongst stakeholders of a system. It is becoming increasingly difficult for stakeholders in a collaborative environment to interact because the collaborative environments are becoming increasingly geographically dispersed. In order for a design team to be fully effective, each stakeholder must possess a clear and efficient expression of the knowledge representation of the system. In traditional design environments, such important issues are handled more easily due to the convenience of working in the same location, and the relevant knowledge can be quickly disseminated to enable stakeholders to easily recognize past instances of current problem states [1]. However, in a distributed collaborative environment, it is more difficult for the knowledge representation of a system to be clearly conveyed due to the increased amount of information and the problems associated with geographic boundaries, organizational boundaries, and departmental boundaries.
Recognizing the importance of communication and knowledge transfer, the Productivity from Information Technology (PROFIT) initiative at the Sloan School of Management at the Massachusetts Institute of Technology (MIT), in conjunction with the Space Systems, Policy, and Architecture Research Consortium (SSPARC) in the Department of Aeronautics and Astronautics at MIT, have developed an integrated paradigm that incorporates appropriate protocols, knowledge representations, and design methodologies to better enable stakeholders of a system to collaborate and to analyze past decisions in the design process. The system that was developed is composed of two tools, SSPARCy and the Multi-attribute Interview Software Tool (MIST). SSPARCy addresses the issue of knowledge management through its capture of major design decisions, thereby allowing for better insights into past design trends. MIST incorporates a formalized means of exploring the tradespace of an endeavor through the application of utility and expense analyses. These tools are related through the stakeholder relationship sub-tool that allows for the combination of both SSPARCy and MIST analyses. While a specific target domain was examined in detail, both SSPARCy and MIST can be adopted to cater to other applications that involve collaboration. The range of potential applications spans from development of global sales strategy to design of training programs and from market analysis studies to emergency crisis management.

This thesis addresses the application of SSPARCy and MIST as a single tool that results in a new methodology to elicit knowledge from collaborative environments in a space that has never been explored before. The tool allows stakeholders of a collaborative system to elicit knowledge of design preferences from an interview process and then utilize these preferences to identify their relationships to design attributes and design properties. By
chaining the pathways of knowledge flow between the two original tools, stakeholders can better understand how design decisions made at any level of the engineering design process will impact the overall collaborative endeavor. Through this process, stakeholders gain unprecedented insights into historical information and thereby become better equipped to make educated decisions regarding future design directions.

This paper delves into the details of SSPARCy and MIST and describes how the analysis of stakeholder relationships to various knowledge representations can provide vital information of the design process. Section 2 investigates the current trends in collaborative engineering and includes comparisons of a representative set of current collaboration tools. Section 3 focuses on the core architecture of SSPARCy and MIST. Section 4 details the knowledge flow through these two systems and highlights the application of these tools to diverse facets of collaborative design. Section 5 describes the value added from SSPARCy and MIST and also delineates potential applications to which they could be applied. Section 6 concludes the paper with some final thoughts on the SSPARCy and MIST framework.
2 Background: Evolution of Collaborative Engineering

Communication is the most important issue in a collaborative engineering environment. The ability to get all stakeholders of a design team working together, understanding the same process with an uniform representation, and being able to convey new ideas are the key foundations for collaborative engineering. In essence, collaboration attempts to leverage the collective intelligence of the group. Collaboration tools have become increasingly important due to the increasing complexity of engineering designs and the globalization of all facets of work.

The application of technology to collaborative initiatives did not occur until the mid-1980's. The rest of this section details the evolution of collaborative engineering tools and processes.

2.1 Origins of Computer-Supported Cooperative Work

In 1984, Iren Greif of MIT and Paul Cashman of Digital Equipment Corporation organized a workshop of twenty individuals from varying fields; these individuals had all expressed an interest in exploring technology’s role in assisting collaboration in the work environment. It was then that they coined the phrase “computer-supported cooperative work” (CSCW) to describe the initiative [2].

CSCW began as an effort by technologists to learn from numerous other fields of study such as economics, social psychology, anthropology, and organizational theory to shed light on how to maximize collaboration amongst individuals in a group endeavor. Since the identification of CSCW as a new field, research has been primarily focused on the issue of maintenance of communication between shareholders. Not until recently has the focus
shifted more towards analysis of design decisions and relations of knowledge representations rather than focusing solely on communication. During the past 10 years, research and development in CSCW has shifted from single user applications such as word processing to group support applications such as integrated product development environments [3]. Nowadays, CSCW systems encompass the following characteristics: interaction between stakeholders, management of knowledge, dissemination of information, and knowledge discovery.

2.2 Effect of Internet on Collaborative Engineering

Before the development of electronic communications networks, stakeholders of a system were required to work in the same room in order to collaborate on projects. In the 1980’s, the process of numerous stakeholders collaborating on different aspects of a project at one location was known as concurrent engineering [4]. By the early 1990’s, as local area networks developed into wide area networks and finally into the World Wide Web, it became possible for stakeholders to utilize the communication and dissemination capabilities of networks encompassing large geographic areas. The majority of the projects that were developed around this time were designed as communication facilitation tools, and very few of these projects involved a knowledge representation of the collaborative environment or the eliciting of knowledge from stakeholders involved.

Speed and connectivity are arguably the Internet’s greatest contributions to collaborative engineering. The fact that the Internet is accessible via computer from any location, any time, and with very little third party assistance, allows for it to serve as a conduit for a generalized tool for knowledge acquisition, knowledge analysis, and knowledge
dissemination. With the increasing globalization of businesses and engineering endeavors, key stakeholders of businesses or engineering processes can rarely be found at the same location at the same time. The World Wide Web offers tremendous potential for collaborative information sharing amongst stakeholders who may be geographically displaced in both time and space. Steady growth in telecommuting and electronic communications, such as email or online messaging, has served to facilitate communication between participants [5]. As technology continues to improve, the capabilities of telecommuting become more advanced, moving from asynchronous processes such as email and fax to interactive processes such as video conferencing.

Since the late 1990's, as use of the Internet has become increasingly prevalent, distance collaboration using teleconferencing and shared media has become a significant area of research and development. Examples of early research prototypes that utilize the Internet are Media Space project at Xerox PARC, Cruiser and Touring machine projects at Bellcore, Argo system at DEC, and the Ontario Telepresence Project [6]. These projects typically involved the use of proprietary systems and analog video to support interaction among stakeholders.

Nowadays, web-based collaboration no longer revolves solely on the transferring of design files over the Internet, but now encompasses systems that can offer interactive, real-time design review and mark-up through the Internet. Technology has come a long way from participants using the Internet to send text messages, to audio/video conferencing, and now to full-blown collaborative tools.
2.3 Development of Collaborative Engineering Tools

As Internet use became more prevalent in the mid-1990’s, small collaborative engineering processes that utilized the Internet began to be packaged together as specific collaborative tools.

Many recently developed tools focus on the development of a collaborative CAD framework in which all stakeholders of a design are able to access a particular CAD design. Other general knowledge management and design tools focus more on the overall high-level understanding of a project rather than on the technical specifications [7]. These higher level collaboration tools are the type of tools that will be later discussed in this thesis.

Collaborative engineering tools support two main functions: knowledge retention and knowledge discovery. Knowledge retention encompasses the capture and archiving of knowledge of past design decision rationale and stakeholder preferences, as part of the evolving knowledge representations of the collaborative environment. Knowledge discovery encompasses the creation of new knowledge and ideas through the analysis of past decision rationale and stakeholder preferences.

The recent incorporation of knowledge-based systems and other artificial intelligence techniques into the basic knowledge management framework allows for the creation of more intelligent systems. The new era of knowledge-based collaborative tools will allow designers to better manage the evolution of products and services. Because of this trend of incorporating knowledge-based properties, more collaborative engineering tools will begin to support knowledge discovery functionalities.
### 2.4 Current Tools in Collaborative Engineering

As mentioned in Section 2.3, collaborative engineering tools can be grouped into expressing two main functionalities: knowledge retention and knowledge discovery. Figure 1 below shows the categorization of the specific collaborative engineering tools that will be discussed in the following subsections.

<table>
<thead>
<tr>
<th>Knowledge Retention</th>
<th>Knowledge Discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>MADE Program</td>
<td>Chung-Hua University Research</td>
</tr>
<tr>
<td>CIS</td>
<td>CDPM Platform</td>
</tr>
<tr>
<td>TeamSCOPE</td>
<td>DICE</td>
</tr>
<tr>
<td>ICEMaker</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Taxonomy of Collaborative Engineering Tools

#### 2.4.1 Collaborative Information Space

The concept of the Collaborative Information Space (CIS) was developed at the University of Karlsruhe in Germany. The impetus behind the tool was to transform individual knowledge into collective useful information through the creation of 'information objects' in the CIS framework [8]. The implementation of information objects in CIS allows the system to represent individual knowledge in a common format and to archive this knowledge in a multidimensional storage structure. The tool is accessible through a web-based user interface.
The classification scheme of CIS identifies three top-level object classes: process, function domain, and information aspect. A process describes a sequence of activities, a function domain covers concrete regions of interest, and the information aspect represents the improved processing and allocation of information in this structure. Figure 2 depicts a screenshot of the multidimensional navigation capacity of CIS with the structure browser interface.

![Multidimensional Navigation of CIS](image)

Figure 2: Multidimensional Navigation of CIS [8]

The Collaborative Information Space tool excels in its knowledge retention functionalities while exhibiting poor knowledge discovery characteristics. The tool mainly serves to act as a knowledge repository, organizing all relevant stakeholder knowledge into a collective
representation where all constituents can access and understand the information. Though
the CIS tool is powerful in its archiving and retention of knowledge, the tool fails to
analyze and extrapolate relationships from the data that it retains.

2.4.2 TeamSCOPE System

TeamSCOPE is a web-based collaborative (WBC) system developed in late 1998 and early
1999 at Michigan State University to respond to the deficiency of information and awareness
casted by geographical separations of globally distributed engineering design teams [9]. The
TeamSCOPE system addressed the need for a common representation for an engineering
project.

TeamSCOPE focused on four aspects: activities of all stakeholders, scheduling and
availability of stakeholders, state of design, and knowledge representation of design.
TeamSCOPE contained a number of features that allowed for the tool to address the
above collaborative design issues including: file manager, message board, calendar,
activity summary, activity notification, usage information, and a knowledge representation
summary. TeamSCOPE relied on the use of an interview process with all stakeholders of
the system to acquire most of the information. Figure 3 shows the file manager and a
stakeholder summary of the TeamSCOPE tool.
Figure 3: TeamSCOPE File Manager and Stakeholder Summary [9]
The interview process used by TeamSCOPE incorporates more functionalities and is more flexible than other WBCs; however, a major drawback of the TeamSCOPE tool is that it is still missing a real-time component. Like the CIS tool, TeamSCOPE focuses mainly on knowledge retention and facilitation of communication between stakeholders and is lacking in knowledge discovery functionalities.

2.4.3 MADE Program

The Manufacturing Automation and Design Engineering (MADE) program, sponsored by the Defense Advanced Research Projects Agency (DARPA), focuses on the need to support teams of specialists from different locations for specific collaborative engineering endeavors focused on knowledge retention [6]. The goal was to develop a highly flexible design environment that would allow stakeholders to evaluate and develop more design alternatives for an engineering system, in order to optimize product characteristics and to quickly prototype complex products and processes based on past decisions and designs.

The MADE program utilizes the World Wide Web to act as a knowledge repository for the stakeholders to reference and build upon. With specific protocol and documentation methods established for each specific engineering design, the MADE program is able to establish an effective knowledge repository to assist in collaborative design. The shared repository of MADE resides online, and this repository is responsible for all CAD models, calculations, test results, and links to other top-level project pages. The MADE platform supports the integration of external applications that are allowed to dock onto the platform to perform analysis on all information retained by MADE. Possible collaborative tools that can be used include authoring, document control, synchronous communication, and asynchronous communication tools.
The MADE program is very versatile and can be applied to numerous collaborative design endeavors. However, the program has not settled on a universal application that can be applied to all collaborative designs. Instead, the program focuses on developing independent web-based tools for each particular collaborative initiative. Like most other collaborative engineering tools, MADE focuses only on knowledge retention and not knowledge discovery.

2.4.4 ICEMaker

ICEMaker is an integration software package that supports network-based data transfer, a system engineering database, and tool linkage. The tool was developed at the California Institute of Technology. ICEMaker allows stakeholders to monitor and communicate mission definitions, configuration definitions, and trade studies [10]. ICEMaker typically begins by guiding a design team through a process of analyzing individual stakeholder needs, and then systematically allocates data needs to each particular stakeholder based on submission requests. In addition, ICEMaker acts as a knowledge repository where stakeholders can search for all technical data developed and used during the design process.

ICEMaker incorporates a single server, multi-client architecture in order to facilitate easy distribution of proposed system specifications. The single server acts as the central command and controller for the system, and each stakeholder is assigned a client and can interact with all other clients through the server. The clients typically employ Excel Workbooks to interact with the server station through input and output pages in the workbooks. Clients can also be written in any software application that has made its API available. For instance, a common non-Excel client is DrawCraft, a spacecraft rendering tool developed with a Visual Basic interface to SolidWorks. Excel/VBA was used as
the default client software due to the familiarity that most individuals would have with the Excel/VBA interface as well as the ease of programming in Excel/VBA. Excel/VBA provides pre-existing graphical interface capabilities as well as familiar data analysis tools such as graphs and figures. Though Excel/VBA may not be as robust as other software tools, it is a practical solution for the server/client architecture.

![Image of Excel/VBA interface]

Figure 4: ICEMaker Project Status [10]

ICEMaker focuses on the capture of mission and system requirements to assist in integrated concurrent engineering. Through the server/client architecture, ICEMaker facilitates collaborative design by allowing for distribution and updating of system specifications by clients who can publish their own proposed specs, read already published specs, and request information from other clients. The server maintains the flow of information by acting as the distributor of information amongst all clients.
Like many of the other collaborative engineering tools, ICEMaker does not address the issue of knowledge discovery. ICEMaker acts as a good conduit for collaborative design but does not explore the relational comparisons of various stakeholder preferences.

One issue of ICEMaker is that each client in ICEMaker has the ability to post its own system specifications and to have a building model of the design in question. Since there are multiple clients in an ICEMaker architecture, the problem arises that there will be numerous copies of different system specifications with no centralized means to differentiate the models. This type of approach is known as multiple concurrent design and is typified by the multiple design alternatives that are available after the analysis process is complete. Typically at the end of the design, stakeholders would decide between the multiple design alternatives by taking into consideration the various tradeoffs of each system.

The opposite of such an approach is the single design approach where stakeholders work together in building a single model of the system. Such a design process requires more compromise of individual stakeholder preferences with the mentality of designing a single collective engineering system. Both design approaches differ in many ways, and it is arguable whether one is more effective than the other.

Another issue regarding ICEMaker is that it is not totally a real time collaboration tool. Though the tool does allow for the interaction of clients to publish and subscribe directly from their design tools, the interaction mechanisms involve asynchronous processes. In order for a system to fully be a real time collaboration tool, it must express the functionalities that allow for real time synchronous communication.
2.4.5 Rule-Based Algorithms from Chung-Hua University

Researchers at the Chung-Hua University of Taiwan have been developing an automated system that analyzes customer goals and desires in an engineering design and implements knowledge discovery of design parameters that match these goals. These rule-based programs analyze stakeholder preferences and attempt to create new knowledge regarding what types of design parameters would be ideal in order to realize the stakeholder preferences in question [11].

The system incorporates a rule-based algorithm for transferring customer preferences directly into application specific parameters by relating the attributes and design parameters through a weighted algorithm. The shortcoming of this system resides in its inability to analyze the effects of multiple customer preferences in determining a set of design parameters that match the intended goal for all constituents. In addition, the tool developed at Chung-Hua University does not support a knowledge repository capacity that would allow customers to maintain evolving states of development of a design process.

This collaborative engineering tool is different from the others mentioned so far because it focuses on the knowledge discovery aspect while implementing very little in terms of knowledge retention. Usually collaboration tools exhibit the opposite of this, acting as very good retention and acquisition tools but not analyzing any relevant knowledge.

2.4.6 Collaborative Dynamic Project Management

The Collaborative Dynamic Project Management (CDPM) system focuses on providing a collaborative project management platform that offers project information and data analysis tools to all stakeholders. CDPM also offers meeting protocols and a knowledge
representation of the design being evaluated [12]. The system was developed at the Massachusetts Institute of Technology’s Intelligent Engineering Systems Lab.

The system architecture of CDPM can be divided into three basic components: client component, collaboration manager, and project resources. The client component consists of the graphical user interface. The collaboration manager handles the flow of information regarding stakeholders and resources employed. The project resources component stores all the knowledge regarding the project itself. This is where the knowledge discovery and analysis occurs.

The problem with most collaborative engineering tools currently designed is that they do not support analysis tools to facilitate knowledge discovery within the design under consideration. The CDPM platform includes analysis tools that can assist in the knowledge discovery process. In addition, the system is flexible enough that it can interact with different kinds of computing devices to interact and share information in a collaborative environment. Unlike many of the other tools, the CDPM system is an example of a knowledge retention and knowledge discovery tool.

2.4.7 DICE

DICE (Distributed and Integrated Collaborative Engineering Environment) is a distributed computer-supported collaborative agent-based framework for engineering design. DICE can be thought of as a network of computers and users where the communication and coordination is handled by a control mechanism and global database [13].

The DICE framework consists of three main components: a Blackboard, Knowledge Modules, and a Control Mechanism. The Blackboard acts as an object-oriented global database or shared workspace and is the medium through which all of the communication
of the framework occurs. The DICE Blackboard has a Negotiation partition that facilitates the interactions between various engineers taking part in the design and manufacturing process. In the design process, the database is partitioned so that each client group can maintain their own private database. While working on a design, each client modifies their own private database, and after a designer is satisfied with their specifications, it is then released to the global database to be shared with all other clients. Knowledge Modules can be thought of as little agents that assist in various aspects of the design framework: aiding in the Control Mechanism, design and construction, system checking, and algorithmic analysis. Knowledge Modules can make changes or request information from the Blackboard - these changes or requests are logged and changes to the Blackboard can lead to the triggering of possible other Knowledge Modules. The Control Mechanism basically performs two main tasks: i) evaluate and propagate implications of actions taken by Knowledge Modules and ii) assist in the coordination process of the system [13].

Main functional components of DICE include a base layer object-oriented database management system (OODBMS), COSMOS which extends C++ to allow for forward and backward chaining, GNOMES which provides a geometric modelling framework, and SHARED which is an information model supporting collaboration. These main components combined with other structures are the building blocks of the DICE system.

DICE is powerful in its flexibility as a tool, and its ability to support numerous tools and functionalities including modelling, design representation archiving, media synchronization, and organizational issues through the framework residing in the main DICE Blackboard. A main difference between DICE and other Blackboard systems is that DICE's Blackboard is more than just a static repository of knowledge, instead, it is a dynamic, knowledge
database that actively responds to different types of interactions and messages that are managed by the Control Mechanism.

The DICE framework can also be considered as another example of a knowledge retention and knowledge discovery tool. DICE enables the acquisition of the knowledge representation of the system and also supports the knowledge analysis and discovery through the application of numerous Knowledge Modules.

2.5 Collaborative Tool Comparisons

After discussing the basic functionalities of a few examples of some of the current collaborative tools in use, Figure 5 below shows a relational comparison of each tool to the collaborative functionalities they exhibit.
Collaborative Engineering Tools

|                      | SSPARCY | MIST | CIS | TEAM-SCOPE | MADE | Icemaker | ° ° ° | ° ° ° | ° ° ° | ° ° ° | ° ° ° | ° ° ° | ° ° ° | ° ° ° | ° ° ° | ° ° ° | ° ° ° |
|----------------------|---------|------|-----|------------|------|----------|------|------|------|------|------|------|------|------|------|------|------|------|
| KNOWLEDGE RETENTION  |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - acquire knowledge  | ●       |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| through interviews   |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - acquire knowledge  | ●       |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| through manual entry |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - archives design    |●         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| states               |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - archives design    |●         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| rationale            |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - common knowledge   |●         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| representation for   |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| tool                 |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| KNOWLEDGE ANALYSIS   |●         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - analyzes stakeholder|         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| preferences          |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - analyzes stakeholder|         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| relationships        |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - analyzes design vs. |         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| stakeholder preferences |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - utility analysis of |         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| design               |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - infers future      |●         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| design sets          |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| KNOWLEDGE TRANSFER   |●         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - ease of access for |         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| tool                 |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - visual/graphical   |●         |      |     | ●          |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| representations      |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |
| - web-enabled        |         |      |     |            |      |          |      |      |      |      |      |      |      |      |      |      |      |      |

Figure 5: Comparative Matrix of Collaborative Tools and Their Respective Features
There are a group of tools that focus mainly on knowledge retention and knowledge transfer. These types of tools include CIS, TeamSCOPE, MADE, and ICEMaker. These tools were mainly designed to be a knowledge repository of the design process in order to archive the evolution of the collaboration endeavor. These tools are lacking in any sort of knowledge analysis and would not be usable for parties interested in performing knowledge discovery. This is the most common type of current day collaboration tools.

Another group of tools focuses solely on knowledge analysis and knowledge discovery. Examples of such tools are the rule-based algorithms developed at the Chung-Hua University. The deficiencies of systems that focus solely on knowledge analysis and knowledge discovery is that they are not stand-alone tools for representing an entire system. These types of tools do not exhibit the functionalities that allow for knowledge capture of the entire design evolution, and subsequently, these tools can only be used for specific instances of design analysis.

Finally the last group of collaborative engineering tools are the tools that exhibit knowledge retention, knowledge analysis, and knowledge transfer. The CDPM platform and DICE are two examples of such systems. They both are very strong as knowledge repositories of the design process, and both tools exhibit some data analysis functionalities.
3 Core Architecture of SSPARCy/MIST

The SSPARCy and MIST research was initiated as part of the Space Systems Policy and Architecture Consortium in the Department of Aeronautics and Astronautics at MIT. The purpose of the consortium was to examine space system design from numerous viewpoints with the objective of developing optimal strategies for choosing between various architectures of space system design. The two tools were designed incorporating many functionalities that existed in collaborative tools [14], along with additional functionalities for data analysis and knowledge discovery.

Even though the SSPARCy and MIST tools were designed with a space system construct in mind, the tools have since been designed to be flexible enough to be able to accommodate numerous other domains and endeavors. Later in Section 5.2, other possible applications for the theoretical constructs of SSPARCy and MIST will be outlined and explored. This section covers the actual theoretical constructs of the two tools under consideration.

3.1 SSPARCy

SSPARCy is geared to capture the rationale concerning major design decisions with the intent of enabling faster and less expensive design endeavors in the future [15]. The SSPARCy system provides efficient access to information regarding MATLAB source code files, and through graphical displays of the information collected, the analysis of these MATLAB code expressions can be viewed. In addition to graphically displaying the knowledge collected, the tool is able to record states of the code throughout the design process, perform integration integrity checks on source code, and aid in the acquisition of design rationale for major system decisions. The SSPARCy system solicits design rationale
by prompting the designer to manually describe major design decisions in a textual format. The original prototype of SSPARCy consists of a myriad of system specific MATLAB scripts that allows for transfer of knowledge regarding design rationale over time and the creation of history reviews, error checking, and system analysis.

SSPARCy represents the system at hand thorough the manipulation of design properties. System design properties are the actual system parameters of the engineering design process, which are based on the system specifications defined by the engineers. Property values change as the design process progresses partly because the engineers of the system change design specifications.

The operation of SSPARCy begins with a capture of the design in question in one major object referred to as the Project object. This Project object contains all the appropriate objects and variables correlating to all the information stored in the system. The other objects of the system are: Function objects, Constant objects, Design Variable objects, and Error objects. A graphical view of the data model is shown in Figure 6.

![Figure 6: SSPARCy Data Model [15]](image)

Function objects refer to actual functions in the source code, Constant objects refer to global constants that are defined in the system, and Design Variable objects are the objects
that contain all relevant information regarding each particular variable in the project.

3.2 MIST

MIST employs the Multi-Attribute Tradespace Exploration (MATE) process to provide a formalized means of exploring the tradespace of an engineering system through the incorporation of preferences into design decision criteria. By incorporating methods developed in economics, operations research, and other disciplines [16], MIST can accurately and lucidly record and analyze each stakeholder’s preferences for the engineering design. In addition to facilitating knowledge capture and knowledge retention of design rationale, MIST provides the functionality to analyze the knowledge garnered from past decisions in order to specify future endeavors and protocols for the engineering process currently under consideration.

The motivation behind the development of MIST was to establish a tool that is able to develop a higher-level representation of the system in question, and for the tool to use this representation to explore the preferences of the various stakeholders involved using cost-benefit and utility analysis methods. In addition to these methods, MIST employs the use of interview techniques to capture basic design attribute utility and expense functions from which all other analyses are based [17].

MIST operates primarily via manipulation of design attributes. Design attributes are defined characteristics of the project that describe the important factors for stakeholders. They are abstract variables that are used to represent and define the system under investigation. Though attributes are defined by the stakeholders and designers of the system, they also have specific values that are calculable at a specific instance of time.
Attribute values are calculated from equations utilizing weighted values of exact design properties.

3.2.1 Relationship Analysis Tool

With the basic functionality developed to capture design iterations and represent stakeholder preferences in MIST, the implementation of data mining technologies to analyze MIST and SSPARCy output appeared to be the next logical progression for the collaboration tools to take. The relationship analysis sub-tool bridges the gap between SSPARCy and MIST by providing for a means of relating design properties and design
attributes. The sub-tool is able to dynamically show the changes and effects of design decisions following the pathway of knowledge flow from system specifications to design properties, to design attributes, to stakeholder preferences, and finally to stakeholder relationships.

In particular, the sub-tool enables designers to examine the relationships and dependencies of each stakeholder’s utility and expense values to the attribute set of a system’s design. This accommodates the discovery of knowledge concerning the derivation of an optimal attribute set that would maximize utility and expense for each and every stakeholder.

The relationship analysis tool offers valuable insights through the analyses of the relationships between stakeholder preferences of a collaborative engineering system. The sub-tool creates knowledge discovery in three ways. First, the tool can analyze and calculate utility and expense values for each stakeholder of a system design based on a given set of design attributes. Second, by taking into account the utility and expense values of a stakeholder, the tool can generate and display the attribute set that generates the values for that particular stakeholder and then show all correlating utility and expense values for every other stakeholder based on the attribute set that is identified. Third, the tool ties together the attribute values from MIST to the properties of SSPARCy allowing for stakeholders to realize how changes to design properties impact design attributes and resulting stakeholder preferences. This powerful feature of the sub-tool allows for real-time analysis of the data flow from design properties to stakeholder relationships.

With the relationship analysis tool, the consequences of each design decision to system attributes can be recorded based on the changes to the utility and expense functions for
all of the stakeholders in the system. The tool captures the evolution of the system design by recording all changes to the system attributes and relates each change to the resulting changes of stakeholder utility and expense. It is this capture of data and the effects of design decisions on attribute scenarios and utility and expense values that adds value to the MIST system. Relationships between various roles in the design process are better explained through their utility and expense data, and future projects benefit from knowing how past projects in the same tradespace operated under similar conditions.

MIST is designed to assist stakeholders in maximizing the utility of a design process. The relationship analysis tool is designed to add the functionality of dynamic representation of dependencies between stakeholders. In order to develop a strong understanding of stakeholder dependencies and relations, the dynamic relationships of utility and expense values to design properties and design attributes needs to be clearly communicated.

3.3 Four-Faceted Knowledge-Based Approach

MIST was designed utilizing the notion of the four faceted knowledge-based approach of knowledge acquisition, knowledge management, knowledge discovery, and knowledge dissemination developed by the PROFIT initiative at MIT [18]. This knowledge-based approach emphasizes that the utility of knowledge can be maximized when it is efficiently captured, understood, and reused in complex endeavors. MIST strives to attain this objective through the leveraging of the four facets as shown in Figure 8. In the figure, the arrows show the flow of knowledge with external knowledge coming in at the knowledge acquisition facet and either leaving after knowledge dissemination or being leveraged further by being circulated back into the knowledge acquisition facet.
To better understand the MIST tool, one needs to understand which functionalities belong to which facet of the knowledge-based approach. Figure 9 shows the design flow within the MIST tool with the functionalities of the tool sectioned into the four faceted knowledge-based approach introduced above.

### 3.3.1 Knowledge Acquisition

Knowledge acquisition involves the process of capturing information from various media, such as people's minds and handwritten documents, into computer accessible media. After acquiring the knowledge of a design system, this information can be used to analyze past design trends and to drive future initiatives. By electronically capturing and archiving knowledge, a system can utilize the computational benefits provided with a software application.

In MIST, the acquisition of knowledge is implemented through the programming of
Figure 9: MIST Knowledge Flow Using Four-Faceted Knowledge-Based Approach

the MAUA interview process into the software tool that enables the system to quickly conduct interviews with more flexibility in terms of data analysis and archiving. During the interview a stakeholder decides between varying scenarios of attribute values to determine the stakeholder’s preferences for the set of design attributes. By conducting the interviews electronically through MIST, stakeholders can easily adjust attribute values, rerun interviews, and conduct analysis using their attribute preferences. Calculations and analyses can be easily displayed and monitored throughout the interview process, allowing stakeholders to take advantage of real time analysis of their preferences and interviews.

3.3.2 Knowledge Management

Knowledge management involves the process of establishing an accurate representation of system knowledge by mitigating issues relating to differences in underlying contexts of information coming from dissimilar sources such as multiple stakeholders, multiple
projects, and multiple states of the process. The management of knowledge is crucial in an engineering design environment because a correct representation of the acquired knowledge is imperative for the understanding and future analysis of shared information between stakeholders. If acquired knowledge is the foundation upon which the system analysis will build from, a poor representation of this knowledge will cause the system to yield poor results.

MIST focuses on knowledge management in two ways: first, through its representation of the system as a set of design attributes and second, through the use of visual representation of stakeholder utility and expense functions.

MIST utilizes the notion of attributes to represent and analyze the system properties of an engineering design. Attributes, in addition to being defined by the stakeholders of a system, are linked to system properties through mathematical functions typically encapsulated in MATLAB. As mentioned earlier, the set of design attributes serve as the basis for all interviews and analysis in MIST. By appropriately representing the engineering design into attributes, MIST is able to effectively acquire and develop knowledge from design decision rationale and stakeholder preferences.

The visualization of utility and expense functions is another knowledge management functionality implemented in MIST. In addition to single attribute functions, multi-attribute cost and utility functions are graphically displayed and dynamically changed as stakeholders proceed through their respective interviews. This functionality allows the stakeholder to visually understand the exact preferences that are being decided as the interview process progresses. Stakeholders can view how their utility and expense functions evolve based on various design decisions over time.
3.3.3 Knowledge Discovery

Knowledge discovery involves the use of emerging artificial intelligence techniques to analyze large amounts of information with the goal of obtaining additional insights from prior knowledge.

MIST demonstrates its knowledge discovery capabilities through its ability to relate utility and expense functions derived from design attributes to actual design rationale. By using the interview process to elicit stakeholder preferences, MIST is able to discover stakeholder specific information regarding what the goals and predilections of each stakeholder are. With such knowledge, the system is able to analyze stakeholder preferences and to conduct sensitivity analysis on how these preferences are impacted by major changes in design parameters. In this process, MIST can also identify which design trends have been effective in the past and the rationale behind them.

MIST's stakeholder relationship analysis tool facilitates knowledge discovery by relating all stakeholder utility and expense values in order to derive an optimal set of design attributes. The ability to relate multiple knowledge representations of the collaborative environment whether its design properties, design attributes, or stakeholder preferences is one of the main functionalities resulting from the combination of SSPARCy and MIST tools. This knowledge discovery functionality differentiates the combined SSPARCy and MIST tool from most other current collaborative engineering tools which focus solely on the retention and dissemination of knowledge.
3.3.4 Knowledge Dissemination

Knowledge dissemination involves the automated extraction of the most relevant pieces of information from the engineering design, coupled with the distribution of such knowledge amongst all constituents of the design process.

MIST facilitates knowledge dissemination by creating an environment where multiple stakeholders are able to access feedback regarding the factors that influence utility and expense for their engineering system. MIST displays the generated relationships between stakeholder utility and expense values to provide knowledge regarding stakeholder dependencies and the effects of design decisions.

Since MIST, in a sense, consolidates information concerning stakeholder preferences and design rationale, MIST can be interpreted as a knowledge repository used for the dissemination of acquired, represented, and created knowledge. With MIST’s comprehensive archiving capabilities, the evolving knowledge repository will gradually encapsulate all design iterations and rationale for the system in question.

The most important outputs of the SSPARCy and MIST tool are the combined relational equations and graphs that depict the relations of all stakeholders of a system to one another, and the dynamic equations that relate design properties to design attributes and to stakeholder preferences. This information is disseminated through the various graphical interfaces of the combined SSPARCy and MIST tools. These relationships are what define the SSPARCy and MIST tools and allows for the better facilitation of collaborative engineering.
4 Pathways of Knowledge Flow in SSPARCy to MIST

The positive synergies that result from the combination of SSPARCy and MIST is the main impetus for the combination of the tools as a singular collaborative engineering tool. When combined, the knowledge flow from system specifications to design properties to design attributes to relational analyses of these components is fluid and comprehensive. The resultant chaining of the pathways of knowledge flow facilitates the understanding of the causal effects of design changes to stakeholder preferences and stakeholder relationships.

This section details the explicit knowledge flow within the SSPARCy and MIST constructs. The process begins with the SSPARCy tool defining all of the design properties from specific system specifications. From here, these design properties are incorporated in order to derive design attributes for use with the MIST system. After design attributes have been established, the stakeholder interview process may commence. After all interviews have been conducted, the knowledge acquired is analyzed in numerous ways resulting in knowledge discovery of stakeholder preferences and relationships. This chaining of knowledge flow is depicted in Figure 10.

![Diagram](attachment:image)

**Figure 10: Pathways of Knowledge Flow in SSPARCy and MIST**

This traversal of knowledge development from system specifications to actual
stakeholder preferences and relationship analyses provides a new methodology for better understanding collaborative engineering designs. This ability to span and assist in collaborative engineering all throughout the design process is one of the primary reasons that the SSPARCy and MIST tools are so innovative. By chaining together each of these knowledge representation types (i.e. design properties, design attributes, etc.), the combined SSPARCy and MIST tool is able to better understand the relationships between each type of knowledge representation. This way, when changes occur to a particular knowledge representation, the chaining knowledge flow model can be used to infer the resultant effects on the system.

In order to highlight the knowledge flow through SSPARCy and MIST, an example of collaborative engineering design involving the design of an orbital satellite is used in the following subsections.

4.1 Relating System Specifications to Design Properties

All engineering designs are based on outlined system specifications. System design properties are the actual system parameters of the engineering design process and help to design the actual values of the system specifications. Design property values change through the design process as the system designers change design specifications.

SSPARCy captures these values and categorizes them as design properties and stores them in Design Variable objects. These objects contain all relevant information concerning the design properties, including the rationale for all updates and changes. As the engineering design process continues over time, stakeholders can view any Design Variable object in order to add, remove, or change its current status. With the error checking
functionality of SSPARCy, one can analyze Design Variable objects and pinpoint the origin of errors that may occur in the engineering design.

In the case of designing orbital satellites, stakeholders would be required to first identify design properties for the system. The properties would be derived from system specs and would serve as the foundation for the SSPARCy system. Figure 11 shows an example of domain-specific design properties.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Units</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Apogee Altitude</td>
<td>Km</td>
<td>Specifies orbit/relationship to ionosphere</td>
</tr>
<tr>
<td>2 Perigee Altitude</td>
<td>Km</td>
<td>Specifies orbit/relationship to ionosphere</td>
</tr>
<tr>
<td>3 Number of Planes</td>
<td>INT</td>
<td>Key to meeting global coverage needs</td>
</tr>
<tr>
<td>4 Swarms per Plane</td>
<td>INT</td>
<td>Key to meeting global coverage needs</td>
</tr>
<tr>
<td>5 Satellites per Swarm</td>
<td>INT</td>
<td>Local coverage resolution</td>
</tr>
<tr>
<td>6 Size of Swarm</td>
<td>Km</td>
<td>Local coverage resolution</td>
</tr>
<tr>
<td>7 Number of Sounding Antennas</td>
<td>INT</td>
<td>Captures functionality trade</td>
</tr>
<tr>
<td>8 Sounding</td>
<td>0-3</td>
<td>Captures functionality trade</td>
</tr>
<tr>
<td>9 Short Range Communication</td>
<td>0-1</td>
<td>Captures functionality trade</td>
</tr>
<tr>
<td>10 Long Range Communication</td>
<td>0-1</td>
<td>Captures functionality trade</td>
</tr>
<tr>
<td>11 On-Board Processing</td>
<td>0-1</td>
<td>Captures functionality trade</td>
</tr>
</tbody>
</table>

Figure 11: Example Design Properties and Rationale [19]

The defining of design properties is one type of knowledge representation within the SSPARCy and MIST tools. By defining the entire collaborative environment in terms of design properties, which are derived from system specifications, this representation of the system captures the main ideas of what the engineers and system designers were intending to capture. This knowledge representation will later allow stakeholders to understand explicitly how changes to the collaborative environment may impact system specifications.

4.2 Relating Attributes to Properties

As mentioned earlier, design attributes are defined characteristics of the project that describe the important factors for stakeholders. Though attributes are defined by the
stakeholders and designers of the system, they also have specific values that are calculable at a specific instance of time. Attribute values are calculated from equations linking weighted values of design properties, which are derived from exact design specifications. The relationships between design attributes and design properties involve complex formulas and equations and are modelled with MATLAB functions.

One of the first steps in MIST is to allocate and categorize all design properties. After attributes have been selected and equations are developed that link numerous design properties to account for an attribute value, the attribute definition phase is completed. Attribute values only change due to changes to the values of their respective design properties.

In the example of orbital satellite design, attributes were selected by the stakeholders and later defined on the basis of the design properties that were chosen by SSPARCy. The set of attributes chosen represents the entire design from a high level design perspective. Each attribute is carefully detailed and the specifications for the attributes need to be clearly spelled out before analysis in MIST is conducted. Figure 12 shows an example of a set of attributes for the case of satellite design.

So far in the above descriptions of SSPARCy and MIST, there are two distinct aspects of knowledge representation: one incorporates design properties and the other utilizes design attributes. The relation between properties and attributes is defined through weighted equations coded in MATLAB that relate the two knowledge representations. A helpful visual representation of these weighted equations can be shown in a quality function deployment. In Figure 13, the design properties are listed as variables to the left of the matrix and the design attributes are listed as attributes in the top of the matrix. The matrix
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Definition</th>
<th>Best</th>
<th>Worst</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial Resolution</td>
<td>Area between which you can distinguish two data sets</td>
<td>1 deg X 1 deg</td>
<td>50 deg X 50 deg</td>
<td>0.15</td>
</tr>
<tr>
<td>Revisit Time</td>
<td>How often a data set is measured for a fixed point</td>
<td>5 minutes</td>
<td>720 minutes</td>
<td>0.35</td>
</tr>
<tr>
<td>Latency</td>
<td>Time for data to get to user</td>
<td>1 minute</td>
<td>120 minutes</td>
<td>0.40</td>
</tr>
<tr>
<td>AOA Accuracy</td>
<td>Error of angle of arrival measurement</td>
<td>0.0005 degrees</td>
<td>0.5 degrees</td>
<td>0.90</td>
</tr>
<tr>
<td>EDP Accuracy</td>
<td>Error of electron density profile measurement</td>
<td>100%</td>
<td>70%</td>
<td>0.15</td>
</tr>
<tr>
<td>Instantaneous Global Coverage</td>
<td>Percentage of globe over which measurements are taken in a time resolution period</td>
<td>100%</td>
<td>5%</td>
<td>0.05</td>
</tr>
<tr>
<td>Mission Completeness</td>
<td>Mission type conducted EDP, AOA, and Turb</td>
<td>EDP only</td>
<td></td>
<td>0.95</td>
</tr>
</tbody>
</table>

Figure 12: Example Design Attributes and Definition [19]

shows the weighted relationships between properties and attributes. Based on these values, the MATLAB scripts are developed to represent the weighted equations.

4.3 Knowledge Through Interviews

Once the design attributes are selected, MIST can be used to conduct stakeholder interviews. Using the Multi-Attribute Utility Analysis (MAUA) interview process, which was developed by the SSPARC team, MIST employs an advanced graphical user interface to speed up and enrich the utility interview process. These interviews can be conducted multiple times over the engineering design process in order to better understand the evolving relations between stakeholder preferences and design decisions.

The interviews provide stakeholders with a scenario using the lottery equivalent probability (LEP) approach as developed in the field of decision theory. The interviews generated by MIST present two situations in which the interviewee is prompted to select
Figure 13: Example Design Properties and Design Attributes Matrix [19]

the more appealing option of the two. After such selection, the value of the variable option changes and the interviewee selects again. This process continues until the interviewee is indifferent between the two scenarios. Once the indifference point has been identified, the interviews focus on other attributes, while the analyses of the identified indifference points are performed in the background mode.
4.3.1 Bracketing Methodology

Every interview conducted in MIST employs the two situation framework to help identify the indifference value between the two scenarios. One of the situations is typically a fixed value and the other situation is a variable value. The interviews are designed so that the variable probability is in the range of between 0% and 50%. The idea behind the algorithm is similar to pruning, where the range of possibilities for the indifference point continually narrows until the range is narrower than the probability resolution set for the attribute [20].

There are two ways for a stakeholder to reach an indifference point. First, the stakeholder could manually select the current value for the variable situation as the point of indifference, or secondly, the stakeholder could continue to select a situation until the range of possibilities is narrower than the probability resolution; at the latter point, the desired value is the median between the two situations.

Figure 14 shows a full bracketing tree with every branch traversed. In the figure, all branches of the tree are gradually selected until an indifference value is reached. The example is based using an attribute resolution of 5%. After a resolution of 5% is attained, the indifference value is automatically determined. The other way to get an indifference value, besides falling below the resolution, is if the stakeholder manually selects the variable as the indifference point. In the figure, the A branch always represents the variable option and the B branch always represents the fixed option.
4.3.2 Single Interview

The interview begins by selecting a random value from the attribute range that is in question, and then lottery equivalent probability method, two situations are presented to the stakeholder. One of the situations represents a 50% chance of getting the random value and a 50% chance of getting the worst possible value for the attribute. The other situation shows a certain probability \( p\% \) of getting the best possible value of the attribute and probability \( 100-p\% \) of getting the worst possible value [19]. Now, the interviewee is prompted to choose between these situations. The value of \( p \) is varied, until the stakeholder
is indifferent between the points.

During the interview, the scenario and definition of the attribute are displayed. The scenario allows for the interviewee to place the question in a context in order to specifically highlight the aspect to be considered. The definition allows for the interviewee to understand and recognize the correct interpretation of the attribute.

Once the indifference point has been determined, another random number is selected, and the process is repeated with the new random value. This process continues until an adequate number of indifference points have been selected for MIST to be able to accurately represent that stakeholder's utility curve for the particular attribute in question.

Figure 15 shows an example of a single interview utility function. The attribute values are on the X-axis, and the utility values are on the Y-axis. The graph shows how utility varies with the value of the attribute being explored. This utility curve shows an inverse relationship where an increasing accuracy value decreases the utility for this particular stakeholder.

![Utility of Accuracy (AOA)](image)

Figure 15: Example Utility Function for Accuracy Attribute from a Single Interview [19]
Stakeholder utility curves for single attributes are important because they show how changes to an attribute effect the utility of the stakeholder. With this knowledge, stakeholders have a better understanding of how to maximize their utility when making design decisions for design properties and design attributes.

4.3.3 Corner Point Interview

The goal of the corner point interview is to determine the relative importance of each design attribute relative to all other attributes of the system. Again, this is done in the interview by presenting the interviewee with two situations from which to choose from. The situations involve the entire set of design attributes with varying values. Instead of using the lottery equivalent method in this interview, the certainty equivalent method is applied here. With this method, the interviewee has a choice between two situations where in the first situation there is a certain particular outcome and in the other situation there are two probable outcomes.

The certain outcome consists of the set of design attributes where the attribute in question is at the best possible value and all other attributes are at the worst possible value. The other situation shows a certain probability \( p\% \) of getting the best possible values for all of the attributes and probability \( 100-p\% \) of getting the worst possible values for all of the attributes.

The indifferent point is again determined between the two outcomes by varying the probabilities of \( p \). At this point, the probability value is the corner point value, ‘k-value’, and it represents the relative weight of the specific attribute being interviewed in respect to all of the other attributes. A higher ‘k-value’ for a particular attribute means that the attribute is more important to the stakeholder than the other attributes that have lower
For the satellite example, single and corner point interviews were run on the attributes by the individual stakeholders of the system. In Figure 16, the indifference values for a single stakeholder are shown for each attribute. These are acquired from single interviews. Next to that, the results from corner point interviews are displayed showing indifference point values when all other attributes are either at their best or worst values.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Initial Interview</th>
<th>Validation Interview</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Indifference Point</td>
<td>New</td>
</tr>
<tr>
<td>Spatial Resolution</td>
<td>32.5%</td>
<td>32.5%</td>
</tr>
<tr>
<td>Revisit Time</td>
<td>42.5%</td>
<td>37.5%</td>
</tr>
<tr>
<td>Latency</td>
<td>37.5%</td>
<td>17.5%</td>
</tr>
<tr>
<td>Accuracy (AOA)</td>
<td>42.5%</td>
<td>12.5%</td>
</tr>
<tr>
<td>Accuracy (EDP)</td>
<td>42.5%</td>
<td>42.5%</td>
</tr>
<tr>
<td>Inst. Global Coverage</td>
<td>48.0%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Mission Completeness</td>
<td>47.5%</td>
<td>48.0%</td>
</tr>
</tbody>
</table>

Figure 16: Example Single and Corner Point Interview Results [19]

The corner point interview is important in the overall analyses of stakeholder preferences. This is because corner point values specify the independence of attributes to a particular stakeholder. If a stakeholder has an attribute that is independent of all other attributes, they would work to ensure that this attribute is maximized regardless of the effects on the other attributes.

4.4 Knowledge Discovery in MIST

Knowledge discovery was shown in the knowledge gained from single and corner point interviews.

In a single interview, the stakeholder would gain a better understanding of their utility function for a given attribute, and they can apply this discovered knowledge to
maximize their utility in respects to the value assigned to the attribute. In a corner point interview, the stakeholder would realize the importance of each attribute relative to all other attributes. This discovered knowledge allows stakeholders to design properties that focus on the particular desired attributes that allow the stakeholder to realize their preferences.

An additional tool for knowledge discovery in MIST is the relationship analysis tool. This sub-tool facilitates knowledge discovery in two ways: i) by deriving utility and expense values for a single stakeholder from a given set of design attributes and ii) by displaying changing utility and expense values for all stakeholders from the utility and expense value of a single stakeholder.

Figure 17 shows the graphical user interface for the stakeholder relationship analysis tool. The stakeholder analysis tool is able to accommodate the analysis of four stakeholders concurrently. Each stakeholder has his/her own possible utility vs. expense values plotted. At each design level, the utility vs. expense values are plotted and displayed on the graphs, and the current design properties and design attributes are displayed on the left of the sub-tool.

The knowledge discovery of particular utility and expense values allows for a stakeholder to better understand the effects a particular set of design attributes has on his/her goals. With these values, a stakeholder can easily determine whether the particular set of design attributes positively or negatively impact their preferences.

In addition to determining utility and expense values, the sub-tool can take derived or hypothetical utility and expense values for a stakeholder and reverse engineer the set of design attributes that would result in the particular utility and expense values. From
Figure 17: Main Form of Relationship Analysis Tool

here, the sub-tool can also calculate all corresponding utility and expense values for all other stakeholders of the system based on the calculated set of design attributes. With this functionality, stakeholders can analyze the effects of design attribute changes to utility and expense values for all stakeholders in the design process.

In the ongoing satellite example, knowledge discovery can be depicted as the identification of numerous attribute set values that result in varying final utility values. In the end, it was discovered that the 5x5, 30 min, 15 min, .005 deg, 55%, EDP/AOA/Turb attribute set resulted in the highest calculated utility. In order to verify the results, the
determined attribute set above could be entered into the relationship analysis tool, and
the calculated utility value of the stakeholder could be compared. Figure 18 shows various
attribute sets and the resulting calculated utility values.

<table>
<thead>
<tr>
<th>Attribute Mix (spatial resolution, revisit time, latency, accuracy, instantaneous global coverage, mission completeness)</th>
<th>Customer Estimated utility</th>
<th>Calculated Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>25x25, 5 min, 60 min, 80%, 45%, EDP</td>
<td>0.169384</td>
<td>0.64647</td>
</tr>
<tr>
<td>50x50, 2 hrs, 5 min, 90%, 30%, EDP</td>
<td>0.44463</td>
<td>0.75227</td>
</tr>
<tr>
<td>5x5, 30 min, 15 min, 0.005 deg, 55%, EDP/AOA/Turb</td>
<td>0.99999</td>
<td>0.99989</td>
</tr>
<tr>
<td>30x30, 4 hrs, 1hr, 0.25 deg, 30%, EDP/AOA</td>
<td>0.91469</td>
<td>0.95719</td>
</tr>
<tr>
<td>10x10, 6 hrs, 20 min, 75%, 95%, EDP</td>
<td>0.27525</td>
<td>0.58432</td>
</tr>
<tr>
<td>20x20, 40 min, 30 min, 0.5 deg, 60%, EDP/AOA/Turb</td>
<td>0.92931</td>
<td>0.98177</td>
</tr>
</tbody>
</table>

Figure 18: Example Discovery of Utility Values for Multiple Attribute Sets [19]

Such analysis cannot be performed using most collaborative tools that are available
today. It is the specific acquisition of stakeholder preferences and the relations of the
varying knowledge representations of the system that allow for the SSPARCy and MIST
tool to discover such relations regarding the design process.

4.4.1 Calculating with Multi-Attribute Utility Analysis

To calculate the multi-attribute utility value for each stakeholder, a time-weighted average
of the attributes is used. Equation 1 below shows the mathematical equation for this
relationship [19].

\[ KU(X) + 1 = \prod_{i=1}^{N} (K k_i U(X_i) + 1) \]  \hspace{1cm} (1)

In the equation, ‘\(K\)’ is the normalization constant and ‘\(U(X)\)’ is the multi-attribute
utility function value - this is the value that is being calculated. ‘\(k_i\)’ is the corner point value,
or relative weight of a specific attribute, and is determined by the corner point interview.
The relative values of \( k_i \) gives a good indication of the relative importance between the attributes. \( U(X_i) \) is the value for a single attribute utility and can be determined from the single attribute utility graphs. In order to solve for the multi-attribute utility value, \( U(X) \) must be isolated to one side of the equation. Therefore, one must be subtracted from both sides of Equation 1 and both sides must be then divided by \( K \), the normalization constant.

The scalar ‘\( K \)’ is a normalization constant that ensures that the multi-attribute utility function has a zero to one scale. In order to derive ‘\( K \)’, Equation 2 must be calculated and held within the bounds of \(-1 < K < 1 \) and \( K \neq 0 \).

\[
K + 1 = \prod_{i=1}^{N}(Kk_i + 1) \tag{2}
\]

To calculate ‘\( K \)’, the ‘\( k_i \)’ values are known to be the corner point values, and the right side of Equation 2 can be multiplied out to a polynomial. Now, if ‘\( K+1 \)’ is subtracted from both sides, the resulting equation would be equal to zero and be easier to solve. This is shown below in Equation 3.

\[
0 = (Kk_1 + 1)(Kk_2 + 1) \cdot \ldots \cdot (Kk_{i-1} + 1)(Kk_i + 1) - (K + 1) \tag{3}
\]

To solve Equation 3, the relationship analysis tool fills in all ‘\( k_i \)’ values with the values of all the respective corner points, which are stored in an array in MIST. Now, to find ‘\( K \)’, the tool solves the polynomial equation through a guess and check methodology. To do so, a specific value for ‘\( K \)’ is entered in order to see how close the solution is to 0. The values used for ‘\( K \)’ range from -1 to 1 and increase with 0.1 increments. Every time that a ‘\( K \)’ value is tried, the tool stores the value for ‘\( K \)’ if the solution to the polynomial
is one of the two solutions closest to zero for all ‘K’ value tried. In order to determine the solutions closest to zero, the absolute values of the solutions are determined, and the smallest absolute value solution is the closest value to zero.

The reason that the two smallest ‘K’ values are saved is that another decimal value for ‘K’ will be determined from these two values because accuracy to the hundredth is desired. Once the two values are found, the smaller ‘K’ is incremented up by 0.01, storing only the new calculated ‘K’ value with two decimals that is closest to zero. This is approximately the value of ‘K’ to the hundredth decimal place. Now that the values for ‘K’, ‘k_i’, and ‘U((X_i))’ are known, the multi-attribute utility value, ‘U(X)’, can be calculated because all other variables are now known. This algorithm is used to calculate all multi-attribute utility values. The same algorithm is used to calculate expense values; here the single attribute expense values are used instead of the utility values.

The SSPARCy and MIST tool differs from other collaborative engineering tools in its ability to calculate these overall utility and expense values by employing the MAUA equations.

4.4.2 Alternatives to Multi-Attribute Utility Analysis

There are many alternative methods available that assist in the analysis of multiple attributes in a design system. One such alternative that was considered was the GINA process. This method is based on information in network optimization theory and allows for stakeholders to compare different designs based on performance and cost [19]. The concept of the process is to convert a design system into an information flow diagram upon which quantifiable performance parameters will be applied. After the architecture is captured and the performance parameters defined, simulation code is designed to test
the performance of the system and to assist in defining an optimal architecture. In this process, the emphasis is more on the transformation of the system into an information network where specific metrics are defined as performance and design vectors.

MAUA is similar to GINA in that both processes evaluate architectures based on the study of the system over a huge trade space as oppose to around a point design. MAUA offers more flexibility and allows for the mapping of customer-perceived metrics to customer utility values. This ability to designate design attributes allows for the MAUA system to adequately explore the trade space and represent the knowledge of the design system in an easily quantifiable manner.

4.5 Additional Satellite Design Example

Another actual satellite collaborative design was undertaken in the spring of 2002 as part of the SSPARC consortium’s X-TOS project [14]. The project was conducted together with members of The Aerospace Corporation and Air Force Research Laboratory. The TOS satellite projects have progressed through many iterations throughout the years, and the spring of 2002 was the first time that the stakeholder interview processes were automated instead of being carried out in a manual basis. The role of the scientific customer in this exercise was played by Kevin Ray, a member of the Air Force Research Laboratory at Hanscom Air Force Base.

After running the SSPARCy and MIST tools, it was found that the MIST software cut down on the time it took to perform the stakeholder interviews by over 50% from previous years. Kevin Ray commended the SSPARCy and MIST tools as “being a great benefit to customers because [the tool] shows them how prioritizing one attribute over another
attribute drives the design of the system” [14]. He also commended the system’s friendly graphical user interface and intuitive nature of the design.

Through the automation of the interview process for MIST, it is now easier for stakeholders to continually update their interviews throughout the design process. This allows the stakeholders of a design to be continually involved in the engineering process. By doing so, stakeholders remain active in the design process and assist in influencing more aspects of the collaborative environment. Through continual adjustment and fine tuning of stakeholder preferences through multiple interviews, a more accurate representation of the design can be developed.
5 Value of SSPARCy/MIST Approach

The value added of the SSPARCy and MIST approach lies in the ability for the tools to capture and represent knowledge of the engineering design process while also exhibiting knowledge discovery functionalities and in-depth analysis of acquired design knowledge. As shown in the previous section, the SSPARCy and MIST tools effectively capture multiple representations of design knowledge such as identifying system specifications, specifying design properties, and determining design attributes.

In addition to knowledge representation, SSPARCy and MIST are able to analyze collaborative environments by employing relational analysis of varying knowledge representations. This is a very important aspect of the system. The tool are able to elicit knowledge from stakeholders regarding their preferences and knowledge representations and incorporates basic cost-benefit and utility analysis to understand relationships of these various representations. By doing so, the tools is able to better understand the effects of design changes on the entire collaborative environment.

5.1 Knowledge Management: Retention and Transfer

Two of the main issues plaguing constituents in a collaborative endeavor are the issues of knowledge retention and knowledge transfer. Knowledge retention is the ability for collaborative tools to be able to acquire and archive knowledge of a system. Knowledge transfer is the ability for collaborative tools to disseminate knowledge and discover new ideas resulting from analysis of archived knowledge.

SSPARCy and MIST address both roles of knowledge retention and knowledge transfer for collaborative engineering tools. Both systems are able to acquire information regarding
knowledge of past design decisions and archived states of a system design. SSPARCy retains knowledge through its variable history review functionality, and MIST retains knowledge through the interview and rationale archiving functionalities. The created knowledge repository allows stakeholders to better understand the evolution of their particular design.

Both systems also incorporate extensive knowledge transfer capabilities. SSPARCy handles knowledge transfer through its many display and variable history representations. In MIST, knowledge transfer occurs in the graphical representations of all utility and expense functions and the graphical outputs of the relationship analysis sub-tool. In the end, what makes SSPARCy and MIST successful is the ability to disseminate the knowledge of the evolution of the design system that is uniformly represented.

5.2 Additional Domains of Application

Originally, SSPARCy and MIST were both designed to assist in the collaborative design and development of space satellites in conjunction with the SSPARC research group here at MIT. The applications of these two tools to the domain of academic research is detailed above in Section 4 of knowledge flow. Since this initial prototype testing, MIST and SSPARCy have been adapted and are currently capable of being applied to numerous other domains despite aeronautical engineering. Outlined in this section are possible applications of SSPARCy and MIST to two separate domains: commercial engineering (i.e. automobile design) and work place (i.e. managerial/personnel tool).
5.2.1 Commercial Engineering Application

First, as globalization continues to become the growing trend in commerce, commercial engineering endeavors are increasingly becoming more distributed. An example of the growing geographic displacement of design teams can be observed in the automobile industry where numerous stakeholders and engineers at various locations throughout the world collaborate together in designing an automobile.

Automobile design would benefit from the application of SSPARCy and MIST in many ways. First, SSPARCy would allow for the proper representation of numerous system specifications into design properties such as ‘length of chassis’, ‘weight of automobile’, etc. SSPARCy would be able to capture design states of an automobile design in various stages of the design process. This allows for the retention of knowledge regarding changes to specific specifications of the design. In order for the design properties to change, they would have to be effected by actual changes to design specifications by engineers. For example, if engineers decided they needed a larger car with more interior space they would increase the length specifications of the car and resultanty would change the value of the ‘length of chassis’ property.

Implementing MIST, design attributes for this application would be ‘handling’, ‘fuel efficiency’, etc. These attributes are a weighted combination of all design properties. For a design attribute to change, take ‘handling’, many separate design properties such as ‘weight of car’, ‘position of wheel’, etc. would have to change in order to adjust ‘handling’ because those are the properties whose weighted values comprise the specific attribute.

System stakeholders would benefit from the archiving aspects of SSPARCy and MIST. Analysis of past design decision could result in the recognition of trends in design and
the avoidance of past engineering errors. Stakeholders could tailor their designs to develop systems that produce maximum utility through the analysis of utility curves and stakeholder preferences.

In order to understand the benefits an automobile maker would gain from the implementation of SSPARCy and MIST, let’s consider the hypothetical example that the Ford Motors Company decided to use the SSPARCy and MIST tools in the design of their next generation of Ford Taurus sedans. SSPARCy would be used to first define the design of the new Taurus in terms of design properties. Also, SSPARCy would incorporate its archiving functionalities in order to keep track of all design changes and the rationale behind them as the design process continues. This information can later be used to understand engineering decisions and to avoid past errors.

In the Ford example, MIST would be used to define the engineering process in terms of design attributes. After the design attributes have been successfully related to design properties through weighted MATLAB equations, stakeholder preferences can be elicited through interviews conducted by MIST. Once all of these representations are developed, the relationship analysis tool can be used to better understand the effects of design changes to the preferences of all stakeholders involved.

From this quick, general simulation of SSPARCy and MIST, it can be observed that there are many ways for the SSPARCy and MIST tools to add value in the automobile engineering design process.

5.2.2 Professional Work Place Application

According to the U.S. Department of Labor Employment, the average employee separation rate in the U.S. is approximately 38.09%, meaning 38.09% of all individuals have voluntarily
or involuntarily experienced a work termination from their employers during 2002 [21]. Figure 19 is based off of data provided by the U.S. Department of Labor and further details the employee total separation rates by various industries.

![U.S. Total Separation Rates for Various Industries in 2002](image)

**Figure 19: Total Separation Rates for Various Industries in 2002**

Such a high rate of turnover causes problems in knowledge transfer and knowledge retention. Knowledge transfer becomes a problem because of the loss of time and resources spent in retraining new employees. Knowledge retention is a problem because when employees leave, the knowledge that they had gained on the job typically leaves with them as well.

With the typical professional work environment becoming increasingly reliant on collaborative work arrangements and with increasingly high employee turnover rates, the importance for employers to use collaborative engineering tools to assist knowledge
retention and knowledge transfer becomes more apparent.

As employees change jobs, and key individuals in collaborative projects depart, it is crucial that the knowledge of the departing individual is retained for future reference. SSPARCy and MIST both exhibit functionalities that allow for this type of design and rationale archiving. For example, in order to retain knowledge from an engineer about a particular engineering project, a MIST interview on the attributes of the project would encapsulate a large majority of the design rationale, and the relationships of the design attributes to the stakeholder’s preferences would also provide insight into the design.

Once the knowledge of a system has been retained and acquired, employers will want to pass on this information to new employees. This is where the functionalities of knowledge transfer in SSPARCy and MIST become useful. Through the use of graphical interfaces and intelligent knowledge representation, SSPARCy and MIST are able to transfer and distill past system knowledge to constituents that require it. Due to these functionalities, SSPARCy and MIST would be successful tools in assisting the professional work place through the facilitation of knowledge management to combat turnover and increasing collaborative environments.

5.3 Collaborative Tools in Use in Aerospace Industry

Aerospace engineering typically involves a complicated system design with a large number of stakeholders due to the highly complex and large systems that are developed. These properties of aerospace design make the industry an ideal candidate for collaborative engineering tools.

The aerospace industry does not have a standard collaboration tool which all aerospace
corporations utilize. Instead, corporations employ various collaboration tools which may even vary within a single company [27].

5.3.1 Collaboration at Avidyne Corporation

Across the aerospace industry, most corporations employ varying types of collaboration tools ranging in diversity of functionalities. Avidyne Corporation, which is based out of Lincoln, Massachusetts, is an example of an aerospace corporation that employs collaboration tools to assist in the collaborative design process. Avidyne is a leading provider of integrated flight deck systems for the next generation of light aircraft. The corporation employs the collaboration tool DOORS to help assist in the CAD development and knowledge management process through the facilitation of communication between stakeholders [22]. DOORS, which is developed by Telelogic, is a requirements management tool that enhances communication, collaboration, and validation across a particular enterprise through the creation of a repository of system knowledge and basic forums for discussion and validation. In the example interface of DOORS shown in Figure 20 [23], the user is able to dynamically update properties of a system attribute and state their rationale for the changes.

Avidyne employs the use of DOORS in order to assist in the communication and dissemination process of collaborative design. By using the tool, all stakeholders of a design system are able to centrally access the current design specifications and goals. The DOORS tool does not add value as a knowledge discovery tool, but the ease of use and facilitation of communication does ease the burden of collaborative design.
5.3.2 Collaboration at The Aerospace Corporation

The Aerospace Corporation has developed an in-house collaboration tool known as the Concept Design Center. The Concept Design Center is a networking tool that was designed to assist in the preliminary design process. The tool facilitates collaboration in a common location by functioning as a central knowledge repository that archives all system specifications designed and edited by the design team. Nicola Nelson, a Principal Engineer at The Aerospace Corporation, emphasized that the most valuable insight to successful collaborative engineering is a clear understanding of all stakeholder preferences in a system [24]. She expressed interest in using a collaborative tool that could clearly elicit and analyze stakeholder preferences in a design system.

SSPARCy and MIST provide functionalities that address the claims for a clear
representation of stakeholder preferences. In addition to this, the knowledge discovery of the combined SSPARCy and MIST tools creates a dynamic understanding of the causal effects that design changes have on stakeholder utility and expense values. This functionality of relational analysis is not commonly found in current commercial collaborative tools.

5.3.3 Collaboration at Boeing

The Boeing Company is the world’s largest aerospace company. The company itself incorporates numerous collaborative engineering tools. The Boeing Satellite Systems group uses the ICEMaker tool described earlier in Section 2.4.4. Boeing has been using ICEMaker since the winter of 2002 to help assist in collaboration of design systems. In addition to ICEMaker, the Boeing Satellite Systems group is looking to acquire an additional collaboration tool. The two commercial tools being considered are Oculus and Phoenix Integration [27]. Both tools integrate and automate various software tools, allow for geographic distribution of stakeholders, and accommodate different computing platforms into a cohesive environment for systems design.

Regarding the topic of knowledge retention of past design knowledge, Stephen Sichi, a Chief Engineer at Boeing, commented that the design cycles of a few aerospace systems are very short, in the ballpark of 5 years. At times, he mentioned, it may be cheaper to maintain an iteration of a current engineering design for 5 years, and after the 5 years has elapsed, to start from scratch on the new generation of the design. He attributed such a short design cycle to advances in either development tools or aerospace technologies that result in enough design change over 5 years that the re-engineering of the system is a more practical solution [26]. However, during the 5 year period that the design iterations are
maintained for, a system that is able to support knowledge retention, knowledge capture, and knowledge discovery would be useful in pushing forward design initiatives. The issue to keep in mind though is that the design cycles for most systems typically are longer than 5 years. This means that typically, knowledge retention for design iterations is even more important and must be adequately handled.

Sichi also mentioned that knowledge retention of employee expertise is slowly becoming more of a problem as the engineering generation responsible for many advancements such as rocket propulsion are slowly reaching the retirement age. In the case of rocket propulsion, in the 1960’s, that generation of engineers were responsible for the majority of the research conducted on rocket design, and in the past few decades, much funding for rocket research has been cut resulting in fewer new engineers becoming experts in the field. This leaves the aerospace industry with the question: What happens to the expert knowledge of rocket propulsion when this generation of engineers retires?

SSPARCy and MIST both address the issue of knowledge retention. Both tools incorporate archiving functionalities that capture design state knowledge for future review. MIST can also archive the knowledge from expert engineers through the stakeholder interview process, and SSPARCy can do the same through its variable history functionality.

Chad McFarland, a Engineering Design Manager at Boeing, recognized that one of the most important added benefits of the SSPARCy and MIST tool is the ability for stakeholders to stay involved in the engineering design process throughout the entire endeavor. Chad recognized the importance of stakeholder involvement throughout the engineering design process, and he understood that with the use of a tool such as MIST, stakeholders could continuously be involved in the design process. This valuable
functionality would be most useful in allowing the customer stakeholder to maintain a consistent voice throughout the engineering design project to develop a more complete design that fits all of the customer's preferences.

5.4 SSPARCy/MIST Comparison to Other Collaborative Tools

A comparison of the SSPARCy and MIST tools to the other collaborative tools mentioned in Section 2.4 will clearly show the different formalized means taken in SSPARCy and MIST. The relevant contributions of each tool discussed and possible improvements are shown in Figure 21.
<table>
<thead>
<tr>
<th>Collaborative Tools</th>
<th>Developed By</th>
<th>Relevant Contribution</th>
<th>Possible Improvements</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSPARCy [16]</td>
<td>Researchers at MIT / Sloan</td>
<td>Defines relationship between system specifications and design properties; captures incremental design states</td>
<td>Stakeholder analysis of preferences; use of Internet</td>
</tr>
<tr>
<td>MIST [18]</td>
<td>Researchers at MIT / Sloan</td>
<td>Defines relationship between design properties and design attributes; captures stakeholder preferences; analyzes all relationships from SSPARCy and MIST</td>
<td>Use of Internet; additional knowledge transfer functionalities utilizing more graphical presentations</td>
</tr>
<tr>
<td>CIS [8]</td>
<td>Researchers at University of Karlsruhe in Germany</td>
<td>Knowledge retention functionality; uniformly represents stakeholder knowledge</td>
<td>Addition of knowledge discovery and additional knowledge transfer functionalities</td>
</tr>
<tr>
<td>TeamSCOPE [9]</td>
<td>Researchers at Michigan State University</td>
<td>Focuses on common representation of design project; incorporation of numerous sub-tools; interview process to acquire information</td>
<td>Additional knowledge discovery needed; ability to analyze and store data of design trends</td>
</tr>
<tr>
<td>MADE [6]</td>
<td>Researchers at DARPA</td>
<td>Utilizes Internet as knowledge repository and knowledge dissemination; allows for quick prototyping of design alternatives</td>
<td>Additional knowledge discovery needed; more flexible tool needed for universal application</td>
</tr>
<tr>
<td>ICEMaker [11]</td>
<td>Researchers at California Institute of Technology</td>
<td>Server/client architecture facilitates collaboration and capture of mission and system requirements</td>
<td>Better facilitation of geographically distributed collaboration; more emphasis on knowledge discovery</td>
</tr>
<tr>
<td>Rule-Based Algorithms [12]</td>
<td>Researchers at Chung-Hua University of Taiwan</td>
<td>Automated system that analyzes customer goals and desires; transfers customer needs to specifications with rules</td>
<td>Knowledge retention functionalities needed to maintain knowledge of design states</td>
</tr>
<tr>
<td>CDPM Platform [13]</td>
<td>Researchers at MIT</td>
<td>Provides project management platform; offers project knowledge and data analysis tools; works as a collaboration infrastructure</td>
<td>More focus on stakeholder preferences and effects of stakeholder decisions on design</td>
</tr>
<tr>
<td>DICE [14]</td>
<td>Researchers at MIT</td>
<td>Distributed collaborative agent-based framework; component based design utilizing control mechanisms and global database</td>
<td>More emphasis on relations of various design aspects; more discovery of knowledge through incorporation of stakeholder preferences</td>
</tr>
</tbody>
</table>

Figure 21: Collaborative Tool Contributions and Possible Improvements
Following the categorization outlined in Section 2.4, SSPARCy and MIST would fall under the tools that support both knowledge retention and knowledge discovery.

The first type of collaboration tools focus mainly on knowledge retention and knowledge transfer. These tools include CIS, TeamSCOPE, MADE, and ICEMaker. Both MIST and SSPARCy not only exhibit archiving capabilities, but they both are capable of conducting analysis on the information archived. MIST utilizes complex utility and cost-benefit analysis techniques in order to understand stakeholder relationships, and SSPARCy analyzes design trends through the application of MATLAB scripts. With these added functionalities, SSPARCy and MIST are more powerful in dealing with collaborative engineering issues by being able to support both knowledge retention and knowledge discovery. Like MIST, TeamSCOPE incorporates an interview process to acquire stakeholder knowledge, but MIST's interview process is more complicated utilizing a lottery preference system as well as basic inference and cost-analysis equations.

ICEMaker differs from SSPARCy and MIST in many ways. One main difference is that ICEMaker doesn’t display the functionalities of knowledge discovery that SSPARCy and MIST so aptly employ. To specifically illustrate this point, during design, ICEMaker uses the server/client architecture where the onus is on the client to post or request system specifications for a design. There are no functionalities built into ICEMaker that allows for the system to realize that a particular specification may work better or that certain clients should post or subscribe certain pieces of knowledge, instead all of these design decisions rest solely on the clients. In SSPARCy and MIST, these systems discover new knowledge through the relational analysis of multiple knowledge representations. When the MIST tool relates design attributes to particular stakeholder preferences, it can be shown that a certain
attribute set would theoretically produce maximum utility for that particular stakeholder. Another difference of ICEMaker from the SSPARCy/MIST construct is that ICEMaker uses a multiple design approach while SSPARCy and MIST use single design approaches when designing systems. Properties of both approaches were discussed in Section 2.4.4, and it was concluded that dependent on the type of design and other outstanding circumstances, either design approach could prove to be more successful.

The second type of collaboration tools focus mainly on knowledge analysis and knowledge discovery. Rule-based algorithms developed at the Chung-Hua University of Taiwan is an example of such a tool. The opposite case exists here where MIST and SSPARCy exhibit knowledge analysis and knowledge discovery functionalities just as this group of collaborative tools, but these tools do not support the knowledge retention and knowledge transfer functionalities that SSPARCy and MIST exhibit. The algorithms developed at the Chung-Hua University focus on the use of rule-based algorithms to only transfer customer needs to system specifications; MIST, however, is not tied solely to such specific design but can also allow for a higher level tradespace exploration of the system.

The final type of collaboration tools to be examined is the type which SSPARCy and MIST belong to themselves, the tools that focus on knowledge retention and knowledge analysis. Other tools of this type include the CDPM platform and DICE. The main difference between CDPM and SSPARCy/MIST is that CDPM focuses less on the preferences of individual stakeholders and more on the knowledge representation of the system. MIST and SSPARCy provide vital information regarding stakeholder preferences and relationship analyses between stakeholders. This added functionality provides valuable information for the stakeholders of the collaborative design.
Like SSPARCy and MIST, DICE enables the acquisition of the knowledge representation of the system and supports knowledge analysis and discovery through the application of numerous Knowledge Modules. However, the difference in the SSPARCy/MIST tool and DICE is that SSPARCy/MIST places more emphasis on the analysis of stakeholder relationships and the discovery of knowledge. DICE on the other hand tends to have a stronger focus on the management of system knowledge instead of the analysis of future knowledge opportunities. DICE facilitates component-based design through the constant interaction and negotiation between its Knowledge Modules. If DICE incorporated the same attribute utility curves or stakeholder preferences that SSPARCy and MIST use, the DICE system would not have to continually ask for inputs in the analysis process.

The combined SSPARCy and MIST tool employs new methodologies to elicit knowledge from the collaborative environment in approaches and techniques that are currently not being utilized by other collaborative tools. These relational comparisons between design properties, design attributes, and stakeholder preferences are what allow the combined SSPARCy and MIST tool to develop a better understanding of effects of design decisions on the collaborative environment.

5.5 Future Work

There are currently many new extensions and improvements to the SSPARCy and MIST tools under consideration.

One such improvement is the added functionality of a real-time interface to facilitate communication between stakeholders. Currently, the tools rely on asynchronous communication where the tools independently work to interview and interact with
each stakeholder individually. A real-time interface would allow for stakeholders to interact and utilize knowledge analyses and discovery functionalities to develop a better understanding of each particular stakeholder’s needs.

In addition to a real-time interface for the tools, another improvement to the accessibility of the two tools is a web enabled implementation of SSPARCy and MIST. An online application of the tools would increase the ease of accessibility for stakeholders. In addition, having only one state of the tool online would serve as a version control system where only one copy of the state of SSPARCy and MIST would be maintained. This way, all attribute values and other design variables would be the same for every stakeholder using the tools.

Also, to better accommodate multiple users of an online tool, a logical addition to the web-enabled version of MIST would be a stakeholder profile functionality that would allow for the packaging of all interviews and analyses for a particular stakeholder in order to facilitate the display of all relevant knowledge of a particular stakeholder on one comprehensive interface. Such a functionality would greatly enhance the knowledge dissemination capabilities of MIST in facilitating communication of stakeholder knowledge.
6 Conclusion

As collaborative endeavors become more popular from the engineering domain to the commercial and academic domains, the need for tools to assist in communication and knowledge management will continue to be a major focal point in collaborative environment research. The problem at hand is that as collaboration becomes much more important in all facets of life, supporting tools will become even more essential to help facilitate such environments. SSPARCy and MIST have been designed to work complementarily to act as a combined tool that facilitates collaborative engineering across the entire spectrum of design representations. The tool employs new methodologies to elicit knowledge from the collaborative environment in approaches and techniques that have not been currently utilized by other engineering tools. Through the chaining of the pathways of knowledge flow of the collaborative systems from system specifications to design properties to design attributes to stakeholder preferences, this combined tool develops an innovative approach to better understanding design system relationships. By linking all of these different facets of design representation, the combined SSPARCy and MIST tool is able to conduct analyses on the entire knowledge pathway by relating various different representations of the system in order to discover how changes in one aspect of design can effect the entire collaborative environment.

The coordinated use of SSPARCy and MIST provides an alternative to collaborative engineering and provides a new methodology using elicitation of system knowledge to better understand relationships between multiple design representations.
7 Acknowledgements

I would like to thank all of the members of the MIT Space Systems, Policy and Architecture Research Consortium Information Technology team who have helped me to develop, fully implement, and test my work. In particular, I wish to acknowledge Professor Amar Gupta and Satwiksai Seshasai for their constructive feedback and perceptive guidance throughout the development and writing processes.

Finally, I would like to thank all of the people who have supported me throughout my years of study here at MIT, I really appreciate it.
References


