MIT Sloan School of Management

Working Paper 4381-02
September 2002

A KNOWLEDGE BASED APPROACH TO FACILITATE ENGINERING DESIGN

Satwik Seshasai and Amar Gupta

A Knowledge Based Approach to Facilitate Engineering Design

Satwiksai Seshasai and Amar Gupta

May 5, 2002

Abstract

This paper presents a knowledge-based approach to facilitate the engineering design process relating to spacecrafts. Because the design evolves over a long time and typically involves individuals working at different locations and frequently for different organizations, the degree of collaboration across temporal and spatial boundaries plays a major role in determining the aggregate time and cost involved in each instance of spacecraft design. A major aspect of such collaboration is the issue of communications – the ability to clearly and efficiently explicate and record the detailed needs of every stakeholder in the process, as well as the major design decisions and the rationale behind these decisions. The approach described in this paper provides a framework for facilitating the decision-making process in engineering design, by eliciting and capturing the goals and requirements of every stakeholder in the design process through utility and expense functions. An interactive system has been designed that incorporates a four faceted knowledge-based framework of knowledge acquisition, knowledge discovery, knowledge management and knowledge dissemination to provide designers and stakeholders with the capability to develop an evolving knowledge repository about all aspects of the design process. This interactive system includes the ability to capture succeeding versions of the detailed design, with zero or minimal human involvement; the capability is provided by a set of algorithms collectively named as SSPARCy. A complimentary tool, called MIST, facilitates the Multi-Attribute Tradespace Exploration process by enabling stakeholders to express their goals and preferences in a formalized manner. The combination of MIST and SSPARCy paradigms enables one to transform crucial applications that are today contingent on geographical proximity to occur with equal or superior effectiveness in a virtual world. While this paper analyzes a situation involving engineering design of spacecrafts, the proposed knowledge-based approach can be readily adopted to facilitate other applications that involve sustained collaboration across geographic and corporate boundaries.

1 Introduction

The spacecraft design environment involves sustained effort by multiple teams and multiple stakeholders based in various locations. In such a distributed environment, it becomes especially important to be able to clearly and efficiently capture the needs of every stakeholder in the process, as well as details of the major decisions and the rationale behind these decisions. By transferring relevant knowledge from one environment to another, one can provide major improvements to the design process, and also enable leveraging of knowledge from one endeavor to another.

This paper proposes a multi-faceted approach to provide a common interface to assist members of the design process to collaborate, to share goals, and to help formulate overall rationale. In addition to helping to create a knowledge repository for the specific design process on an incremental basis, the proposed approach facilitates knowledge discovery by immediately presenting and processing data from all stages of the design process. Detailed material on the "why" in spacecraft design is incorporated by capturing the true cost and utility to the customer.

2 Four Faceted Knowledge Based Approach

In the specific context of design of spacecrafts, knowledge-based techniques can be applied to more effectively achieve engineering goals by leveraging multiple facets on a concurrent basis: These facets are as follows [1]:

- **Knowledge Acquisition**, which represents the process of capturing information from various media, including people's minds and handwritten documents, into computer accessible media.
- **Knowledge Management**, which deals with mitigating issues relating to heterogeneities in underlying contexts of information coming from disparate sources such as multiple stakeholders, multiple projects and multiple stages of the process.
- **Knowledge Discovery**, which involves using emerging techniques to analyze huge amounts of information and to get better insights into such information than is possible using the best human domain experts.
- **Knowledge Dissemination**, which provides the automated extraction of the most relevant pieces of information from a huge computer based information infrastructure, with such extraction being tailored to the needs of different constituencies of users in each of the relevant set of organizations.

Based on the above framework, two sets of tools have been developed specifically for the spacecraft design environment. These have been named as SSPARCy and MIST.

SSPARCy is designed to perform automatic capture of vital information as the design evolves over time. It then allows users to view this information graphically, records the history of the states of the code, performs integration integrity checks and facilitates the capturing of design rationale associated with important code elements. SSPARCy incorporates a suite of features that provide users with a centralized source of information regarding a design simulation. It presents the current state of a simulation. In addition, it records the history of the various states of the entities of a simulation to allow the designer to examine and analyze the evolution of a simulation over time. Furthermore, it performs automatic analysis of system integration integrity to alert an integrator of potential problems.

The MIST system is an interface for capturing, processing, analyzing and storing information about utility characteristics for every stakeholder in the design process. It provides a framework for expressing utility through the creation of attributes for every stakeholder, as described in the MATE process [2]. The MIST system allows designers and stakeholders to use the system to create, define and analyze attribute information. It structures the knowledge, and builds a series of rule-based interviews to elicit the stakeholder's utility. In addition to making the design process more efficient by conducting the interviews, the MIST paradigm incorporates analysis tools that allow users to observe links between multiple stakeholders, multiple design phases and multiple projects. Further, emerging data mining techniques are being incorporated into the MIST system to provide knowledge discovery capabilities geared specifically to the spaccraft design community.

3 Previous Efforts

Before we present details of our knowledge-based approach, we present information of related efforts by other researchers.

3.1 Vehicles Knowledge-Based Design Approach

The Vehicles knowledge-based design environment, developed at the Aerospace Corporation [3], is a framework for managing knowledge related to the design environment of space systems. Analysis and modeling tools are combined with a historical database of previous projects to assist in building new architectures that benefit from the experiences of old architectures. The issues addressed in Vehicles are similar to the issues discussed in this paper, relating to what the best methods are for formalizing the knowledge related to the design environment. Vehicles proposes a flexible environment in which to describe systems and subsystems, and to analyze the effects of various changes to the design. The software environment provides a platform for designers to build their own tools, tailored for the specific types of analysis they wish to conduct. The design rationale, history and utility capture capabilities described in this paper can provide a complimentary value proposition to the Vehicles system.

3.2 Rule-based Algorithms from Chung-Hua University

One system, under development at the Chung-Hua University in Taiwan [4], uses a rule-based algorithm for transferring an individual customer's needs directly into specifications, by developing a matrix of weights between attributes and design factors. The weights are then used to determine rules, or relations, indicating how certain design parameters should change their respective values based on other values of parameters.

The proposed paradigm can build on this approach by incorporating information on the goals and needs of multiple stakeholders, as well as by using the evolving knowledge repository to facilitate decisions on how specifications can be derived from customer preferences.

3.3 AIDA

The Artificial Intelligence Design of Aircraft (AIDA) effort at the Delft University of Technology applies a case-based reasoning method to delineate initial values for an aircraft lay-out using knowledge from previous cases [5]. Once again, however, the AIDA approach is dealing with the goals and desires of only one stakeholder. Additionally, the AIDA approach seeks to find one optimal architecture, rather than to provide a knowledge-based framework to be used by the designer to explore the various options.

3.4 SPOOL

The SPOOL project at the Université de Montréal takes a slightly different approach; it uses reverse engineering techniques to analyze existing projects and to determine which patterns in the design could be used to infer the rationale behind recurring patterns [6]. This minimizes the need for the designer to provide the rationale. By combining utility and expense data with the design history, our proposed approach will be able to make similar inferences regarding the design rationale at every stage of the design process, thereby offering an improvement over both the SPOOL approach and the other approaches mentioned above.

3.5 C-DeSS

The C-DeSS system developed by Klein proposes a geometric design rationale tool [7]. This system provides a language through which a designer can describe the design geometry and express reasons for decisions made in forming this geometry. Elements of such a structured language can be integrated into the framework described in this paper, to form a more concrete representation for the entire design, not just for the geometry.

3.6 DICE

The DICE (Distributed and Integrated Collaborative Engineering Environment) methodology initiated by Sriram offers a platform for collaborative engineering by decomposing each engineering project into a set of modules and allowing work to be conducted in parallel on each section of the project [8]. When the system encounters conflicting decisions about a particular design decision from engineers in different modules, it uses the design rationale to help negotiate the outcome. The approach described in this paper would complement the DICE approach by providing a concrete relationship between the history of the design parameters and the associated utility and expense functions which led to these decisions. This would allow engineers, working in a set of collaborative enterprises, who have conflicting solutions to a given problem to gauge the exact implications of each option on the different stakeholders of the system.

3.7 WAVE

WAVE is an algorithm to learn information extraction rules [9]. Since it is intended to be an algorithm, it does not offer the broad functionality that is available in SSPARCy. However, the incorporation of the WAVE algorithm into SSPARCy would augment the latter’s flexibility and ability to adapt to changes in the syntax of the designer code. Further, this could potentially enable the application to analyze other programming language by allowing SSPARCy to learn the information extraction rules for a new language over time.

3.8 Utility Evaluation from the University of Massachusetts

The Trade-off Based Robust Modeling and Design group at the University of Masschusetts has developed an online utility evaluation tool which conducts interviews to determine stakeholder preferences in a similar fashion to the MIST approach [10]. The system incorporates mechanisms to deal with preference consistency and uncertainty and risk. These issues are addressed in the context of a given stakeholder for a given project. The MIST approach builds on this methodology by capturing information which will be useful in determining the relations between preferences of multiple stakeholders, at multiple stages of the design process, for multiple projects.

3.9 ICAD System

The ICAD system is a Knowledge Based Engineering software solution used by world class manufacturers in aerospace, automotive and industrial equipment manufacturing, such as Boeing, British Aerospace, Pratt and Whitney, GM, Ford, Jaguar, Lotus and others, to automate system-level design, product design, tooling and product configuration [11]. This system uses generative technology to capture and apply generic product design knowledge, which includes product structures, development processes, and manufacturability rules. Such a system represents a potential application for the knowledge captured and managed by the approach described in this paper.

3.10 NASA

The Virtual System Design Environment at NASA addresses the importance of facilitating collaborative environments in spacecraft design [12]. One of the focuses of the system is to enable the modification of a spacecraft structural component by various members of a team in different geographic locations. This issue can be extended by the MIST approach to incorporate the direct involvement of the stakeholders in the system. If designers are able to interact at the component design level over a network, the same mechanisms can be used to demonstrate concepts, explain the status of the project, and gather input from the stakeholders of the system.

3.11 Design Repositories at NIST

The National Institute of Standards and Technology Design Repository Project addresses the importance of developing formal representations of knowledge in the design environment [13]. A specific language is being developed to represent design models, and interfaces for creating, browsing and searching for information on these models are being developed as well. The integration of design history and rationale with utility information from multiple stakeholders represents a complimentary means of representing a design model which addresses many of the same issues. For systems like these to work together, a set of standards must be developed, as described by the Engineering Design Technologies group at NIST [14].

3.12 Other Design Rationale Tools

The existing field of design rationale capture tools spans the spectrum from fully unstructured rationale to completely modeled rationale. Meeting minutes represent an unstructured, time delineated capture. QuestMap [15] and DRAMA [16] are examples of the next step – they provide basic structural elements and enable the user to devise a useful structure. At the other end of the spectrum from meeting minutes is DRIM [17], which is a completely specified model for the rationale underlying the design process. As a design rationale capture tool, SSPARCy lies somewhere on the spectrum between QuestMap and DRIM. SSPARCy creates a simple structure for design rationale by associating rationale with each simulation entity. Additionally, SSPARCy captures this rationale over time. Since this rationale can evolve at any level from project to variable over time, a minimal logical structure is provided for the user to specify rationale in the manner and the level he or she perceives as being most beneficial. SSPARCy does not impose the rigid conceptual structure that DRIM proposes. Therefore, SSPARCy is a compromise in terms of design rationale capture between inflexible structure and amorphous disorder. Furthermore, the basic structure it provides is most appropriate for the domain-specific design process it endeavors to capture.

Design reuse is an obvious application for design rationale. Work at the Tsinghua University in Beijing demonstrates a prototype for using design rationale to support design reuse [18]. The Tsinghua system incorporates many intelligent mechanisms to process design rationale information and relate it to the design of the system. The knowledge captured by the system is used in future projects as a means of decision support.

4 Knowledge Based Module for Design Rationale Aspect

Consistent with our multi-faceted knowledge based approach, our research team has developed concept demonstration prototype tools to illustrate how such tools can incorporate growing knowledge over time and offer increasing value to the customers.

4.1 Architecture for Design Rationale Concept

We first describe SSPARCy, which is a knowledge-based tool that can facilitate design and development steps over time by ensuring that vital design decisions and rationale have been encapsulated. By providing the user with detailed system analysis, history reviews, and error checking, SSPARCy attempts to address the void that currently exists for methods to automatically capture crucial information with no or little human intervention.

Each simulation exercise can be encapsulated in one major object, which is referred to as a project. The Project object contains all the necessary objects and variables that represent the information stored in the application, such as functions, constants, design variables, and errors. The Function objects refer to actual functions in the simulation source code. Similarly, the Constant and Design Variable objects represent each global constant and design variable that is defined in the system. Lastly, the Error object refers any possible error in the simulation design that can lead to redundant, unused, or misrepresented code in the code files. A graphical view of the architecture of the data model is shown in the figure below.

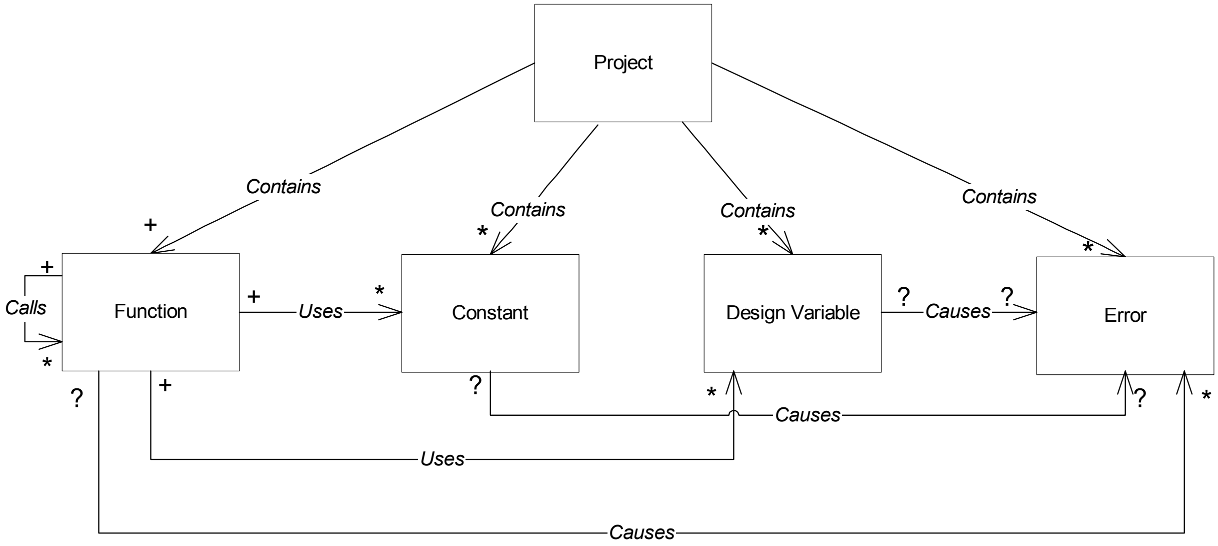


Figure 1: A graphical representation of SSPARCY Data Model which shows how major system objects can interact over time [20].

Each Function, Constant, Design Variable, and Error object can be referred to as a general Variable object. The architecture has been designed so that each Variable object can contain valuable information regarding the Variable's name, value, units, valid range, author, date of creation, and possible aliases which can refer to it in the project. Also, each Variable stores its rationale, so the user can record design decisions and changes that relate to each object in the system. The data that are stored in the object model provide significant possibilities for greater functionality in the graphical user interface.

4.2 Knowledge Acquisition Aspect of SSPARCy

The graphical user interface has been designed to facilitate knowledge acquisition through intelligent capture of simulation exercises. For example, the history table feature visually conveys the degree of stability of parameters in the system. Users can easily identify the volatilities of different parameters across iterative design sessions. Also, by storing Error objects in the data model, the user can analyze potential system problems that are currently very difficult to analyze or even recognize.

Even though the functionality of the system is continuously evolving, several of the major components can be seen through the graphical user interface. Currently, SSPARCy has been designed to be compatible with multiple types of simulation files. SSPARCy facilitates project management efforts by allowing multiple projects, specified in different formats, to be opened, viewed, and saved simultaneously. The system also enables the user to view any Variable object in the current project, and to see all the vital information fields that are stored along with that Variable. As the system design evolves over time, variables may be added, removed, or changed from the current status. The graphical user interface enables the user to view the manner in which the project has evolved over time, in terms of any of the variables that characterize the project. Finally, the current implementation incorporates functionality for error checking so that users can see what possible errors might exist in the project and where those errors might have occurred.

4.3 Knowledge Management Aspect of SSPARCy

4.3.1 Current Variable Data

As mentioned before, it is essential that the user be provided with an easy and effective way to view the current state of any Variable object in the system. Whether the user wants to see the value and the rationale of a global constant in the project or just the author of a specific design variable, SSPARCy provides appropriate display options. By using tables to display a collection of Variable objects, the application allows for quick review of essential data in the source files and the rationale behind their existence and their respective values.

A typical variable data table would show the name of each constant in the project listed in the first column in alphabetical order. Next to the name, the second column lists the subsystems in which these parameters belong to, followed by the current values of these parameters. This setup represents the display at the default situation. Note that only the parameters in use are kept in this table. If a parameter had been removed in the latest design sessions, it would not show up on this list but would appear on the history table instead.

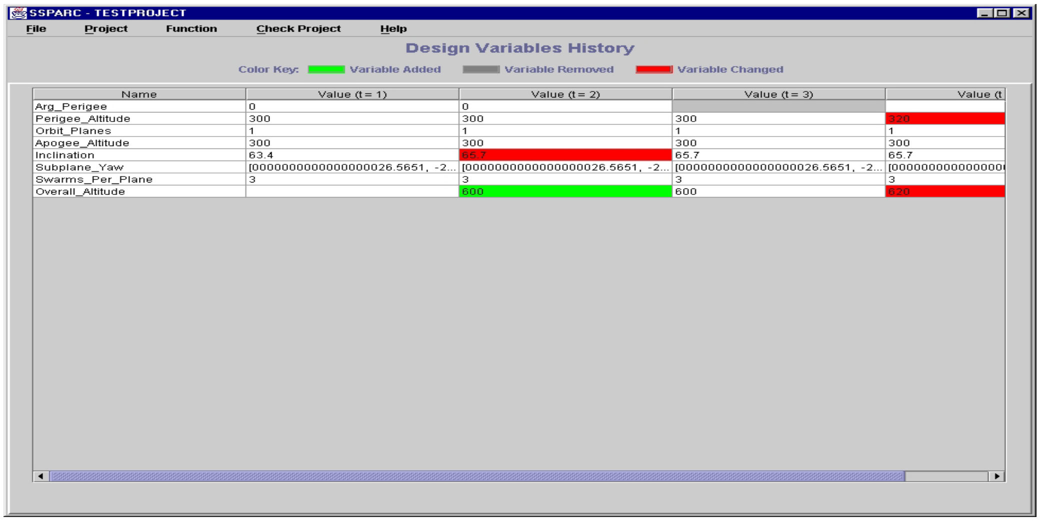
4.3.2 Variable History Review

For every Variable object in the system, the user is able to quickly review how that variable has changed over time and what variables have been added or removed from the current project. As seen below in Figure 2, a tabular format is once again utilized to display such information.

In this example, the history of the project's design variables is presented to the user. The first column of the table lists the names of every design variable that has existed in the project since it was first created. The rest of the columns in the table represent the state of the project over time. Each column gives the value of the design variables at that time, or leaves the cell blank if that design variable was not present in the project at that point in time.

By adding color codes to the table, the graphical user interface gives the user an easy-to-read look at the history for the particular variable. Based on the coloring of a cell in Figure 2, one can tell if something has changed for the design variable in row x at time

y. If the background of the cell is colored green, it indicates that the design variable was added to the project at this time.   In the same manner, a dark gray cell indicates that the specific design variable in row x was removed from the project at time y. Finally, a red background notifies the user that the design variable has been neither added nor removed, just that its value has changed from time y-1 to time y.



Design Variables History

Color Key: 🟩 Variable Added ⬛ Variable Removed 🟥 Variable Changed

Name	Value (t= 1)	Value (t= 2)	Value (t= 3)	Value (t
Arg_Perigee	0	0	⬛	
Perigee_Altitude	300	300	300	🟥 320
Orbit_Planes	1	1	1	1
Apogee_Altitude	300	300	300	300
Inclination	63.4	🟥 65.7	65.7	65.7
Subplane_Yaw	[0000000000000000026.5651, -2...	[0000000000000000026.5651, -2...	[0000000000000000026.5651, -2...	[0000000000000000
Swarms_Per_Plane	3	3	3	3
Overall_Altitude		🟩 600	600	🟥 620

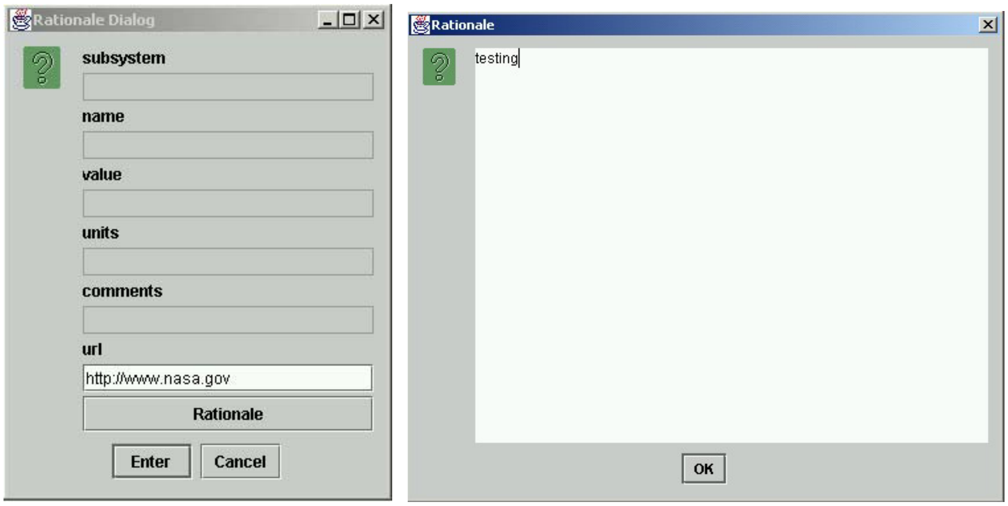
Figure 2: The history of Variable values over time can be viewed in a color-coded table

By allowing the user to view the history of any Variable object in the system, SSPARCy provides an extensible tool for comprehensive analysis of successive simulation exercises.

4.4 Knowledge Discovery Aspect of SSPARCy

In order to ensure that the design rationale capture aspect imposes zero or minimal overhead on the designers, SSPARCy automatically parses parameters' values, units, comments, and timestamps from their source files. However, if a designer wishes to enter additional details regarding a parameter, he or she can do so by selecting the parameter from the table and then clicking the button "Edit Info." New pop-up windows will appear, and the user can enter supplementary information such as rationale and URL references (see figure below). Other users can later access these information by choosing the button "View Rationale" or "View URL."

A table can be generated for any of the Variable objects in the system. Therefore, with just one selection from the menu bars at the top of the screen, the user can be presented with a table that displays all the functions used in the simulation, the global constants that exist, and the design variables that are used throughout the project.



Rationale Dialog

subsystem

name

value

units

comments

url

http://www.nasa.gov

Rationale

testing

Figure 3: Rationale Dialog Windows allow users to input additional details regarding specific parameters.

In order to provide better inspection and analysis of the future simulation files, a new approach was developed to provide the following functionality:

- Permit the viewing and storage of important information (name, value, rationale, author, etc) relating to the *functions* that are used in the project

- Enable the viewing and storage of important information (name, value, rationale, author, etc) relating to the *global constants* that are used in the project

- Facilitate the viewing and storage of important information (name, value, rationale, author, etc) relating to the *design variables* that are used in the project

- Permit storage of the history of the above data as it changes over time in order to enable the user to have information regarding what information has been updated since the last system design

- Provide for error checking of the current system design to inform the user as to where possible errors may exist in the simulation and modeling files and how those errors may be fixed.

5 Multi-attribute Interview Software Tool (MIST)

Multi-attribute Tradespace Exploration and its follow-on Concurrent Design analogue (MATE-CON) are designed to facilitate intelligent interaction with the designer and customer. Using the cumbersome face-to-face Multi-Attribute Utility Analysis (MAUA) interview process previously developed by the SSPARC team as a foundation, MIST uses an advanced Graphical User Interface (GUI) and a web-based computer interface with graphics to speed up and enrich the utility interview process. The designer is able to describe attributes, ranges of values, units and scenarios developed with input from the customer. The system then prepares an interview based on the attributes of the tradespace, and allows the designer to conduct the interview and to enter the responses. Further, the customer can take the interview independent of the utility facilitator. The MATE interview process consists of multiple stages, dealing first with single attribute utility parameters and then with multiple attribute utility parameters. Each stage also contains a set of validation questions to ensure that the variables being considered are independent of each other.

The automated and customized interview sessions present the customer with a scenario using the lottery equivalent probability (LEP) approach as developed in the field of decision theory. Each option is a situation with probabilities of two states for attribute analysis. The user will usually express preference for one of those states. In response, the system will modify one of the states and ask the question again in a manner very similar to the series of dual-value choices given by an optometrist when fine tuning the power of the lens. The system continues in this form until the customer answers that he/she is indifferent to the two options. The software then tallies the indifference points for later calculation and moves on to analyze the next attribute. This entire process can be conducted on a single day or over time. The results of each interview serve as the basis for developing functions to assess the utility and the cost of design based on the set of the attributes specified by the customer or the set of customers. The integration of these data with the design parameter data collected by the SSPARCy system provides a comprehensive assessment of the rationale at various levels of design detail throughout the process.

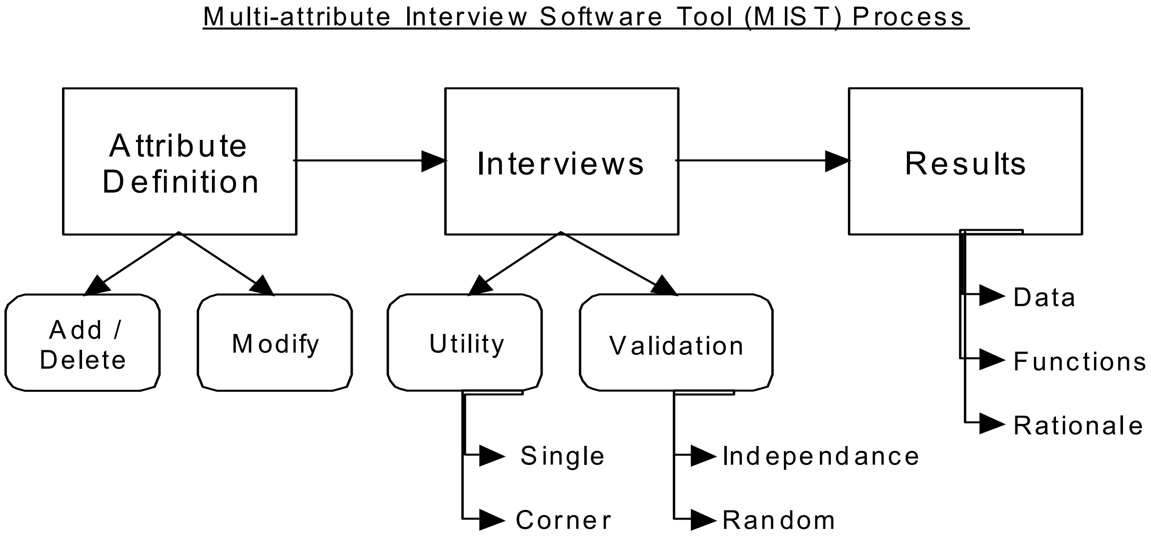


Figure 4: Overview of major steps which MIST can automate.

5.1 Knowledge Acquisition Aspect of MIST

By conducting the interview with a software tool, the process can be made fully interactive and tailored to the individual responses. Computations are performed while the interview session is proceeding. Designers are provided with the options for adding and modifying attributes, and also for changing the scenario used to describe an attribute to the customer. As an improved communication interface is the core goal of this endeavor, the system allows not only the designer, but also the customer to provide feedback about the structure of the interview process. By systematic analysis of the choices made during the interview process, one can develop an evolving knowledge repository about the design of the system as well as the design of the interview itself. The latter allows for future interviews to be conducted in a more efficient manner. The major issue raised in this area is how much control and flexibility is appropriate in customization of the interview session in real-time. Allowing excessive variability in the interview process may reduce the reliability and consistency of the results.

5.2 Knowledge Management Aspect of MIST

In this concept demonstration prototype endeavor, we have emphasized the visualization aspect. The paradigm of customized visualization alone adds value to customers by helping them to understand the way in which their utility estimates evolve over time. The multi-attribute cost and utility functions can be created in graphical format. One can graphically witness how answers elicited during the interview process impact the design process. This impacts the integrity of the interview process in both positive and negative ways. The positive aspect is that with increased knowledge in real-time, the customer may make more rational decisions: when faced with a graph which plots each of their answers for a given attribute, a customer may be alerted to reconsider the answer. On the negative side, the customer may tend to give conventional and risk averse responses, as many individuals probably prefer to give answers which appear to be consistent and rational. Further, a customer's true preferences may be changing over time, and it might be harmful to allow the customer to "game" the system. This is another issue which can be explored with the introduction of an interactive tool to facilitate the interview. The optimal level of customization will depend on the characteristics of the particular problem domain.

5.3 Knowledge Discovery Aspect of MIST

One of the major intellectual questions that MIST attempts to address is the role of utility and attribute based design to design rationale. The two systems, SSPARCy and

MATE, have a common interest in collecting information on design parameters over time, and reacting to the major trends and changes. By using one integrated system to collect both forms of data, one possibility is to relate each moment of the design process to a specific interview, thereby associating an interview with each value of a design parameter. Major changes in the design parameters can then be easily mapped against the changing preferences elicited during the interviews.

5.4 Knowledge Dissemination Aspect of MIST

The MIST system also contains a set of analysis tools to process the data collected from the interviews and the SSPARCy design parameter capture process. First, a set of reports can be produced to document both the customer's preferences and the various stages of the design process, as well as perhaps the design rationale specified by the designer. The use of an integrated tool allows for a consistent template to be used and for a comprehensive knowledge repository to evolve over time. When a new person joins the team, such a knowledge repository will provide an invaluable mechanism for transferring the knowledge about the design process to the new member.

In addition to the reports, algorithms for utility and expense function generation are being integrated with the system to produce meaningful and useful tools with the interview results. The results of the interviews will be used to design a specific function for each stakeholder at the end of each interview. This will allow all iterations of an architecture to be assessed, and the changing utility of an architecture to be related to the changing preferences of the customer. The algorithms for developing these functions have already been delineated by the design team.

Finally, other data mining tools will be implemented to use the knowledge repository as a vehicle to facilitate knowledge discovery within the design process. Once adequate data have been collected about a specific design process or multiple design processes, interpreting the patterns within these data can lead to better decisions in future endeavors.

5.5 Attribute Interface

The MIST system utilizes the notion of *attributes* to represent the user-defined characteristics of the project that describe the important factors for the stakeholder. Before the interview process can commence, the engineers and the stakeholder must agree on a set of attributes to build the utility functions. This process involves a series of decisions, and a set of discussions between the different members of the engineering team, as well as the stakeholder. The attribute interface has been designed to structure the important pieces of knowledge related to each attribute, so that the discussion can be focused. Since the questions for the interviews are generated while the interview session is actually "on", it is technically possible to change the properties of an attribute, or add an attribute, at any time during the process. However, since an interview is given in the context of the current state of attributes, any chance in the properties of the attributes could invalidate the previously collected interview responses. For this reason, the attributes in the system must be defined before the interview session begins.

An attribute is represented in the system as a hybrid data type with various properties as shown in the figure below. The figure shows the properties related to an attribute, the data type of each property, the description and comments. Some of these properties are directly related to the description of the system, and others were implemented in order to facilitate the software system. When an attribute is created by the design team, an instance of the attribute data type is instantiated and modified as appropriate. The interview user forms and the analysis tools base their actions on the status of the attribute properties.

Attribute Property	Data Type	Description	Comments
`Name`	`String`	Name of attribute	Understandable to all parties involved
`Min`	`Double`	Lowest value in range	
`Max`	`Double`	Highest value in range	
`Units`	`String`	Units for attribute values	Must be very specific
`Increment`	`Integer`	Number of indifference points to be collected	Enough to avoid large jumps in utility
`Direction`	`Boolean`	Direction of increasing utility	Towards max or min value?
`Scenario`	`String`	Scenario for single attribute interview	Describes context in which only the single attribute is considered
`Resolution`	`Double`	Probability resolution to which utility can be distinguished	Usually around 5%
`Format`	`String`	Format in which value of attribute is presented to user	Examples: “5x5”, “6”
`UnitsInc`	`Boolean`	Include units in format?	Usually `true`.
`Independent`	`Double`	Attribute value for independence interviews	Usually near middle of attribute range
`Definition`	`String`	Definition of attribute	Must be very specific
`LinearScale`	`Boolean`	Increase indifference point questions on linear or logarithmic scale?	
`Threshold`	`Double`	Difference in utility between adjacent indifference points which requires more questions	

Table 5: Table of properties for the attribute data type in MIST

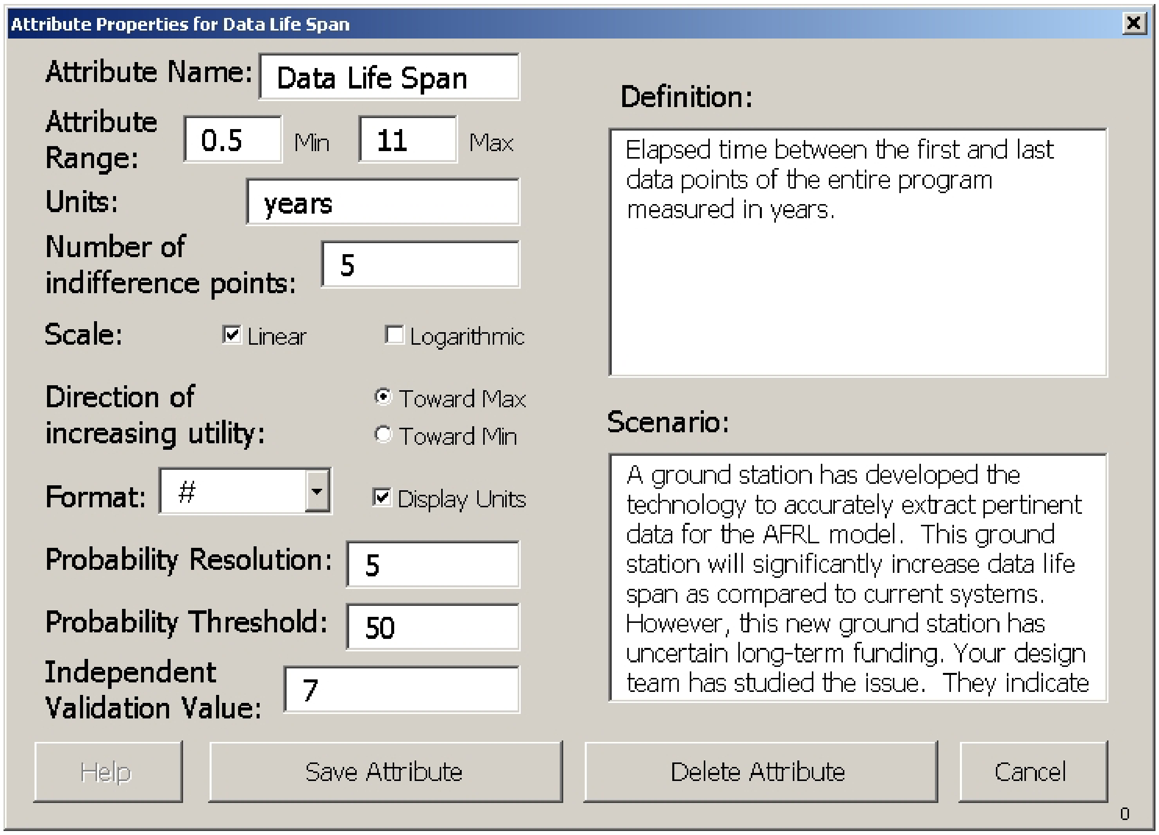
5.5.1 Attribute Operations

A user can add, modify, or delete an attribute as described in the following paragraphs.

The user can add an attribute to the system from the main page. When the “Add Attribute” button is clicked, the Attribute Properties form is presented to the user, and the user can enter the properties for the attribute. Since part of the purpose of this form is to enable the engineers to conduct the discussion during the attribute definition period, it is not necessary for the user to put in a value for every property of the attribute. This process must be completed by the time the interviews are ready to be conducted – otherwise, the interview generation module will search for a value of the attribute which will not be there. The only required element for the attribute is the Name, since it is stored based on the name. After the properties are entered, the user can click on the Save button, and the attribute will be added to the system. At this point, the Attribute Rationale form is displayed and the user is asked to enter a rationale for adding the attribute to the project. The name of the attribute is added to the list of attributes on the main page, both in the completed interviews matrix and the random values section.

To modify an attribute, the user can click on the Modify Attribute button from the main page. This brings up the Attribute Navigator form, which shows a list of attributes currently in the system. The user selects the attribute they wish to modify, and clicks on the Modify button. The Attribute Properties form is displayed with whichever values of the attribute are in the system. Any of these values can be modified, however since the attribute is defined by its name, changing the name would result in the system creating a new attribute with the new name. When the Save button is clicked, the user is asked if the change to the attribute is a major change, worth cataloging. If it is, the user is prompted for a rationale for the change in the Attribute Rationale form.

To delete an attribute, the user must click on the Modify Attribute button from the main page, select the attribute to be deleted, and then click the Delete button. Deletion is only available under the Modify Attribute form to ensure that the user is certain that he or she wishes to delete the attribute, instead of simply modifying it to make it appropriate. When an attribute is deleted, the user is prompted for the rationale behind deleting the attribute. This rationale is stored along with the final values of the attribute, in the attribute's history page. While the attribute name is deleted from the main page and all interviews, the history of values is still stored within the project, to preserve knowledge of the decision making process which led to this deletion.



Attribute Properties for Data Life Span

Attribute Name: Data Life Span

Attribute Range: 0.5 Min 11 Max

Units: years

Number of indifference points: 5

Scale: ☑ Linear ☐ Logarithmic

Direction of increasing utility: ● Toward Max ○ Toward Min

Format: # ☑ Display Units

Probability Resolution: 5

Probability Threshold: 50

Independent Validation Value: 7

Definition:

Elapsed time between the first and last data points of the entire program measured in years.

Scenario:

A ground station has developed the technology to accurately extract pertinent data for the AFRL model. This ground station will significantly increase data life span as compared to current systems. However, this new ground station has uncertain long-term funding. Your design team has studied the issue. They indicate

Figure 6: Attribute Modification interface

5.5.2 Attribute Storage

When an attribute is still active in the system, the current states of the properties of the attribute are stored in a worksheet. At all times, the user can know what the current values are. In the leftmost columns, each of the fields in the Attribute Properties form is stored in the appropriately labeled cell. The "Attribute Options" column represents the list of values for which the single attribute interview will seek indifference points for this specific attribute. This list is generated by the system every time the properties of the attribute are saved, either by adding or modifying the attribute. It is done by splitting the attribute range, specified by the minimum and maximum values, into the number of equal segments specified by the number of indifference points, either linearly or logarithmically. This list is then randomized and stored in the worksheet.

Along with the properties of the attribute, the worksheets are also used to store all responses given in interview sessions with respect to this specific attribute. These worksheets help to establish a complete history of the decisions made during the interview process. The history is used by the system to determine which interviews have been conducted, and also by the designers.

Although it is not designed to be modified by the users of the system, the values in the attribute worksheet can be modified as a backup method for resolving issues that may come up during the interview process. For example, if the attribute options generated are not round numbers, the designer might prefer to edit the values to provide the stakeholder with more reasonable questions during the interview process. Also, during the interview, if the stakeholder wishes to go backwards and revise his or her answer, the worksheets provide the means for editing or deleting responses. As the system evolves further, it will become less necessary to access the attribute worksheets.

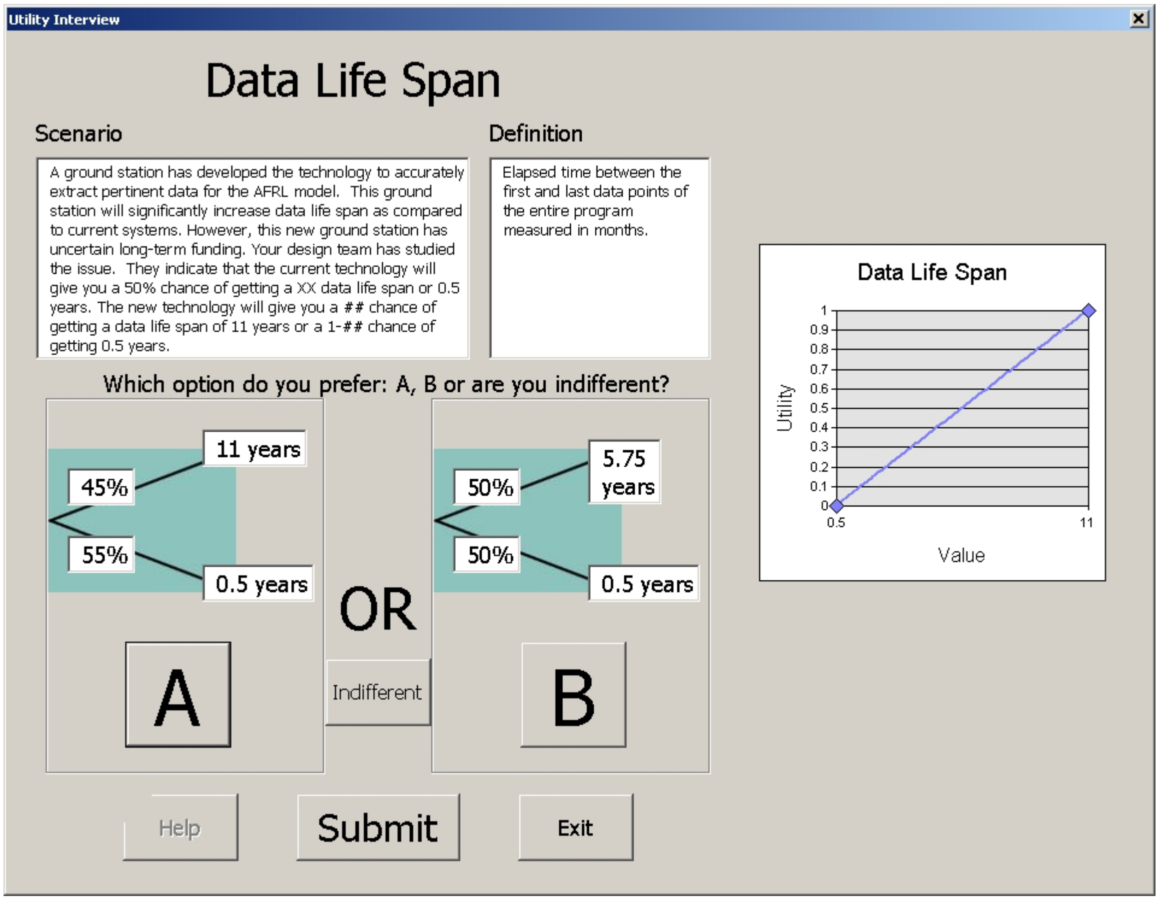
Each attribute has a corresponding Attribute History worksheet that is created and modified along with the attribute. Since this worksheet can never be deleted, it provides a complete history of the entire attribute definition period. Each column of this worksheet contains a list of the properties of the attribute. It also contains a cell for the user-defined rationale for the specific iteration of attribute property values. These values are only stored when the attribute is created, deleted, or the user specifies that a change made to the attribute was a major change deserving cataloging.

5.6 Characteristics of Interview Process

Each interview is designed to acquire a piece of knowledge intended to help build or validate a utility function for the stakeholder being interviewed. The interviews are meant to take place after the attribute definition phase has occurred, and the attributes have been approved by the stakeholder. The interviews can be conducted at multiple points throughout the design process, to provide a continuing notion of the important goals and requirements of the stakeholder being interviewed.

5.6.1 Visualization

The issue of visualization is one that was examined in great detail by the designers of the MIST system. Using the MIST concept demonstration software, it is possible to display the curves representing the utility functions for the attributes being discussed as the interview is taking place. The figure below demonstrates this functionality for the single attribute interview, where the user's responses each help build the utility function for a single attribute. Similarly, the utility curves generated for all of the attributes can be displayed in any of the multi-attribute interviews.


Utility Interview

Data Life Span

Scenario

A ground station has developed the technology to accurately extract pertinent data for the AFRL model. This ground station will significantly increase data life span as compared to current systems. However, this new ground station has uncertain long-term funding. Your design team has studied the issue. They indicate that the current technology will give you a 50% chance of getting a XX data life span or 0.5 years. The new technology will give you a ## chance of getting a data life span of 11 years or a 1-## chance of getting 0.5 years.

Definition

Elapsed time between the first and last data points of the entire program measured in months.

Which option do you prefer: A, B or are you indifferent?

Figure 7: Single attribute interview with utility curve displayed.

If a response during the single attribute interview leads to a drastic change to the utility curve, the interviewee can see this immediately and think again about whether the response accurately represents his or her preferences. During the multi-attribute interviews, when the user is answering questions dealing with the relationships between attributes, having utility curves displayed might allow them to help evaluate in their minds how they actually feel about the different attributes.

When this idea was presented to the fellow designers, the decision was made to withhold the visualization features from the stakeholder, because it might provide too much information leading to bias in the responses. For example, during the latter part of a session, a stakeholder might be inclined to provide responses which fit the initially defined utility curve, to avoid appearing irrational. Another possible downfall of the visualization feature is the ability of the user to pre-determine a utility curve in their mind, and to provide answers which fit this curve. Based on further tests, one will decide whether to offer the visualization capability as an option in future versions of MIST.

5.6.2 Bracketing

Every interview generated by this system is designed to find the value at which the interviewee is indifferent between the two situations presented. The decision is basically between one option where the probabilities of certain outcomes are fixed, and a second situation where the probability of one outcome as opposed to another is varied. The interviews are designed so that this varied probability is between 0% and 50% -- a probability greater than 50% would result in one outcome always being better for the user.

The method for obtaining successive values is based on the algorithm used in DeLeqeue's thesis [21]. At any given time, there is a known range of probabilities in which the indifference point exists. For example, when an interview question begins, it is known that the indifference point is within the range of 0% to 50%. The software module in the MIST system which generates potential indifference points is described in the flow chart below:

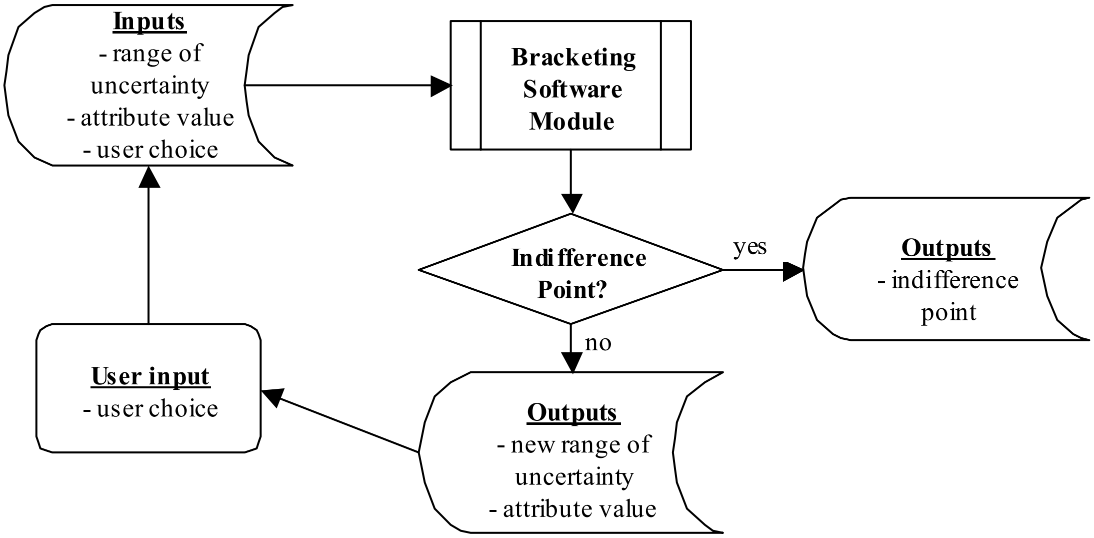


Figure 8: Flow chart describing bracketing software module.

Each successive question asks the user for his or her preference at the midpoint of this range. Based on the user's response, it will be clear whether the indifference point is either equal to, above or below the probability in the interview question. Either the interview ends with an indifference point selected, or the range is modified to reflect the latest response, and a new question is asked. Each probability chosen is a multiple of the attribute's probability resolution, which represents the degree to which the customer can distinguish between two situations. This process continues until the stakeholder declares a certain probability as the indifference point, or the size of the range is less than the resolution. At this point, the stakeholder is informed that the indifference point will be set to the value exactly between the two endpoints of the range. A modification to this bracketing procedure was made for the implementation described in this paper, as described in the Test Implementation section. The figure below describes the example from the X-TOS project. The probability resolution for this example was 5%, and instead of using the midpoint of the range, values were chosen closer to the endpoints. Future versions will incorporate a more general version of this strategy.

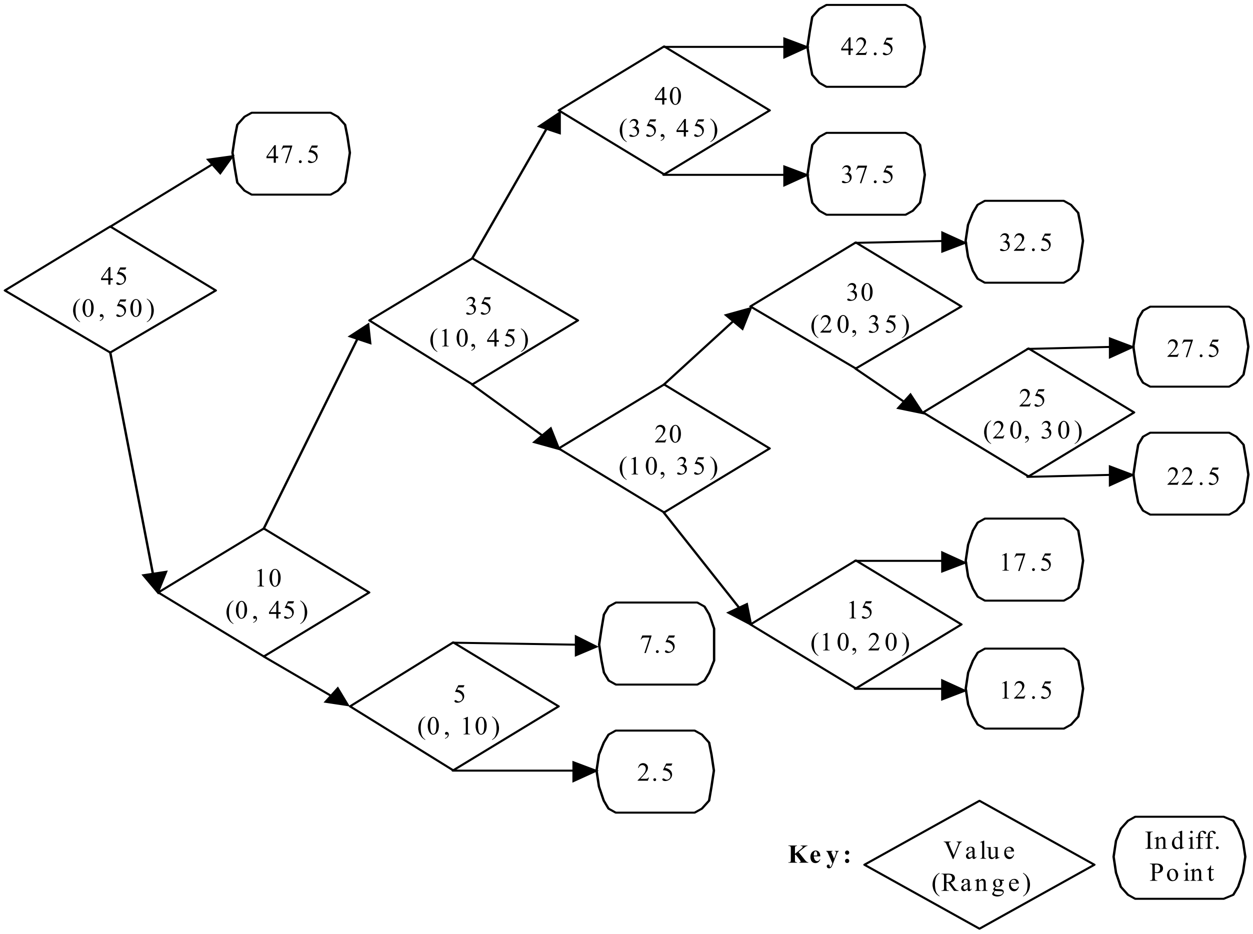


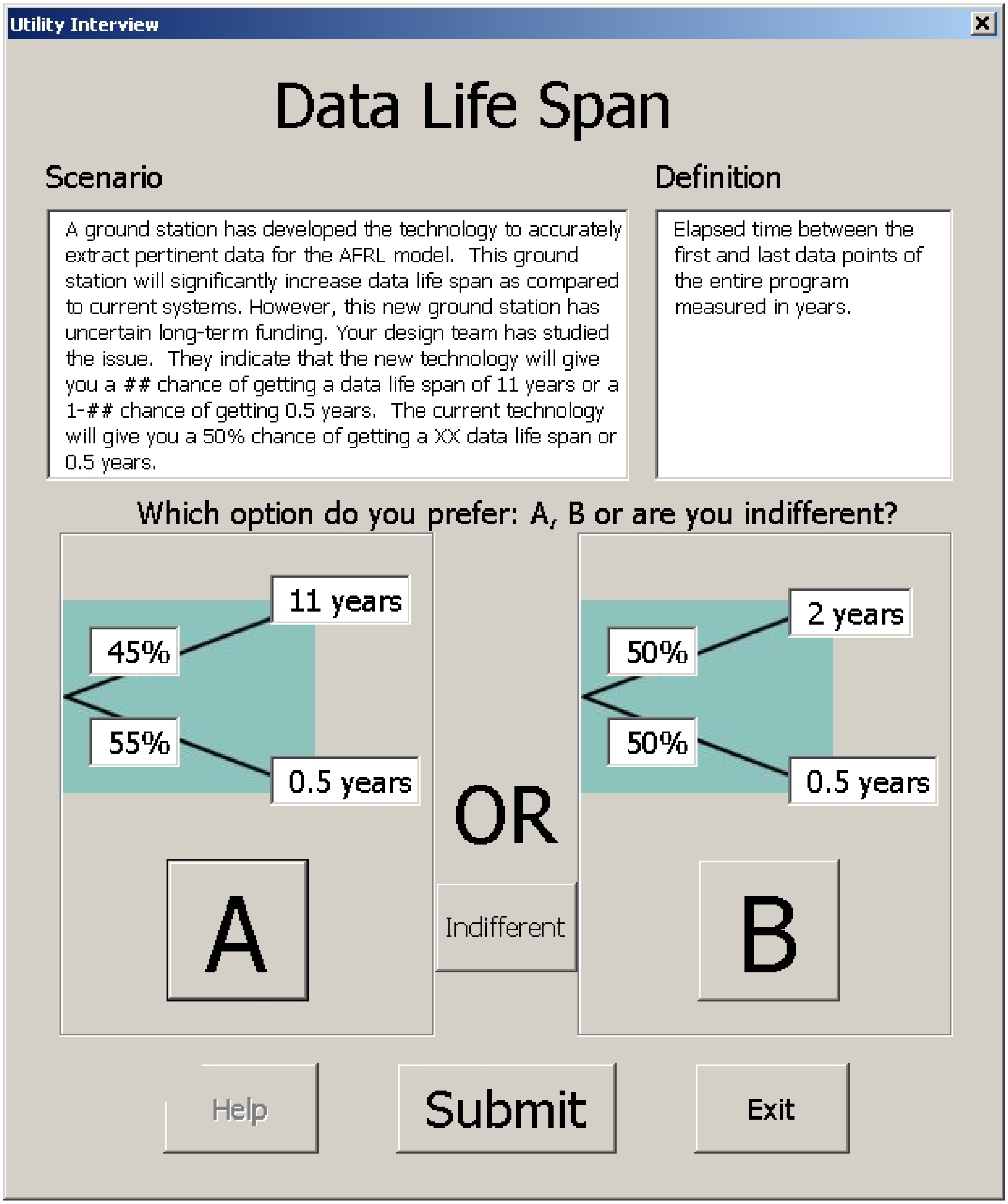
Figure 9: The X-TOS bracketing decision tree.

5.6.3 Single Attribute Interview

The goal of this interview is to build a utility function for each attribute, similar to the graph shown in the Output section of this paper. This function provides the mechanism for assessing the utility provided by differing values of the attribute for any proposed architecture. It provides decision makers with knowledge about whether one can obtain a significant gain in utility by increasing values at the lower end of the attribute range or at the high end of the range. The utility functions provide both a visual notional idea of the nature of the attribute, as well as a concrete input into a concurrent engineering simulation model. A concurrent engineering simulation can use the utility function, combined with a module which calculates the values of each of the attributes based on the architecture, to assess the utility provided for each attribute in any proposed architecture. This offers the potential to provide real-time feedback to engineers about any considered design decision without having to consult the various stakeholders each and every time.

After the attributes are defined and the properties are agreed upon, the system is deployed to the stakeholders with whom the attributes were designed. Depending on the nature of the roles involved, a single set of attributes may apply to a single stakeholder or a set of stakeholders with similar roles. The stakeholder begins an interview session with the single attribute interview.

The stakeholder navigates through the single attribute interview, as shown in the figure below, for each attribute. The user is provided with two pieces of information: the scenario and the definition of the attribute. The scenario is written to place the question in a context meant to emphasize that the specific attribute in the interview is the only aspect of the system to be considered at the time. Thus, a situation is described, such as the discovery of a new technology which has the potential to affect the value of the attribute, but carries with it some risk of failure. Identifying the level of risk that the user is willing to take is one of the purposes of the single attribute interview. The definition of the attribute is also displayed because often in engineering situations, the specific interpretation of a concept varies among different parties.



Data Life Span

Scenario

A ground station has developed the technology to accurately extract pertinent data for the AFRL model. This ground station will significantly increase data life span as compared to current systems. However, this new ground station has uncertain long-term funding. Your design team has studied the issue. They indicate that the new technology will give you a ## chance of getting a data life span of 11 years or a 1-## chance of getting 0.5 years. The current technology will give you a 50% chance of getting a XX data life span or 0.5 years.

Definition

Elapsed time between the first and last data points of the entire program measured in years.

Which option do you prefer: A, B or are you indifferent?

Figure 10: Single attribute interview form

The interview begins with a random value extracted from the list of values generated with the Attribute Property form. As described in an earlier section, these values represent increments along the attribute range at which indifference points are desired. Using this value, two situations are presented to the user. Each contains a certain chance of a favorable outcome, and a certain risk of an unfavorable outcome. Taken from multi-attribute utility theory, these questions are designed in the lottery equivalent manner [19]. One situation presents a 50% chance of obtaining the attribute value for which the indifference point is desired, and a 50% chance of obtaining the worst possible value for the attribute, as defined in the Attribute Property form. The other situation represents a certain probability *p%* that the best possible value will be obtained, and the probability *(100-p)%* that the worst possible values will be obtained. The value of *p* is varied based on the bracketing principle described above. Each time the user selects an option, the choice is recorded and the next probability in the bracketing sequence is displayed. This continues until an indifference point is reached – either by the user selecting the Indifferent button or by the system declaring an indifferent value. At this time, the indifference point is recorded, and the system displays the next random attribute value for which it desires an indifference point.

Once all of the indifference points are collected, the system uses the Utility Threshold collected by the Attribute Properties form to determine whether new indifference points need to be collected. If any two consecutive indifference points have a difference that is greater than the Utility Threshold, the system goes through the interview process for the midpoint value between the two indifference points. This process is continued until no two consecutive indifference points have a difference in utility which is greater than the Utility Threshold, thus ensuring that the utility curve is captured to a desirable level of fidelity.

When this process is complete, the single attribute utility interview ends and a check mark is placed in the appropriate cell on the main page. The user is returned to the Attribute Navigator form, to conduct further interviews for remaining attributes, as appropriate. The following figure describes this process flow:

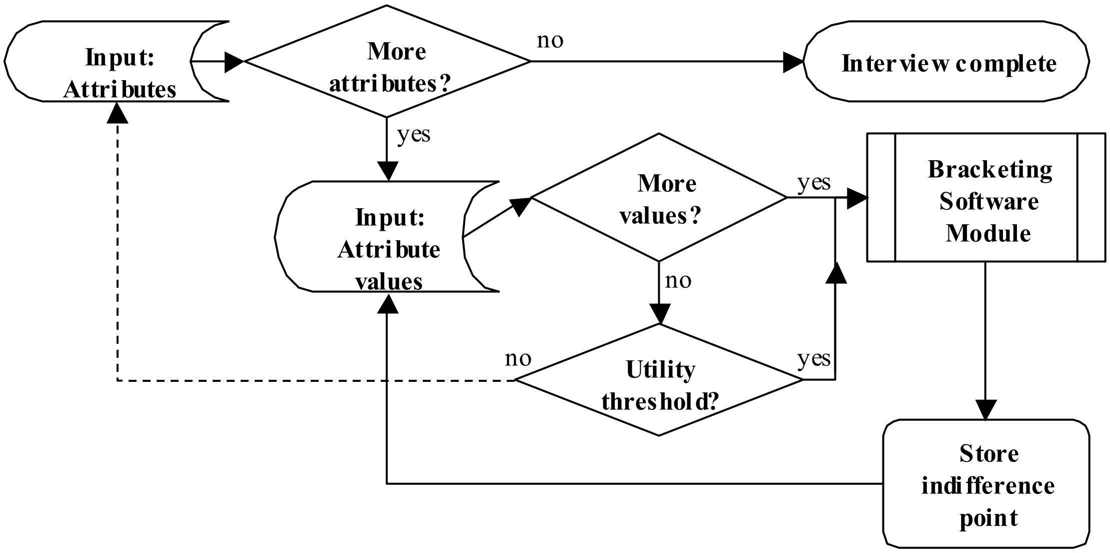
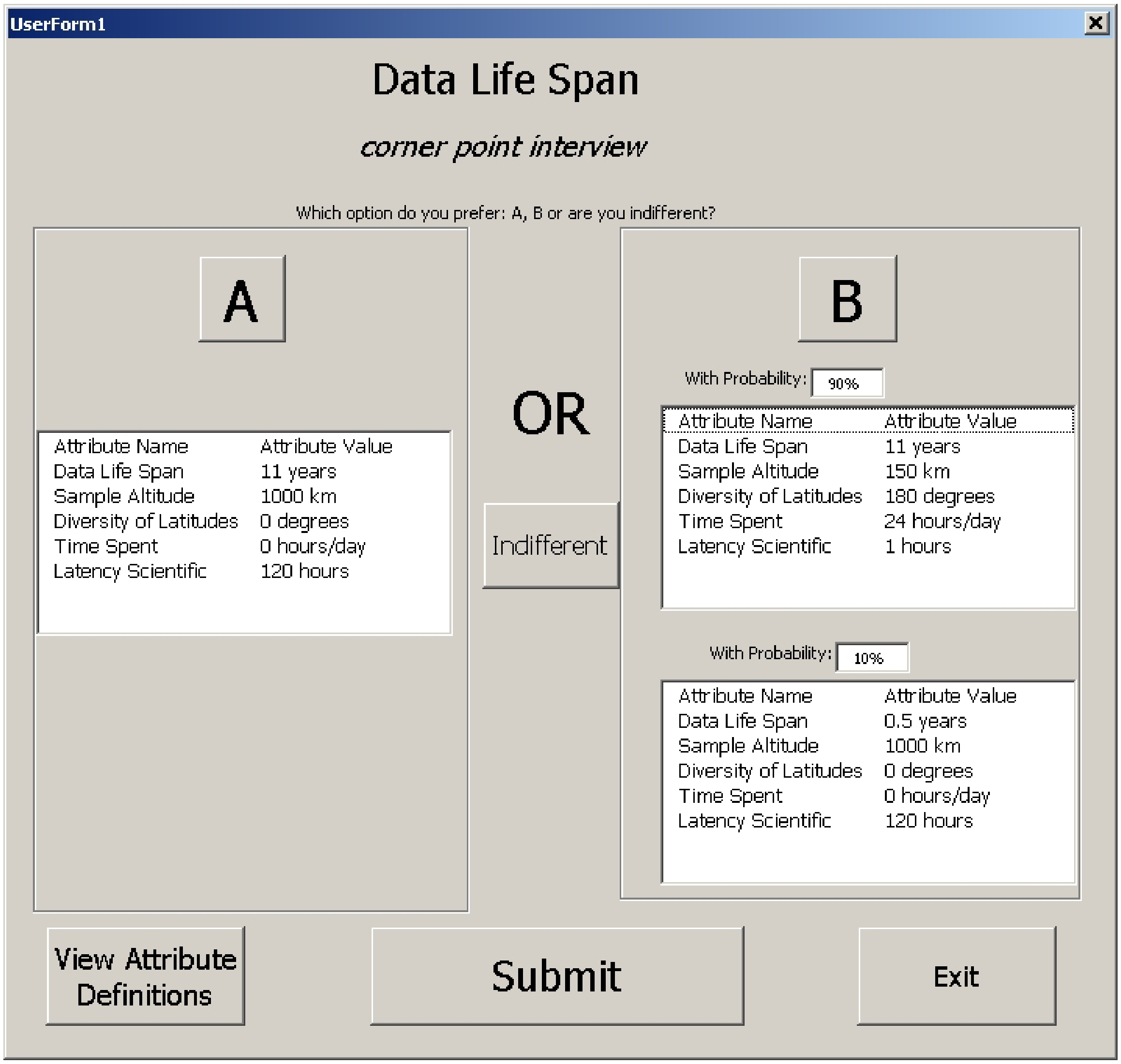


Figure 11: Process for single attribute interview

5.6.4 Corner Point Interview

The goal of the corner point interview is to determine the relative importance of each attribute with respect to the other attributes in the system by finding the "corner points" described in the MATE process [19]. This is done for each attribute by presenting the stakeholder with a situation similar to the one shown in the figure below. The user interface used for the corner point interview is similar to the interface for the single attribute interview, except that individual values are replaced with lists that contain values for each attribute.



Data Life Span

corner point interview

Which option do you prefer: A, B or are you indifferent?

A:

Attribute Name	Attribute Value
Data Life Span	11 years
Sample Altitude	1000 km
Diversity of Latitudes	0 degrees
Time Spent	0 hours/day
Latency Scientific	120 hours

B:

With Probability: 90%

Attribute Name	Attribute Value
Data Life Span	11 years
Sample Altitude	150 km
Diversity of Latitudes	180 degrees
Time Spent	24 hours/day
Latency Scientific	1 hours

With Probability: 10%

Attribute Name	Attribute Value
Data Life Span	0.5 years
Sample Altitude	1000 km
Diversity of Latitudes	0 degrees
Time Spent	0 hours/day
Latency Scientific	120 hours

Figure 12: Multi-attribute interview form, used for corner point and random mix interviews.

Further, the certainty equivalent method is used, in place of the lottery equivalent method [19]. In the certainty equivalent method, the user has a choice between being certain of a particular outcome, or having a chance of another outcome. In the corner point interview, the two choices are as shown below:

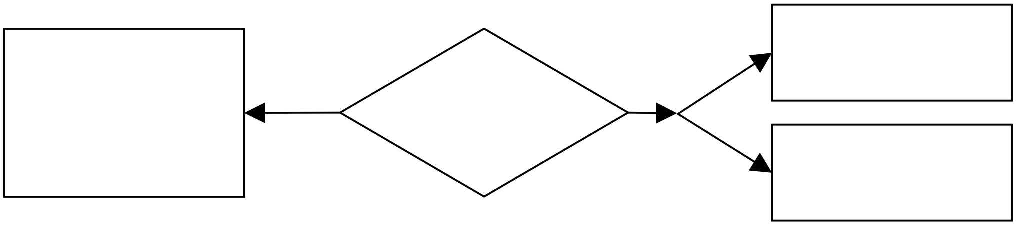


Figure 13: User choices in corner point interview

For each attribute, the above situations are created by populating the interview form's ListBoxes with the appropriate data from the attribute. Then, the bracketing module is used as described in the previous section, to find the percentage at which the stakeholder is indifferent between the two situations. At this point, the probability value, or "k-value", is stored with the attribute's parameter data. The "k-value" represents the relative weight, from 0 to 1, of the specific attribute being interviewed. A higher indifference point in a corner point interview means that the stakeholder was willing to give up a chance at a perfect system at a higher probability in exchange for the certainty of having a system where only the given attribute is perfect. The higher this probability, the more important the attribute, and thus the higher the k-value.

5.6.5 Attribute Independence Interview

After the data are collected, it is important to validate the selected attributes by ensuring that they are independent of each other. This is done by assessing and comparing the utility of two architectures, as described in the table below.

	Architecture 1	**Architecture 2**
Attribute	Intermediate value from att. def.	Intermediate value from att. def.
All Other Attributes	Set to **highest** possible values	Set to **lowest** possible values

Figure 14: Two architectures used to compare utility and determine independance

In each architecture, the value for the attribute being tested is left constant, at a value specified by the users in the attribute definition stage early in the process. The difference between the two architectures is that the remaining attributes are all set to the highest values in one, and the lowest values in the other. The stakeholder is asked to consider each system's utility with respect only to the attribute in question. If the attribute is truly independent, the utility of both architectures should be the same.

5.6.6 Random Mix Interview

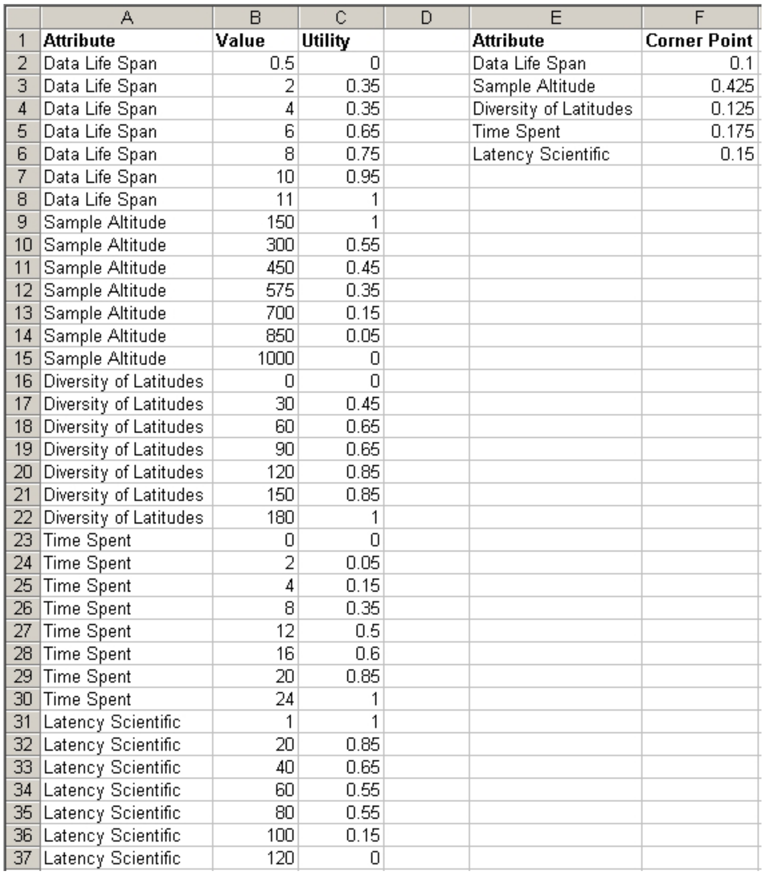
The Random Mix module was developed to provide data to validate the utility functions developed by the MIST system. Users of the system can specify the number of random sets to generate. For each set, the system employs a random number generator which selects a value for each attribute along its range, as specified in the attribute definition stage. The resulting random sets, each representing a random architecture in the tradespace defined by the stakeholder and designers, are presented to the stakeholder as the culminating questions of the interview. Using the same interface as the corner point interview, the system uses the bracketing module to determine the utility of each of the random sets. These utilities are compared to the utilities generated by calculating the utility of each architecture with the utility functions. It is often the case that these utilities do not correlate, leading to the conclusion that humans cannot properly comprehend the multi-dimensional problem of determining their own utility. Accordingly, a system such as MIST is invaluable in more accurately distilling the stakeholder's true utility values [19].

5.7 Output of MIST

The MIST system produces a set of outputs, which can be used to facilitate the knowledge discovery and knowledge dissemination facets described earlier. The outputs described here are for one session of the MIST interview process. These results are provided in a generic format to the users, so that they can be imported into any third-party application for further processing. Work has begun on extending the MIST system to develop knowledge management tools as described in a later section.

5.7.1 Attributes and Interview Reports

Data in the MIST system are stored with respect to each attribute. As the interview is conducted, the responses are stored in the same spreadsheet which contains the attribute's properties as conceived in the attribute definition stage. Every choice is recorded; however, only the indifference points are used to calculate the utility functions and k-values. At any time, the user can choose to have all indifference points collected and stored in a separate table for analysis, as shown below.



	A	B	C	D	E	F
1	**Attribute**	**Value**	**Utility**		**Attribute**	**Corner Point**
2	Data Life Span	0.5	0		Data Life Span	0.1
3	Data Life Span	2	0.35		Sample Altitude	0.425
4	Data Life Span	4	0.35		Diversity of Latitudes	0.125
5	Data Life Span	6	0.65		Time Spent	0.175
6	Data Life Span	8	0.75		Latency Scientific	0.15
7	Data Life Span	10	0.95			
8	Data Life Span	11	1			
9	Sample Altitude	150	1			
10	Sample Altitude	300	0.55			
11	Sample Altitude	450	0.45			
12	Sample Altitude	575	0.35			
13	Sample Altitude	700	0.15			
14	Sample Altitude	850	0.05			
15	Sample Altitude	1000	0			
16	Diversity of Latitudes	0	0			
17	Diversity of Latitudes	30	0.45			
18	Diversity of Latitudes	60	0.65			
19	Diversity of Latitudes	90	0.65			
20	Diversity of Latitudes	120	0.85			
21	Diversity of Latitudes	150	0.85			
22	Diversity of Latitudes	180	1			
23	Time Spent	0	0			
24	Time Spent	2	0.05			
25	Time Spent	4	0.15			
26	Time Spent	8	0.35			
27	Time Spent	12	0.5			
28	Time Spent	16	0.6			
29	Time Spent	20	0.85			
30	Time Spent	24	1			
31	Latency Scientific	1	1			
32	Latency Scientific	20	0.85			
33	Latency Scientific	40	0.65			
34	Latency Scientific	60	0.55			
35	Latency Scientific	80	0.55			
36	Latency Scientific	100	0.15			
37	Latency Scientific	120	0			

Figure 15: Sample table of outputs for one interview session

5.7.2 Single Attribute Utility Functions

The core output for the MIST interview system is the utility functions for each attribute. As described in the MATE process, these utility functions describe the changing utility of an architecture as the value of the attribute changes. When the MIST user selects the Generate Reports button, they can generate a utility function for any attribute. The utility functions are represented as curves with each indifference point plotted and linear interpolation used to determine the function between indifference points. An example of such a curve is shown below.

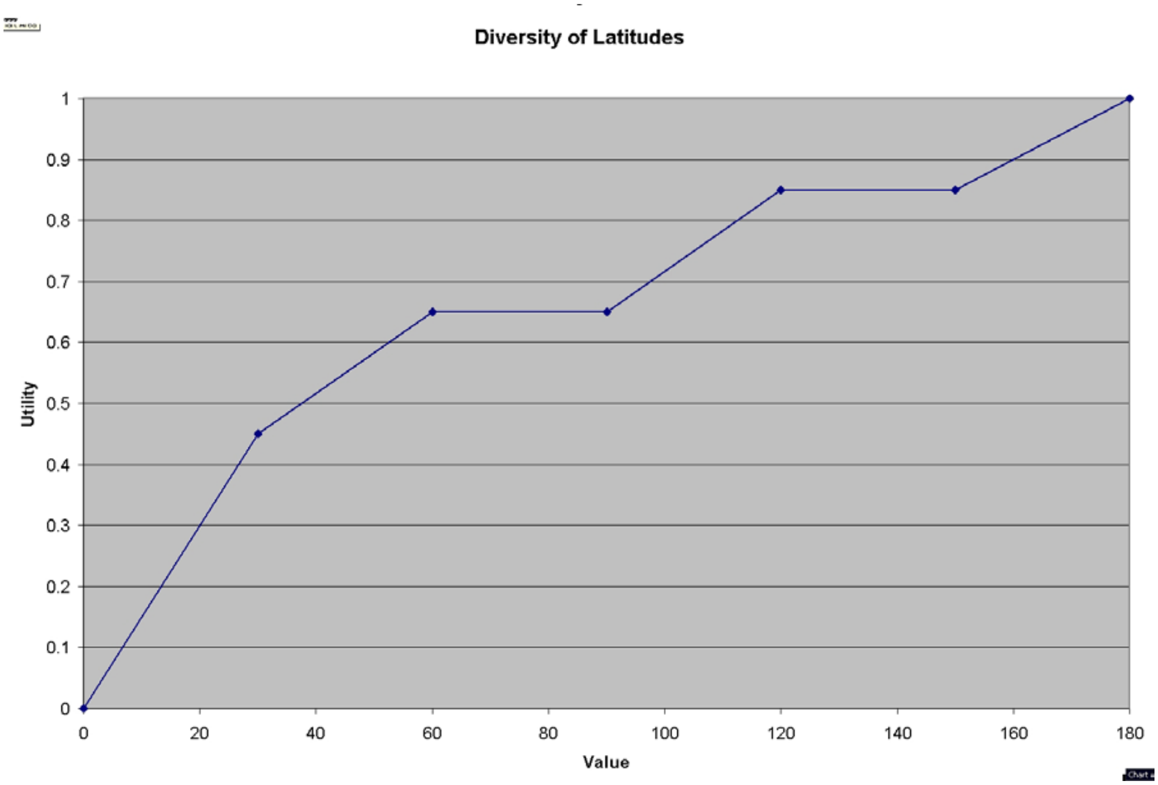


Figure 16: Sample utility curve for single attribute

5.7 3 Multi-Attribute Utility Function

The final output for the MIST system is the multi-attribute utility function, which calculates the utility of an architecture, given values for each of the attributes. The derivation of the actual function is described in the MATE process and shown in the equation below. In this equation, k_i and $U_i(X_i)$ represent the k-value and utility function of a single attribute, and K is the aggregate scalar calculated as described in the MATE process [19].

$$KU(\underline{X})+1=\prod_{i=1}^{n=6}\left[Kk_iU_i(X_i)+1\right]$$

Figure 17: Multi-attribute utility function

In MIST, code modules carry out each step of the derivation on the fly, as shown in the flow chart below. Basically, once the value of each attribute is known, the utility of the system for each attribute can be determined by using linear interpolation between the two closest indifference points. Each attribute's utility is then multiplied by the k-value, and then a normalized product is produced; this represents the utility for the system with the given set of attribute values. This function can be used in conjunction with the knowledge management tools being developed for future versions, as described in a later section of this paper.

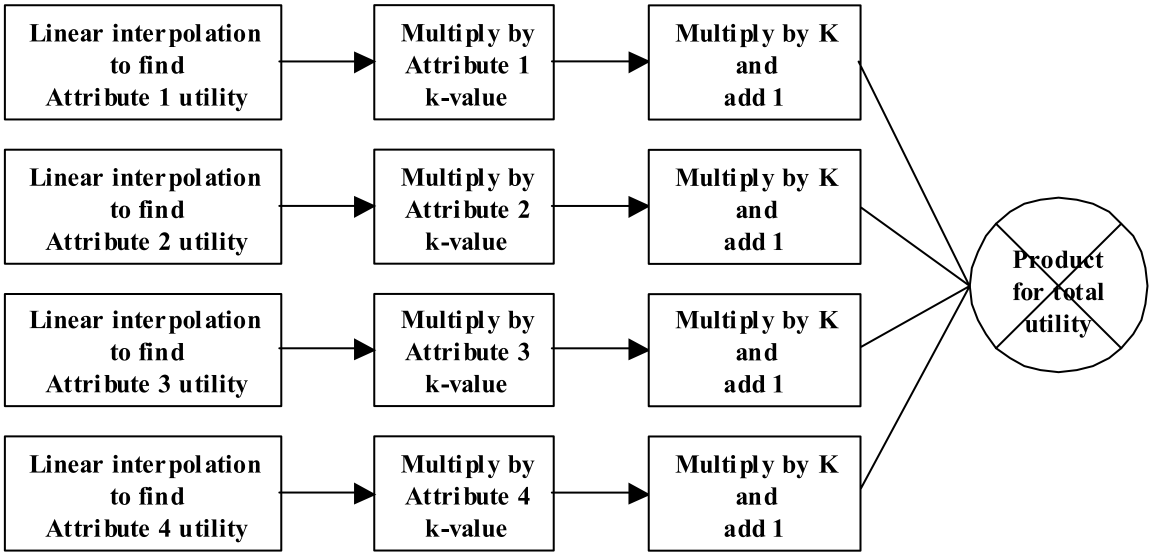

Linear interpolation to find Attribute 1 utility
Multiply by Attribute 1 k-value
Multiply by K and add 1
Linear interpolation to find Attribute 2 utility
Multiply by Attribute 2 k-value
Multiply by K and add 1
Linear interpolation to find Attribute 3 utility
Multiply by Attribute 3 k-value
Multiply by K and add 1
Linear interpolation to find Attribute 4 utility
Multiply by Attribute 4 k-value
Multiply by K and add 1
Product for total utility

Figure 18: Multi-attribute utility calculation by software

5.7.4 Design rationale and history

One of the goals of the MIST system, especially taken in conjunction with the SSPARCy system, is to provide an efficient interface for capturing the history of decisions which are made along the design process. Many of the decisions made in defining the attributes and developing the interview process are crucial design decisions. The rationale captured for major design decisions by the Attribute Rationale module, as described above, is stored in a spreadsheet form very similar in format and structure to the SSPARCy method of storing design history. This allows the SSPARCy suite of tools to be used to process the discussion of attribute definition, thus preserving and improving this important phase of knowledge exchange.

6 Test Implementation

6.1 Project Description

The approach described in this paper was tested in the Spring of 2002 as part of the SSPARC consortium's X-TOS project. The project was conducted with support and inputs of members of The Aerospace Corporation and Air Force Research Laboratory.

In previous years, members of the design team assigned to assess the utility of the proposed system conducted manual face-to-face interviews with the customer. By using the prototype software system, the gathering of utility data was accomplished in a much faster and more efficient manner. The role of the scientific customer in the X-TOS project was played by Kevin Ray, a member of the Air Force Research Laboratory at Hanscom Air Force Base. The knowledge acquisition process was much smoother, as the interview process took one-sixth the time required in previous years. By having the data and questions displayed in a visual manner, it was much easier for the Air Force user to translate the notions of utility, contained within his mind, to the concrete data structure of the MATE approach.

The software system also allowed this interview to be conducted without the need to have all the interacting parties be present in the same geographical location. In fact, the interview session was conducted in stages, as it fit in with the user's personal schedule, and communication between the user and engineers during the interview was made via telephone and electronic mail. The user was also able to take time off to consult with other members of the customer team, since he was working on his own schedule and his own location. The flexibility in terms of both time and location enabled higher quality of responses by the user. In previous projects, the burden of conducting the interview led to the utility information being collected from one stakeholder only, and that too in a piecemeal manner. With the implementation of the approach described in this paper, the utility data are being collected with a higher fidelity, with more stakeholders, and at frequent intervals throughout the design process.

By increasing the frequency of entry of utility data, and by processing of successive streams of data, one can accomplish significant levels of knowledge discovery. This approach has also allowed our design team to engage in knowledge management, and knowledge dissemination activities as the utility information became a continuous driver for the design process rather than being a simple one-time input into the system.

6.2 Issues Encountered

After using the system, Kevin Ray observed that the system was very intuitive and the interview process was much more understandable. Our experience with him and other test users have led to several design refinements in SSPARCy and MATE, and the major issues are mentioned here.

The largest issue revealed by the implementation was the importance of the attribute definition stage, and the need for the SSPARCy and MIST paradigms to be better applied to this stage. The knowledge contained within the discussions leading to the final attributes is valuable and should be preserved. Another change necessitated by the X-TOS implementation was the incorporation of an interview override mode, which raised the issue of the balance of manual input versus automation required by the system. This implementation also underscored the importance of a clearer and networked interface, to allow the designers to better communicate with the user before and during the process. As the system evolves, it will incorporate the ability to observe the state of the interview over the network. Finally, a scheduling system would reduce the amount of knowledge the user must possess about the process itself, and make the system more automated and flexible.

6.3 Lessons Learned

The issues raised above all suggest one very important theme: the effectiveness of knowledge acquisition is heavily dependent on the nature of the software interface. In designing such a system, it is important to consider the type of knowledge being sought, the form in which it exists prior to using the system, and the form in which it is to be disseminated. Much of the knowledge being acquired in this system exists on multiple levels. If users of the system are asked the questions our system is trying to answer, such as what the relative importance of an attribute, they might give a certain set of answers. However, when the interview questions are framed in the certainty equivalent method employed by this system, the user may give answers which are closer to their true preferences. Effectively designing software that can negotiate the boundary between these two levels is the goal of this system. The X-TOS implementation helped demonstrate that this goal is not as simple as it may seem. Future versions of this software must ensure that it creates an environment that not only enables communication across locations but also elicits knowledge which the users may not even be aware of.

7 Extending MIST

The Multi-attribute Interview Software Tool described in this paper reflects the initial phase of an effort to build a complete and valuable knowledge-based tool to facilitate collaboration and informed design processes. This initial phase has built the core functionality to capture the utility of one stakeholder, at one point in time, and for one project. Each of these dimensions must be extended to enable this knowledge acquisition tool to transform into a source of knowledge management and discovery. As the element of time is incorporated into the system, it will be integrated with the SSPARCy system to provide a framework for relating utility and expense information to the design data and rationale of each project. Finally, for any of these software modules to be effective, the system must include a more user-friendly interface and help functionality for all users of the system. Work has begun in each of these areas and will be integrated into the next version of the system.

7.1 Network Mode

As one of the main goals of the MIST system is to enable collaboration over geographic borders, functionality is being developed to allow the interview to take place over a network. This will reintroduce the possibility of real-time interaction into the interview, solving some of the issues described in the test implementation section, without requiring the users to travel.

The network mode has two main roles: observer and user. The user is the person directly interacting with the system, usually the stakeholder taking the interview. Observers are interested parties wishing to monitor and provide feedback on the discussion surrounding the attribute definition and utility, such as other members of the stakeholder's team, designers, and engineers.

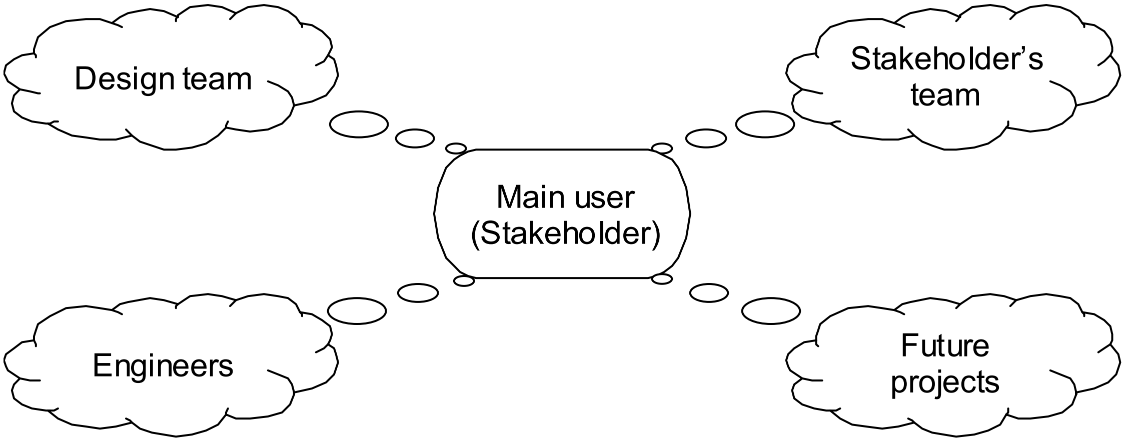


Figure 17: Potential roles for observers.

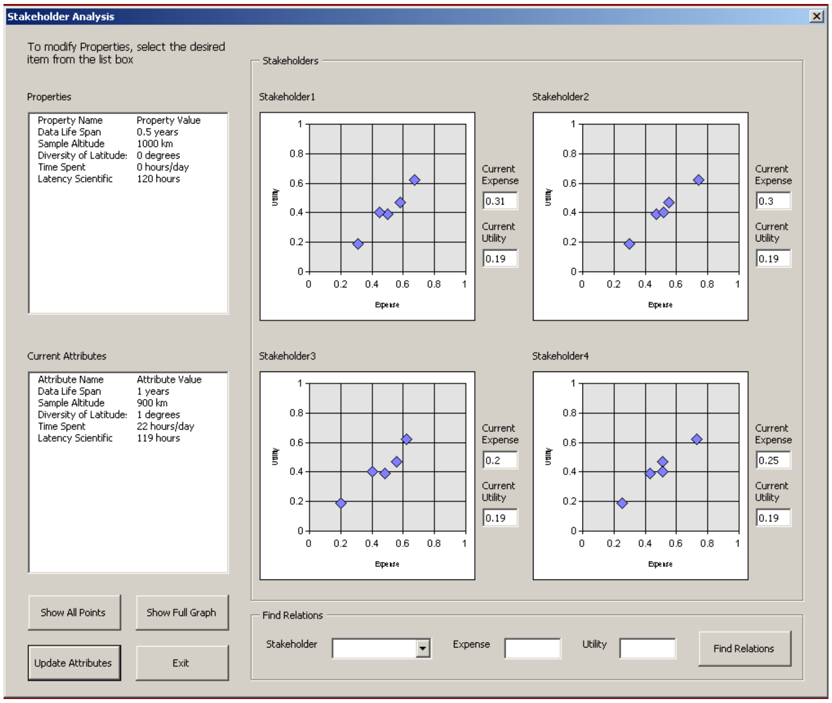
This mode is accomplished by building a library of possible actions which could take place during the various stages of the interview, and then using this as the basis for capturing the flow of the interview. The library includes a means of representation for commands such as opening a form, making a choice, clicking a particular button, etc. An entry in the library has the action, as well as the objects which were acted upon. When the user enables the observation mode, a text file is created and every action performed by the user is transcribed using a command from the library. This effectively creates a log file which can be used either in real-time or later in the process to observe the progress of the interview. The log file is made available in a secure manner to the other members of the team who wish to observe.

The observers run in a mode which does not allow user interaction, beyond being able to step forward and backwards through the interview. An observer selects the location for the log file, and then watches as the actions taken by the user are "played" on their local system. This allows observers to comment on the choices being made, and provide advice on both the interview process and the utility decisions. Future versions will pursue the ability for multiple users to actively interact with the system, as well as integrated means of communication such as instant messaging, embedded within the MIST system.

7.2 Multiple Stakeholders

In order for the MIST system to become an integrated part of the concurrent engineering process, functionality must be developed to enable designers to examine the relationships and dependencies of each stakeholder's utility and expense values to the attributes of a system's design. This would also allow for the discovery of knowledge concerning the optimal attribute set that would maximize utility and expense for the entire stakeholder set.

The figure below demonstrates the main interface to the relationship analysis tool. This facilitates knowledge discovery in two 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. This is done by using the utility equation as described in an earlier section. Secondly, by taking into account the utility and expense value of a stakeholder to a system design, the tool can generate and display all correlating utility and expense values for all stakeholders based on the attribute set that is under consideration.



Stakeholder Analysis

To modify Properties, select the desired item from the list box

Properties

Property Name	Property Value
Data Life Span	0.5 years
Sample Altitude	1000 km
Diversity of Latitude:	0 degrees
Time Spent	0 hours/day
Latency Scientific	120 hours

Current Attributes

Attribute Name	Attribute Value
Data Life Span	1 years
Sample Altitude	900 km
Diversity of Latitude:	1 degrees
Time Spent	22 hours/day
Latency Scientific	119 hours

Stakeholders

Stakeholder1: Current Expense 0.31, Current Utility 0.19

Stakeholder2: Current Expense 0.3, Current Utility 0.19

Stakeholder3: Current Expense 0.2, Current Utility 0.19

Stakeholder4: Current Expense 0.25, Current Utility 0.19

Find Relations

Stakeholder, Expense, Utility

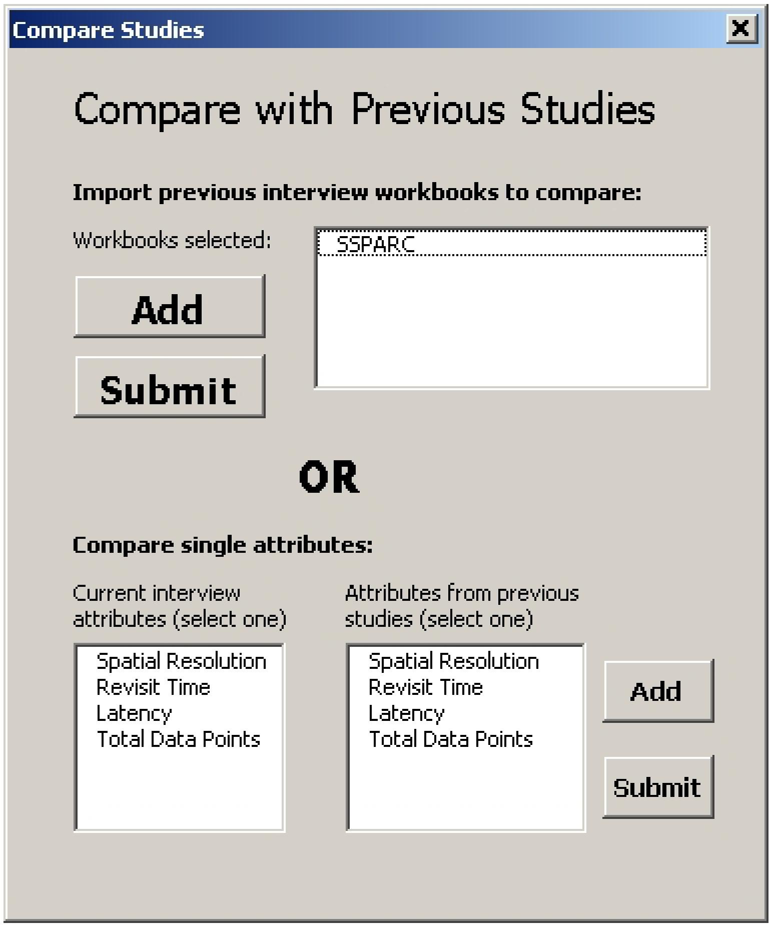
Figure 18: Interface for analyzing multiple stakeholders

With the relationship analysis tool, the effects 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. Relationships between various roles in the design process are better explained through their utility and expense data, and future projects benefit from knowing how previous members in the same role operated under similar conditions.

Such a system simulates real-time feedback for designers in a concurrent process without having to consult every stakeholder with every design change. With a strong knowledge regarding stakeholder dependencies and relations, systems can be designed to optimize utility and expense values for all stakeholders involved.

7.3 Multiple Projects

Data from multiple spacecraft projects can be analyzed to develop models for future spacecraft project direction. Using data from the entire suite of tools, a complete set of design history, rationale and utility can be compiled for each project. An interface is being developed to process data from multiple projects and produce results which help determine similarities and differences. An initial version of the interface is shown below. Note that a user can select to compare either specific attributes, or entire projects. The utility curves for the attributes are compared using the least square method, which determines how closely the utility of one attribute resembles another. As a user discovers similarities to past projects, he or she can use the design history and lessons learned from similar design patterns to make their current design process more efficient.



Compare Studies

Compare with Previous Studies

Import previous interview workbooks to compare:

Workbooks selected:

SSPARC

Compare single attributes:

Current interview attributes (select one)	Attributes from previous studies (select one)
Spatial Resolution	Spatial Resolution
Revisit Time	Revisit Time
Latency	Latency
Total Data Points	Total Data Points

Figure 19: Interface for analyzing multiple projects

8 Future Research

With the base functionality developed in each approach, the next step is to incorporate "intelligent" functionality to process the knowledge captured by each approach. The ultimate goal is to combine approaches into a single, powerful framework for knowledge-based engineering design. As the extensions described in the previous section are developed to provide the functional basis for pursuing these tasks, research must be done on effective methods for incorporating emerging methods into this process.

8.1 Data Mining Techniques

A major motivation for pursuing this goal is the staged implementation of data mining technology to analyze data from multiple perspectives. One of the major knowledge discovery efforts will be to analyze data from multiple design projects, to gain better insights into how the ultimate design is related to the set of spatially and temporally distributed inputs from the stakeholders. This would help to formulate the strategy for having the system provide “automated” baseline designs that might best satisfy a stakeholder's preferences based on past projects and past successes and failures. One aspect of this capability is to manage data over multiple stages throughout the design process. As requirements and situations change throughout the design process, it is important to capture the changing goals and desires of the stakeholders, and examine these data to discover trends in how these changes happen and how the project is affected by these changes.

Another issue to be addressed by data mining techniques is to compare the preferences of similar stakeholders. Relationships between the various roles in the design process will be better explained through the preference data, and future projects will benefit from knowing how previous members in the same role operated under similar conditions.

8.2 Multimedia

Another issue for employing these approaches in a virtual design world is the need for multimedia forms of input to ease communications over geographic boundaries. A text-to-speech engine in the interview system would make the interview process easier for those not familiar with the computer interface, as well as provide a human voice to a discussion which is usually carried out with humans. The other component of this is voice recognition - once again, this would allow the users to communicate with the system in a much more human manner. The functionality has already been developed to allow users from multiple locations to view the status of an interview in real-time. By integrating real-time communication aspects such as chat sessions, audio and video conferencing, the interview session could involve a large number of locations. It would then be feasible to involve more stakeholders in the process, because the need for geographical proximity is reduced.

9 Conclusion

Design environments such as the spacecraft environment are dependant on the successful interaction of multiple teams and multiple stakeholders based in various locations. Capturing the needs of every stakeholder in the process, as well as the details of the major decisions and rationale, provides a means of enabling knowledge transfer between teams without requiring intense interaction. Such a knowledge-based framework can have a tremendous impact on the design process, by providing designers with information which saves time, money and effort. The model presented in this paper provides a means for innovating the spacecraft design process by truly exploiting the underlying knowledge within a traditional design environment.

10 Acknowledgements

We thank fellow members at the Space Systems, Policy and Architecture Research Consortium, especially Daniel Hastings, Hugh McManus, Joyce Warmkessel, Adam Ross, Nathan Diller and Quincy Scott. Additionally, members of the SSPARC IT team who assisted in the development of the software described in this paper are Alexandro Artola, Presley Cannady, Winston Chang, Shane Cruz, Anna Konfisakhar, Carolyn Ng, Frank Reyes, Tara Sainath, Connie Tao and Jason Yeung.

11 References

[1] Gupta, Amar. "A Four-Faceted Knowledge-Based Approach for Surmounting Borders", Journal of Knowledge Management, Vol. 5, No. 4, December 2001.

[2] Ross, Adam. MATE: Defining Roles, Stakes, and Decision Makers. Presentation to SSPARC group, November 20, 2001

[3] Gillam, April. "Vehicles Knowledge-Based Design Environment." Journal of Spacecraft and Rockets: Volume 30: Number 3, May-June 1993. pp 342-347.

[4] Chiu-Chi Wei, Ping-Hung Liu and Chie-Bein Chen, "An automated system for product specification and design". Assembly Automation: Volume 20: Number 3, 2000 pp 225-232. MCB University Press.

[5] Rentema D.W.E., Jansen F.W., Torenbeek, E., "The application of AI and geometric modelling techniques in conceptual aircraft design", in: *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St.Louis, Missouri, USA, September 1998, pg. 928-935.

[6] Keller, Rudolf K. et. al. "Pattern-Based Reverse-Engineering of Design Components." In Proc. of ICSE'99, IEEE, Los Angeles, California, April 1999.

[7] Klein, Mark. "Capturing Design Rationale in Concurrent Engineering Teams." IEEE, January 1993.

[8] Sriram, Ram D. *Distributed and Integrated Collaborative Engineering Environment*. Book to be published, 2002.

[9] J. Aseltine. "WAVE: An Incremental Algorithm for Information Extraction." *In Proceedings of the AAAI 1999 Workshop on Machine Learning for Information Extraction*. 1999.

[10] Wan Jie and Krishnamurty, S., "Comparison based Decision Making in Engineering Design," 1999 ASME Design Theory and Methodology, Las Vegas, Nevada.

[11] Katragadda, Prasanna. Knowledge-Based Systems Interoperability Workshop 9, Concentra Inc.

[12] Mapar, Jalal, et. al. "NASA Goddard Space Flight Center Virtual System Design Environment." IEEE, 2001.

[13] Szykman, Simon, et. al. "Design Repositories: Engineering Design's New Knowledge Base." IEEE Intelligent Systems, 2001.

[14] Sriram, Ram D. "Standards for the Collaborative Design Enterprise Response to Object Management Group", http://www.omg.org/homepages/mfg/mfgppe.htm

[15] QuestMap v3.12. The Soft Bicycle Company, 2000.

[16] A. Brice and B. Johns. "Improving Process Design by Improving the Design Process." *A DRAMA white paper*. QuantiSci, Oct. 1998.

[17] F. Pena-Mora, R. Sriram, and R. Logcher. "Conflict Mitigation System for Collaborative Engineering." *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*. pp. 101-124, 1995.

[18] Xin, Wang and Xiong Guangleng. "Supporting Design Reuse Based on Integrated Design Rationale." IEEE, 2001.

[19] Diller, Nathan, et. al. "B-TOS: Terrestrial Observer Swarm." Final Report, 16.89 Space Systems Engineering, MIT. Spring 2001.

[20] Scott, Quincy R. "SSPARCy: A Software Integration Support and Design Rationale Capture System." Master's thesis, MIT, July 11, 2001.

[21] Delquie, Phillippe. Master's Thesis, Department of Civil Engineering and Technology and Policy, MIT, 1986.