Department of Computer Science
University of Toronto

Technical Report #CSRG-612

Released October, 2011

Submitted November, 2005

Applying the i* Framework, Scenarios, and Prioritization Methods to a Requirements Analysis for Kids Help Phone

Practical Project for CSC2106 Requirements Engineering

Jennifer Horkoff

jenhork@cs.utoronto.ca

Abstract

In applying the i Modeling Framework, scenarios of system usage, and the i* evaluation procedure to the Strategic Requirements Analysis for Kids Help Phone research project, we have made useful discoveries concerning the use of these methods in a medium-sized real-world project. Our findings indicate the need for effective methods to deal with the scalability issues of i*, including alternative visualizations and tool support. The support of software tools is also needed in order for the i* evaluation procedure to be practically applicable to a large scale model. The use of scenarios in conjunction with i* models and i* evaluation has produced an effective way to evaluate and prioritize organizational processes, but such evaluation may have limited usefulness for new system design. Finally, we have developed a method to prioritize system features, represented in i* models, using the i* evaluation procedure and goal prioritization information.*

1. Introduction

The involvement of our research team in a project with Kids Help Phone, a not-for-profit organization offering counseling to Canadian children, has given us the opportunity to apply and evaluate research methods and tools in the field of requirements engineering. Specifically we are using the i* Modeling Framework [1, 2], introduced as a method of capturing the intentionality of social actors in a system domain through the use of goals, softgoals, tasks and resources. In addition, we are applying scenarios as a means to represent requirements in concrete sequences of actions. In conjunction with these methods, we apply a qualitative i* evaluation procedure, to evaluate the effectiveness of scenarios and their components in terms of the quality criteria captured in the i* models. In the application of these methods to our project, we have made useful discoveries, including the creation of a methodology which uses the information in goal models to prioritize system features or processes.

2. Project Background

The Strategic Requirements Analysis for Kids Help Phone (KHP) research project had been active for approximately a year and a half when this particular project began. The aims of the overall project were to help Kids Help Phone understand its objectives and competencies while facing an expansion of its internet services. The project commenced with an elicitation stage in order to understand the organization and the aspects of KHP which had to be maintained regardless of service changes. The information collected originally focused on the organization as a whole, and was recorded in large i* models. These models were used to evaluate the effectiveness of various internet technologies on the goals of KHP, to investigate the utility of viewpoints modeling [3], and to explore the results of applying i* to a real-world project. After presenting some of the new technology possibilities, the organization indicated that their current primary focus was to deal with the overwhelming and increasing traffic volume in their current internet service, Ask a Counselor, before introducing new web-based services.

The Ask a Counselor service uses specialized proprietary bulletin board software to provide an online question and answer counseling service to kids. The posts are entered into a two-tier system, the first tier involving editing and a reply from counselors, and the second tier providing further editing and an approval by counseling supervisors. The question and answer then appears live on the kid's website, with the entire process taking two to four days. The high post traffic has forced KHP to turn off the service, in the form of blocking the receipt of new posts, for days at a time, allowing them to catch up with demand. As a result of these efficiency and workflow problems, the temporary focus of our project became improving the efficiency of the Ask a Counselor service.

To this end, we commenced a more detailed elicitation period focusing specifically on counseling and especially on web counseling. It became clear that the underlying software for the web counseling system possessed many operational flaws which diminished efficiency. In addition, counseling supervisors had many ideas about new features which would improve performance and make the job of web counseling easier. Our intermediate goal became to create a software requirements specification for a replacement system, and to see that system implemented. This paper focuses on the application of research methods in the creation of this specification.

3. Research Goals

The project provided us with the opportunity to apply methods which attempted to use the wealth of information captured in the i* SR model in the production of a specification. Focusing on the application of i*, we wanted to further test the ability of i* to capture objectives for a real-world project of medium size. In addition, we intended to assess the application of the i* qualitative evaluation procedure to such a project. This procedure, based on a similar procedure from the NFR Framework [4], propagates qualitative labels representing levels of satisfaction or denial throughout the graph, using a combination of rules and human judgment. It is typically used to measure and reason about design alternatives in order to make the most appropriate choices. This evaluation procedure will be described in detail in [5]; for example applications see [6, 7, 8].

The use of scenarios in requirements engineering has seen much attention and development due to their ability to describe system usage in realistic situations, aiding in the discovery of specific requirements and the communication of system functionality with stakeholders. See for example [9, 10]. We intended to use the scenario idea in conjunction with our i* model and the i* evaluation procedure. We hoped to represent scenarios of the current and future systems in the i* model, and to evaluate each scenario in order to determine its effectiveness in satisfying organizational goals. The evaluation of the current system scenarios would help guide our design of the new system, and the evaluation of scenarios for the future system would confirm that our design had sufficiently met the criteria captured in our models. We hoped that our findings would include other useful discoveries about the effectiveness of scenario use within this project. Table 1 provides a summary of the research goals of the specification creation project.

4. Project Plan

A project plan was conceived in order to give the project direction and focus, and to produce an estimation of completion time. As we chose not to follow a precise pre-existing methodology, our plan was left deliberately flexible.

From previous work within the larger KHP project we had already acquired data from interviews, group sessions and walkthroughs of various KHP systems, including a wish list of desired system changes and features. From these sources we were able to construct a large i* model depicting the intentionality and functionality of the current phone and web counseling process. This model contained quality criteria which represented both abstract organizational softgoals like "Efficiency", and goals relating to more specific system needs like "Have Bilingual Spellchecking". We included a lower level task structure, placed in rough temporal order, of the actions performed during phone or web counseling. The tasks were divided up into three sections corresponding to the three system users: counselors in tier one, counseling supervisors in tier two and kids on the web. The model contained approximately 485 links and 350 elements, 230 of which represented quality criteria and system goals, the rest of which represented specific tasks in the current system.

Our first task within this project was to perform an analysis of the current system. We intended to extract scenarios of current system usage from the task structures in the i* model, and then to use the i* evaluation procedure to evaluate each of these scenarios in terms of the quality criteria within the model. In the design of the new system, we would use this measure of effectiveness to direct our focus in the creation of new system scenarios. The new scenarios would then be evaluated in the i* model to determine how well they met the system goals, with a reworking of scenarios which proved to be ineffective. Our intention was to then validate our work with the KHP stakeholders by presenting the scenarios, using the feedback we received to make necessary adjustments to our design.

We would then create a specification for the new system which incorporated the relevant information contained in the i* model and the new scenarios. The specifics of this process were left undefined. We would attempt to verify that the operationalizations contained in the specification sufficiently satisfied the quality criteria in the i* model via evaluation. Finally we would hand the specification to a group of undergraduate students who were to implement a first version of the system. Table 2 provides summaries of our original plan of action. The entire process was estimated to take 4 weeks work with full to part-time work of two team members and the potential input of principle investigators throughout all steps.

5. Actual Project Steps

During the application of the original project plan, factors such as time limitations and unforeseen qualities and effects of the methodologies resulted in adjustments to the planned project steps.

5.1. Current System Analysis

Our first steps involved analysis of the current Ask a Counselor System. We extracted scenarios from the i* model of the current system, using the idea of a scenario as a potential sequence of tasks in each of the three system sections. The tasks for each section were analyzed to determine which tasks were conditional, in that they did not always have to be performed in a scenario. The tasks were converted to a list, with indentation based on the super and sub-task structure in the i* model. An attempt was made to explicitly list the conditions for optional tasks with the textual version of the scenario. The output of this activity was a set of “meta”-scenarios for each major section of the system, or a list of all possible tasks that could be performed. From this meta-scenario one could extract numerous scenario instances by choosing subsets of the optional tasks. One could view the meta-scenario as a use case which stores information on all possible use case paths. As the list of all tasks for one of the main sections of the system contained 116 tasks, 73 of which were optional, the specific extraction of all possible scenario instances was unfeasible by hand. Figure 1 and Figure 2 show a section of the task decomposition for the current system and the corresponding meta-scenario section, with brackets representing optional tasks and parenthesis describing conditions.

Before starting the evaluation of the scenarios in the i* model, it became apparent that the current model was lacking potential contribution links. These links had been added to the model in a haphazard manner. When a new element was added to the model, the modeling group would attempt to determine if this element had an effect on other elements present. However, as there were approximately 200 elements in the model, exhaustive searches of possible contributions were never performed. Although the potential contribution links can likely never be complete in a model of such size, the quantity and quality of such links has a potential effect on the results of the evaluation procedure. Therefore, before evaluation, one of our group members spent approximately 3 to 4 hours adding roughly 40 contribution links to the 485 pre-existing links. At this point we thought the model was sufficiently complete enough for a relatively accurate evaluation.

The evaluation of scenarios in a model representing an existing system had not been attempted before, and required some adjustments to the ideas involved in i* evaluation. The i* evaluation procedure was originally intended to evaluate the effects of potential design choices on non-functional requirements represented as softgoals. These effects would be used to make trade-offs and decide amongst operationalizations. In this project we needed to analyze the effectiveness of a current, and not potential, system. What we were evaluating was the effects of performing or not performing optional tasks. This turned out to be more complicated than originally envisioned, as the effect of a particular optional task can differ depending on the conditions. For example, take the task “Reject Message if Necessary”, as shown in Figures 1 and 2. This task is considered optional, as messages are only rejected in some scenarios. However, the effects of rejecting or not rejecting messages will vary depending on the conditions of the rejection, divided into four cases: the message is rejected with the correct conditions, the message is rejected under incorrect conditions, the message is not rejected when the conditions were correct for rejection, and the message was not rejected when conditions for rejection did not apply. Each of the four situations could produce a different set of contribution links, for example the second and third case will likely cause problems for system supervisors, while the first and second cases will likely annoy users (kids), and the third case may have a negative effect on user anonymity. In the context of this project, taking into account resource limits, the evaluation was performed focusing on the first and last cases, assuming that users followed the conditions correctly.

As the number of scenario instances for each section of the system was enormous, instead of evaluating the effectiveness of each individual scenario we choose a subset of scenarios which represented related sections of optional tasks. For example there was a scenario for moving posts, a scenario for rejecting messages, a “minimum” scenario where no optional tasks were performed, making a total of 25 scenario instances over the three sections of the system. Each of the scenarios was evaluated manually against the approximately 200 quality criteria in the model. However, a maximum of only 62% of the quality criteria received values from these evaluations.

It became apparent that a way to make relative comparisons of the evaluation results was needed. In previous evaluation examples with smaller models, a comparison was done visually by simply examining the results of the evaluation on the quality criteria. In this case, the large amount of criteria made a visual comparison difficult. Therefore, we created a quantitative score which assigned values to each potential evaluation result and added the results for a scenario together. Although the score is quantitative, as it is based on qualitative evaluation data, the intention was to provide a relative measure of scenario effectiveness, as opposed to a definite or authoritative measure. The scores for each of the current system scenarios, including a brief title for each scenario, are given in the first two columns of Table 3. The results indicate that the scenario which has the most negative effect on quality criteria is the editing of the post in the first tier.

Although this method may offer insights into the effectiveness of particular scenarios, the scores are based only on the quantity of criteria touched by the scenario tasks, and not on the relative importance of each of those criteria. In a related project in the same domain, we collected prioritizations for the subset of the quality criteria relating to items on a new system wish list. We decided to incorporate this data into the i* model and the evaluation score. The prioritization data consisted of the allocation of system wishes into five priority buckets by six research team members. We intended to receive similar prioritization information from KHP stakeholders, but were unable to acquire this information in time. We used this information to calculate an average bucket for each wish, and assigned these buckets to the model elements corresponding to the wishes. Elements not involved in the prioritization exercise were assigned a default bucket. Each bucket was given a corresponding numerical weight, and these weights were incorporated into the evaluation score. The scores and ordering which incorporate this data are shown in the last two columns of

Table 3.

5.2. Future System Analysis and Design

In designing the future system we intended to redesign the current system scenarios based on their performance in terms of the quality criteria and knowledge needs captured in the i* SR models. However, it became apparent that knowing precisely how to modify scenarios was difficult without making reference to the elements in the i* model. In order to improve the effectiveness of a scenario, we needed to understand why it was problematic. If reference to the i* model was necessary, then instead of adapting old scenarios to better fit quality criteria, we believed it was more effective to operationalize the criteria and build new system scenarios from these operationalizations. This way we could attempt to sufficiently ensure that quality criteria was sufficiently met, by systematically considering each criteria, and brainstorming ways in which it could be partially or completely satisfied.

The process of considering each of the approximately 230 quality criteria for operationalization produced about 200 new or modified model elements, although some elements were repeated across more than one system section. From the existing 121 operationalizations in the current system model, we removed 43, with the decision that their functionality was replaced by one or more of the new operationalizations. As a result of these additions and further changes, the final element count for the future system model was approximately 520 elements with 760 links. Our next step was to take the roughly 200 new elements with the remaining 78 operationalizations from the current system and place them in rough sequential order to form new system meta-scenarios. In this manner, we designed the new system using both a top-down and bottom-up method, using the scenario tasks remaining from the current system model as well as new tasks derived from quality criteria.

Many of the operationalizations did not fit well into the context of scenarios for the major sections of the system, as they represented states or actions in isolation. For example "Archive posts with flexible time boundary" or "Make reply space adjustable". Instead we focused on scenarios for the three complex areas of the system. Of the 200 new or modified operationalizations, about 135 were placed in such scenarios.

Our next step involved analyzing these scenarios in isolation from the i* SR model, with the intention of finding potentially missing or inappropriate actions. During this phase two of our group members, over a period of roughly five hours, changed or moved 28 tasks, removed 7 tasks and added 63 new tasks, resulting in a total of 245 actions. Of the 63 new tasks 35 were related to a single feature, case files, which had somehow avoided operationalization in the i* SR model. The new tasks added to the scenarios were then added to the task structure of the i* SR model, to keep the model consistent with the scenarios.

Our intention was next to evaluate the new system scenarios against the quality criteria in the i* SR model. As we started the evaluation, it because apparent that the information gained via this step was not the critical information that was needed. Evaluating scenarios provides a way to compare the effectiveness of processes, isolating action sequences which may be problematic. This information could be useful to new system design by indicating sequences of actions which perform poorly, indicating that new design alternatives for this process should be considered. However, it was apparent that there were far too many new system tasks or features to be reasonably implemented in one term, as was the intention. Before we analyzed the effectiveness of individual system processes via scenarios, we needed to determine which tasks within a scenario would actually be possible via our design choices. As a result, we changed our project plan and instead used the i* evaluation procedure to determine the effectiveness of each individual feature, leaving the new system scenario evaluations for future work.

In order to proceed, we had to determine what "features" were present in our high level design of the new system. We examined the scenarios tasks and determined which tasks were necessary to provide a base system functionality (referred to as "base" tasks), and which tasks were "additional" in that the system would function sufficiently without them. In choosing the base tasks we tried to provide functionality which was at least equivalent to the current system, with the addition of some simpler features that we felt were critically needed, such as the ability to claim a post for replying. Of the 245 scenario actions, 100 were considered necessary or base actions, with 145 classified as additional. These 145 additional tasks were grouped together into 37 sets of related functionality, creating 37 additional features. An examination of the operationalizations not included in scenarios produced only a few of the 37 features, as most features involved at least one task in a scenario.

The first evaluation of the i* model involved the assumption that all of the additional features would be implemented, attempting to confirm the effectiveness of our operationalizations. The next evaluation was to determine the effectiveness of implementing only the base functionality, assuming that no additional features were implemented. As with the evaluation of the current scenarios, we incorporated the individual goal priorities into the evaluation results and used a relative numerical score. These evaluations resulted in a score of 197.5 and -117, respectively, leading us to believe that our operationalizations had been effective. The significant different between these scores can be at least partially explained by the nature of the i* model. The goals and softgoals within the model reflected strongly the stakeholder's desires for new system features. In contrast, the current functionality which performed satisfactorily, was not stated explicitly, and therefore not included in the model. In other words our model was built upon statements such as "It would be nice if we had this"; as opposed to "We already have this and it works well". This effect also helps explain the poor coverage of the quality criteria in the evaluation of current system scenarios.

In evaluating the effectiveness of the optional features, we had the option to evaluate the tasks associated with these features in isolation, or in the context of the satisfaction of the base set of features. Evaluating the features in isolation would produce results with a higher variance, as the overall results depend only on the feature-specific tasks. Despite this, we decided to evaluate the features in the context of the base tasks, with the reasoning that contributions would be evaluated in terms of how much difference they make in conjunction with base feature contributions. For example, a positive feature contribution to an element which is already satisfied under the base functionality is less useful than a positive contribution to an element which is denied under the base functionality. If the features were evaluated in isolation, these positive contributions would be given equal weight.

As model evaluation must currently be performed by hand, we felt it was infeasible to evaluate all 37 additional features. We compromised by choosing a subset of 28 features which we felt were the most likely to be implemented, postponing features which we felt were controversial or whose payback was obviously not worth the cost. We then proceeded to evaluate the effects of satisfying the tasks associated with each of the 28 additional features. The difference between the resulting scores and the score for the base functionality was used to create a priority of features as shown in Table 4.

The next step was to create the new system specification based on the derived base and additional features. Adapting the outline recommended by the IEEE [11], we included sections for both necessary and additional requirements, grouped by feature. We also included information not typical to the IEEE outline, based on ideas contained in [12]. Each feature contained a list of related goals, in order to provide the developers with the intentions behind the features and a better understanding of the proposed system domain. Two sets of meta-scenarios were provided, with base features only and with additional features, for each of the system sections. The model tasks which were grouped under certain features were listed with feature descriptions, and these tasks came with a reference to their position in one or more scenarios. We envisioned that a developer could look up the tasks associated with a feature within a scenario, in order to gain a better understanding of how this feature fits into the overall functionality of the system. The priority information shown in Table 4 was included with the additional features in the specification to give the developers a relative comparison of feature importance.

It is our suggestion that the developers of the system use the priority scores, along with an estimation of implementation cost, to plot the features in terms of cost and importance, as is done in the AHP prioritization method [13]. This plot will produce an ordering for implementation.

Table 5 contains a summary of the actual project steps, and Figure 3 shows the evolution of the project artifacts. The actual time to complete the project was 7 to 8 weeks of full-time to part-time work for one to two people.

5.3. Validation of Results

Due to the presence of multiple research questions and multiple project outputs, there are several results which should be validated. Ideally, we would like to validate the requirements in the system specification, the system scenarios, the goals in the goal model, and the priorities of the optional features with the stakeholders. We would also like to validate our methods by evaluating the success of the overall system. We would like to determine the effectiveness of the specification contents for developers, particularly whether the inclusion of goal and scenario information was helpful.

Due to the limitations of time, we will have to postpone the validation of the overall success of the system and the effectiveness of the specification to future work. However, we were able to orchestrate a partial validation of our requirements, goal models, and new system scenarios with KHP stakeholders. A full validation of all items was not feasible, as it was not reasonable to expect a charitable organization to commit enough time to review a 70 page requirements specification, 485 links, 350 elements of an i* model, and 245 actions within scenarios. Instead we arranged two two-hour group sessions with stakeholders, one session with counseling management, and another with counselors. During these sessions we presented abstract versions of the new system scenarios for all of the major sections of the system. With these actions was associated "motivations and concerns" expressing quality criteria from the i* model which was positively or negatively effected by a task, according to our model. We also presented the 28 optional features, giving a brief description of the feature, how they may fit into the system scenarios, as well as motivations and concerns. Instead of presenting a numerical score representing the priority with the feature, for simplicity, we divided the 28 optional features into 5 categories of priorities, and presented these classifications with the optional feature. As well as making notes on stakeholders feedback, we gave out a handout listing each scenario and optional feature, asking questions about the accuracy and completeness of the scenarios, motivations and concerns; the usefulness of the optional features; and the accuracy of the priority levels.

Unfortunately, we were only able to acquire completed handouts from four stakeholders: three supervisors and one counselor. Although this sample is too small to completely validate our methods, analyzing these results is still interesting. We have counted the number and the length of priority category adjustments by the stakeholders, relative to the prioritizations with and without individual goal priorities, and averaged the results together, shown in Table 6. A relative score, assigning higher weights to higher adjustments, has also been calculated, as shown in the bottom row of Table 6. Taking into account these results, we have created a new ordering of optional features for our specification

6. Analysis of Process and Results

6.1. Large Scale Application of i*

The application of i* to a real world problem of medium size reveals several difficulties, the most prominent being the scalability of i* to larger domains. The graphical layout of our large model made it very difficult to work with when following links and finding items. See Figure 4 for a high level view of the final new system i* SR model. The visual complexity of the model make it hard to reach sufficient completeness in terms of contribution links, as each node should be compared with every other node to find potential contributions.

Actors and the dependencies between them are fundamental constructs in the i* Strategic Rational (SR) model. However, one may be able to see that our SR model does not use actors. This can be partially attributed to the small number of potential actors and roles in our domain, but the primary reasons for the exclusion of actors are related, to scalability. Physically laying out 500 elements in the circles used to represent boundaries of an actor's element ownership would cause a considerable increase the size it occupies. In addition, elements are often shared amongst many actors in the organization. For example, goals such as "Reduce Difficulty of Software Use" or "Add Functionality to Current Search" are shared by both counselors and supervisors. To show that an element is shared between users in an i* model, one could create a common role that represents shared intentionality, or one could repeat goals in multiple actor boundaries. The repetition of elements is obviously problematic for graphical reasons, as the i* model is already very large without such repetition. The creation of an additional actor shows more promise, but in a model with many actors, there could be a very large number of potential shared intentionality actor combinations.

Work has been done to suggest various tools to help with i* scalability problems. For example, the use of "slices" to show bottom-up or top-down linked paths of elements [14], and the use of "business services" as collapsible related segments of i* models [15]. Other tactics may be employed to reduce the size of the i* model, by either performing a careful pruning of the model, removing elements which are no longer relevant or important, or by making models which focus only on one specific topic or concern, such as “Efficiency”. However, alternative ways of visualizing i* models may be more effective, for example, tables of related elements, such as in a database schema. Although ideas to address i* scalability are present, the essential missing element is the implementation of these ideas into publicly available and easy-to-use tools. There is currently research software available for i* models, such as OpenOME [16], or REDEPEND-REACT-BCN [17], but, to our knowledge, these tools are not currently able to adequately address scalability issues.

Despite the problems apparent with the application of i* to real world problems, we can see rewards that lead us to believe that the benefits of i* still outweigh the costs. Although we were able to learn a lot about the domain through our interviews, group sessions and system walkthroughs, we feel that the process of synthesizing this information to extract important elements and create an i* model provided us a better understanding and a better retention of domain information. The increase in domain knowledge derived by the creation of models is hard to measure or quantify, but it can be seen that the process of spending many hours reviewing transcripts of domain interactions will produce a greater understanding of the domain, no matter the output of the sessions.

In addition, the recording of stakeholder goals, actions, and the relationships between them, provides a way to capture such knowledge in a synthesized form. Although the model is large and difficult to read, we believe it is still a preferable way to store domain knowledge, in comparison to the hundreds of pages of transcripts recording our interactions with stakeholders. The ability of i* to store important domain information is especially valuable in a project such as this which is stretched out over a considerable period of time. Although certain group members were present for the original interviews, we would be unlikely to remember some of the important details over a year later. However, by trusting the group's ability to extract such important information and store it in the i* model, we save the effort in having to go back and re-read the transcripts. In addition to storage, the production of an i* model can facilitate methods of useful analysis, explored in the following sections.

6.2. Large Scale Application of the i* Evaluation Procedure

Prior to this project, the i* evaluation procedure had only been applied to models with roughly 100 elements. To facilitate the evaluation of over 500 elements in this project, the model was divided into layers. The fundamental benefit of such layers was visual, enabling the evaluator the ability to better see and pick out the individual contribution links and to ensure all values were propagated. However, combining the evaluation results of the layers turned out to be problematic, as it was possible that an element would have many labels coming from the same source, via a separate label from each layer. It became necessary to trace back the immediate children of the element to determine exactly how many labels the element was receiving, and from where. For elements with many contributions it became necessary to make a textual list of contribution links, as they were graphically very difficult to count, see Figure 5 for an example. In retrospect, it may have been more effective to use an alternative system of layering.

In applying human judgment in the evaluation procedure, it is not only the number and type of contributions to a node which should influence the final node label, but also the specific sources of these contributions. For example, an evaluator could decide that the contribution of "Reduce Difficulty of Software Use" to "Efficiency" is stronger or more important than "Improve Typing Skills" and this decision will be reflected appropriately in the decision for the final label of "Efficiency". However, when one is making a decision based on 15 contributions, such as the example in Figure 5, it is much more difficult to incorporate the sources of the labels into evaluation judgment.

A significant difficulty with the large scale application of the evaluation procedure was the time it took to complete. The evaluation of the new system features, for example, took one person more than a week of full time work.

As a result of these findings, it is clear that there is a critical need for adequate tool support. Although the human judgment component of this qualitative procedure makes it impossible to fully automate, the procedure can be partially automated. Information required to make judgments can be made more easily available, perhaps in an automatically generated table with names and contributing labels of children, (a more detailed version of the table in Figure 5). As the evaluation procedure often involves many evaluations for a single model, the implementation of multiple layers for different evaluations would be helpful. Automatic comparison of evaluation layers, a type of "diff" function, would make the job of analyzing the varying effects of operationalizations much easier. In addition, the ability to automatically migrate evaluation results to some sort of spreadsheet or other tool which enables analysis would save a significant amount of time.

In order to save time, the evaluation of each individual feature and scenario was performed by making modifications to the "min" evaluation, which evaluated the satisfaction of only the required features or actions. Therefore, having the ability to start a new evaluation by making small changes to a previous evaluation would also save time and effort. In addition, it became apparent that changes to a small number of operationalizations in a model of this size often did not have a significant effect on the overall evaluation, especially to higher level goals. For example, in Figure 5, it's apparent that changing the contributions of one or two of the children may not have any effect on the final label judgment. This effect, of small changes being "drowned" in the larger model, relates to the trade-offs between evaluating features or tasks on their own versus evaluating them in the context of base features or tasks. Despite the results, the original decision to evaluate the actions or features in the context of other features still seems reasonable. As this "drowning" effect is a direct result of the large size of the model, any method which makes the model smaller may mitigate this effect.

As a result of our exclusion of actors in the i* SR model, we have lost the potential to evaluate elements in relation to a certain actor. The questions posed by the evaluation procedure would then move from "What are the effects of these operationalizations?" to "What are the effects of these operationalization for a specific actor?" This level of analysis may have proved very interesting for our project, but we unfortunately lacked the time or tool support to accomplish it.

In this type of qualitative evaluation procedure the qualitative labels are not weakened over a long path of evaluation. As the contributions get further and further away from the source, their effects often seem less sensible. For example, in our current system model "Post link in reply" has an indirect positive effect on "Emotional Connection" by tracing through contribution links. It is clear that this issue becomes more important in larger models when long contribution paths are possible. As a potential solution to this problem, the evaluation procedure could be adjusted to include the idea of propagation length in human judgment. At a certain point it can be judged that a specific contribution is no longer applicable to the recipient elements.

Despite the apparent difficulties with the large scale application of the evaluation procedure, it is clear that some sort of evaluation method is needed in a model of this size. Answering questions such as "What are the effects of a certain feature on the quality criteria of the model?" or "Which feature is more effective over all" is very difficult without some sort of systematic procedure.

The possibility of applying other evaluation methods, such as those described in [18, 19, 20], was ruled out at the beginning of the project, due to the perceived need for human judgment due to the proliferation of softgoals in the model. However, it is possible that a fully automated procedure such as the one described in [20] may have eliminated some of the problems experienced. A comparison of these procedures using the model in this project was not feasible due to time restrictions.

6.3. The Use of Scenarios

In this particular project, we had success in extracting scenarios from the task structure of the current system model. However, due to the limitations of the i* model syntax, these scenarios did not contain information on the conditions necessary for certain tasks. It is also possible that other i* models may not contain such a detailed description of task structures, added explicitly to our model in order to evaluate workflow.

Creating scenarios for the new system from i* operationalizations also shows promise. As the tasks in the scenarios were directly derived from the quality criteria, we could be relatively confident that the scenarios produced by ordering tasks adequately addressed the goals in the model. In spite of this, only 68% of the new tasks and goals fit well into a sequence of action. When examining the new system scenarios, almost 40% of the 245 final actions were added to, or modified from the original model operationalizations. This indicates that although extracting scenarios from i* models may be a useful first step, further checks and iterations may be needed in order to ensure adequate scenario correctness and completeness.

Regarding the usefulness of the extracted scenarios, in the current system the goal of evaluating scenarios was to determine which scenarios were detrimental to system goals and needed to be focused on in redesign. As it turned out, this information received only a small amount of attention during the new system design, as the focus was on the design of a new system and not the redesign of an existing system.

Despite the relative ineffectiveness of the current scenario analysis for this project, this method has potential for application in different situations. If the goal of the project was to redesign or make changes to a current system instead of building a new one, the ranked analysis of current operational processes provided by this method would be useful to direct the focus of system redesign. However, well-defined methods to deal with the evaluation of conditional tasks are needed in order to effectively deal with the complexity created by the possible combinations of actions and conditions.

As described, the evaluation of new system scenarios was abandoned in favor of the evaluation of potential new features. It is likely that the circumstances of this project, having many potential features and only a small amount of implementation time, are common in software projects. However, the evaluation of the new system scenarios may still be useful for the design of processes for the new system. Once the set of new features have been determined, scenario evaluation could be used to determine which sequences of potential tasks are the most effective in achieving goals, including which tasks should be optional or mandatory.

While reviewing the new system scenarios extracted from the i* model, we were able to come up with 63 new tasks, indicating the usefulness of the scenario format as an effective tool for brainstorming. It is possible that the team members, upon further examination of the model, or upon brainstorming without the aid of any tools, may have been able to come up with most or all of these tasks, but it was the consensus of the group members involved that that the use of scenarios made this process easier.

The evaluation of scenarios in i* models raises the question of the value of evaluating scenarios versus evaluating individual features or tasks. Although the evaluation of features seemed the most immediately useful for this project, there still appears to be value in evaluation of both scenarios and tasks. Scenario evaluation provides a view of the effectiveness of a process, while the evaluation of a task or feature provides a view of the effectiveness of a design choice or individual action. The views provided by each type of evaluation are complementary, and could be used together to produce a useful analysis.

One of the purported benefits of using scenarios in system design is the ability to present concrete sequences of actions back to the stakeholders for a validation of design. In our case we felt that the meta-scenarios describing all possible actions were too long and complex to present. We instead presented high level scenarios of common actions and alternatives, extracted from the complex scenarios. The use of scenarios in this context seemed to provide a useful means to present and validate the base functionality of the new system.

In the validation of the optional features, including their priorities, we had the option to present each feature in the context of a scenario, or via a brief description of its functionality. In light of the fact that we had approximately 60 to 90 minutes to present and receive feedback on 28 optional features, we opted to present the features via brief descriptions. We were also concerned about the potential repetitiveness of presenting scenarios for the optional features. As the features occurred in the three sections of the system we were focusing on, we would likely be presenting the same three scenarios multiple times, adding only a few new tasks each time for the feature. However, even though we did not explicitly use scenarios to present the optional features, the contextual information we gained from placing these features into a concrete set of ordered tasks was occasionally embedded in the description.

6.4. Prioritization of Features via i* Evaluation

The combination of i* models, model evaluation and feature extraction offered a method to prioritize the importance of optional features based on model quality criteria. In addition, the incorporation of individual goal prioritization in the relative prioritization scores of features provides an effective way to represent the relative importance of goals in the prioritization.

A comparison of the prioritization results with and without individual goal prioritization information, with the results of the validation sessions should help to shed some light on the effectiveness of feature and goal prioritization. As the results from Table 6 show, 71% and 64% of our prioritizations were either correct or off by only one prioritization level, for the results with and without individual goal prioritization, respectively. Despite the small size of our sample, we believe that the results for our prioritization method are encouraging. It is also interesting to note that the prioritization incorporating individual goal prioritization seemed to perform better when compared to stakeholder opinions.

It is necessary to consider factors, other than the effectiveness of our methods, which may have affected our results. The quality or completeness of the i* model, and not necessarily the quality of our prioritization method, could cause inaccuracies. In addition, presenting the stakeholders with our prioritization likely had an anchoring effect [21]. If we had asked them to prioritize the features without first looking at our ranking, there would likely be an increased difference between their prioritizations and the results of this method. It is also relevant to note the large gap in time, 6 to 17 months, between the elicitations for our i* models and the collection of the prioritization feedback. In addition, the individuals from whom we elicited the information for our i* model were not the same individuals who provided us feedback on our prioritization. The differences in priorities could be attributed to personal differences or a general shift over time of organizational goals not adequately captured in our model.

6.5. Creation of a Specification Using i* and Scenarios

In converting the products of our analysis into a specification we made interesting observations. The task and feature information derived from the model criteria was often not of sufficient detail to translate directly into requirements. Specific details, as well as information concerning conditions, had to be produced at the time of creating the specification. There were also necessary requirements that were not derived from analysis artifacts, but had to be inserted based on the domain knowledge of the specification authors. Out of the 56 base and optional features and 222 requirements included in the specification, at least two features and 13 requirements were specifically identified by the authors as not being derived from the scenarios or goal models. These results indicate that although these artifacts are useful tools in crafting a specification, human intervention and ingenuity is still required. We can also speculate, as the authors of the specification were heavily involved in the modeling exercises, that the domain knowledge acquired through these exercises may have helped them acquire the knowledge necessary to "fill in the gaps" for the specification.

7. Related Work

Methodologies which incorporate the knowledge captured in i* models into the software development process have been previously described. In the Tropos methodology [22], i* models are used to assign responsibilities to the new system. The details in the model are then decomposed to create an agent-oriented design. Further work has incorporated model checking into Tropos, by adding formal temporal information to entities in i* and using them to derive scenarios which satisfy or do not satisfy requirements [23]. In this method scenarios are represented as formal representations of entity creations and conditions, captured at discrete time intervals, different from the relative ordering of tasks used in the current project.

Martinez et al. have developed a methodology to convert constructs of an i* model into formalized specification statements by converting them into formal structures in the KAOS framework [24]. Although this method shows potential for the conversion of i* operationalizations into requirements statements, as our i* model made extensive use of softgoals, not used in this method, this methodology was not applied.

In the RESCUE method different streams of modeling, including activity diagrams, i*, and use cases/scenarios, are created in parallel with stages of synchronization [25]. These artifacts are then used to create specific requirements. Further work in this method describes the use of patterns to convert structures in an i* SD diagram to textual requirements, focusing mainly on dependencies between actors [26]. As our SR model did not contain actors or dependencies, it was impossible to apply the RESCUE method's SD patterns.

The projects associated with the RESCUE methodology serve as examples of an application of i* to a substantial real-world problem. The KHP project's previous work with viewpoints and i* provides us another example [3]. Other work which has applied i* to real world projects include the SKwyRL project, using catalogues of patterns in i* [27], an application of i* to hospital bed management and to steel making [28], an application of i* in agriculture [29], and in projects involving automatic software creation from models [30]. Several of these projects are consistent with our findings in identifying the scalability of i* models, particularly in SR models, as a problem.

Much attention has been paid in the literature to the potential relationships between goals and scenarios. Liu and Yu suggest combining scenarios with i* models using Use Case Maps [7]. In this work, alternative tasks in i* models are associated with a scenario in order to better understand choices in functionality. Our usage of scenarios and i* differentiates itself from this work by explicitly representing the sub-tasks of each scenario in the i* model, showing their individual contributions to quality criteria.

Work by Kavakli et al. has suggested the operationalization of scenarios from goals; developing the goals and scenarios in an iterative process [31]. The GBRAM method, described in [32], focuses on the refinement of goals to system requirements using obstacles and scenarios. This method uses the scenarios derived from goal models to validate the goals, evaluate alternatives, identify new goals, and elaborate requirements. The work of Rolland et al. focuses on the iterative discovery of goals from scenarios [10], developing "Requirements Chunks", a pairing of a single goal and a scenario.

In the work of Santander and Castro [33] a methodology is provided for deriving use cases from i* models. They recommend using the structures of the i* SR models such as decomposition and means ends to derive the details and sub cases of use cases. This is similar to the steps in our project where we extract scenarios from the current and new system i* models.

There has been extensive work on the prioritization of requirements, such as the work of Karlsson and Ryan [13]. Using goals in such prioritizations is explored in [34], where user goals, skills and preferences are used to rank alternatives in customizable software.

8. Conclusions

The project has produced interesting findings concerning the application of i* and an i* evaluation procedure to large projects, namely the need for tool support and methods to address scalability. We feel that we have been successful in creating a specification which incorporates the artifacts of analysis, including scenarios and the i* model. The application of scenarios has been met with mixed success, but shows overall promise for the evaluation of organizational processes, capturing of contextual task information and presentation to stakeholders. In addition, our method to prioritize features and processes using i* evaluation and goal prioritization has potential to provide a means to utilize goals in prioritization.

References

[1] E. Yu, "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering", *Proceedings of the 3rd IEEE Int. Symp. on Requirements Engineering (RE'97)*, Washington D.C., USA, Jan. 6-8, 1997, pp. 226-235.

[2] E. Yu, *"Modelling Strategic Relationships for Process Reengineering*", Ph.D. Thesis, also Tech. Report DKBS-TR-94-6, Dept. of Computer Science, University of Toronto, 1995.

[3] S. M. Easterbrook, E. Yu, J. Aranda, Y. Fan, J. Horkoff, M. Leica, and R. A. Qadir, "Do Viewpoints Lead to Better Conceptual Models? An Exploratory Case Study", *13th IEEE International Requirements Engineering Conference (RE'05)*, Paris, France, Aug 29-Sep 2, 2005.

[4] Chung, L., B.A. Nixon, E. Yu, J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, 2000.

[5] J. Horkoff, "An Evaluation Procedure for the i* Framework", (in progress), Masters Thesis, Dept. of Computer Science, 2005.

[6] L. Liu, E. Yu and J. Mylopoulos, "Security and Privacy Requirements Analysis within a Social Setting", *International Conference on Requirements Engineering (RE'03)*, Monterey, California, September 2003, pp. 151-161.

[7] L. Liu, E. Yu, "Designing Information Systems in Social Context: A Goal and Scenario Modelling Approach Information Systems", Elservier 29(2), 2004, pp. 187-203.

[8] E. Yu and L.M. Cysneiros, "Designing for Privacy in a Multi-Agent World", in *Trust, Reputation and Security: Theories and Practice*, R. Falcone, S. Barber, L. Korba and M. Singh (eds.), Springer-Verlag, 2003.

[9] A. Sutcliffe, "Scenario-based requirements engineering", Proceedings, *11th IEEE International Requirements Engineering Conference (RE'03)*, Monterey Bay, USA, 8-12th September, 2003, pp. 320- 329.

[10] C. Rolland, C. Souveyet, C. Ben Achour. "Guiding Goal Modeling Using Scenarios," *IEEE Transactions on Software Engineering*, vol. 24, no. 12, December 1998, pp. 1055-1071.

[11] Software Engineering Standards Committee of the IEEE Computer Society, *IEEE Recommended Practice for Software Requirements Specifications*, Approved 25 June 1998, IEEE-SA Standards Board IEEE, Std 830-1998.
[12] A. I. Antón, C. Potts, "The use of goals to surface requirements for evolving systems", In *Proceedings of the 20th international Conference on Software Engineering*, Kyoto, Japan, April 19 - 25, 1998.
[13] J. Karlsson, K. Ryan. "A Cost-Value Approach for Prioritizing Requirements," *IEEE Software*, vol. 14, no. 5, September/October, 1997, pp. 67-74.
[14] M. Leica, "Scalability Concepts for i* Modelling and Analysis", Masters Thesis, Dept. of Computer Science, 2004.
[15] H. E. Esquivel, O. Pastor, P. Giorgini, J. Mylopoulos, (Presentation) "Extensions to the i* Framework to fit with a Model-Based Code Generation Environment", 2nd Inter-Galactic i* Workshop, RESG Meeting, 21-22 April 2005, London, Retrieved October 2005 from http://www.cs.toronto.edu/%7Eeric/istar-at-RESG05.htm.
[16] *OpenOME, an open-source requirements engineering tool*, Retrieved October 2005, from http://www.cs.toronto.edu/km/openome/.
[17] GESSI: Software Engineering for Information Systems Group, *REDEPEND-REACT an Architecture Analysis Tool*, Retrieved October 2005, from http://www.lsi.upc.edu/~ggrau/REDEPEND-REACT/.
[18] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, R. Sebastiani, "Formal Reasoning Techniques for Goal Models", *Journal of Data Semantics*, 2004.
[19] P. Giorgini, J.Mylopoulos, R. Sebastiani, "Simple and Minimum-Cost Satisfiability for Goal Models", *16th Conference On Advanced Information Systems Engineering (CAiSE*04)*, Springer Verlag, 2004.
[20] P. Giorgini. J. Mylopoulos, E. Nicchiarelli, R. Sebastiani, "Reasoning with Goal Models", *21st International Conference on Conceptual Modeling (ER2002)*, Springer Verlag, Tampere, Finland, October 2002.
[21] J. Aranda and S. M. Easterbrook, "Anchoring and Adjustment in Software Estimation", *European Software Engineering Conference / ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE'05)*, Lisbon, Portugal, Sept 5-9, 2005.
[22] J. Brinkkemper. John Mylopoulos. A. Solvberg. Eric Yu, "Tropos: A Framework for Requirements-Driven Software Development", *Information Systems Engineering: State of the Art and Research Themes*, 2000, Springer-Verlag.
[23] A.Fuxman, J. Mylopoulos, M. Pistore, P. Traverso, "Model Checking Early Requirements Specifications in Tropos", *The 9th IEEE International Requirements Engineering Conference (RE2001)*, Toronto, Canada, August 2001.
[24] A. Martínez, O. Pastor, H. Estrada, "Isolating and Specifying the Relevant Information of an Organizational Model: A Process Oriented Towards Information System Generation", *Lecture Notes in Computer Science*, Volume 3046, Jan 2004, pp. 783 - 790.
[25] N. A. M. Maiden, S. V. Jones, S. Manning, J. Greenwood, L. Renou, "Model-Driven Requirements Engineering: Synchronising Models in an Air Traffic Management Case Study", *16th International Conference, CAiSE 2004*, Riga, Latvia, June 7-11, 2004, pp. 368-383.
[26] N. A. M. Maiden., S. Manning, S. Jones, J.Greenwood, "Towards Pattern-Based Generation of Requirements from Systems Models", *Proceedings REFSQ'2004 Workshop, in conjunction with CaiSE'2004*, 7-8 2004, Riga, Latvia, 2004.
[27] T. T. Do, M. Kolp, S. Faulkner, "Agent Oriented Design Patterns: The SKwyRL Perspective" *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS 2004)*, Porto, Portugal, April 2004.
[28] M. Kolp, (Presentation) "Bed Management Organizational Analysis with i*: The Case of the Saint-Luc University Clinics", 2nd Inter-Galactic i* Workshop, RESG Meeting, 21-22 April 2005, London, Retrieved October 2005 from http://www.cs.toronto.edu/%7Eeric/istar-at-RESG05.htm.
[29] A.Perini, Understanding the Requirements of a Decision Support System for Integrated Production in Agriculture, *2nd Inter-Galactic i* Workshop*, RESG Meeting, 21-22 April 2005, London, Retrieved October 2005 from http://www.cs.toronto.edu/%7Eeric/istar-at-RESG05.htm.
[30] O. Pastor, "Some Lessons Learned from using i* Modelling in Practice", *2nd Inter-Galactic i* Workshop*, RESG Meeting, 21-22 April 2005, London, Retrieved October 2005 from http://www.cs.toronto.edu/%7Eeric/istar-at-RESG05.htm.
[31] E. Kavakli, P. Loucopoulos, and D. Filippidou, "Using Scenarios to Systematically Support Goal-Directed Elaboration for Information System Requirements", *Proceedings of the IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'96)*, Friederichshafen, Germany, March, 1996.
[32] A. I. Anton, C. Potts, "The Use of Goals to Surface Requirements for Evolving Systems", *Proc. of the 1998*

Int. Conference on Software Engineering, 1998, pp. 157-166.
[33] V. F. Santander, J. Castro, "Deriving Use Cases from Organizational Modeling", *Proceedings of the 10th Anniversary IEEE Joint international Conference on Requirements Engineering*, September 09 - 13, Washington, DC, 2002, pp. 32-42.
[34] B. Hui, S. Liaskos and J. Mylopoulos, “Requirements Analysis for Customizable Software: A Goals-Skills-Preferences Framework”, *11th IEEE International Conference on Requirements Engineering*, Monterey, CA., Sept. 2003.

Appendices

Table 1: Project goals

- Create a Specification for the replacement Ask a Counselor System which utilizes the information captured in the i* counseling model
- Further evaluate the effectiveness of applying i* to a medium sized real world project
- Evaluate the application of the i* evaluation procedure to a real world project
- Apply scenarios and the i* evaluation procedure to the i* model in order to:
    - Evaluate the effectiveness of current system scenarios
    - Confirm the effectiveness of future system scenarios
- Evaluate the usefulness of scenarios for our project in general

Table 2: Original project plan

Analysis of the Current Ask a Counselor System	**Previously Completed**
1. Interviews to extract organizational and system goals	Yes
2. On site system walkthroughs	Yes
3. Creation of i* Model which captures system and organizational goals, current system tasks and their effects on system and organizational goals	Yes
4. Extraction of Scenarios of current usage from i* model	No
Analysis of the Future Ask a Counselor System	**Previously Completed**
5. Create new system scenarios based on previous scenarios, and the scenario evaluation results	No
6. Evaluate the new scenarios in the i* model to ensure they meet system and organizational goals effectively	No
7. Present new scenarios to KHP for confirmation and adjustment. Make needed changes to scenarios and the model.	No
8. Use new scenarios and goals in i* models to create a specification of the system	No
9. Verify that system specification sufficiently satisfies goals in the i* model	No
10. Give Specification to Students to Implement	No

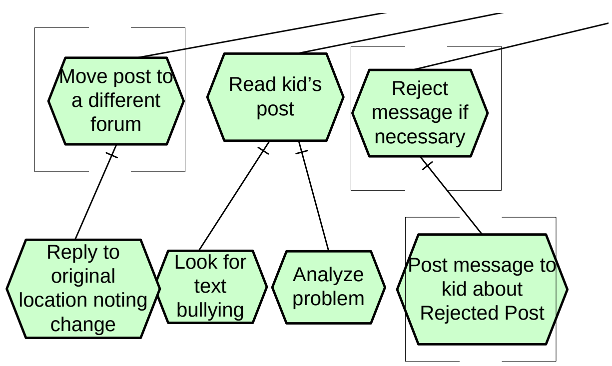

Figure 1: Part of the task structure in the i* model for the current system

- **(If post in wrong forum)** [Move post to a different forum
  - Reply to original location noting change
-]
- Read kid's Post
  - Look for text bullying
  - Analyze Problem
- **(If message meets rejection criteria)** [Reject message if necessary
  - [Post message to kid about rejected post]
-]

Figure 2: Part of the Meta- Scenario corresponding to Figure 1

Table 3: Scenario sores for the current system

Scores without Goal Prioritization Weights		Scores with Goal Prioritization Weights	
Score	Scenario Name	Score	Scenario Name
-145.5	1.7 Edit Post	-140	1.7 Edit Post
-139.5	1.3 Move Post	-134	1.3 Move Post
-136.5	1.4 Reject Message	-131.5	3.1 All
-136.5	3.1 All	-131	1.4 Reject Message
-133.5	1.2 Min Actions	-128	1.2 Min Actions
-133.5	1.9 Authority Referral	-128	1.9 Authority Referral
-127.5	All of 4 but 4.5	-122.5	All of 4 but 4.5
-127.5	4.5 Post	-122.5	4.5 Post
-124.5	1.10 Share Experiences	-119	1.10 Share Experiences
-124.5	1.11 Feedback from Supervisors	-119	1.11 Feedback from Supervisors
-123	2.4 Move Post	-118	2.4 Move Post
-121.5	1.8 Reply with Extra Actions	-116	1.8 Reply with Extra Actions
-120	1.5 Search for Posts	-115	2.10 Reject
-120	1.6 Communicate about Repeat	-114.5	1.5 Search for Posts
-120	2.10 Reject	-114.5	1.6 Communicate about Repeat
-118.5	2.5 Edit Kid Post	-113.5	2.5 Edit Kid Post
-117	2.2 Min Actions	-112	2.2 Min Actions
-115.5	2.1 All Actions	-110.5	2.1 All Actions
-112.5	2.9 Edit Reply 2	-107.5	2.9 Edit Reply 2
-111	2.3 Search	-106	2.3 Search
-111	2.6 Return 1	-106	2.6 Return 1
-111	2.7 Return 2	-106	2.7 Return 2
-111	2.8 Edit Reply 1	-106	2.8 Edit Reply 1
-109.5	All Scenarios	-104	All Scenarios
-99	1.1: All Actions	-93.5	1.1: All Actions

Table 4: Priorities of future system optional features

Scores without Goal Prioritization Weights		Scores with Goal Prioritization Weights	
Score	**Feature**	**Score**	**Feature**
33.75	Link to Previous and Pending	27	Link to Previous and Pending
22.5	List of Logged in Moderators	26	Provide and Manage Best Answers
21.25	Provide and Manage Best Answers	21	List of Logged in Moderators
17.5	Feedback sections for each post	19.5	Personal Space for Kids
15	Counselor Space/Feedback	18	Question and Answer one entity
15	Automatic Message Assigning	17.5	Optional Public/Private Threads
13.75	Optional Public/Private Threads	16	Automatic Message Assigning
13.75	Question and Answer one entity	15	Sorting
13.75	Personal Space for Kids	14	Feedback sections for each post
12.5	Timeouts	13.5	Counselor Space/Feedback
11.25	Print	12	Automatically Save and View Edits
10	Sorting	10.5	Estimate Response Time
10	Automatic Moving Notice	10	Automatic Moving Notice
7.5	Automatically Save and View Edits	10	Timeouts
7.5	Auto save	9	Print
7.5	See Messages in Both Tiers	7	Spellchecking
6.25	Blank Forum	7	See Messages in Both Tiers
6.25	Spellchecking	6	Auto save
6.25	Language viewing control/filter	6	Post Archiving
6.25	Estimate Response Time	5	Language viewing control/filter
5	Post Locking	4	Post Locking
5	Post Archiving	4	Country Filtering
5	French moderator view	3	Blank Forum
3.75	Show Relevant Internal Links	3	Show Relevant Internal Links
3.75	Country Filtering	3	Record of Counselor Picks
3.75	Record of Counselor Picks	2	French moderator view
2.5	Prompt kid for updates	2	Prompt kid for updates
-2.5	Case Files	0	Case Files

Table 5: Actual project steps

Analysis of the Current Ask a Counselor System
4. Extraction of Scenarios of current usage from i* model.
5. Added more links to the i* SR model.
6. Application of evaluation of current scenarios onto i* model in order to compare and analyze effectiveness of current scenarios.
7. Created scores out of scenario evaluation results in order to acquire a prioritization of scenarios.
8. Added goal priorities into goal model and evaluation in order to get a more accurate prioritization of scenarios.
Analysis of the "to-be" Ask a Counselor System
9. Operationalized goals in the i* SR model.
10. Ordered the operationalizations in the order they would be performed in order to form new system scenarios.
11. Analyzed scenarios apart from model to find missing tasks.
12. Determined the required “base” and optional features for the system by looking at scenario actions.
13. Evaluated the “min” and “max” set of system features.
14. Evaluated optional features and compared the result to the “min” evaluation to derive a prioritization of optional features.

15.	Added goal priorities to the model and incorporated these priorities into the prioritization of optional features.
16.	Used required and optional features to create a specification for the new system.
17.	Gave Specification to Students to implement.
18.	Presented base scenarios, optional features, and feature priorities to KHP for validation. Adjust the specification based on their input.

Ask a Counsellor System System Specification Creation Process

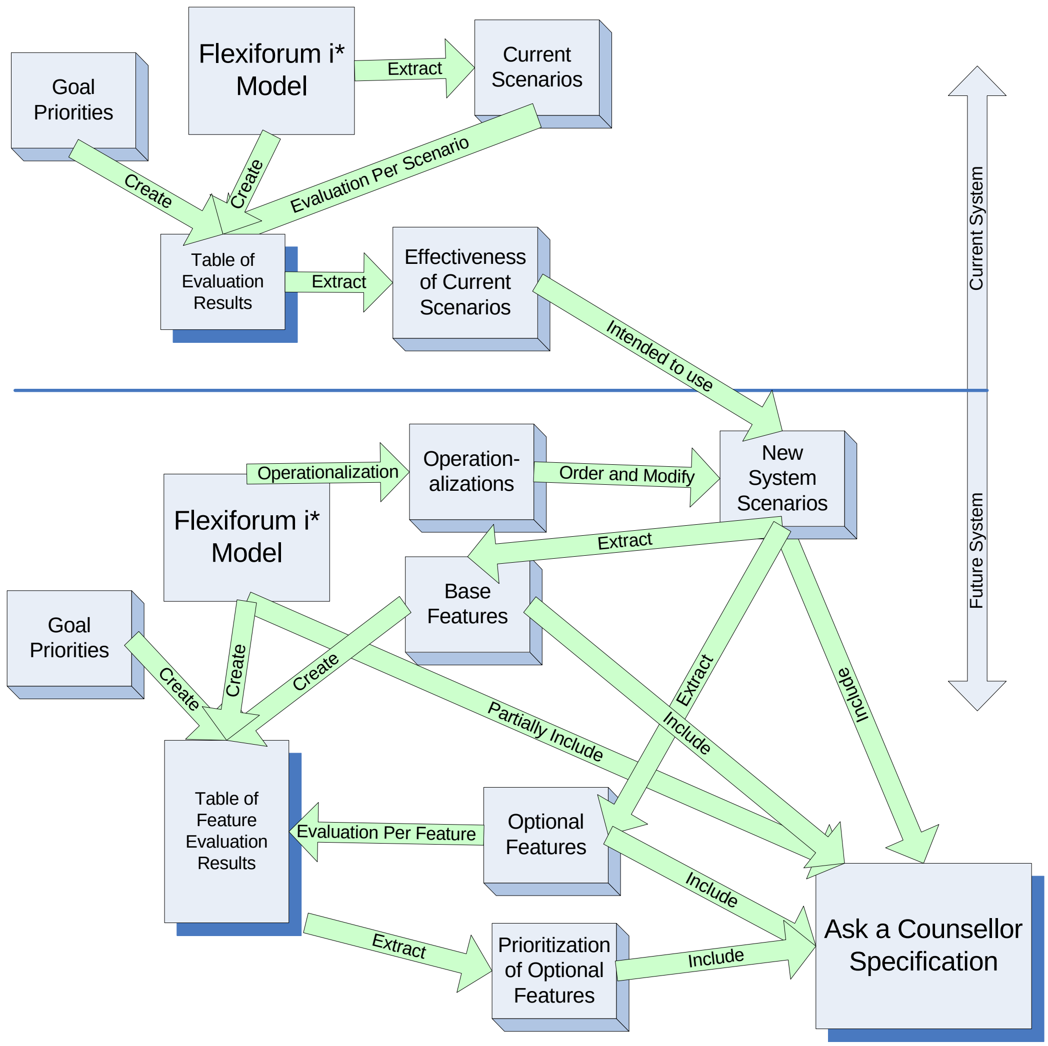

Figure 3: Artifacts of actual project steps

Table 6: Stakeholder priority adjustment

Priority Adjustment	**Comparison to Prioritization with Individual Goal Prioritization**		**Comparison to Prioritization without Individual Goal Prioritization**	
	Number of Features	**% out of 28**	**Number of Features**	**% out of 28**
Correct	11	39.29	8	28.57
Off by 1	9	32.14	10	35.71
Off by 2	6	21.43	7	25.00
Off by 3	2	7.14	3	10.71
Off by 4	0	0.00	0	0.00
Off by 5	0	0.00	0	0.00
	Score	27	Score	33

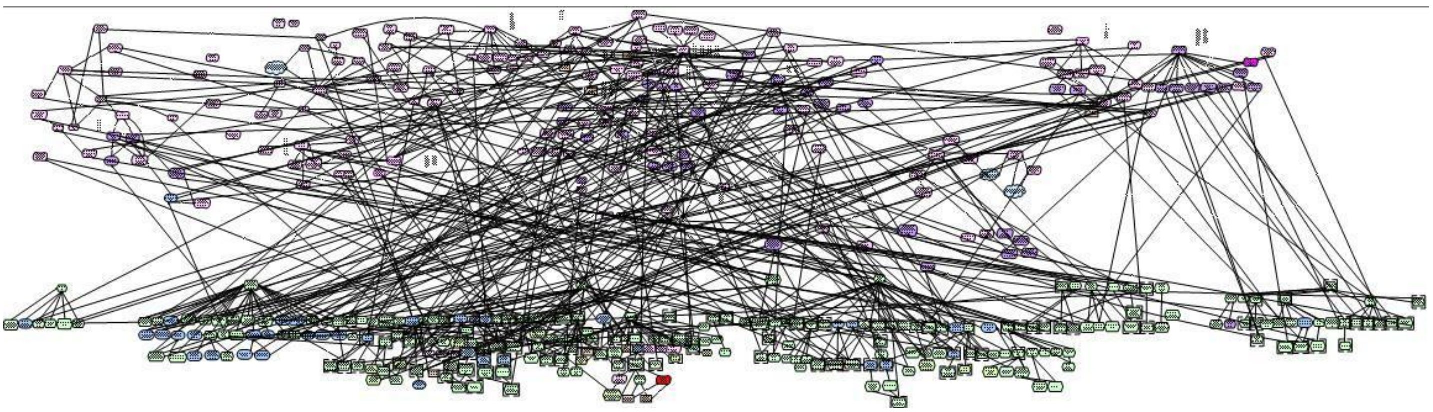

Figure 4: The i* SR model for the future ask a counselor system

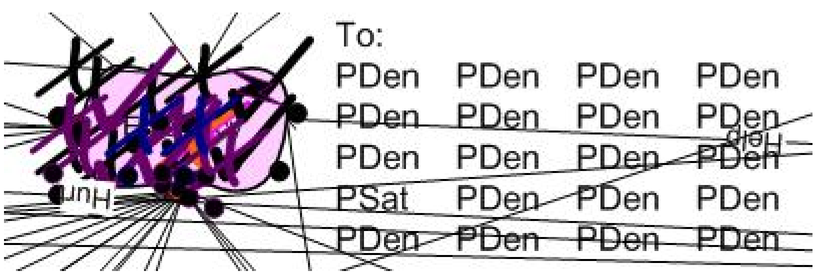

Figure 5: Example of the evaluation of a high-level element (efficiency)