Towards Improved Visual Support for i* Modelling

by

Christopher Cocca

A thesis project submitted in conformity with the requirements
for the degree of Master of Information Studies
Faculty of Information Studies
University of Toronto

Supervisor:
Professor E. Yu

Faculty Readers:
Professor J.M. Cherry
Professor S. Easterbrook (Computer Science)

Your file Votre référence
ISBN: 978-0-494-27330-2
Our file Notre référence
ISBN: 978-0-494-27330-2

Towards Improved Visual Support for i* Modelling
Christopher Cocca, MISt, 2007
Faculty of Information Studies
University of Toronto

Abstract

The i* framework (Yu, 1995) offers a graphical notation for modeling and analyzing socio-technical systems, particularly during early stage requirements engineering for software systems. A number of software tools exist to support i* modeling. Nevertheless, there is a strong need for better visual support in i* modeling. This study aimed to identify areas for improvement and to develop prototypes to address these areas. A review of existing i* tools and observation of users revealed that current tools provide inadequate support for maintaining an overall view of a model, filtering extraneous data and extracting sub-collections of the model data.

Prototypes were then developed based on these needs, and an evaluation by users of i* was carried out to test the validity of the concepts and requirements included in the prototypes. User feedback suggested these prototypes would be useful when using software for i* modelling and offered suggestions for further improvements.

Contents

List of Tables

List of Figures

1. Introduction
1.1 Background
Graphical notations are widely used in requirements engineering to represent the data captured during requirements gathering. The use of graphical notations provides analysts with a visual representation of the captured data, allowing analysts to extend their working memory by transferring the data held implicitly to an explicit visual form. The visual nature of graphical notations enables analysts to quickly identify relationships, dependencies, and conflicts amongst various stakeholders. This information is necessary for developing requirements that will best satisfy the stakeholders of the system.

To this point, however, many tools supporting the visualization of requirements-related data have struggled to cope with the inherent complexity and scope involved in many requirements engineering projects. As a result, the benefits of visualizing requirements data decrease; modellers must increase their effort to gain insight from the models they create.

1.1.1 The i Framework*
The i* framework (Yu, 1995) is a requirements engineering technique used to capture and document socio-technical factors surrounding a system. Modellers using i* are interested in looking beyond *what* a system should do, to discern *why* a system should do what they do. The i* framework explores these issues by visually depicting the goals of system stakeholders and the means for achieving these goals. Model development is performed using two kinds of models in the i* framework: the Strategic Dependency (SD) model and the Strategic Rationale (SR) model.

1.1.2 Strategic Dependency Model
The Strategic Dependency (SD) model is used to describe the relationships between various 'actors'. For each relationship between actors, a link is established. This link denotes a dependency between the two actors at either end of the link. The depending actor is called the depender, with the actor who is being depended upon named the dependee. The reason or object that has created this dependency is described as the dependum. Goals, tasks, resources and softgoals comprise the dependency types used to form dependums between actors. The diagram below depicts a simple goal dependency between two actors. The car owner (depender) requires the body shop (dependee) to repair his car (the dependum). In other words, the car owner depends on the body shop to repair his car, thus creating this relationship between the actors.

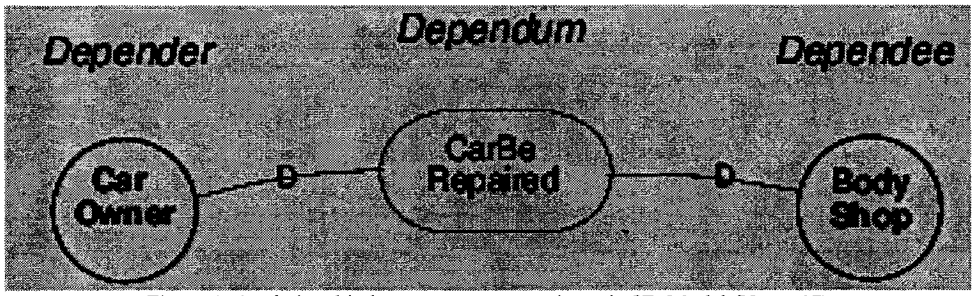


Figure 1: A relationship between two actors in an i* SD Model (Yu, 1997)

The intent of the SD model is to provide a deeper understanding of the business process by isolating and focusing on the intentional dependencies among actors, irrespective of current or potential work processes and entity flows (Yu & Mylopoulos, 1994). Focusing on these dependencies helps modellers and other evaluators to distinguish "what is at stake, for whom, and what impacts are likely if a dependency fails" (Yu & Mylopoulos, 1994, p. 5).

1.1.3 Strategic Rationale Model

The Strategic Rationale (SR) model also takes the form of a graph consisting of nodes and links. Nodes denote goals, tasks, resources, or softgoals; identical to the node-types used in the SD model. Nodes are linked in one of two ways; using means-ends relationships or task-decomposition relationships. A means-ends link can be explored in either direction. Examining a link from the top-down perspective may raise questions as to how an actor is employing a particular means to meet an end. Alternatively, exploring the means-ends link from the bottom-up perspective allows modellers to explore why a task is performed, namely as a means to satisfy a certain end. Task-decomposition links allow for tasks to be broken into sub-elements and form a parent-child relationship. A sub-element can be a task, goal, resource or softgoal.

The SR model is intended to document the rationale organizational actors have about the relationships described in the SD model. Further, the SR model is useful for producing alternative approaches to the status quo ('as-is') scenario. Each goal described in the SR model has the potential to be achieved in many different ways by asking questions such as "how can we achieve this goal?" or "why is this goal achieved this way?". These alternative approaches can be documented using means-ends and task-decomposition links. Further, contribution links can be applied to these alternatives in an attempt to evaluate which alternative offers the most benefit. Contribution links are used to describe how a certain subtask or subgoal affects a parent softgoal. Contributions can either 'help' or 'hurt' the achievement of a goal. If the contribution of a subtask or goal is unknown, this too can be documented. Models can then be evaluated by assessing the affect of contributions as they are propagated through the model.

1.2 Context and Purpose of Research

The i* framework has proven useful in documenting the goals and intentions of actors and analyzing opportunities, threats, and alternative approaches for a given system or organization in many instances. However, it appears that as i* models grow in complexity and scope, the benefit of producing these models is reduced. Modellers must exert more effort to construct models and spend an increased amount of time attempting to gain insight from the models through analysis.

Strategic Dependency models become cluttered and links between actors untraceable. Modellers are forced to expend considerable effort constructing models in a legible way and attempting to trace dependencies from one actor to another. The effort and time spent on such tasks detracts from the central aim of the SD model; to distinguish "what is at stake, for whom, and what impacts are likely if a dependency fails" (Yu & Mylopoulos, 1994, p.5).

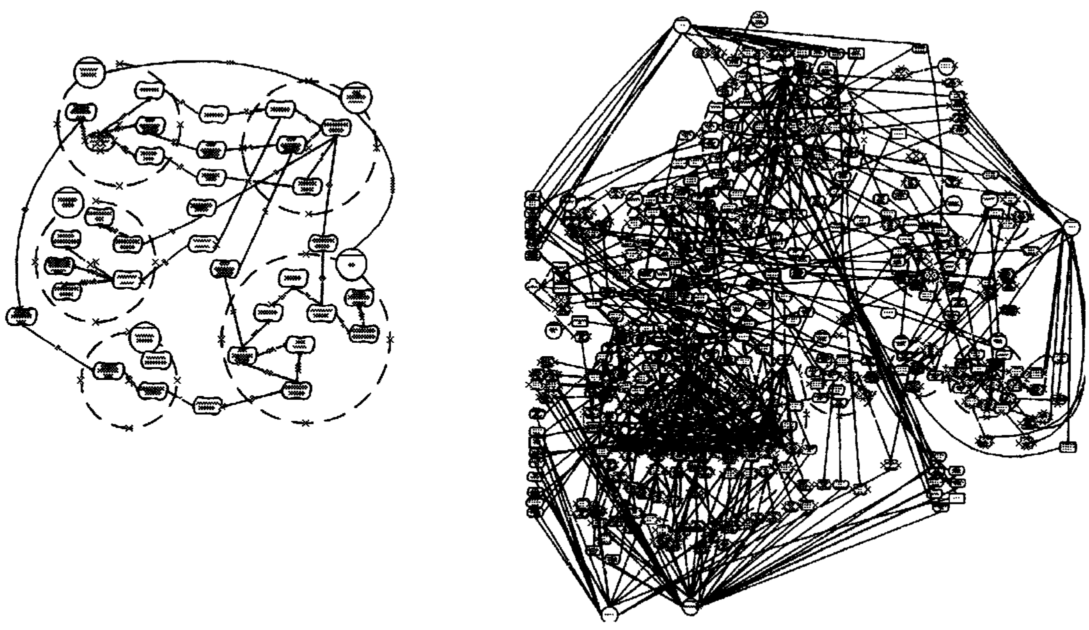


Figure 2: Comparison of Small and Large i* Models

The amount of effort required to create Strategic Rationale models in large-scale projects has caused some researchers to limit their uses (Maiden, 2004). SR models are useful for documenting the actors' rationale for the relationships, goals and tasks they hold. However, constructing SR models is problematic due to the level of complexity involved in displaying such models using existing i* software (Maiden, 2004). For example, it often becomes difficult to track links through the model, as they too become cluttered and unclear, or to differentiate one alternative from another. Specifically, it becomes unclear how nodes relate to particular alternatives, and what alternative best 'satisifices' the largest number of actors.

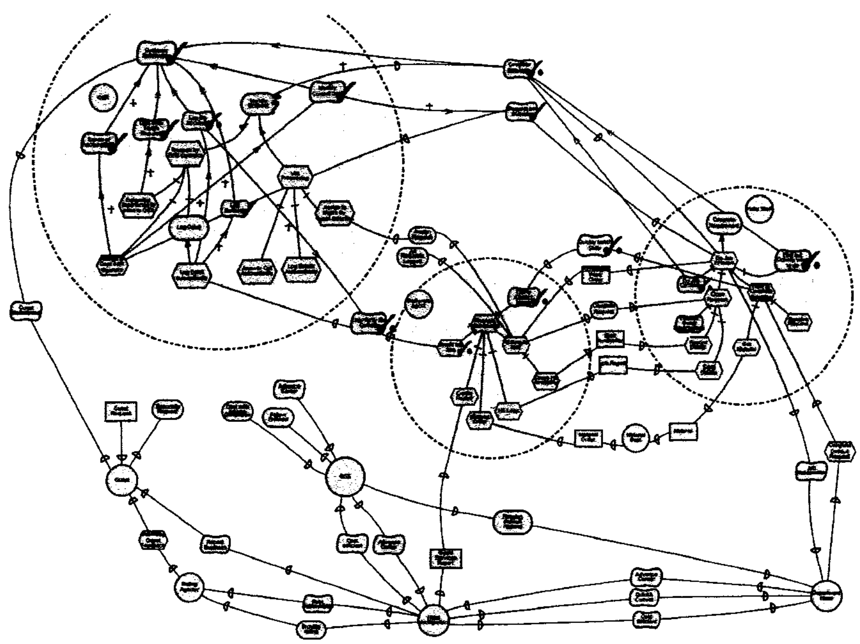


Figure 3: Example SR Model (Chung, Cocca, Samavi, & Sutherland, 2005)

In order to benefit i* modellers, tools supporting i* models must be better equipped to visualize the data captured during i* modelling. This study attempts to identify mechanisms for improved visualization. First, user observation and interviews are used to uncover users' visualization challenges in developing and evaluating i* models. Based on these challenges, visualization concepts discussed in the literature review and mechanisms used within existing applications are sought out in an attempt to overcome the challenges discovered during user observation. Concepts and mechanisms that offer the potential to overcome these challenges are developed as prototypes. Finally, prototypes are evaluated by persons with i* modelling experience to determine their usefulness.

1.3 Scope and Definitions

It is important to define a number of terms that are commonly used throughout this paper. First, the word *tool* is used synonymously with the term software or application, to describe "the totality of software required to produce a system acceptable to end users" (Oxford University Press, 2004). The term *mechanism* is used throughout the paper to describe both the layout and behaviour of devices that are incorporated within software to produce some kind of visual effect on the i* models.

The term *canvas* refers to the total visual space occupied by the model. In many cases, the size of the canvas of an i* model can extend well beyond the screen size, making it important to distinguish between the canvas and the screen. Model *elements* refer to the goals, softgoals, tasks, resources, actors, agents, roles that comprise i* models.

1.4 Significance of Study

This study examines the visual challenges users face when interacting with i* models and explores visual mechanisms through prototypes useful for improving model clarity in an attempt to reduce the effort required by modellers to gain insight from the models under analysis.

A review of existing i* tools is conducted to provide a succinct description of the features currently available to support i* modelling. The prototypes produced in this study aim to build upon the existing tools in order to provide additional or improved capabilities for i*. Ideas for prototypes are derived from user observation. The potential value of the proposed features is then judged by another group of i* users, as they were presented with mockup demonstrations of the prototypes.

These prototypes are software-independent. In other words, each prototype described in this paper could be incorporated into any existing i* tool.

1.5 Outline of Study

Following this chapter, literature and software relevant to this study are reviewed. Information Visualization, Requirements Engineering Visualization, and previous work relating to the Visualization of i* Models is discussed. Existing software supporting i* models is discussed, with the intent of understanding the current state of the art and areas yet to be addressed in tool development. Chapter 3 outlines the methods employed for the project. The techniques used in user observation, prototype selection and development, and prototype evaluation are discussed. Chapter 4, discusses the findings from user observation, provides rationale for the prototypes selected, describes how they were developed, and describes the results from user testing of prototypes. Chapter 5 outlines the limitations of the study, offers concluding remarks and suggestions for future work.

2. Literature Review

2.1 Information Visualization

Information Visualization is a field of study with much to offer when searching for mechanisms or techniques for improved interaction support. Card, Mackinlay, and Shneiderman (1999) define Information Visualization as, "using a graphical means to create or discover the idea itself: using the special properties of visual perception to resolve logical problems" (p. 1). Visualization allows a person to extend their working memory by moving it from outside of the mind and into visual form.

Information Visualization (IV) transforms data in order to represent it in a visual way. In *The Eyes Have It*, Shneiderman (1996) classifies and explains seven broad IV data types that encompass the field of IV: one-, two-, three-dimensional data, temporal and multi-dimensional data, tree data and network data. These data types are not exhaustive; there are many variations of these data types, and software often employs combinations of these data types. Below is a brief description of data types that best match the types of data contained within the i* framework, and an explanation as to why each data type has the potential to benefit i* visualization.

Tree data types represent hierarchies or tree structures where each item in a collection is linked to a parent item. Items and links may have multiple attributes. Examples of IV for tree data types are indented outlines (Egan et al., 1989 as cited in Shneiderman, 1996), 3D cone and cam trees (Robertson et al., 1993; Carriere & Kazman, 1995 as cited in Shneiderman, 1996), and dynamic pruning (Kumar et al., 1995 as cited in Shneiderman, 1996). Treemaps have also been used in computer directories and business decision making (Ashai et al., 1995 as cited in Shneiderman, 1996).

Network data types are used when relationships between items in a collection are not necessarily structured as parent-child. In these cases, items may be linked to any number of other items. Visualization techniques related to network data types may include highlighting paths that connect various items or methods for traversing the network. Network data types have been used in an attempt to visualize the World Wide Web (Andrews et al., 1995, Hendley et al., 1995 as cited in Shneiderman, 1996).

Network data types are closely associated with the structure of SD Models and SR models. In both SD and SR models, i* objects may be linked to any number of other i* objects, and relationships need not take the form of parent-child. Exploring mechanisms that have been built upon network data types may be a great resource for potential mechanisms suitable for tools supporting i*.

The use of planes, described by Shneiderman as two dimensional data types, may also be useful when visualizing i* models. Planes may be used to represent distinct layers by which to group various i* objects. Layers have the potential to be used in i* software for grouping i* objects by type on separate layers, allowing users to control the layers that are shown and hidden as necessary. Additionally, layers could be used to organize labels and contribution links used during model evaluation. These approaches may reduce diagram clutter by isolating specific alternatives or types of objects. For example, Adobe's Photoshop and Macromedia's Flash applications use planes in the form of a 'layers' component within their software, allowing objects to be hidden and displayed by the user.

2.2 Requirements Engineering Visualization

Requirements Engineering Visualization (REV) is a new area of research (14th IEEE International Requirements Engineering Conference, 2006) that attempts to apply IV principles and techniques to the field of requirements engineering. Improvements to the visual representation of requirements are essential for allowing analysts and modellers to develop a deep understanding of the data captured during the requirements-gathering process.

Ozkaya (2006) discusses the necessity of capturing structural relationships effectively. The author suggests it is necessary that REV researchers gain an increased understanding of the mental model best suited for visualizing requirements. In order to do so, more work must be conducted in determining who will produce and use visual representations.

Feather, Cornford, Kiper and Menzies (2006) discuss the utility of the visualization techniques they have utilized in past projects. They found that no single visualization technique will serve all purposes and relied on a mix of several. They also found that many simple visualization techniques such as TreeMaps and bar-charts, were sufficient for their needs. Though the authors employed these techniques using quantitative models, and i* is primarily a qualitative modelling system, the relevance of their findings is applicable for this research project. Many visualization techniques are required to satisfy the various tasks undertaken by i* modellers, and many traditional IV techniques may prove useful should they be incorporated as features within tools supporting i*. However, these techniques may need to be adapted to meet the unique characteristics of the i* modelling framework.

2.3 Visualization for i* Modelling

Issues of scalability in i* have been explored in previous studies. Leica (2005) suggests i* modelling follows a lifecycle, with each phase of the lifecycle containing unique scalability challenges. In evaluating the 'reasoning' phase of i*'s lifecycle, Leica proposed slicing of models for better visualization. Leica developed techniques for slicing a model top-down and bottom-up. Slicing "allows one to find semantically meaningful decompositions of programs, where the decompositions consist of elements that are not textually contiguous" (p. 22). Slicing offers a means of filtering model data and may be particularly useful for improving the clarity of models by removing extraneous data from the modeller's view.

You (2004) also explores scalability in i* and proposes a view extension for addressing this issue. Views are "extensions over a model according to some criteria" (p. 7) and allow the models to be broken down into segments that are self-contained, with the rest of the segments easily available. The notion of views has a great deal of potential for producing alternative displays of i* model data. Horkoff (2006) created an evaluation procedure adapted and expanded from the Non-Functional Requirements (NFR) Framework to improve model quality by detecting semantic flaws within.

Grau's (2006) work has produced software utilizing tabular formats and tree hierarchies to represent various process activities, associated actors, and i* models. This perspective represented an alternative perspective to visualizing i* models and was considered when developing prototypes in later stages.

2.4 Scope and Limitations of Previous Research

Previous i* researchers have explored a variety of methods for improving or extending the i* framework. While most researchers have focused on the notation itself, this study examines the tools used to facilitate the creation of i* models as a means for improved interaction and visualization of i* models. The literature and the functionality of existing applications provide a strong foundation for identifying potential mechanisms to be incorporated in future i* tool support.

2.5 Research Objectives

This study aims to improve visual support for i* modelling by identifying features with the potential for improving clarity and reducing the clutter in regards to i* models. First, challenges encountered by i* modellers during the modelling process are identified. As challenges are discovered, requirements for overcoming these challenges are documented. In order to address challenges in i* model visualization, various mechanisms are selected in an attempt to improve tool support for i* modelling, with a focus on improved model clarity, especially during the analysis of i* models.

3. Method

The purpose of this chapter is to present and discuss the specific phases of the overall research design, and to provide a detailed discussion of the various procedures and methods used for collecting and analyzing the data. The project included four phases. In the first phase, tools supporting i* modelling were reviewed. As well, observation and interviews were conducted with participants as they worked on i* modelling projects. The second phase involved the identification of concepts or mechanisms for overcoming the challenges uncovered in Phase 1.. The third phase involved the development of prototypes. The fourth phase involved user evaluation of the prototypes. In advance of participant involvement (in phase 1b and phase 4) a detailed description of the procedures and instruments used in the study was approved by the University of Toronto's Social Sciences and Humanities Ethics Review Board (Appendix 1, Appendix 3).

3.1 Phase 1a – Review of i* Tools

A review of existing software supporting i* modelling was conducted. The purpose of this review was to understand the current state-of-the-art for i* modelling and the various mechanisms included in each tool. Gaining an understanding of current tool support was important in this study as its intention was to improve or extend current functionality beyond that which already existed for i* modelling.

Tools were reviewed first by locating and installing each i* tool. Once installed, the researcher interacted with each tool in order to understand how the various features of each tool worked. When available, documentation was consulted as a means of ensuring that all features of a tool had been reviewed.

After each tool had been reviewed, the various features of the tools were categorized by Shneiderman's (1996) Visual Information Seeking Tasks. Shneiderman suggests that these tasks aid users in identifying the visual information they seek. Categorizing the features by these tasks helps to demonstrate what types of tasks have been strongly addressed, weakly addressed or not addressed by the various tools. This comparison revealed areas to focus on when developing new features for i* modelling tools.

3.2 Phase 1b – Observation and Interviews

This study used a qualitative approach in order to gain insights into modellers' experiences with tools supporting the development of i* models. In qualitative studies, a researcher attempts to "collect open-ended, emerging data with the primary intent of developing themes from the data" (Creswell, 2003, p. 18). Exploring the practices of i* modellers in order to develop common themes was particularly important for this study, as little research had been conducted in regards to the difficulties relating to the visualization of i* models using i* software. As this phase focused on the discovery of challenges faced by modellers, qualitative methods such as observation and open-ended interviews were deemed more likely to be effective than quantitative methods. Qualitative studies are more conducive to exploration of new topics yet to be studied than quantitative studies.

In order to identify the full range of challenges relating to the modelling process, multiple methods must be used (Maiden & Rugg, 1996). This is because each method possesses various strengths and weaknesses that make them useful for capturing different types of knowledge from the participant.

Maiden and Rugg (1996) present a variety of methods for requirements acquisition in the ACRE framework. Its core guidelines for method selection are derived from theories of cognition and social interaction, and empirical data which support these theories. The theories provide a rigorous, well-founded starting point for method selection. Of the methods outlined in the ACRE framework, the two methods most useful for the given problem domain were observation and open-ended interviews. Their simplicity and flexibility were well-suited to the exploratory nature of the problem that was studied. Additionally, the use of a technique called Abstraction, described below, is well-suited to the participants of the study, given their ability to think conceptually and their experience with abstract modelling.

First, observation of modellers was undertaken to better understand the experiences of modellers using i* modelling software. Observation sessions were typically followed by short (15 to 30 minute) open-ended interviews. The purpose of these interviews was to probe the participants about interesting events that took place during the observation session and to discuss ideas for improving their experience with i* modelling software.

Interviews were also used as a means of employing a technique described by Robertson (2001) as Abstraction. Abstraction requires participants to make useful generalizations about the problem domain. These generalizations are particularly useful for identifying commonalities between various activities. "People who are good at abstract thinking are usually comfortable looking at class models, data models and other models that focus on the "essence" of the subject matter" (2001, p. 3). It was anticipated that expert modellers would likely be skilled in Abstraction, as they are highly educated and have significant experience working with models such as i* models and Entity Relationship Diagrams that focus on the 'essence' of a problem domain.

3.2.1 Participants

Participants involved in this study included i* modellers from the Knowledge Management (KM) Lab at the University of Toronto and students from the University of Toronto's Faculty of Information Studies (FIS) enrolled in *FIS 1341 – Analyzing Information Systems.* The former have a great deal of experience in using and developing i* models, and are familiar with tools supporting i*. These modellers seemed likely to generate a great deal of insight into how tools can be improved to better support the work they are conducting. Students of FIS 1341 were also included as participants of the study. In FIS 1341, students are introduced to the i* modelling framework and use i* tools in course assignments. These users represent 'novice' users; those who may have a strong technical background but are unlikely to have developed i* models prior to FIS 1341.

To recruit novices, the researcher arranged to speak with the FIS 1341 class during a lecture. At this time, the project was described to the class, outlining the requirements for participation. The class was informed that the instructor of the course would not be made aware who was participating in the study. The students were then provided with instructions for contacting the researcher should they be willing to participate.

Expert modellers were recruited through the Knowledge Management (KM) Lab at the University of Toronto. The researcher spoke to faculty overseeing the KM Lab about potential participants to be included in the study. KM Lab faculty then provided contact information for students in the early stages of a project that utilized i* modelling. The researcher contacted each student individually, requesting their participation in the study.

Participants of the study were asked to volunteer; they were not chosen at random. Participants were chosen from a convenience sample based on their experience with the i* modelling technique. It seemed likely that modellers of varying levels of i* experience may have different approaches to developing i* models and therefore may uncover distinct challenges in their work. By incorporating a range of expertise in the study, it was hoped that a wider variety of challenges would be uncovered.

3.2.2 Research Instruments

Observation and interview data was captured by the researcher in Microsoft OneNote (Appendix 1-B). A new OneNote file was created for each observation session. Within OneNote, different sections of the document were dedicated to different types of data that were anticipated to surface during the observation sessions. Observations, Key Observations, and Key Tasks were used to categorize data. Observations were used to capture the actions taken by the user. If an important observation was noted, it was either directly recorded in the Key Observations section or copied and pasted into the Key Observations section from Observations. If a task conducted by the participants was noted as significant by the researcher, it was placed in the Key Tasks section.

3.2.3 Observation Sessions

Expert Modellers. Three expert modellers were each observed once individually, and once as a group. Each observation session took place within a meeting room on the University of Toronto campus. Consent to participate in the study was obtained at the beginning of the initial observation session.

Novice Modellers. Novice Modellers were observed as a group of three over four separate sessions. Observation took place in a meeting room within the Faculty of Information Studies, at a time convenient to the participants. Consent to participate in the study was obtained at the beginning of the initial observation session.

During the observation of expert and novice modellers, the researcher observed the actual practices of the participants in their domain. The researcher attempted to take the role of the complete observer when observing participants as they used i* tools for work on their current project or assignment. The complete observer does not participate and aims for minimal disruption to the natural setting of the participants, with the intent of capturing the 'true' actions of the participants within the problem domain (Creswell, 2003).

The advantages of complete observation include the ability to gain firsthand experience and record information as it comes available. Additionally, activities or requirements that are implicit to the user may surface during observation that would not be stated explicitly (tacit knowledge). For example, an i* modeller may instinctively organize or rearrange elements within the model without being aware of doing so. However, the observer may be seen as intrusive and thus may cause participants to act in ways they normally would not (Creswell, 2003). Observation notes were taken using OneNote, and when permitted by the participants and deemed appropriate by the researcher, audio recording devices were used to capture the events that took place during sessions.

Each session ended with a brief, open-ended interview. Participants were asked open-ended questions relating to the events that took place during the session. The questions were used to clarify or refine for the researcher any of the interesting items that arose during the session. Questions were asked so that users might explicitly describe various activities that were noted during observation.

Interviews, particularly with expert participants, were used as a means to conduct what Robertson (2001) calls 'Abstraction'. Abstraction requires participants to make useful generalizations about the problem domain. These generalizations are useful for identifying commonalities between various activities. "People who are good at abstract thinking are usually comfortable looking at class models, data models and other models that focus on the "essence" of the subject matter" (2001, p. 3). The expert modellers were skilled in Abstraction, as they were highly educated and had significant experience working with models such as i* models and Entity Relationship Diagrams that focus on the 'essence' of a problem domain.

3.2.4 Data Management and Analysis

The goal of data analysis is to "come up with reasonable conclusions and generalizations based on the preponderance of data" (Taylor & Bogdan, 1984, p. 139). In order to develop conclusions, themes and generalizations about the tasks associated with visual challenges, a number of steps were taken. First, the data was manually organized and prepared for review. Next, the data was reviewed by the researcher. As the data was reviewed, it was reorganized into chunks, depending on the data's subject matter. Data was organized according to categories which surfaced during the researcher's review of the data. Grouping similar data by category helped to abstract common tasks and difficulties from the data. Finally, an attempt was made to interpret these common tasks and difficulties in the context of i* model visualization. These difficulties played a key role in selecting appropriate prototypes for modellers.

3.3 Phase 2 – Exploration of Proposed Solutions

The challenges identified in user observation drove the exploration of potential solutions. Concepts from IV and REV, as well as existing applications in a number of domains were explored with the hope of uncovering potential mechanisms that met requirements for the challenges identified during user observation and interviews.

Domains dealing with large scale models were of particular interest in this phase. In looking for innovative solutions to overcoming the challengers faced, it is often useful to look across domains for inspiration (Preece, Rogers, & Sharp, 2002). Mechanisms included software for domains such as graphic design, architecture, and biomedical research offered potential. These fields must deal with large scale models.

When new mechanisms were discovered, they were considered in light of the unique characteristics of i* modelling. Adaptations or reconfigurations of the techniques were necessary to match the benefits of the technique to the tasks and properties of i* modelling.

3.4 Phase 3 – Prototyping Proposed Solutions

Prototyping aims to test the features and functionality of a mechanism. In Phase 3, low fidelity prototypes were developed in Microsoft Visio, and presented to the user as paper prototypes.

Paper prototyping is a method of communicating the look and behaviour of mechanisms to users (Snyder, 2003). There are many advantages to using paper prototypes for gaining feedback. Such prototypes are particularly useful in gaining feedback from potential users early in the process of design without investing a great deal of time and effort in developing robust prototypes. Paper prototypes are also quite useful for facilitating discussion amongst the designer and the user and acting as a proof-of-concept (Preece, Rogers & Sharp, 2002). Paper prototypes remove any of the technical requirements, making these types of prototypes accessible to people lacking technical capabilities. Producing paper prototypes is also helpful for identifying missing functionality and requirements.

Paper prototyping is less useful for identifying issues with interaction, keystroke or mouse errors, or when dealing with long documents and lists. Though interaction is an important part of prototype development, the study of interaction between the mechanisms and between mechanisms and users was outside the scope of this project. Instead, gathering feedback regarding the functionality of the prototypes was paramount in this study. Using this approach, one can determine if the prototypes are useful to modellers before focusing attention on the usability of the prototype.

3.5 Phase 4 - User Evaluation of Prototypes

In order to determine the utility of the prototypes developed in Phase 3, a second user study was conducted. This study aimed to qualitatively explore the usefulness of the prototypes developed in Phase 3 by having participants with i* modelling experience review the functionality of the prototypes. The study aimed to provoke feedback from the participants relating to the usefulness of the prototypes for various i* model tasks relating to the analysis of i* models.

The study conducted in Phase 4 explored the usefulness of the prototypes developed. Because this Phase aimed to gain feedback in regards to the functionality and utility of prototypes, paper prototypes seemed best suited to the task.

3.5.1 Participants

The participants for this study were recruited from the Faculty of Information Studies (FIS), based on their prior experience with i* modelling. In order to recruit participants, an email was broadcast to the FIS community via the FIS Student's Council mailing list. Those interested in participation were directed to respond to the researcher via email. The researcher explained in the broadcast email that participants would be paid $20 for the one-hour session and participation in the study would be confidential. Participants were required to be familiar with the i* modelling framework in order to be eligible for participation. Students who participated in the observation and interview phase of the project were not eligible for participation in this study, as their previous experience and discussion with the researcher in regards to the project may have been a source of bias.

Those who responded to the call for participation were pooled together to form a group and selected for participation at random by the researcher. The researcher then corresponded with each participant individually to set an interview time convenient for both the researcher and the interviewee. The FIS 1341 course instructor played no role in the recruitment of students.

The nature of the study was not to obtain information about the interviewees, but to understand how participants felt the prototypes could be used for i* modelling and how i* models could be improved.

No personal information was collected. At the beginning of each interview session, the researchers explained that participants could decline to answer any question they felt uncomfortable answering.

3.5.2 Procedures

After obtaining consent for participation and audio taping of the session, each interview began with the interviewer asking the participant about their experience with i* modelling and the types of software they had used to conduct i* modelling.

For each of the six prototypes developed, participants were provided with an explanation of the prototype's functionality, read by the interviewer from a script. The interviewer then presented each prototype to the participant using paper prototypes. The participant was free to ask any questions about the prototype relating to its functionality. If the interviewer felt that responding to a particular question might influence the participant's opinion of the prototype, he declined to answer the question. Prototypes were presented in random order to each participant, and it was explained to the participants that prototypes were presented in no particular order. This step was taken so as not to suggest the superiority of one prototype over another. Once a prototype was presented to the participant, it was placed in front of the participant, allowing the participant to revisit any prototype at their convenience.

Once all prototypes had been presented to the participant, the interviewer presented the participant with a case study. The case study took the form of an i* model, produced by a KM Lab modeller. The case study's data was used as a basis for building the prototypes as a means of providing context for the prototypes, with the intention of clarifying each prototype's functionality and purpose. The case study i* model was likely to be familiar to some participants, as it was presented in FIS 1341 by the teaching assistant as an example of i* modelling. The case study also provided a consistent set of data with which to present the prototypes to all participants. Using one set of data for all participants aided in the consistent presentation of each prototype and its functionality. The case study's SD model was presented to participants. The SD model was less complex than SR models and simplified the presentation of the case study with the intent of reducing the amount of time participants would need to gain familiarity with the different aspects of the model. The lack of coverage of SR models in this phase is a limitation of the study, further discussed in Section 5.2.

The interviewer explained the case study to the participant and allowed the participant up to five minutes to review the initial i* model (the Montreux Jazz Festival i* model, (Appendix 3-J) constructed for this case. The user was then informed that he or she could ask any questions pertaining to the contents of the model, to ensure understanding.

Following the review of prototypes and the i* model case study, the interviewer posed four tasks to the participant. The first task required participants to analyze the relationship between the actors 'MJF Artists' and 'MJF'. The second task asked participants to compare the relationship of 'MJF' to 'Ticket Corner' with the relationship of 'MJF' and 'General Festival Partners'. The third task required participants to determine the most important actor in the model. Finally, the fourth task required participants to determine all goals for the actor 'MJF'. Each task related to analysis of i* models and were drawn from the modeller observation study conducted earlier.

The interviewer began by explaining each task to the participant. The participant was allowed to ask for clarification about the task. When the objective of the task was clear to the participant, the interviewer then asked the participant to review the prototypes and select the prototype they felt would best assist them in completing the task. The participant was able to ask for clarification about the functionality of the prototype; in this case, the researcher would re-read the functional description to the participant. Following their selection of a prototype, the interviewer then would ask the participant a series of questions related to the suitability of the prototype for the task. For example, how they would utilize the prototype they selected for the task, what improvements they would make to the prototype they selected, and if they felt any other prototypes would be applicable for the task. The prototypes presented to participants, functional descriptions for each prototypes, tasks presented to participants, and questions posed to participants can be found in Appendix 3.

3.5.3 Data Management and Analysis

As stated above, the goal of data analysis was to "come up with reasonable conclusions and generalizations based on the preponderance of data" (Taylor & Bogdan, 1984, p. 139). Interview notes and audio recordings were reviewed by the researcher at the conclusion of all interviews. First, for each task, participant feedback was reviewed to determine the prototypes most often discussed as useful. Second, participant feedback was reviewed to identify the improvements suggested by the participants for each prototype. These suggestions would be useful for iterating the design of the prototypes in future work.

First, the audio recording of each session was reviewed by the researcher. While listening, the researcher took notes for each task and also recorded comments made by the participants regarding each prototype. Once notes were taken from each interview, the researcher grouped similar data in order to discern trends and patterns in user feedback.

4. Results

4.1 Review of Existing i* Tools

To discover novel features for improved visual support for i* modelling, it was necessary to first identify features currently implemented by tool developers. Several tools currently support i* modelling. The primary intent of the tool review was to gain knowledge of how each tool provided support for the analysis of i* models. Less focus was given to the mechanisms that assisted in creating and constructing models.

Tools supporting i* were found at the i* Wiki (i* Wiki: i* Wiki Home, 2006), a website used by the international i* community, dedicated to the collaboration and exchange of i*-related research. Researchers using the i* Wiki have shared information about the tools they have developed for i* modelling. Typically, a tool is available for download with permission of the group responsible for developing the tool. These tools are developed as research prototypes, used for research experimentation and 'proof-of-concept'. One would not expect each tool to possess a comprehensive set of features or be fit for everyday usage. However, the unique perspective and research objectives of each research group contributing to i* tool development did result in unique mechanisms incorporated in i* modelling tools.

Tools were selected for review by visiting the General Information page for each tool on the i* Wiki. These pages are maintained by the developers of each tool and contain relevant information about the capabilities of each tool. If the tool supported the i* notation and the modelling of SD and SR models, the tool was included as part of the review. In two cases, tools that did not directly support the i* modelling framework were included to explore the applicability of novel mechanisms they had developed for i* modelling. The GR-Tool incorporated a tabular view of model data and significant professional development had been undertaken for the TAOM4E tool, which made the tools interesting candidates for review.

Seven tools were included in the review. Each tool reviewed was downloaded and installed on a computer running Microsoft Windows. The researcher then ran the application and interacted with the variety of features each tool incorporated. In addition to interacting with the tool, the documentation associated with each tool was used when available. The documentation outlined the features incorporated in the tool, and in some cases, discussed the rationale for including features. In cases where documentation was provided, the researcher attempted to test each of the features discussed in the documentation.

Shneiderman's taxonomy of seven information-seeking tasks served as a means for evaluating the mechanisms within each tool. Included in this taxonomy are seven distinct tasks helpful to users seeking visual information: overview, zoom, filter, details on demand, relate, history, and extract. Each task represents an action useful to the user of a system when attempting to understand information represented in a visual form. As i* modellers seek to gain insight from the visual information represented in the models they produce and analyze, the tasks outlined by Shneiderman appeared to be quite applicable. The taxonomy served as a method for assessing each tool's features as they related to these visual information seeking tasks.

First, OME3, openOME and the i* stencil are covered. Next, REDEPEND-REACT is discussed, followed by JPRiM. Tools closely related to i* modelling, TAOM4E and the GR-Tool, round out the review.

4.1.1 OME3

OME (v3.13) is a general, goal-oriented and/or agent-oriented modeling and analysis tool developed by researchers at the University of Toronto (Yu, 2000). OME is a tool to aid users in the development of goal-oriented models by providing a graphical interface for which to build these models. The tool is designed to support multiple frameworks and to connect with a knowledge base for sophisticated computer-aided analysis. I* is one of the frameworks currently supported. Other frameworks supported by OME include the NFR Framework, and GRL, a variant of i*. OME is constructed in Java and is executed as a .bat file. To install the tool on a Windows machine, users must download Java Run-time Environment (JRE). Once Java has been correctly installed, users can download the OME .zip file. Users extract the .zip file to a directory and run the 'ome3.bat' file.

OME contains a number of features for developing i* models. OME employs a toolbar with iconic representations of i* element types, such as the i* actor, task, goal and softgoal. Once an icon is selected from the toolbar, the user can click anywhere on the canvas to create an instance of this element on the canvas. When an element has been added to the canvas, it can be repositioned by selecting the element with the mouse and dragging it to the desired location. Deleting elements that have been added to the canvas can be done by pressing the 'delete' button.

OME allows for elements to be contained within actor boundaries, consistent with the method for creating Strategic Rationale models using the i* framework. Elements located within an actor's boundary are shifted as the actor element is moved within the screen, simplifying the movement of these elements within the model.

When users right-click with their mouse on the OME canvas, OME displays a menu of commonly used features in i* modeling. The availability of menu options varies with the elements selected at the time of right mouse click. For example, OME limits the availability of menu options like 'change dependency link' by making them inactive when they are not applicable. Actors' boundaries can be expanded and collapsed to display or hide the elements located within actor boundaries.

The following table categorizes the features of OME3 as they relate to Shneiderman's Visual Information Seeking tasks.

Shneiderman's Visual Information Seeking Tasks	Feature Supporting Task
Overview	N/A
Zoom	N/A
Filter	N/A
Details on demand	Actors' boundaries can be expanded and collapsed to display or hide the elements located within actor boundaries.
History	N/A
Extract	N/A
Relate	When an element has been added to the canvas, it can be repositioned by selecting the element with the mouse and dragging it the desired location, allowing elements to be grouped by proximity to one another.

Table 1: OME3 Summary of Features Supporting Shneiderman's Visual Information Seeking Tasks

4.1.2 openOME

OpenOME was built by researchers at the University of Toronto's Knowledge Management Lab, extending the previous OME version by adding functionality and integrating OME as a 'perspective' within Eclipse. Eclipse is an "open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle" (The Eclipse Foundation, 2007). The purpose of openOME is to extend and improve upon the existing OME application, making it more robust, flexible and open-source.

To install and run openOME, a variety of software is required. First, the Java Runtime Environment (JRE) must be installed on the local machine. Second, users must download the open-source Eclipse software. When both JRE and Eclipse tools have been installed, users have two options. The user can choose to install openOME as an Eclipse Rich Client Platform. This requires users to download the openOME zip file and install it within the 'eclipse' directory. Alternately, users can choose to install openOME as a plug-in.

All of the features included in OME are included in openOME, with some additions. The integration of openOME into the Eclipse environment allows openOME to leverage existing Eclipse features such as cut and paste and the undoing and redoing of user actions. OpenOME also leverages pre-built Eclipse features for navigating models, such as the Eclipse 'hand' tool which is useful for panning across large models. The Eclipse 'zoom' feature allows users to focus on specific aspects of the model when zooming in or view the entire model by zooming out.

A novel feature integrated into the openOME tool is the ability to hide various elements based on some criteria. For example, users can choose to filter out unsatisfied elements from view on the canvas, focusing on satisfied elements and their dependencies. Users also have the ability to select a number of elements they would like to view in isolation and hide all other elements. This may be useful for evaluating specific alternatives, decompositions or relationships amongst model elements.

The following table categorizes the features of openOME as they relate to Shneiderman's Visual Information Seeking tasks.

Shneiderman's Visual Information Seeking Task	Feature Supporting Task

Overview	Hand tool for panning across a model's canvas.
Zoom	Zooming in and out of the model.
Filter	Hide unsatisfied elements or unselected elements from view.
Details on demand	Actors' boundaries can be expanded and collapsed to display or hide the elements located within actor boundaries.
History	Undo/redo of user actions
Extract	N/A
Relate	When an element has been added to the canvas, it can be repositioned by selecting the element with the mouse and dragging it the desired location, allowing elements to be grouped by proximity to one another.

Table 2: openOME Summary of Features Supporting Shneiderman's Visual Information Seeking Tasks

4.1.3 i Stencil for MS Visio*

The i* stencil for Microsoft (MS) Visio was created by the University of Toronto's KM Lab. The stencil was originally created to handle large-scale models, as OME3 suffers performance difficulties as models grow in size. Developers of the i* stencil were able to leverage the stability of Microsoft Visio to better facilitate the creation of large-scale models.

Microsoft Visio allows for customized stencils to be created by users. These stencils can be imported when the MS Visio program has been started. Once the i* stencil is downloaded, it is ready to be imported into Visio.

The i* stencil approach relies on the existing functionality that is included in MS Visio. MS Visio provides the ability to cut and paste objects and undo and redo user actions, as well as drag and drop stencil objects on to the canvas. MS Visio connection point algorithms allow users to easily adjust the position of objects on the canvas without having dependency connections break apart and be redrawn. MS Visio also provides mechanisms such as a 'pan & zoom' window, allowing for improved mobility within the canvas area. However, the i* stencil provides only graphical shapes for the i* modeller, and maintains no underlying metamodel for i* concepts.

The i* stencil supports the construction of i* models by taking advantage of existing Visio functionality. Users are able to construct models by dragging elements from the i* stencil and dropping them onto the Visio canvas. The tool also supports the straightening of dependency links and fully supports the i* notation, including the 'role' and 'agent' elements. Most other tools do not support these element-types. Visio's 'Pan and Zoom' feature is a particularly useful feature for navigating large-scale i* models.

The following table categorizes the features of the i* stencil for MS Visio as they relate to Shneiderman's Visual Information Seeking tasks.

Shneiderman's Visual Information Seeking Tasks	Feature Supporting Task
Overview	Pan & Zoom window for gaining an overview of the model.
Zoom	1. Zooming in and out of the model. 2. Pan & Zoom window for navigating a model's canvas.
Filter	Elements can be assigned to layers; users can choose to make layers either visible or invisible, filtering elements in the process.
Details on demand	N/A
History	Undo/redo of user actions.
Extract	N/A
Relate	1. When an element has been added to the canvas, it can be repositioned by selecting the element with the mouse and dragging it the desired location, allowing elements to be grouped by proximity to one another. 2. Groups of elements can be assigned a 'fill' colour, demonstrating that they are related in some way. 3. Additional Visio shapes can be used as containers, to group elements together.

Table 3: i* Stencil for Visio Summary of Features Supporting Shneiderman's Visual Information Seeking Tasks

4.1.4 REDEPEND-REACT

REDEPEND was developed by a team of researchers and developers at the Centre for HCI Design at City University (Maiden, Jones, & Ncude, n.d.) and continues to be developed as an independent tool. REDEPEND-REACT, developed by researchers at UPC Barcelona, operates as a Microsoft Visio plug-in, is programmed using Visual Basic for Applications (VBA) and extends REDEPEND's functionality in order to support the REACT (Grau, 2006) and PR*i*M (Grau, n.d.) methodologies.

The REACT method aims to "apply i* in the context of components selection, in order to identify, reconcile and evaluate the different, and often contradictory, forces that drive the selection of the software components to meet complex system requirements" (Grau, 2006). PRiM (Process Reengineering i* methodology) contains five phases and uses the i* framework for constructing and analyzing alternative system architectures for a given system.

REDEPEND-REACT (R-R) is a mature prototype supporting the definition and evaluation of architectural properties, where the information system to be evaluated is modeled by using the i* framework. This tool expands on REDEPEND, a Microsoft Visio plug-in that enables users to graphically model system goals using the i* formalism (Grau, 2006). In this review it is not possible to distinguish between the features of REDEPEND and REDEPEND-REACT, as only R-R is available on the i* wiki, while REDEPEND is not.

REDEPEND-REACT(R-R) is used as a plug-in for MS Visio. The tool includes a number of stencils, and macros created using Visual Basic for Applications (VBA) are used to implement i*-specific features for the Visio environment. R-R supports construction-phase activities in many ways. The tool takes advantage of built-in Visio features such as dragging and dropping of i* elements from the stencil onto the canvas, the ability to cut and paste single and multiple elements, undo and redo user actions.

R-R also extends the functionality of the tool beyond Visio's standard functionality by creating specific features for improved visualization of the models. R-R supports the automatic reorganization of i* elements within the canvas. Users are also able to duplicate the current page by right-clicking on the Visio canvas. The main advantage to duplicating the page is the persistence of stencils and functionality associated with R-R from the current page to the new page.

By recognizing the importance of decomposition in SR models, R-R enables users to right-click on an element and add a decomposition link and child element, reducing the number of clicks necessary for creating these relationships. R-R also incorporates a number of tools for evaluating models. Users can change the direction of dependencies by right clicking on these types of elements and selecting this option from the menu. R-R allows users to convert SR models to SD models, hiding some of the complexity of the model and allowing for REACT evaluation techniques to be applied to the model.

R-R possesses the ability to evaluate current and alternative models for certain properties contained within the model. As well, alternative models can be generated by R-R using an 'architectures wizard' feature. The wizard solicits users for input as to how models should be restructured and generates the resulting alternatives. To improve visualization and reduce complexity, these alternatives are also easily deleted if incorrect or unused.

The following table categorizes the features of REDEPEND-REACT as they relate to Shneiderman's Visual Information Seeking tasks.

Shneiderman's Visual Information Seeking Task	Feature Supporting Task
Overview	Pan & Zoom window for gaining an overview of the model.
Zoom	1. Zooming in and out of the model. 2. Pan & Zoom window for navigating a model's canvas.
Filter	Elements can be assigned to layers; users can choose to make layers either visible or invisible, filtering elements in the process.
Details on demand	N/A
History	Undo/redo of user actions.
Extract	1. Automated generation of requirements from the model to alternate formats, such as MS Excel and Word. 2. Extraction of SD model elements to produce SD model from SR model.
Relate	1. When an element has been added to the canvas, it can be repositioned by selecting the element with the mouse and dragging it the desired location, allowing elements to be grouped by proximity to one another. 2. Groups of elements can be assigned a 'fill' colour, demonstrating that they are related in some way. 3. Additional Visio shapes can be used as containers, to group elements together.

Table 4: REDEPEND-REACT Summary of Features Supporting Shneiderman's Visual Information Seeking Tasks

4.1.5 JPRiM

JPRiM was developed by researchers at Universitat Politècnica de Catalunya. JPRiM is created to support to the PRiM method for reengineering processes, "where the specification of a new system starts from the observation of the current system and ends with the achievement of the specification of the system-to-be" (Grau, n.d.). PRiM is an iterative process consisting of five phases. The JPRiM tool is used to support each phase and approaches the evaluation and generation of alternative systems from a textual perspective.

JPRiM represents i* models in a non-traditional format; utilizing tabular formats and tree hierarchies to represent various process activities, associated actors, and i* models. The traditional approach to i* modelling – SD and SR models constructed using i*'s graphical notation – is not supported.

The tool prescribes a methodology that allows users to define the contents of i* models, construct i* models, generate alternatives to the current process, and evaluate i* models following the PRiM methodology.

JPRiM is constructed on the Eclipse platform and uses MySQL as a back-end database. First, users must download and install MySQL. Once installed, users must download the JPRiM .sql file and construct the database using the MySQL command line feature. Following the successful construction of the JPRiM database, users must download and unzip Eclipse. Once Eclipse has been successfully downloaded, users should download the JPRiM tool.

The first phase of the PRiM methodology requires users to analyze the current process and define activities and the associated actors within the process. Users input process information into Detailed Interaction Scripts (DIS). These DIS organize information on an activity-level, relating actors, resources, triggering events, and conditions to the activity (Grau, n.d.).

In this phase, i* models are generated based on the data entered during analysis phase activities. These models are represented using Eclipse's tree-hierarchy feature, rather than graphical notations. The rationale for a structured, hierarchical representation is to better manage i* models as they grow in scale. JPRiM developers suggest this representation may improve the ease with which modellers can focus on specific elements of the models (Grau, n.d.). JPRiM allows for the generation of alternative processes to the current, and the ability to run evaluations on both current and potential configurations. JPRiM uses the "reasoning capabilities provided by i* to explore different ways to carry out a process" (Grau, n.d.). The tool does this by establishing what actors should be added in a reconfiguration, and how responsibilities and dependencies amongst actors should change.

Tradeoff analysis is used to determine the most viable alternative during evaluation. Users can choose what properties should be evaluated, and what metrics should be used for evaluating these properties. Metrics can be either actor-based or dependency-based (Grau, n.d.).

The following table categorizes the features of JPRiM as they relate to Shneiderman's Visual Information Seeking tasks.

Shneiderman's Visual Information Seeking Task	Feature Supporting Task
Overview	Collapsed tree-hierarchy for overview of model contents.
Zoom	N/A
Filter	N/A
Details on demand	Expand and collapse tree-hierarchy for element details.
History	N/A
Extract	1. Generation of alternative processes to the current process
Relate	1. These Detailed Interaction Scripts organize information on an activity-level, relating actors, resources, triggering events, and conditions to the activity 2. Tradeoff analysis is used to determine the most viable alternative during evaluation

Table 5: JPRiM Summary of Features Supporting Shneiderman's Visual Information Seeking Tasks

Related Tools

Aside from tools directly supporting the i* framework, other tools closely related to i* are also worth reviewing. These tools support goal-oriented modelling following the Tropos methodology (Bresciani et al, 2004; Tropos, 2007), and present unique mechanisms that have the potential to be quite useful when considered for i* modelling software.

4.1.6 TAOM4E

The TAOM4E (Tool for Agent Oriented visual Modeling for the Eclipse Platform) tool was developed by researchers at the University of Trento. TAOM4E was developed to support the *Tropos* (Bresciani et al, 2004) methodology. Tropos provides a "modelling language based on a multi-agent paradigm; it supports analysis techniques and a structured modeling process" (Bertolini et al, n.d., p.1).

The tool was developed in Eclipse, because Eclipse offers a "flexible solution to the problem of component integration" (Bertolini et al, n.d., p.1).

There are a number of technologies needed to run TAOM4E. First, Java JDK 1.4.2 must be installed. Eclipse software must also be installed. Further, two Eclipse extensions must be installed. First, the Eclipse Modeling Framework (EMF) must be installed. This tool allows for applications and tools to be constructed using a structured data model, in this case XMI (XML Metadata Interchange). Also required is Eclipse's Graphical Editing Framework (GEF). The GEF "allows developers to take an existing application model and quickly create a rich graphical editor" (The Eclipse Foundation, 2007c). The GEF contains a number of features useful for modeling, such as the Palette tool, the 'tree' representation of data, the overview window (a thumbnail of the model within scrolling), for example.

Though TAOM4E does not directly support all aspects of the i* modeling framework, many of the features it has incorporated could be very useful for i* modelling. The Palette tool allows users to easily access and select elements and instantiate them on the TAOM4E canvas. TAOM4E also takes advantage of many of the features already contained within Eclipse. Cutting and pasting elements, selecting multiple elements and undoing and redoing user actions are gained by developing TAOM4E within Eclipse.

TAOM4E also uses the GEF's tree-hierarchy to visualize the model as a hierarchy of elements, providing users with a different view of the model's data. Each node in the tree can be expanded and collapsed, allowing users to focus on specific elements. A 'properties' window in the bottom of the tool is useful for providing further details about the selected element.

Finally, TAOM4E represents different types of elements with different colours and allows modellers to insert their comments within models. The use of colour is another feature that may be useful in i* modeling. Colours may help modellers to distinguish items from one another or to associate various elements with certain alternatives and processes developed by modellers.

TAOM4E allows users to add comments, which may be useful for describing the rationale of the modeller when creating some aspect of the model. This is particularly useful if models are being developed in groups or over a long period of time. Modellers can also review comments made by other modellers or themselves.

TAOM4E uses the GEF's 'panning' window for model navigation. This feature provides a small image of the entire model and an area of focus. Moving the area of focus in the small image also moves the full-size model to the same location. This feature is different from Visio's 'Pan and Zoom' feature because it only allows for panning. Zooming can be done only via a drop-down menu located on the top toolbar within Eclipse.

The following table categories the features of TAOM4E as they relate to Shneiderman's Visual Information Seeking tasks.

Shneiderman's Visual Information Seeking Task	Feature Supporting Task
Overview	Panning tool used to navigate the model
Zoom	Drop-down zoom menu allows users to select percentage of model being shown.
Filter	N/A
Details on demand	Properties window provides further details about an element that is selected by the user.
History	Undoing and redoing user actions.
Extract	Generation of alternative processes to the current process
Relate	TAOM4E assigns different default colours to different i* element-types, making it easy to identify different elements.

Table 6: TAOM4E Summary of Features Supporting Shneiderman's Visual Information Seeking Tasks

4.1.7 GR-Tool

The GR-Tool (GR-Tool, 2006) was also developed at the University of Trento. This tool supports the Tropos methodology, and provides users of the tool with the ability to conduct forward and backward reasoning on the goal models that have been developed. The GR-Tool was developed using the Java programming language, and runs as a .jar file. Users must have the Java Run-time Environment (JRE) installed. In order to conduct top-down analysis, Lindo (Lindo System's index page, 2006) must also be installed.

Though the GR-Tool tool does not fully support the i* modeling notation, many features included within the tool may be useful in supporting modellers using the i* framework. In one view, the GR-Tool represents model data in a tabular form. Though there is no direct connection between the tabular mechanism employed in the GR-Tool and i* modelling, Maiden et al (n.d.) suggest that dependency tables may be useful for creating and viewing the dependencies between elements within an i* model.

The GR-Tool is similar to other tools in that it employs a toolbar with iconic representations of various elements commonly used when modelling. An innovative feature incorporated by the GR-Tool is the use of an 'element pane'. When an element is selected from the model on the canvas, the object pane displays further information about the element, such as related elements (visualized using a tree hierarchy). The GR-Tool supports forward and backward reasoning over goal models. While not directly applicable to i* models, the visualization of evaluation results may be useful. Other tools such as JPRiM and REDEPEND-REACT incorporate evaluation features, and may benefit from the visualization techniques used by the GR-Tool.

The following table categorizes the features of the GR-Tool as they relate to Shneiderman's Visual Information Seeking tasks.

Shneiderman's Visual Information Seeking Task	Feature Supporting Task
Overview	Represents model data in a tabular form, user can scroll through tabular data.
Zoom	Drop-down zoom menu allows users to select percentage of model being shown.
Filter	N/A
Details on demand	When an element is selected from the model, details of the element are displayed in the 'object pane'. Further element details can be added through the object pane.
History	Undoing and redoing user actions.
Extract	N/A
Relate	1. Relates model data using tabular form, in addition to traditional graphical notation. 2. Forward and backward reasoning can be conducted for evaluation of the model.

Table 7: GR-Tool Summary of Features Supporting Shneiderman's Visual Information Seeking Tasks

	OME	openOME	i* stencil	REDEPEND-REACT	jPRIM	TAOM4E	GR-TOOL
Overview	◔	◔	◕	◕	◕	◕	◕
Zoom		●	●	●		●	●
Filter		◔	◑	◑			
Details on demand	◔	◔			●	◑	◕
History		●	●	●		●	●
Extract			◑	◑	◑		
Relate	●	●	◕	◕	◕	◕	◕

◔ ◑ ◕ ●

Weakly supports ◄———————► Strongly supports

Table 8: i* Tool Review Summary Describing the Degree to Which Tools Support Shneiderman's Visual Informaiton Seeking Tasks

4.1.8 Conclusion

The review of tools yielded useful insights into the current state of i* tools. A review of table 8 suggests few tools support the tasks of Filtering and Extraction, while most tools make an effort to provide users with features that support the task of Zooming. Table 8 describes the degree to which each tool supports the various Visual Information Seeking tasks, according to Shneiderman's taxonomy. Appendix 2 summarizes the capabilities of the tools in comparison to the seven visual information seeking tasks.

At first glance, the summary table suggests that most tools contain features or mechanisms that address a high number of visual information seeking tasks. However, the adequacy of these mechanisms for users of the tool is not clear. To assess the utility of the mechanisms and suggest alternative or new techniques for improving i* model visualization, it is important to understand how users interact with the features.

Though the tools supported i* modelling, few tools provided extensive i*-specific features for the user. Tools built in Visio and Eclipse took advantage of standard features such as undo and redo, for example, but did not offer features unique to i*'s needs. Tools that did provide extended functionality, such as REDEPEND-REACT, focused mostly on features that produced automated evaluations of the models. While useful, these tools did not provide features aimed at improved model clarity.

The review of i* and related tools above demonstrates that tools typically focus on providing features for evaluation in relation to analysis. However, evaluations are generated automatically in JPRiM, R-R and GR-Tool, allowing users to analyze the results of the evaluations.

4.2 Analysis of User Observation Sessions and Interviews

This section presents findings regarding the challenges novice and expert users experienced when producing i* models. From observation and interview sessions, various challenges emerged. First, the challenges discovered when observing 'novice' participants are discussed. Second, challenges that arose during expert observation and discussion are described. Third, an effort is made to document the challenges common amongst novice and expert users are discussed.

4.2.1 Novice Participants

The 'novice' participant group was comprised of three students from FIS 1341. The students worked together on a course project, in which i* modelling was used to analyze a number of actors in a business environment. Students were to model the current environment and suggest potential reconfigurations amongst relationships in order to better satisfy the goals of the actors involved.

Interviews with the participants indicated that none of the students had used the i* modelling technique prior to the course project. Prior to the course, one participant possessed experience with other modelling techniques, such as Business Process Modelling. The two other students had not used modelling techniques prior to the course.

Four observation sessions were conducted with these participants during the month of November, 2006. Each session lasted approximately three hours. For three of the sessions, screen capture software and audio recording were used, and notes were taken by the researcher. In the other session, only notes were taken because the researcher was unable to secure the necessary equipment for this session. Participants were instructed at the start of the first observation session to work as if a researcher was not present.

The first three sessions concluded with informal interviews of the participants by the researcher. In a group, participants were asked to describe their experiences during the session. If additional questions arose or participants tired, questions were placed in a document and sent to the participants, allowing them to complete the questions individually on their own time. An effort was made to conduct the interviews in-person rather than via email.

For each session, the students worked as a group. The group chose to produce their models using the OME tool. The group explained that this tool had been recommended for use by the course instructor and teaching assistant. One student was responsible for interacting with the OME via a laptop, while the other two students made comments and suggestions about what should be modeled, and how. For each session, the group used a projector to display the models on the wall. The group explained that it was easier to see and discuss the models when projected on the wall, rather than huddle around the laptop screen.

From the novice participant observation and interview sessions, three themes emerged. Themes were discovered by reviewing notes taken during the sessions, grouping similar data, and identifying prominent trends or patterns in the data. Each theme represents a broad challenge encountered by the participants while using the i* modelling framework to complete their course project. The challenges are presented in no particular order.

4.2.1.1 Challenge 1: Developing Strategies for Model Organization

Model organization proved to be a significant challenge for the novice participants. The central task in organizing the i* model required novices to decide how elements would be arranged. This included determining how elements should be physically laid out on the 2-dimensional canvas, the amount of space between the elements, and what elements would be positioned in the same area of the canvas. During the final observation session, modellers complained that a great deal of their time is spent rearranging and organizing the model, as opposed to further developing and analyzing the model. To address these challenges, modellers employed a number of techniques in an attempt to organize the model in a coherent fashion.

Before organizing the model, the participants carefully considered what elements would have relationships, and grouped these elements together. The modellers placed elements they felt had strong ties to one another close together on the canvas. When elements were not perceived to have a strong connection, they were placed further apart. For example, one group of closely-related elements was placed in the top left corner of the canvas, while another group of elements was arranged in the bottom right corner of the canvas. As models grew in size, a coherent organization scheme for the model became increasingly difficult to maintain. In reference to this issue, one modeller suggested that, "this is going to be messy, and there's just no way around it." Another modeller remarked, "we need to look at strategies for getting it all in there… think about this strategically".

Early in modelling, the participants attempted to organize elements and dependency links in a way that avoided intersection between links. This too became more difficult as the models grew. Modellers realized that "lines are going to have to cross".

Another interesting technique employed by the novice group when modeling involved the arrangement of certain groups of elements so as to describe their meaning. One modeller explained that he was attempting to create a visual hierarchy within the model, in order to convey the increased importance of some goals relative to others. He noted that in doing so, it may be possible to understand the relationship between elements without using links, thus reducing the clutter of the model and improving visual clarity.

It is interesting to note that all of these strategies for building the models were developed by the group *ad hoc*. These techniques for organizing the models were not prescribed by process or instructions for conducting i* modelling; such instructions do not exist. Modellers spent considerable time and effort developing their own strategies for organizing and visualizing the models to reduce the cognitive load or "working memory" required for remembering where elements were within the models they created.

When analyzing the model, the modellers would undertake significant effort to restructure the model to improve the clarity of dependency links when reviewing the relationship amongst actors. If many lines crossed one another, the participants would attempt to move dependency links and elements that were not currently under consideration so that a clear path between the two actors was clearly visible.

4.2.1.2 Challenge 2: Displaying the Contents of a Model

The second challenge novice modellers encountered related to the number of model elements that could be displayed on the screen at one time. Modellers found the OME tool would not support zooming in or out on the model. "Can we zoom? [OME] doesn't let me zoom".

This was not problematic early in the project, when the model size was small enough that all elements were capable of fitting within boundaries of the screen. However, as the model grew, the canvas size required to hold all model elements quickly outgrew the size of the screen. As a result, modellers could view only a portion of the overall model they were constructing. This presented significant challenges to the modellers.

The inability of the tool to display all parts of the model at once required the participants to maintain a mental model of the i* model's structure. Recalling where certain elements were within the model allowed the participants to scroll to an element when necessary, such as when an element was to be included in a dependency link or was under analysis by the group. According to one member, this task put "a premium on working memory". Having significant parts of the model off-screen, "increases the cognitive load because everything you're considering isn't on the screen at once".

Modellers often discussed the desire to view the model from an overall perspective. "I would like to see the whole picture, how things fit together". When analyzing the model, they mentioned that they were confident they had addressed all dependencies and goals between the actors they could see together on screen, but were concerned that some dependencies may have been missed between actors that were not on viewable on the screen at the same time. For example, if actor A and actor B both were placed on the canvas in a way that was viewable to the participants, they were confident they had identified all dependencies between the two actors. However, if Actor A and actor C were far apart in the model, such that significant scrolling was required to view the relationship between actors, the participants felt it was more likely they did not consider and document all dependencies.

Another difficulty as a result of increasing canvas size related to the difficulty in tracing links. Dependency links may span a distance greater than the screen size, requiring the modeller to scroll through the canvas in an attempt to follow this link. Tracking these types of links proved difficult for the participants.

A feature in OME allowed users to hide the Strategic Rationale part of the model. When used, the feature would 'collapse' all elements contained within an actor's boundary, hiding them from sight. The participants originally attempted to use this feature as a means of reducing the complexity of the model and identifying relationships between various actors, but new challenges arose. Collapsing the actor boundaries did not adjust the position of other elements in the model. As a result, collapsed elements maintained their original positions, and scrolling was still required to analyze relationships. Adjusting the positions of these elements would cause significant problems when expanding the actor boundaries, as the layout and dependencies within would now be affected. "Decomposition doesn't really help. You can collapse a number of actors and still not see all of the overall (parent) level; to do this you would have to rearrange all the closed elements and it would screw everything up when you opened them back up again".

When analyzing the contents of a model, novices mentioned that expanding and collapsing was useful for reducing the amount of information present, allowing them to focus solely on the dependencies between actors. The participants collapsed all actor boundaries to review the Strategic Dependency model.

4.2.1.3 Challenge 3: Incorporating External Aids for i Modelling*

The participants relied upon a variety of external aids to assist in i* modelling. These aids helped participants in a variety of ways, such as placing the i* constructs into a format they understand, fact checking, and compensating for missing tool functionality.

The novices first used a spreadsheet to create a table of dependencies to determine the goals, actors and dependencies to be included in the i* model. The participants' first experience with OME involved translating these relationships established in the spreadsheet into i* syntax. This process was interesting for two reasons. First, the participants used the process of translating and transferring the data in the dependency table to the i* model to discuss the validity of the relationships. In many cases, these discussions led to the modification of relationships. What was represented in the spreadsheet was subsequently changed as modellers referred to another external aid, i* literature, when producing the models.

Novice modellers began the modelling process quite unfamiliar with the i* modelling framework. When modelling, the participants frequently referenced various external documents explaining the syntax of i* models and various modelling rules. As the participants were often unsure if a relationship or decomposition they created was valid, they referred back to original i* papers and course materials on the topic in an effort to confirm their work.

The literature outlining the syntax and rules for i* modelling was consulted frequently by the participants. The OME tool did not contain help content, and thus participants consistently referred to i* papers (Yu, 1994; Yu, 1997), as well as course lecture slides for examples on the common structure of i* models and what types of relationships could and could not be created.

During the phase of the project that required the students to consider alternative configurations between actors, the students decided to use a drawing program to sketch potential reconfigurations. The participants extracted important elements from the i* model and redrew them in the drawing program. One of the participants drew the elements, while the other group members made suggestions. The program allowed the participants to include elements by drawing them freehand and quickly change colours. The drawing program contained no support for i* modelling in regards to graphical shapes or dependency links.

With a concept of the initial model and relationships in each of the students' minds, a number of alternative configurations were suggested. Each alternative configuration was provided with a colour. Students seemed to greatly enjoy this technique. Describing their experience with the drawing programs one student explained, "that was so much easier". "[We were] spending most of our time trying to figure out the tool".

4.2.1.4 Summary

Observation of novice participants uncovered two key challenges relating to the visualization of i* models. First, arranging elements on the model canvas quickly became a challenge for the participants. Second, attempting to display contents of the model became increasingly difficult as the models grew in size. In response to these challenges, the participants developed their own ad-hoc strategies for organizing the model, displaying model content and brainstorming potential reconfigurations of the model, in an attempt to maximize readability and visual clarity. Next the observation of expert participants is discussed.

4.2.2 Challenges for Expert Participants

Three expert modellers volunteered to participate in the study as participants. Participants were recruited by contacting faculty at the KM Lab, who in turn directed the researcher to students in the midst of a study using i* modelling. Two 'experts' were Ph.D. students at the University of Toronto's Department of Computer Science. The third participant was conducting post-doctorate research at the Computer Science department. All three were members of the University of Toronto's Knowledge Management (KM) Lab. The KM Lab's research focuses on the "representation, organization, acquisition, retention and analysis of knowledge" (Lapouchnian, n.d.).

Each participant had extensive experience using the i* modelling framework. Two participants had taken part in previous i* modelling projects with external organizations. The i* modelling framework was the focus of one participant's thesis. All participants had strong backgrounds in using other modelling techniques, such as Data Flow Diagrams (DFD), and Unified Modelling Language (UML).

The three participants were observed as they conducted an analysis of an external organization. The central purpose of this analysis was to identify areas within the organization where knowledge is transferred, and propose methods for best capturing this knowledge. i* modelling was a central component of this analysis.

In earlier stages of their study, the participants had interviewed various stakeholders within the organization. One member of the group was responsible for conducting the interviews. A second member captured the stakeholders' responses by taking notes. The third group member rapidly constructed initial i* models using the i* stencil for Microsoft Visio. All interviews the experts conducted were audio recorded.

When stakeholder interviews were complete, the group members met to discuss what types of information should be considered 'in scope' for their project. Modellers intended to limit the scope of their modelling to focus on the central issues of their project.

Once the scope for the project had been defined, the expert modellers decided to divide the initial models amongst themselves. Each group member was responsible for reviewing a set of the initial models constructed during stakeholder interviews and revising them according to scope agreed upon by the group. Once all models had been reviewed and revised, the participants met as a group in an attempt to combine the individual components into four overall models. Each overall model was intended to represent one of the four areas of focus defined by the group.

For this study, each expert participant was observed individually as they reviewed the initial models constructed during stakeholder interviews. Each observation session lasted approximately three hours. For each of the sessions, notes were taken by the researcher; audio and video recording devices were not used. Participants were instructed to work as if a researcher was not present, however, in all cases participants voluntarily 'thought aloud' during the sessions. Finally, participants were observed as they worked in a group. During this session, participants attempted to merge the individual models together.

For each of the individual observation sessions, participants were observed as they reviewed the models they were responsible for revising. In each session, modellers revised the models by listening to the audio recording of the stakeholder interview and reviewing the model to make changes. Prior to listening to the audio recording, participants each familiarized themselves with the model. To familiarize themselves with the model, participants rearranged and untangled model elements, spreading them across the canvas. Each participant used the i* stencil for MS Visio to support their i* modelling.

Once the participants appeared comfortable with their overall understanding of the model, they listened to the audio recording corresponding to the model. While listening to the interview, each participant edited the model, rearranging, renaming or deleting elements. Two of the participants used a colour scheme to signify an element's relation to a particular scope category. The other participant created large blocks for each area of scope and placed relevant elements within these blocks.

4.2.2,1 Challenge 1: Scoping and Selective Pruning of Data

The experts collected a great deal of information from organizational stakeholders during their stakeholder interviews. Each of these interviews was audio recorded by the expert group. During the stakeholder interviews, one expert attempted to capture data provided by each stakeholder 'on-the-fly' in an i* model. When all stakeholders had been interviewed, a great deal of data had been captured using through audio recordings and the i* models constructed during the sessions. The experts hypothesized that a great deal of the data captured was unlikely to directly apply to their area of focus, namely knowledge transfer within the organization. Experts also believed that retaining additional data within the i* models would add clutter and make it more difficult to work with the models and gain insights useful for developing new approaches to capturing knowledge transfer.

After the initial models were constructed during stakeholder interviews, the expert participants relied upon external aids for defining what data should be captured in the models, and what could be removed. The experts decided to include data that related to 'knowledge transfer agents' (KTAs), as the project was focused on the current ways knowledge is transferred within the organization and methods for improving it. The experts conducted this analysis and scope definition without the assistance of the i* stencil tool. A text document setting out the scope and summarizing key discussion points was developed by the group. This document was used by each member as a reference when individually reviewing and revising the i* models. The scope document was developed prior to the participants' involvement in this paper's study.

Each expert relied upon the scope document and audio recordings from the stakeholder interview when revising each model. Model elements were deleted if deemed out of scope by the expert reviewing the model. However, it is interesting to note that when asked, experts admitted that if they deleted an element later determined as important, it would be difficult to trace the deletion of that element without referring to the previous model. Modellers also mentioned that working individually to revise each of the models required a high level of trust in the other modellers' abilities to prune the models effectively. When deleting elements from the models, the experts consistently referenced the scope document and audio recordings to ensure their decision matched the scope criteria and what was said by the interviewee.

4.2.2.2 Challenge 2: Tracking Issues

The modellers used many different techniques to note issues that arose during modelling. One approach modellers took involved adding comments to models. One modeller used Visio's 'comment' feature, embedding comments within the model. Another modeller simply added a textbox and text near the element being discussed in the comment. The content of the comments often consisted of the modeller's rationale or unresolved issues the modellers had with an aspect of the model. One expert noted problem areas by making the outline of unresolved elements thicker. When the issue was resolved, the modeller restored the thickness to its original state. Modellers used colour to mark elements with "weird things going on". For example, one modeller highlighted an element when they believed the audio from the interview did not correspond to that which was modeled, so the modeller could discuss this issue with the group.

4.2.2.3 Challenge 3: Organizing i Model Data*

Modellers varied in the methods used to organize the models in a visually coherent manner. The first action each modeller took when reviewing a model for revision was to reorganize the model in a way that was meaningful to himself or herself. In some cases, the modellers would first rearrange elements to improve clarity and understand the relationships within the model. Modellers mentioned that reorganizing the model was a good way to remember what the model was about. It seemed likely that while doing so, each was building a mental model of the model's structure. Each modeller then organized the model according to the scope document. For example, two modellers designated a unique colour to represent each key area outlined in the scope document and while reviewing the model, assigned a colour to each element within the model. Another modeller created designated areas within the models for each of the key areas and added text to give a title to each area. Regardless of their approach for identifying elements, modellers grouped elements that were related in scope together in the models. One of the modellers mentioned that organizing the model elements by proximity "eases cognitive processing".

Modellers also grouped elements within actor boundaries to specify importance or relevance to other actors. One modeller discussed his intentions to put the most important goals of the actors at the top of actor boundaries. Other modellers considered the relationships elements had to other actors when organizing internally. Often, elements with external relationships would be placed near the edge of the boundary. Modellers also sought to create hierarchical relationships within the actor boundaries, so that it was easy to visualize the decomposition of goals and tasks.

4.2.2.4 Challenge 4: Model Complexity

All expert participants had a high degree of familiarity with i* modelling. It was clear that experts understood the strengths and weaknesses of the technique, particularly in regards to the tendency for models to quickly become extremely complex and the difficultly in working with models as they grew in size. Particularly complex was the visual aspect of large models. As elements were added and the model grew well beyond what was capable of being displayed on the screen, it grew increasingly difficult to recall the location of elements within the model.

Each expert spent a great deal of time reorganizing model elements for the purpose of improving clarity. One expert described this task as "not separating them in any particular way, [I'm] just trying to make it easier to read and work with later on". Each expert started by moving elements away from the center of the model, enlarging the actual size of the model, but also providing a great deal of space between model elements. One expert paid particular attention to untangling links between elements, for the sake of clarity.

Experts also dealt with visual complexity by strategically positioning model elements manually on the canvas, as discussed above. It appeared that this helped the modellers to construct a mental image of where elements were located within the model, allowing them to navigate to these elements as they revised the model.

Modelling each stakeholder separately allowed the experts to manage the complexity of the models. Modellers were able to divide the models amongst group members for reviewing purposes. This allowed modellers to focus on each stakeholder individually, to ensure the information within each stakeholder's model was accurate (model matched the audio recording) and within scope. Experts mentioned that attempting to model all stakeholders together initially would have been too difficult. Before individual reviews of the models were conducted by the individual members, the scope was set in a group meeting. Additionally, experts discussed the need to trust other group members to do an accurate job of revising models at the appropriate level of scope and detail.

4.2.3 Common Challenges Amongst Novice & Expert Participants

When comparing the challenges encountered by each group to one another, common issues surface. Each group had to decide what to include within their i* models. Both groups spent a great deal of time organizing and grouping model elements and experienced visually complex models. Finally, each group spent time rearranging model elements, particularly in an attempt to analyze the models. To combat these challenges, both groups used features within the tools, external aids and their own problem solving abilities.

4.2.3.1 The Use of External Aids

Both groups relied upon external aids when deciding what data should be included in a model, how the data should be included, and the relationships the data has to other information. The novice group relied upon text description of the case under analysis and a group member's own expertise with the case and processes in place. Prior to modelling, the group created a spreadsheet to help them track the data to be included in the model. The group often referred to various i* literature and examples, such as i* academic papers, course notes and presentations on the i* modelling framework to determine how the data should be included in the model. The expert group also relied upon external aids to help them determine what should be included in their model. The experts agreed upon a scope to limit the data to be included in the models, and outlined this in a text document, listing the four key areas of focus and describing the key characteristics for each. Experts then listened to the audio recording from each stakeholder interview in its entirety as they individually reviewed the models that had been created, referring back to the scope document to determine whether elements were inside or outside the pre-determined scope. While novices relied upon i* literature to help them determine how data should be modeled, experts did not. Instead, experts relied upon their previous i* experience to make this assessment.

4.2.3.2 Organizing and Grouping Model Data

Novice and experts both spent a great deal of time organizing, rearranging and grouping elements within the model. As well, participants spent a significant amount of time considering how these models should be built and developing strategies for laying out the model visually.

Both groups appeared to strongly consider the relationships between elements as they organized the model. Elements that had strong (or multiple) relationships with other elements were positioned closer together, to minimize the length of dependency links across the canvas. Novice users often referred to the spreadsheet when considering the elements most likely to have numerous links between them. Experts appeared to take a more intuitive approach to organizing the model. Dependency links were organized in an attempt to minimize overlap. Minimal overlap improved the clarity of the models, making it easier to trace links between elements.

Both sets of participants also aimed to group similar elements. The novice group, using OME, tended to group by proximity. Similar elements were placed closer together on the canvas, while elements that had less in common were placed farther apart. The expert group, using Visio, also grouped similar elements. The Visio tool allowed the experts to change the colour of elements. Two experts used colours to relate elements to one of the four key scope components. Another expert used other Visio features to create and label boxes for each scope area, then place related elements in the correct area. However, experts encountered some difficulty with this technique, as they noted some elements belonged to more than one scope area. In these cases they chose the most appropriate area for the element, because they could not assign multiple colours to elements.

Both groups employed a hierarchical approach to demonstrate relationships between elements. Expert and novice participants both attempted to demonstrate the relationship between elements using a hierarchical approach. Experts and novices often positioned top level goals near the top of an actor's boundary, with lower level goals, tasks and resources below. This was perhaps because there is currently no way in i* notation nor in supporting tools to designate the importance of one goal in relation to another. Instead, the participants attempted to use a hierarchical approach to compensate.

4.2.4.3 Analyzing the model

In both groups, modellers needed to overcome the visual complexity of the models in an attempt to analyze the models. The analysis of models is particularly of interest in this study, as it seems that this area of the modelling process is a particularly fertile area for incorporating additional features. The central intent of producing models is to gain insight through analysis of the models; mechanisms that aid in the extraction of insight from the models will greatly benefit the user and increase the utility of i* modelling in general.

Both groups spent considerable time rearranging elements in order to improve clarity when assessing the model. Dependency links in particular were moved to minimize overlap. Experts rearranged the models they were reviewing, untangling elements and their dependencies to get a strong sense of the model contents, commit the structure of the model to memory, and to analyze the relationships between model elements.

As elements were added and the model grew well beyond what was capable of being displayed on the screen, it was more difficult for modellers to remember where the elements were within the model. As models grew in size, neither group could gain a legible overall view of the model. The OME tool used by novice participants did not support zooming, and thus the group was only able to view the area of the model that fit within the limits of the screen. To produce a cohesive, overall view of the model, the novices took screenshots of various parts of the model and patched them together using Adobe Photoshop. The experts, using Visio, were able to zoom in and out of the model, and did so often. In many cases, experts would zoom out to maximize the amount of the model viewed on the screen with the text still legible. However, once models reached a certain size, the entire model could not be legibly viewed.

Novices in particular mentioned the difficulty in analyzing the model without being able to easily view the relationships between elements (particularly actors). A great deal of time was spent vertically and horizontally scrolling while attempting to trace dependency links between actors. This increased the cognitive load on the modeller, as they were forced to remember what elements off-screen may affect the linked element. Novices noted that they felt they were missing dependencies because of this. The horizontal and vertical scrolling also made it more difficult to keep focus on the link they were attempting to follow when tracing dependency links through the model.

Relationships between elements that have been placed far apart in the model are more difficult to evaluate, and in some cases, the distance between elements caused modellers to overlook relationships. To compensate, the novice participants relied upon a drawing program to extract the key elements under consideration, and evaluate alternative ways for achieving goals. The novices noted their satisfaction with the speed and ease of this technique, in comparison to OME.

4.2.3.4 Ad hoc solutions to challenges

When the novice and expert groups had issues with the i* modelling framework, both developed ad hoc solutions as a means of dealing with these challenges, using a variety of tools. In some cases, the ad hoc solutions were produced using the i* modelling tools, while at other times external aids were used to facilitate the solution. It appeared that in many cases, if the tool offered adequate support, participants would develop visualization strategies using the tool to overcome the challenge. In situations where tool support was lacking or did not exist, participants relied upon support external to the tool to overcome challenges. The need for participants to create ad-hoc solutions suggests that the tools used during modelling were missing some key mechanisms required to aid the modellers in visualizing model data.

Experts using Visio relied upon a variety of its features for dealing with the challenges they faced. The ability to change element colours and add shapes that did not directly relate to i* allowed modellers to group various elements by their corresponding scope area. Visio also allowed experts to place comments within the i* model as a means of noting the rationale of the modeller or issues that arose during model reviews. One expert took advantage of the ability to vary the thickness of dependency links when attempting to highlight an issue within a model he was reviewing. Changing the thickness allowed the dependency and involved links to stand out from other elements, drawing the modeller's attention to it later. Both groups of users adopted conventions for organizing model elements within the tool by proximity and hierarchically to give meaning to data.

When determining what to include in the model, both groups relied upon external aids. Novice groups used a spreadsheet to organize model data and supporting literature to understand, and validate their model construction, whereas experts relied upon a scope document created prior to model reviews to limit the complexity and scope of their models. When novices attempted to develop and assess alternatives, they relied upon a drawing program to extract key elements within the model. They then discussed, constructed, and analyzed alternatives, using a variety of colours. Experts decided to limit the complexity of models by modelling each stakeholder separately rather than attempting to begin with one large model. Each of the approaches was necessary because tool support was lacking.

Reviewing the results of user observation, many important insights were gained:

- Gaining an overall view of the model was difficult for both groups. Novice modellers could not zoom in or out because it was not supported by the OME tool, while experts decided to model individual actors separately in an attempt to manage the visual complexity of the models. The separation of stakeholders did not allow them a comprehensive view of the model;
- Users spent a great deal of time reorganizing and adjusting the model to improve clarity when assessing relationships between elements or trying to understand how elements were related to one another;
- Users found it difficult to determine the completeness of relationships between elements, particularly when the elements were placed far apart in the model.

4.3 Generation of Requirements for Proposed Solutions

4.3.1 Introduction

The findings above suggest that in order to better satisfy the needs of the modellers, additional features should be incorporated into future i* software that provide them with an overall perspective of the model, enable them to better compare various model actors and allow them to distinguish between the data that is essential and non-essential given a specific context or task.

Shneiderman (1996) proposes a guideline to aid users in their search for information through visualization. He includes seven distinct tasks helpful to users seeking visual information: Overview, Zoom, Filter, Details on Demand, Relate, History, and Extract. Each task represents an action useful to the user of a system when attempting to understand information represented in a visual form. This guideline may be particularly useful for identifying and developing mechanisms to improve the visual information gained by users of i* software. A review of current i* tools and the challenges modellers face when visualizing models suggests that the Visual Information Seeking tasks of Overview, Filtering and Extraction would offer significant benefit to i* modellers.

4.3.2 Overview

In Shneiderman's guideline, Overview refers to the task of providing an overall view of the entire collection. Shneiderman suggests strategies such as including a "zoomed out view with adjoining levels of detail" (p. 339), fisheye views, and a "moveable, field-of-view box to control". The investigation of current i* tool capabilities found i* tools native to Microsoft Visio with the ability to support the task using a pan and zoom mechanism, allowing modellers to quickly navigate around the model.

Novice modellers used a tool that did not support panning thus had difficulty navigating the model. To compensate, modellers scrolled vertically and horizontally across the model. However, they often noted this scrolling to be time consuming and difficult to maintain their focus on the dependency link they were attempting to track through the model.

Modellers also felt it was likely that some key dependencies were missed within their model. This notion stemmed from the inability to quickly move from one actor on one side of the model to the other. In other words, the modellers felt that the further elements were from each other physically on the canvas, the less likely all current or potential dependencies were considered. This was particularly important when developing alternatives to the current environment. To reduce this effect, modellers placed elements they felt were strongly related closely together.

A number of requirements should be considered when developing an overview of the model. First, non-essential detail must be hidden in this view of the model. Hiding non-essential data improves the clarity of the model, reducing clutter. In the case of an overview, non-essential data may be considered anything that does not contribute to providing the user with an overall view of the entire collection. Examples of essential data that modellers may want to consider when attempting to gain an overview involve determining what actors are the most important to the model, what actors depend on each other and the strength of these relationships and the type of environment the model is describing. An additional requirement for an overview of i* models is to provide additional detail, on-demand. For example, an modeller reviewing a summarized view of an i* model may require additional information about an aspect of the model; this information should be readily available.

Key Requirements for mechanisms contributing to the Overview task:

1. Hide data that does not contribute to an overall description of the model;
2. Group similar data and summarize;
3. Display details of hidden or summarized data when requested by the user;
4. Provide an easy transition from this view to the standard view.

During observation, it was clear that modellers gained a strong mental understanding of where elements were located within the model. This understanding was essential to the modeller's ability to quickly navigate the model to locate areas of interest. Therefore, it is important that any graphical reorganization of the model does not disrupt the modeller's mental picture of the model, particularly as it is likely that the modeller would switch quickly between the 'standard' and 'overview' views of the model. A complete reorganization would force the modeller to relearn the locations of model elements for each view, reducing the effectiveness of the technique.

4.3.3 Filtering

Shneiderman describes Filtering as, "filtering out uninteresting items" (p. 339). The act of filtering represents an opportunity for improving the clarity of i* models by removing unnecessary information from the user/modeller's view. Currently, i* tools lack support for filtering features. Tools such as openOME and tools supported by Visio support some means of filtering model data. OpenOME allows users to hide all elements unselected by the user or to hide all unsatisfied goals from the user. Visio supports the use of layers. Model elements can be assigned to layers, and layers can be shown or hidden by the user. However, users must create and specify their own layers. Using the 'layer' feature in Visio opens a dialog box that prohibits the users from other activities while the dialog box is open. To resume interacting with the Visio tool's other functionality, the layer dialog box must be closed by the user.

Both novice modellers and expert modellers would benefit from the ability to filter items, particularly when their goal is to analyze aspects of a model. Modellers constantly rearranged elements not directly related to the elements of the model they were focusing their analysis on at the time. Modellers also spent considerable time and effort considering strategies for minimizing overlap and clutter in the model. Filtering mechanisms may reduce the time spent on such tasks, if the modeller is aware that unnecessary data can be filtered from view without rearranging elements. To move beyond current tool support for filtering of i* models, a number of techniques were explored that have been implemented in other software and domains and prove useful.

Key Requirements for mechanisms contributing to the task of Filtering are:

1. Allow the user to specify the criteria by which to filter model data;
2. Ensure that users understand what will happen when the filtering mechanism is applied;
3. Ensure that user actions can easily be undone or altered;
4. Hide OR reduce the prominence of data that does not match the criteria determined by the user;
5. Display OR highlight data that does match the user-specified criteria;

6. Reduce the space between model elements, while maintaining the integrity of the graphical layout of the model;
7. Provide an easy transition from the filtered view to the standard view;
8. Allow this view to be stored by the user.

4.3.4 Extraction

Shneiderman refers to Extraction in the Visual Information Seeking Guideline as the ability for a user to extract a sub-collection of data from the overall collection. This technique has particularly strong implications for use in i* modelling.

As witnessed in observation, both novice and expert modellers spent a great deal of time attempting to identify, clarify and analyze the relationships between various actors in a system and developing strategies to minimize clutter. In doing so, modellers spent a great deal of time reorganizing the structure of the models to clarify the relationship. For example, modellers constantly adjusted or reconfigured the unrelated dependency links that impeded their view of the dependency links between two or more actors currently being considered. Novice modellers spent a large amount of time scrolling horizontally and vertically across the model in an attempt to analyze the relationships between actors that were far apart in the model. Extracting actors that are physically far apart in the model would alleviate the scrolling currently required.

The ability to extract actors under analysis appears to offer a great benefit to the modeller. In order to produce a prototype to support this task, some of the key requirements need to be considered.

Key Requirements for mechanisms contributing to the task of Extraction are:

1. Allow the user to specify the elements that should be extracted;
2. Display extracted elements in a new window that can be moved around the screen, minimized, or closed by the user;
3. Support multiple windows displaying different extracted elements;
4. For all elements that are extracted, also extract any items directly related to the element;
5. Ensure that user actions can easily be undone or altered;

6. Automate the spatial layout of extracted elements;
7. Provide an easy transition from extracted view of the model to the standard view of the model;
8. Allow the user to save or store each view.

Some of the requirements above require further clarification. First, it is important to allow users to produce multiple Extraction windows. This facilitates the comparison or viewing of two parts of the overall model. In a large model, this might be particularly useful. The next important note involves the automation of layout of the extracted elements. Automation is important in this case because it will allow the reduction of space between the extracted elements. It also seems that automated layout configuration will have a minimal effect on the user's ability to understand the extracted data, due to the fact that the amount of data in the extracted window will be significantly less than the entire model. Storing these views may be particularly useful when presenting model data to those not involved with the project. This technique will allows the modellers to 'carve up' models and display pieces at a time, reducing the initial cognitive load required to analyze a full i* model.

4.4 Prototype Development

As discussed above, the Visual Information Seeking tasks of Overview, Filtering and Extraction offer potential areas for prototype development. In the following section, the six prototypes that were developed are described. Each prototype was developed in Microsoft Visio with the intent that it would be printed as paper prototypes to be used in feedback sessions. The Montreaux Jazz Festival case study presented to participants during the Prototype Evaluation study served as a basis for populating the prototypes. Prototypes were developed only for the SD model associated with the case study, as explained in section 5.2.

4.4.1 Overview

4.4.1.1 Overview (Graphical Format) Prototype

This prototype relates to the Overview task. Many tools supporting i* modelling used a zooming dropdown menu in combination with a panning window. These techniques focused solely on quickly moving through the model *as it exists.* Viewing complex i* models using standard visualizations does not provide a clear, concise view of the contents of the model.

For example, expert modellers with pan and zoom capabilities still spent a great deal of time rearranging model elements, for the purpose of improving clarity when reviewing a model for analysis. When the overarching goal of the modeller is to analyze the contents of an existing model, creating a mechanism that produces a simplified, alternate view of the i* models may prove useful in providing modellers with the content required to make rapid decisions regarding the relationships between various actors in the model, and the strength of these relationships.

When the 'graphical overview' prototype was developed, it was envisioned as a 'view' that could be added to existing i* tools. Users selecting the view would be presented with the Overview in a separate window or tab. The design of this prototype was inspired by the notion that many elements within i* models contain multiple dependency links between one another. Modellers often described following these links through the model as a primary contributor to the clutter and reduced clarity. In order to overcome this challenge, this prototype aimed to merge redundant links between actors into one link and denote the number of links using text, as described in Figures 4, 5 and 6. The prototype is displayed in Appendix 3-C.

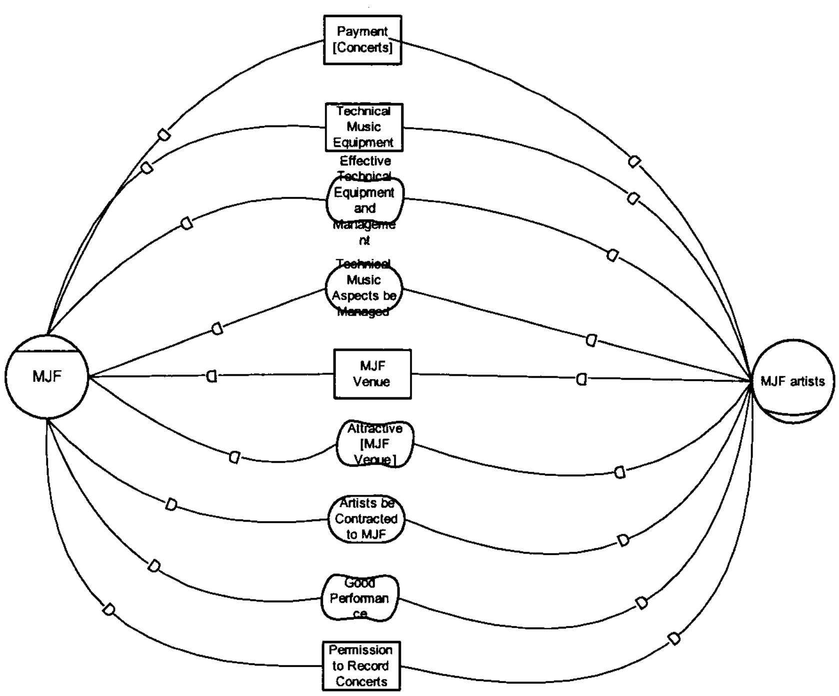


Figure 4: Traditional i* View of Model Using Montreax Jazz Festival Data

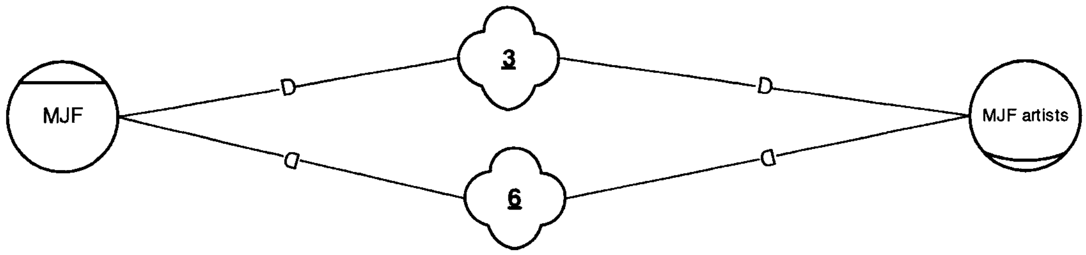


Figure 5: Overview Prototype using Montreaux Jazz Festival Data

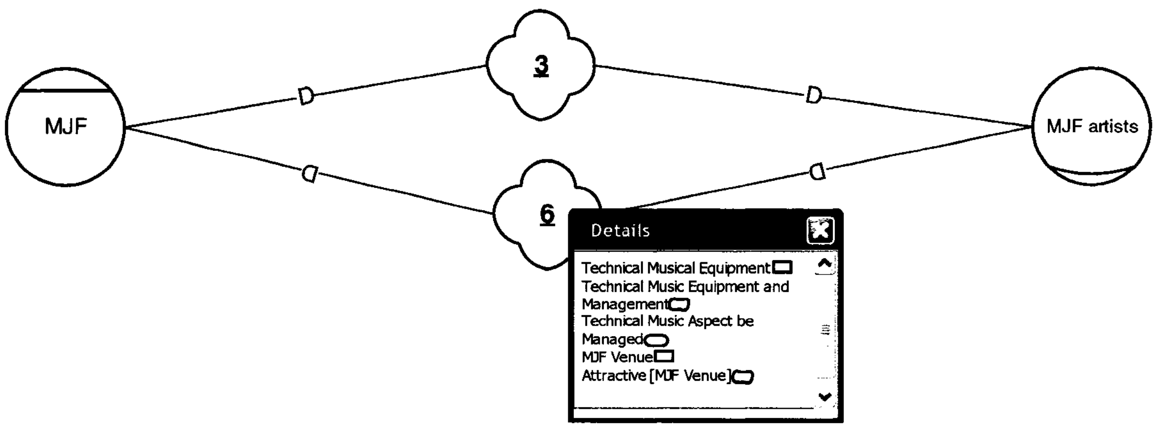


Figure 6: Overview Prototype using Montreaux Jazz Festival Data with Details

4.4.1.2 Tabular View: Prototype

An alternative representation of the model may also prove useful for modellers interested in gaining an overall view of the model. In this case, a tabular format of model data is proposed. A tabular view may satisfy some of the requirements of the Overview and Filtering tasks. It provides a summarized view of data, as well as allowing data to be sorted by various categories. This prototype is inspired by the use of tabular formats by novice modellers using 'dependency tables' as a starting place when beginning i* models and the work of Grau (2006) on JPRIM. JPRIM represents i* models in a tabular format to cope with the increased complexity of i* models as they grow in size and scope.

A tabular view of the model involved converting the visual dependencies to tabular layout. The headings used in the table are 'depender', 'dependee', 'dependum' and 'type', where type refers to the type of the dependum (e.g., goal, task, resource, softgoal). Users can sort any of these headings in descending or ascending alphabetical order. As well, sub counts are produced for each actor, displaying the total number of dependencies they have within the model. Below is an example of the tabular view prototype. The entire tabular view is displayed in Appendix 3-D.

Depender	Dependee	Dependum	Type
Affiliated Festival	**MJF**	**Permission to use MJF Brand and Franchise**	**resource**
Affiliated Festival	**MJF**	**Promotion [MJF Brand and Franchise]**	**soft goal**
	2		
General Festival Partners	MJF	Revenue [Products]	resource
General Festival Partners	MJF	Exclusive Rights	resource
	2		
Individual Consumer	**MJF Covering Media**	**News Media**	**resource**
Individual Consumer	**Montreuxjazz.com**	**MJF Merchandise**	**resource**
Individual Consumer	**MJF Program**	**MJF Information**	**resource**
Individual Consumer	**MJF Sponsor**	**Products**	**resource**
	4		
Infrastructure Partner	MJF	Payment [Infrastructure]	resource
	1		
MJF	**MJF Artists**	**Artists be contracted to MJF**	**goal**
MJF	**MJF Program**	**Information be Provided**	**goal**
MJF	**Ticket Corner**	**Information be Provided**	**goal**
MJF	**Ticket Distributor**	**Tickets be Distributed**	**goal**
MJF	**MJF Food and Beverage Provider**	**Jazz be Used**	**goal**
MJF	**Musical Partners**	**Technical Music Aspects be Managed**	**goal**

Figure 7: Tabular View: Sample using Montreaux Jazz Festival Data

4.4.2 Filtering

4.4.2.1 Filtering (Slicing) Prototype

First, models may be filtered using a technique called slicing. Slicing relates to the task of Filtering because it allows for the filtering of items that are not of interest to the user to be hidden from view. Leica (2005) explored the concept of slicing in relation to addressing the scalability issues of i* modelling. He applied this technique to i* models from top-down and bottom-up approaches. In slicing top-down, all elements that do not relate to the element of interest are filtered out of the model. Bottom-up slicing traces the contributions an element makes to higher level actors. Additionally, he explored model leveling, which took into account both the distance and strength of relationships between elements when slicing the model. A tool that incorporates the slicing feature is the Osprey software package. Osprey (2003) is a network visualization system used to for the visualization of genome biology. The mechanisms incorporated in the Osprey system are particularly applicable to i* modelling, as they both contain network data-types (in reference to Shneiderman's (1996) taxonomy of data-types). Particularly interesting in the Osprey tool is the way in which model slicing is implemented. While Leica explored top-down and bottom-up slicing and model leveling, the filtering (slicing) prototype serves to complement this work by allowing the user to specify the number of elements away from the selected element they would like to filter from the model.

Osprey allows users to filter models in multiple ways. Most interesting in relation to i* is the filter category named 'depth'. Implementing this mechanism allows users to filter elements (in Osprey's case, called nodes) *x* number of nodes away from the selected nodes, where *x* is the number of nodes. Figure 8 below demonstrates the filter's technique.

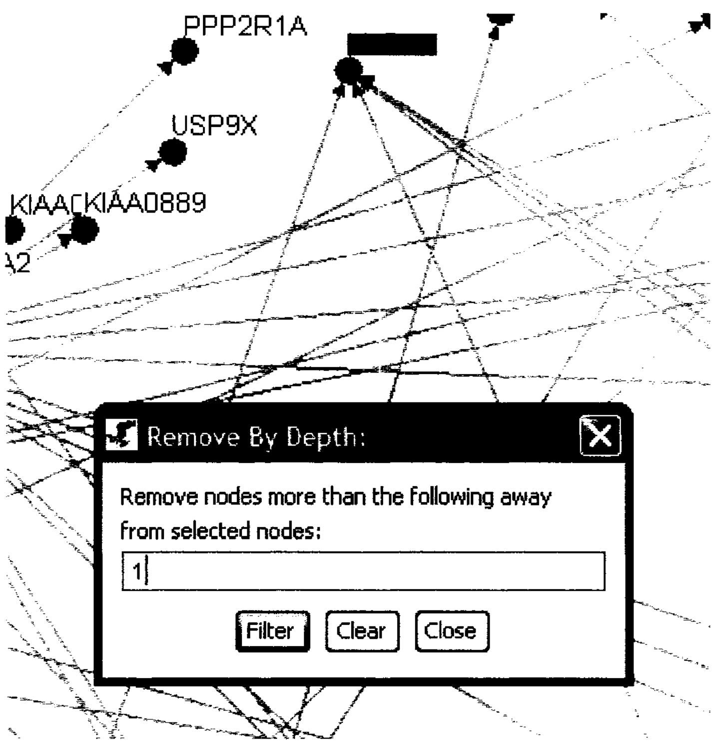


Figure 8: Filtering by Depth in Osprey

This technique could be quite useful for i* modelling, as it would allow users to select one or more elements of interest and explore the relationship with other elements. This would dramatically improve clarity and provide users with a quick and effective way to produce multiple views of the model data. As well, because context is essential when analyzing i* models, slicing allows the user to control the extent to which other elements indirectly affecting the relationships under analysis are displayed. This technique would be particularly useful when the goal of the modeller is to analyze a model.

The slicing prototype closely followed the design of Osprey's mechanism. One outstanding issue to further be explored in user testing is whether to completely filter all model data that does not meet the criteria or to highlight the model data meeting the criteria and reduce the visibility of other data by altering its appearance. Retaining this data provides the user with an increased level of context for the model, but also adds clutter and reduces model clarity. However, it may be possible to alter the appearance of this data so that clarity is improved while maintaining the context of the model's data. Figure 9 displays the key component of the slicing prototype. The Slicing prototype is displayed in Appendix 3-E.

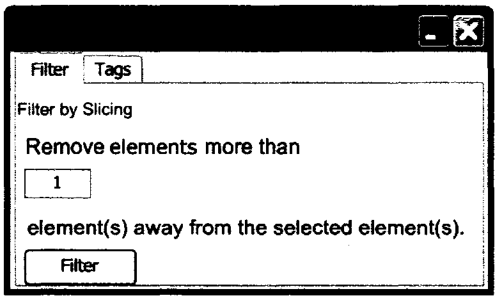


Figure 9: Filtering (Slicing) Prototype

4.4.2.2 Filtering (Layering) Prototype

Another approach to filtering appropriate for i* may be the use of layers. i* tools currently implemented with MS Visio have access to this feature. A layering mechanism allows model elements to be distributed to different layers. Layers are created by the user and elements are assigned to these layers by the user also. Filtering occurs when users alter the visibility of these layers. Making a layer invisible hides all of the contents of the layer from view on the canvas. This prototype relates to the Visual Information Seeking task of Filtering, as it allows uninteresting items to be hidden from the view of the user, so they may focus on items of interest.

Visio has a number of default layers built into Visio and provides users with the ability to create their own layers. Model elements can be moved to these layers by right-clicking on one or more element and assigning it to a layer through the right-click menu 'layer…' option. Users can create and define new layers within Visio. However, interviews with expert modellers familiar with this technique did not use it often because of its time consuming nature. While they recognized the utility of filtering through layers, the time required to assign data to layers in Visio did not justify its use. Figure 10 demonstrates the use of layers in MS Visio.

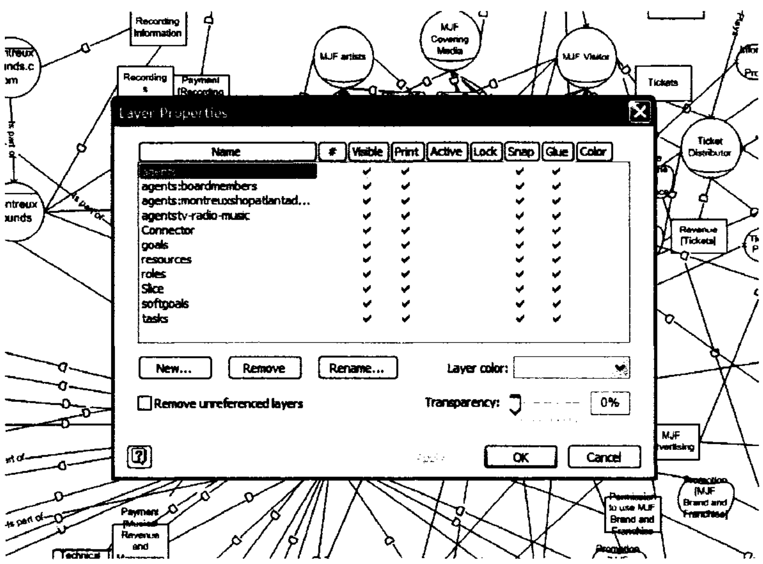


Figure 10: Layering in Visio

Exploring across a number of domains uncovered a variety of products that incorporate 'layering' functionality in their software. Graphic design packages from Macromedia and Adobe (Adobe, 2007) product suites employ layers as a means of organizing objects within a drawing or movie. GIS programs such as ArcView (ArcView: The Geographic Information System for Everyone, 2006) also incorporate layers within their software. AutoCAD, software commonly used for architectural design, also incorporates layers.

Though the domains of these software programs vary greatly, one common theme amongst the tools relates to their need to handle large-scale data. Adobe Illustrator may be used for the design of billboards, ArcView can be used to display large maps, and AutoCAD provides designers with the capabilities to build skyscrapers. As i* models grow in size, they too must cope with the scalability of the models, and likely can borrow techniques from these domains. In this case, the 'layer' functionality for organizing and filtering model data is quite useful.

Adobe's use of layering was particularly useful for i* modelling. The figure below displays the layering mechanism contained within Adobe's Photoshop program.

Adobe's layering mechanism satisfies many of the key requirements derived from observation and interviews of modellers. First, it can be quickly and easily accessed. Users do not have to open a new window to interact with the mechanism and interacting with the layering mechanism does not prohibit users from using other components of the software. This makes hiding, closing or moving something to a layer much easier than the mechanism incorporated into Visio. Layers can be made visible and invisible, layer opacity can be modified, and layers can be grouped into folders. Users can apply opacity and visibility to folders rather than individual layers, and expand or collapse the contents of folder or folders to reduce the amount of space layers take up within the component. This provides a user with control over how much or how little of the model's data is shown.

However, Adobe's layering technique is not without its shortcomings when considered for use in i* modelling. The first shortcoming relates to the inability of adding elements to multiple layers. This may be useful for filtering purposes. For example expert modellers attempting to assign colours to categorize elements often encountered elements belonging to more than one category, depending on the context. Modellers may want to assign the element to multiple layers, and this is not possible using Adobe's technique without actually duplicating the item of interest and placing the duplicated copy on a separate layer. Second, Adobe asks the user to define new created layers. In the case of i* modelling, perhaps it would be useful to have commonly used elements such as goals, softgoals, and resources as pre-existing layers, with the ability for users to add additional, user-defined layers. It may also be useful to have elements automatically assigned to the appropriate layer based on their type (e.g., goal, softgoal, task). For example, if a modeller adds a 'goal' element to the canvas, it may be useful for the goal element to automatically be associated with the pre-defined 'goal' layer. Users could then specify alternate layers with which to associate the goal element, for example, a business process.

The Layering prototype was developed using Photoshop and Visio. Developing this prototype aimed to capture the utility of the layering tool. First, the original i* model data was redistributed to layers using Visio's layering mechanism. The layers would be used to emulate the results of user interaction with the Photoshop layer mechanism. The Photoshop layering mechanism was captured in various states (all layers visible, some layers invisible, varying levels of opacity). Each state was incorporated into a Visio drawing of the i* model. Each drawing represented a different usage of the layering mechanism, and displayed the results of this usage. First, for example, the i* model would be shown unmodified, with the Photoshop layering mechanism included. The next drawing would then display only some elements of the model, hiding the rest. This was achieved using Visio's layering technique. The Photoshop layering technique would emulate this, making the appropriate layers invisible. Figure 11 below displays a partial view of the Layering prototype. Appendix 3-F demonstrates further details for this prototype.

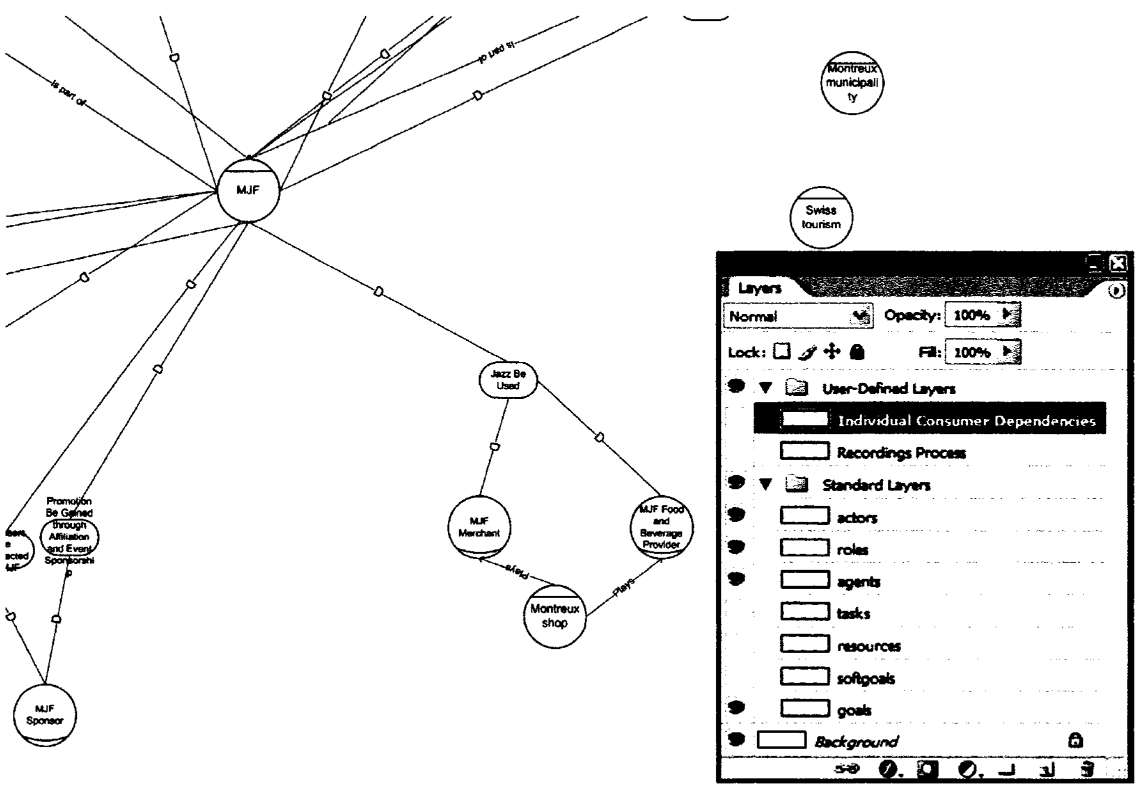


Figure 11: Filtering (Layering) Prototype Sample using Montreaux Jazz Festival Data

4.4.4.3 Filtering (Tagging) Prototype

Current techniques for filtering have been limited to the existing content of the model. However, in many cases that surfaced during observation, users demonstrated the need to further describe the data within models according to project-specific schemas or taxonomies. Filtering by tagging also relates to the task of Filtering; tags can be used to filter uninteresting items from the view of the i* modeller.

In the case of the expert modellers, although they worked independently to review models, they met beforehand to develop a taxonomy that helped to define scope. Each user then visually applied this taxonomy to their models. Novice modellers also attempted to further describe the data in their models, using colours to visualize element's inclusion in various alternatives.

Both groups had a similar problem when trying to describe their data visually. Two of the experts noted that it was often difficult to group elements into one group because in other contexts, one element could belong to many groups. For example, an element belonging to Alternative 1 may also be included in Alternative 2. Thus, for describing data it is important that data can be associated with more than one tag.

To further investigate this problem, the practice of tagging was examined. In this paper, tagging refers to the practice of associating keywords with some type of data. For i*, tagging would involve the association of keywords with model elements, such as goals or actors. Tagging has become popular in social networking sites, such as Flickr and del.icio.us. This type of tagging places the onus on the user to define and associate keywords with an object.

In many cases, placing the burden of defining the taxonomy on the user can be problematic. This is particularly the case when users do not have an agreed upon definition of keywords and are not operating in a coordinated fashion. However, in the case of i* modelling projects, users are typically working alone or in small groups. As witnessed during observation, groups, especially those with experience, meet beforehand to discuss the scope of the project and work together to determine important aspects of the case they are modelling. The cooperative style of work and development of a common taxonomy prior to applying metadata to the model indicate that the technique of tagging may indeed be useful for filtering and extracting data within i* models. Incorporating a pre-defined taxonomy may have limited use because of the unique domain of each modelling project.

The tagging prototype was developed in MS Visio. A component was constructed allowing users to tags elements, add new tags to the taxonomy, and filter elements using tags. The component's design was based on Adobe's dockable panels, used in Photoshop and Illustrator. This gives the user flexibility in showing the panel, the potential to lock the component to a fixed position within the application or move the component within the application.

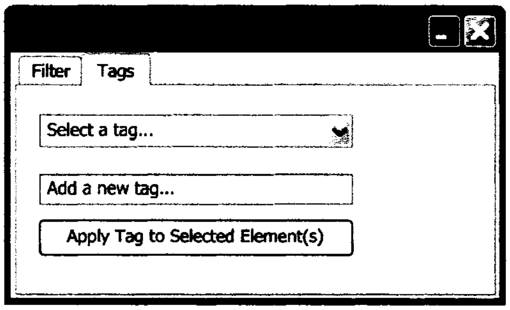


Figure 12: Tagging Prototype: Selecting or Adding Tags

Tags can be applied to model elements by selecting elements and either selecting an existing tag already created by the user or adding a new tag. Multiple elements can be selected and applied with the tag at once. It may be useful for the component to contain a default taxonomy available to users, populated with commonly used i* terms and allow for additional terms to be added by the user. It may also be useful to allow users to import their taxonomies into new models when they are being created. The ability for users to remove tags from elements must also be included in this component. This may be achieved by implementing an 'element information' component, similar to TAOM4E, where details about each element are shown. If multiple elements are selected, only information shared between elements would be available for display and editing.

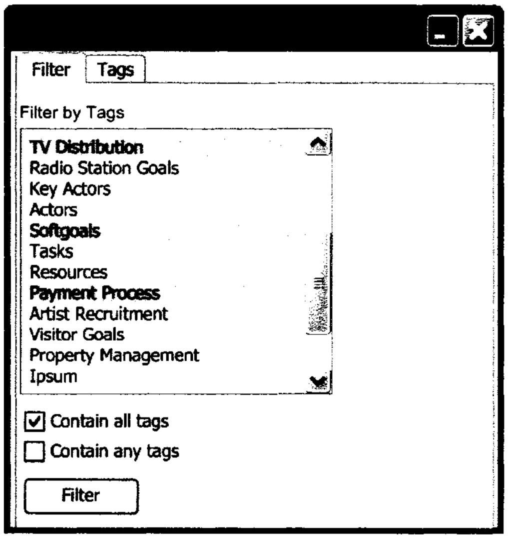


Figure 13: Tagging Prototype: Filtering Alternative 1

As Figure 13 above describes, users can filter both SD and SR models by selecting the tags they wish to view. All other elements will be removed from the user's view. Users can choose to filter for elements containing all of the tags they select, for fewer results, or expanding the results by filtering for elements that contain any one of the tags. However, the limited options for filtering offered by this alternative do not allow the user to display only softgoals that have been tagged with payment process or TV distribution.

To resolve this shortcoming, an alternative approach is presented in Figure 13 below. This filtering feature allows users to select elements first, and then choose the tags they would to use as filters. In this example, all actors, goals and softgoals that have been tagged with either 'payment process' or 'visitor goals' will be displayed in the model.

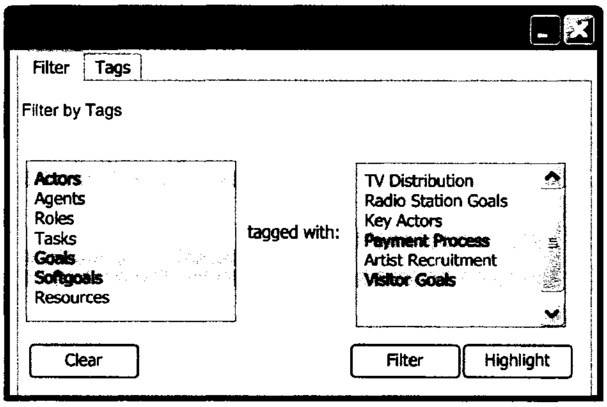


Figure 14: Tagging Prototype: Filtering Alternative 2

Filtering can be quite a useful tool for reducing clutter. However, filtering may hide elements that may be useful when analyzing current items of focus, such as the goals of other actors currently hidden by filters. For this reason, two alternate approaches are proposed. First, instead of using filter, a 'highlight' button is also incorporated into the component. Using the highlight button changes the colour and thickness of the shapes currently matching the filtering criteria. This allows users to maintain a high level of context while emphasizing the current areas of focus. The Tagging Prototype is outlined in Appendix 3-G.

4.4.3 Extraction

4.4.3.1 Extraction Prototype

When developing the Extraction prototype, the requirements in the section above were carefully considered. First, users must be able to specify the elements under consideration. This could take place by allowing the user to select multiple items and select an option from a menu to extract the current selection. The extracted data could be viewable to the user in a new window, allowing users to easily adjust the position of the new window. The extracted view may also be saved or printed by users.

Equally important is for the mechanism to incorporate all elements that are directly related to the elements that selected by the user. In this case, slicing could be applied to extract all elements directly related to the elements that have been selected by the user. Figure 15 displays the Extraction prototype. This prototype is included in Appendix 3-H.

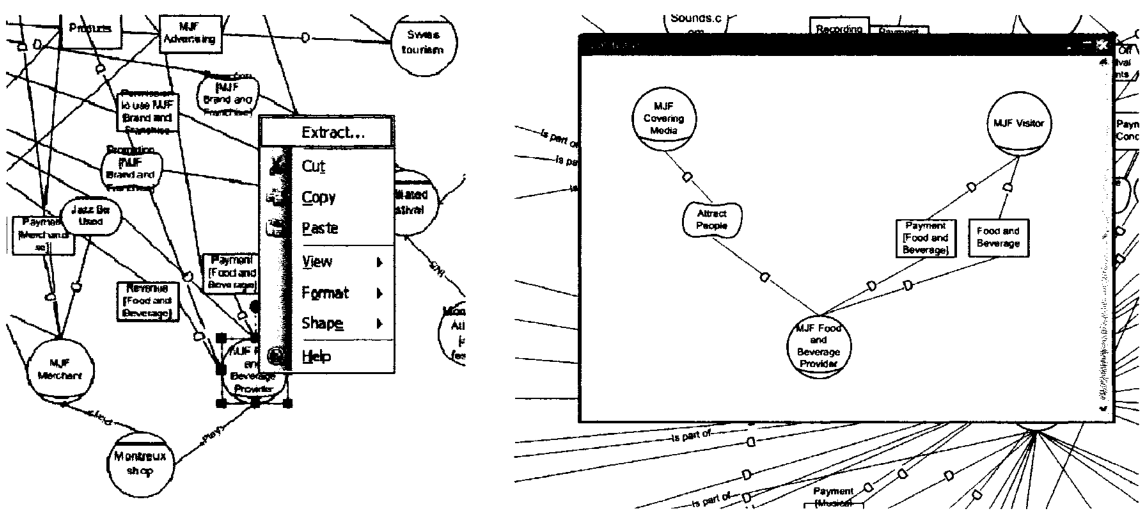


Figure 15: Extraction Prototype: Extracting and Displaying Extracted Results in a New Window

4.5 Other Considerations for Prototype Development

4.5.1 Mechanisms Working in Combination

While it is important to consider the effect of each of these prototypes on the model, it is also necessary to anticipate the impact of their functionality on other prototypes. It seems plausible that users may want to combine the functionality of the prototypes to produce a combined effect on the model. In this case, it is important to consider each prototype's capabilities in light of the others'. For example, tagging and slicing may be used to filter by certain tags and then display all elements one node away from the elements. Tags and layers may be used to display only the goals of all elements tagged with a certain term.

The result of using these prototypes in combination is increased flexibility for the user. However, it is important for developers to anticipate problems, such as software crashes or illogical data, that may be caused by using these features in combination. In this case, the use of certain combinations of mechanisms may have to be limited by the tool developers.

4.5.2 The Requirement of Context in i Modelling*

Previous experience with i* modelling and the observation of modellers using the i* framework underscores the importance of context when analyzing i* models. i* models are a large network of dependencies. Though elements may not be directly linked together, there is potential for these elements to impact each other indirectly through other elements. Evaluating parts of models in isolation may result in the exclusion of important data that may indirectly affect your analysis. For this reason, maintaining the context when analyzing i* models is very important.

An attempt to balance the effective display of filtered or extracted data with data to be hidden from view was carefully considered. Rather than completely hide non-essential data, a potential solution is to reduce the visibility of other data. At the same time, essential data under analysis would be highlighted. Taking this measure would provide users with the context often needed in i* modelling while still highlighting the essential data directly under analysis.

4.5.3 Considering Automation in Prototype Development

The incorporation of automation appears to be a valid solution to many of the existing organization and clutter issues surrounding i* modelling. However, as observation has demonstrated, the ability to manually build models has many benefits. First, novice modellers manually translating dependencies from the dependency table to the modelling notation found a number of errors and omissions in their documentation. Had the model been automatically generated from their table, these errors would not likely have surfaced until much later in the modelling project. In a separate example, the manual creation of the model by modellers built a strong mental picture of the contents of the model, in terms of the location of elements. Automating changes to the model made in other 'views' (e.g., tabular, extraction, and graphical overviews) may disrupt the mental model; this was the primary reason for making these items static views instead of editable.

Automation is used only in the Extraction technique, because a limited number of elements are included in the newly extracted window, and the underlying model layout is preserved. In the condensed overview, automation is considered only from the perspective of reducing the space between elements, while preserving their positions in the model relative to other elements.

With six prototypes developed in this stage, feedback was needed to determine the usefulness of these prototypes from the perspective of i* modellers. The following section describes the feedback received during these sessions.

4.6 User Evaluation of Proposed Solutions

4.6.1 Introduction

In this section, the feedback from participants involved in prototype evaluation is described. Participants evaluated the Graphical Overview, Tabular View, Filtering (Slicing), Filtering (Layering), Filtering (Tagging) and Extraction prototypes. First, the characteristics of the participants are described in regards to their experience with i* modelling and i* modelling tools. Next, feedback gathered for each task is discussed. The prototype selected by participants for each task, key functionality that made the prototype useful, and alternative prototypes that were mentioned by users but not selected for each task are outlined. Finally, user feedback and suggested improvements or additions are discussed for each of the prototypes.

4.6.2 Participants

Eight participants were selected for participation in this study. Participants were selected based on their response to an email requesting participation in a study involving i* modelling. All participants were Master's students at the Faculty of Information Studies. Seven participants had acquired experience with i* modelling through course work. These participants had used the i* modelling technique in at least two courses, for a number of assignments. The other participant possessed an understanding of the principles and purpose of i* modelling but had not used the technique for courses or projects. Three participants had used i* modelling in more than three courses, and two had employed i* techniques for research papers. Participants had experience with both the i* stencil for MS Visio and OME3. Six participants explained that their experience with Visio was far greater than that with OME3.

Each participant was provided with a functional description of each prototype, as well as a corresponding paper-based prototype to review and further clarify the functionality of the prototype using example data and scenarios. When all prototypes had been explained to the participant, the interviewer described four tasks for the participant to consider. Tasks that commonly arose during modeller observation were selected for inclusion in the study. An attempt was made to vary the types of tasks presented to the user. It was hypothesized that tasks with different characteristics would require different prototypes. It was thought that including different types of tasks would encourage users to discuss a variety of the prototypes presented, providing richer feedback about the strengths and weaknesses of each prototype.

4.6.3 Task 1

The first task required participants to select the prototype that would best assist them in analyzing the relationship between the MJF artist and MJF (see Appendix 3-K). Seven of eight participants selected the extraction tool as most useful for this task. Reasons for selecting this prototype varied. Many participants mentioned the simplicity of the extraction tool as a reason for selection. "You just point and click… it's really simple". Others mentioned the fact that they were able to remove all of the clutter and look only at the items directly related between the two actors. "It cuts out all of the noise – I get a clear view of the relationships between the two agents. With the regular view it's pretty confusing; there's so many lines all over the place".

In terms of the functionality, participants mentioned the ability to view the isolated information in the new window and the ease of use as main reasons for selection. Having to select only the relevant actors and having the mechanism automatically select related dependencies was also mentioned by a number of the participants as very useful. Participants also noted the automated layout feature of the prototype to be quite useful, particularly if the elements selected for extraction were far apart in the standard view of the model.

One participant selected the slicing tool rather than the extraction tool, because he felt that context was very important for analyzing any relation and that slicing best maintained the context. Four participants mentioned the possibility of employing the slicing tool to assist with the task before choosing extraction. Many said they chose not to select the slicing technique because it included a great deal of additional information, making it less efficient than the extraction prototype. Other prototypes that were mentioned by participants included the graphical overview and the tabular view of the model.

4.6.4 Task 2

The second task required participants to select the prototype that would best assist them in comparing the relationship between the MJF and Ticket Corner and MJF and General Festival Partners. Again, extraction was the overwhelming choice, favoured by seven of eight participants.

Participants enjoyed the simplicity of the technique, combined with the ability of the extraction prototype to produce multiple windows. Six of seven participants who selected this prototype explained that they would extract each pair of actors in an individual window and compare the relationships between windows.

Three participants mentioned the tabular view as potentially useful but did not select this prototype. Those mentioning the tabular view suggested that it could be useful if filtering was available in such a way that only the three actors in question were displayed. However, they also cited the lack of a visual representation and inability to view indirectly related relationships as reasons for not choosing this prototype.

4.6.4 Task 3

The third task asked participants to select the prototype that would best assist them in determining the most important actor in the model. Though one participant suggested they would not require a prototype to assist in completing this task, they were asked to choose a prototype they thought would be useful. In this case, the prototype most often selected was the tabular view, selected by six of eight participants. The other two participants selected the graphical overview prototype for this task.

Many equated the most important actor to be the one with the most dependencies in the model. Following this line of thinking, the ability to group model data by depender or dependee and count the total number of dependencies for each was the primary reason participants selected the tabular prototype. "[The tabular view] quantifies it for me". "This way you have the number of dependencies and it's accurate". Others mentioned the efficiency of the tabular view in determining the most important actor. "It's really efficient and you don't have to sort it out [visually] – it was really quick".

Two participants took a more nuanced approach when considering the most important actor in the model, noting that because some goals are more important than others, the actor with the most dependencies may not necessarily be the most important actor in the model. Of these two participants, one selected the tabular view anyway but was careful to note that further analysis was necessary. The other decided the graphical overview would be best for this task because it combined the quantitative display of numbers with the visual layout. This participant mentioned that the numbers between each actor "distinguish critical areas of the system to look at for analysis".

4.6.5 Task 4

The final task required participants to select the prototype that would best assist them in determining all the goals of the MJF. This task caused the greatest variation in response amongst participants when asked to select the best suited prototype to the task.

Four participants selected the layering prototype, three selected the tabular prototype and one selected slicing. One participant mentioned that he selected layering over the tabular view because it did not require him to "visualize a list". Others suggested they would use layering simply to filter out all unnecessary elements from the model and believed this would provide a clear enough picture to pick out the appropriate goals related to the actor. Most participants suggested that they would create a new layer and manually place all relevant items on this layer, filtering everything else out. Participants were particularly attracted to the ability to easily add or remove other information from the model by hiding and showing layers. "[You can] hide and view selectively the information that provides context, so you can bring it in and out".

The participants that selected the tabular format were attracted to its simplicity. These participants appreciated the fact that all goals were displayed in a list format, and were easy to identify quickly. One participant who ultimately selected the tabular view over the tagging prototype stated "I would use tabular because I wouldn't have to go and tag everything first. Because things are right there [in the tabular view], it's easier."

Besides the layering, tabular, and slicing prototype, two participants mentioned the tagging prototype before selecting others.

Tables 9 and 10 summarize the task selection portion of the study.

Prototype	Task 1	Task 2	Task 3	Task 4
Graphical overview	0	0	2	0
Tabular	0	0	6	3
Slicing	1	0	0	1
Layering	0	0	0	4
Tagging	0	1	0	0
Extraction	7	7	0	0

Table 9: Prototypes Selected, by Task

Prototype	Task 1	Task 2	Task 3	Task 4
Graphical overview	1	1	3	0
Tabular	1	3	0	1
Slicing	4	1	0	0
Layering	0	1	0	3
Tagging	1	1	1	2
Extraction	1	0	0	0

Table 10: Prototypes Mentioned but Not Selected, by Task

4.6.6 Participant Feedback by Prototype

Next, comments and feedback about the specific prototypes are summarized. Feedback was gained from participants as they considered the prototypes for tasks, when asking for clarification about their functionality, or when asked for general comments regarding changes, improvements or additions in relation to the prototypes at the conclusion of the interview.

4.6.6.1 Graphical Overview

A number of interesting points were made in regards to the functionality of the graphical overview. One participant suggested that weights be applied to the dependency links between actors, increasing the thickness of the line to indicate a higher number of dependencies between actors. Another participant suggested that a total number be displayed for each actor, indicating the total number of dependencies the actor has within the model.

Two participants mentioned that they did not like the fact that the context was hidden from view, and suggested as an alternative that this view be applied partially to the model. They suggested that the ability to expand pieces of the model to standard view while condensing areas of the model they are not currently focusing on would be useful. The techniques incorporated in the graphical overview could be used to visualize the condensed portions of the model.

Finally, participants suggested that it would be useful if this view was editable, meaning that users could add new data to the model from this view and have the changes be reflected in the SD model. When suggested that this may disrupt the mental model they developed when constructing the model, they agreed but said if the changes were highlighted in some way, it would outweigh the cost of disrupting the mental model.

4.6.6.2 Tabular Overview

Participants mentioned that the ability to filter data from the table based on user-defined criteria would make this prototype more useful. Examples of criteria included "show me all actors with more than *x* number of dependencies" and "display only data for actor *x*".

Again, participants suggested that it would be useful if this view was editable, in that users could add new data to the model in this view and have the changes be reflected in the SD model.

4.6.6.3 Filtering (Slicing)

The most interesting feedback regarding slicing was the call by participants for slicing to work from a top-down approach. A number of participants suggested that top-down slicing would be particularly useful when analyzing SR models, and mentioned that adding this functionality would be ideal. When told that this functionality existed (Leica, 2005) and that this prototype was seen as complementary to that technique, they felt that slicing would be quite useful. Users also favoured the decision to retain data not meeting the slicing criteria by making it transparent. They found this to be essential in evaluating relationships, as context is a key component of the i* modelling framework. "I really like the ability to see the relationships around [the one under analysis]… I want to see who the actors talk to".

Another comment frequently offered by participants was the call for a higher degree of specificity when slicing. Users would have liked to be able to specify only certain element types be included when slicing. For example, two participants mentioned that limiting the slicing filter to only show the goals related to the selected actors would have been very useful.

4.6.6.4 Filtering by Layering & Tagging

The layering and tagging prototypes were less popular choices for the tasks. This was perhaps because layering and tagging often apply to model-wide visualizations, such as showing all goals for the model, or all tasks for the model. The inability to specify the goals for an actor without creating a new layer or tagging a number of elements was a source of concern for the modellers.

Additionally, these techniques required foresight in regards to how the model will be analyzed. As one subject succinctly noted, "with tagging and layering, you need to know beforehand what you're going to be looking at [during analysis] in order to build the correct layers or apply the right tags". Two participants felt that tagging would be much more useful if a set of tags were developed that were generic enough to be applied to a number of domains. When asked to elaborate, the participants felt that pre-defined tags may be a source of inspiration for analysis; if for example, the tags would suggest a way of analyzing the model that had not been previously considered by the modeller.

Layering was the more popular choice between the two, because, as participants stated, it required less effort. "[Tagging] creates more work for the designer, because you have to label and choose the elements you're going to label, and then filter. It's too much effort." Participants liked the amount of control over the layers; they noted that it would be very easy to show and hide a variety of information very quickly using the layering prototype.

4.6.6.5 Extraction

The majority of feedback for the extraction technique was overwhelmingly positive, particularly when analyzing relationships between actors. However, a number of useful suggestions for extending its functionality were discussed by the participants.

The first suggestion involved highlighting selected items in the underlying model in combination with extracting the items to a new window. This would allow the user to view the context in which the direct relationships take place, while maintaining an extracted view of the isolated window. Though two participants suggested that they would like to view the context within the extraction window, the suggestion of highlighting the items in the underlying window seems the better solution, as it provides alternate views of the information. Figure 16 demonstrates a method for noting highlighted items in the underlying model.

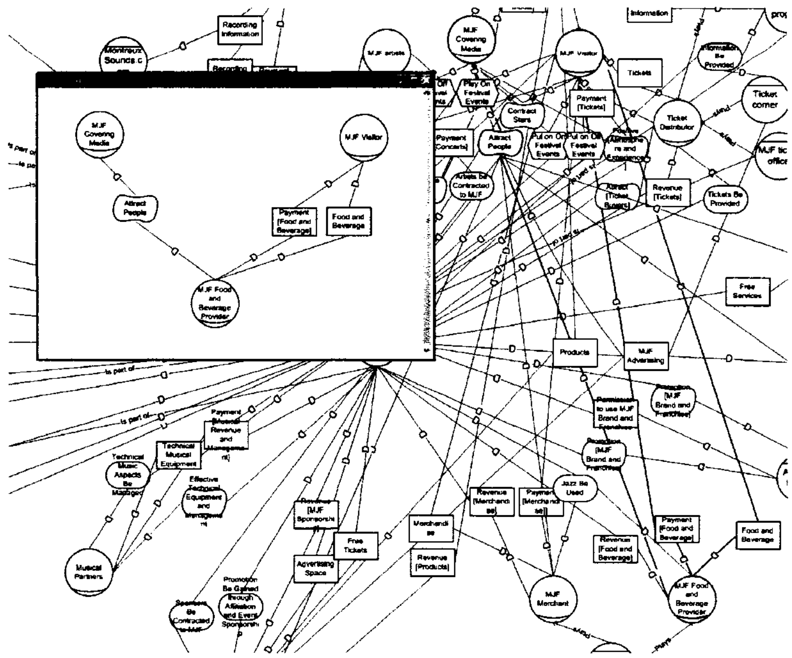


Figure 16: Extraction with Selected Items Highlighted in Original Model

The second improvement suggested by participants related to the editing capability of the extraction prototype. The participants felt that the utility of the prototype would be greatly increased if they were able to make edits in the extracted window and have these changes committed to the underlying model automatically. While they agreed that this would disrupt their mental picture of the model, to compensate they suggested that all changes in the original model be highlighted for the modeller to see. They felt this benefit would outweigh the cost of any disruption to their understanding of where elements were located in the model. Others suggested that because i* models are typically so cluttered, modellers are constantly rearranging elements anyway. "The model is so messy anyways, what's the difference… you have to move things around anyways and change the format anyways".

4.6.6.6 Conclusions from Prototype Feedback

For each task, the prototype selected by the participants varied. This supports the notion that there is no one mechanism appropriate for all tasks. Rather, the unique properties of different tasks require unique prototypes for assistance. This may be the reason certain prototypes appeared more popular than others; it seems plausible that had the tasks selected for use in the study differed, the popularity of prototypes would have also differed.

Participants indicated that more powerful results could be achieved if the prototypes worked in combination with one another. Surprisingly, participants felt that the benefits of automation outweighed the cost, especially if a technique could be included that identified the changes made to the model.

4.6.7 Implications for All Prototypes

Two key points were noted during evaluation sessions. Discussion with the participants revealed that users should be provided with control over how much context should be shown in conjunction with the filtering prototypes. The use of context amongst participants varied greatly, depending on the task and the prototype they were currently interacting with. When discussed with the participants, all agreed that it would be useful to control the degree to which data not meeting the filtering criteria is displayed. One way of implementing this technique would be using a slide bar. Depending on the direction the slide bar is dragged, transparency of extraneous data would increase or decrease. Slide bars have been incorporated in a variety of tools, such as Adobe Photoshop and Illustrator.

The second key notion with implications for all prototypes is the utility of using various prototypes in combination to produce unique results. Using prototypes in combination was exciting to the participants, as they described that using the mechanisms together would produce higher levels of control over what they can display in the model, particularly when filtering.

4.6.7.1 Implications for Overview

The evaluation of the prototypes had one important implication for prototypes developed with the intention of providing an overview of model data.

From evaluation, an additional requirement to those discussed in section 4.3.2 surfaced:

1. Allow this view to be editable and have changes automatically committed to the original model; all changes made as a result of editing should be made clear to the user.

4.6.7.2 Implications for Filtering

The evaluation of the prototypes had one important revision to the requirements for Filtering.

Feedback from participants during interviews suggests changes are necessary to requirement 4 (Hide OR reduce the prominence of data that does not match the criteria determined by the user). Instead of prescribing whether extraneous data should be hidden or shown, allow the user to control the degree to which it is shown.

4.6.7,3 Implications for Extraction

Feedback from participants yielded two important additions to the requirements for Extraction.

Additional requirements are necessary when considering prototypes performing the task of Extraction:

1. Highlight the direct relationships of selected items in the underlying model in conjunction with opening the selected items in a new window;
2. Allow this view to be editable and have changes automatically committed to the original model; all changes made as a result of editing should be made clear to the user.

4.6.8 Summary

Feedback on the prototypes was discussed in this section. Participants' selection of prototypes varied based on task. This supports the notion that there is no one mechanism appropriate for all tasks. Rather, the unique properties of different tasks require unique prototypes for assistance. User feedback also was helpful for generating further insight into what is required for each prototype relating to the Overview, Filter, and Extract tasks. The initial requirements developed by the researcher were altered or expanded as a result of user feedback, improving requirements.

Selecting and developing prototypes based on Shneiderman's taxonomy proved quite useful, based on positive feedback from i* modellers included in this study. Focusing on particular tasks helped to identify areas where functionality was lacking in current tool support and helped to identify potential mechanisms incorporated into prototype development.

5. Conclusion

5.1 Implications of Findings

This study aimed to identify areas for improving the visualization of i* models and develop prototypes to address these areas. As discovered through a review of i* tools and observation of i* modellers, there is a strong need for visual support within tools supporting i* modelling. Current tools weakly support the tasks of providing an overall view of the collection, filtering extraneous data and extracting sub-collections of the model data. During observation, modellers constantly rearranged elements to achieve clarity when analyzing relationships between actors and struggled to gain a clear overall view of the model.

Prototypes were then developed to meet these needs, and user evaluation of prototypes was employed to test the validity of the concepts and requirements included in the prototypes. User feedback suggested these prototypes would be useful when using software for i* modelling and offered suggestions for further improvements.

Although this study focused specifically on i* modelling, many other RE techniques rely upon graphical notation for documenting requirements and face visualization issues as models grow in size. The methods of observation, prototype selection and development, and user evaluation outlined in this study provide an approach for exploring difficulties software users face and searching for solutions to overcome these difficulties. As a result, the methods employed in this paper may be applicable when identifying useful prototypes for software supporting other Requirements Engineering techniques.

Future releases of software may choose to consider Shneiderman's Visual Information Seeking tasks when looking for areas to extend the functionality of future software. These broad tasks may serve to focus the efforts of developers as they evaluate the current state of software and look for areas of improvement and expansion. The Overview, Filtering and Extraction tasks, and subsequent requirements and prototypes proposed in this study offer a firm foundation for the further development of i* tools.

5.2 Limitations of the Study

Five principle limitations of this study warrant discussion. First, the number of modellers involved in the observational study was limited. Observing only one group of 'novice' and one group of 'expert' modellers limits the generalization of results. However, the noted utility of these prototypes by the participants in the Prototype Evaluation study suggests that the initial study did capture challenges encountered by a broader population of modellers. An additional shortcoming of the sample is that all modellers were recruited from the University of Toronto. It is possible that i* modellers external to the University of Toronto may employ different methods when conducting i* modelling, and thus face dissimilar challenges.

Second, the tasks selected for prototype evaluation may have favored certain prototypes more than others, skewing results. A better approach may have been to include a larger number of varied tasks and extend the session length from one hour to one hour and thirty minutes. This may have uncovered tasks for which the less popular prototypes were useful and facilitated increased discussion in regards to these prototypes.

Third, Strategic Rationale (SR) models were not included in prototype designs. Though it was envisioned that the prototypes developed in this study would be compatible with the characteristics of SR models, user feedback was not gained in this regard. The lack of feedback limits this study's ability to demonstrate the prototypes developed would prove useful for supporting modeller visualization of SR models.

Further, paper-based prototypes may have favoured the representation of some prototypes over others. For example, the Extraction paper prototype may have made a more favourable impression because the utility of its features was more clearly obvious on paper than other prototypes, such as the tagging prototype.

Finally, modellers observed in Phase 2 did not use all tools addressed in the i* literature review. Instead, use was limited to the tools selected by the participants for their modelling projects (OME3 for novices, and the i* stencil for Visio for experts). This may have influenced the types of challenges that arose during observation, thus influencing the types of prototypes selected for development. An improved approach may involve requiring users to try all i* modelling tools prior to the observation session and select the one they felt would work best. Modellers could then have discussed the reasons for their selection, strengthening the review of i* modelling tools with complementary data.

5.3 Future Research

Several aspects of this project can be built upon and extended in future research. First, researchers could observe modellers from outside the University of Toronto and compare the behaviours between groups. As the set of challenges identified in Phase 1b played a significant role in the identification of prototypes for development, it may be interesting to determine if the challenges for modellers outside of the University of Toronto community are congruent and if the prototypes proposed in this study remain applicable for other groups.

Second, a systematic approach to documenting and analyzing the challenges of modellers could be undertaken. Using goal-based modelling such as the Non-Functional Requirements (NFR) Framework (Chung, Nixon, & Yu, 1995), challenges could be converted into goals to guide the exploration of potential solutions. The NFR goal model could also be used to evaluate potential solutions regarding their ability to overcome the identified challenges.

Directly related to the prototypes, future work could develop an interface to host the various prototypes, and to consider the interaction design between each mechanism and between the mechanisms and the user. A next step may involve examining how these mechanisms can be presented in a way that allows the users to move fluidly between mechanisms. Also, the complexities involved in allowing the mechanisms to work in concert with one another to produce additional capabilities may be explored. Further, iterations of prototypes may be developed, with higher fidelity, to test the interaction between the mechanisms and the user. Finally, tool developers may choose to select some prototypes for implementation in future releases of their software.

Should prototypes be developed and included in i* tools, future work may include measuring the effect of the prototypes that have been implemented to determine their ability to positively impact the visualization of the model, from the modeller's perspective.

Appendices

Appendix 1: User Observation Research Instruments

A –Consent Form

Human Participants Research Consent Form
University of Toronto
Letter of Informed Consent

Date

I, (print name in full) ______________________ acknowledge in signing this consent form that I agree to volunteer in the research project titled "Towards Improved Tool Support for the i* Modelling Framework" being conducted by Christopher Cocca between September, 2006 and April, 2007.

I understand that the research being conducted relates to the improved tool support for i* modelling, and that my **involvement consists in my participation in interview and/or observation sessions**. I am aware that observation sessions consist of researchers observing the manner in which I use i* tools. I understand that information from my interview and/or observation sessions with the researcher will be studied and may be used in a project report and in future papers, journal articles and books that will be written by the researcher.

I grant authorization for the use of the above information with the full understanding that my anonymity and confidentiality will be preserved at all times. I understand that my full name or other identifying information will never be disclosed or referenced in any way in any written or verbal context. I understand that transcripts, both paper and electronic versions, will be secured in the privacy of the researcher's home or office and that any audio or video tapes of my conversations with the researcher will be erased no later than December, 2009. **I also understand that I may ask for audio or video taping to stop at any time I desire, without negative consequences for myself.** I have been explained that researchers do not foresee any risks in my participation in this study. If you have questions about your rights as a research subject, please contact: ethics.review@utoronto.ca or call 946-3273.

I understand that my participation is entirely voluntary and that I may withdraw my permission to participate in this study without explanation at any point up to and including, the last day of December, 2005.

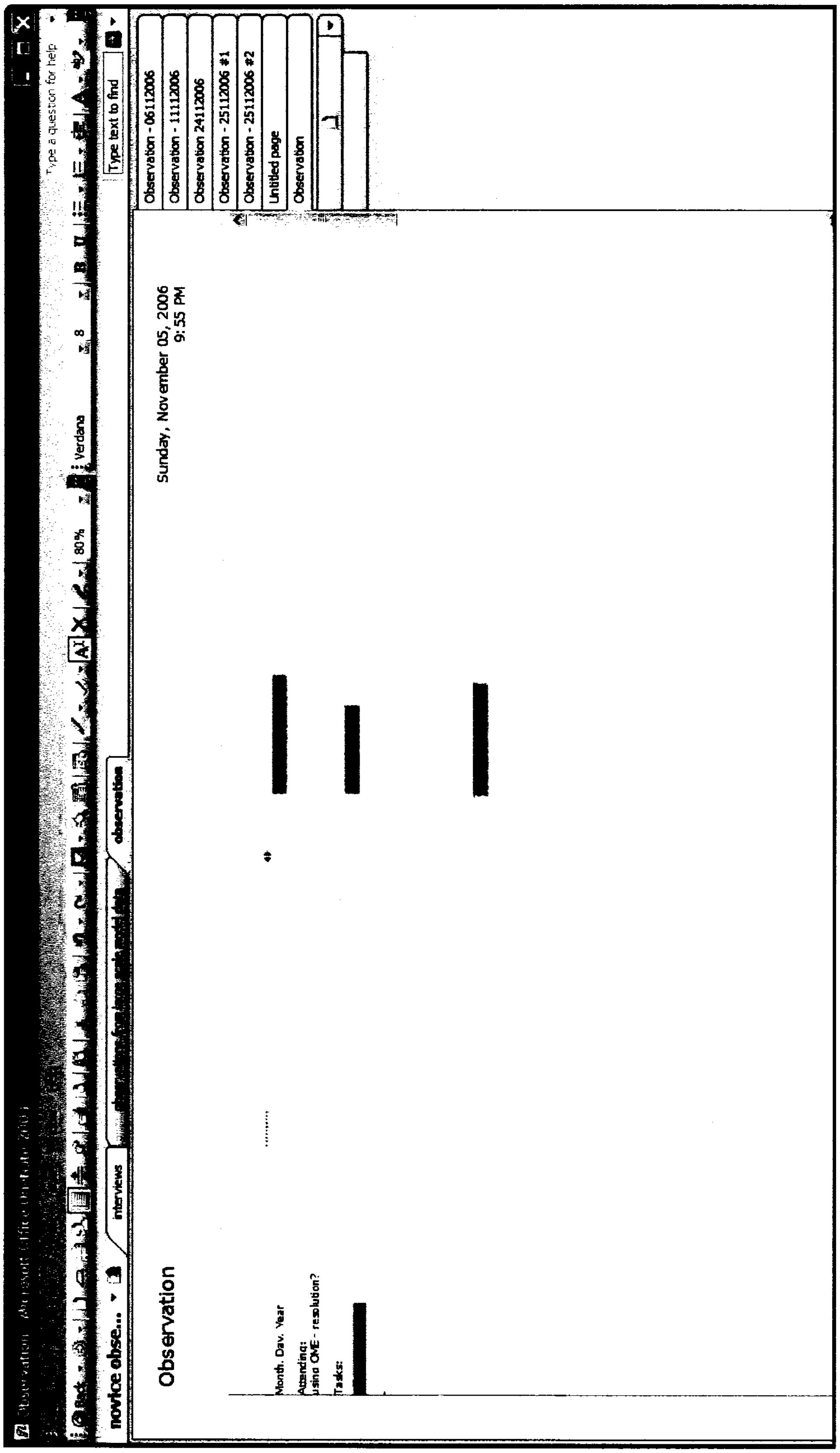
I grant permission to use the following:

____ **Audio taping**
____ **Video taping**

____ My first name only
or
____ Only a pseudonym

____
Signature

____
Date


Type a question for help
novice obse...
interviews
observation
Verdana
80%
Type text to find
Observation - 06112006
Observation - 11112006
Observation 24112006
Observation - 25112006 #1
Observation - 25112006 #2
Untitled page
Observation
Observation
Sunday, November 05, 2006
9:55 PM
Month. Day. Year
Attending:
using CME - resolution?
Tasks: [redacted]

Appendix 2: i* Tool and Related Tool Summary Table

Overview	N/A	Hand tool for panning across a model's canvas.	Pan & Zoom window for gaining an overview of the model.	Pan & Zoom window for gaining an overview of the model.	Collapsed tree-hierarchy for overview of model contents.	Panning tool used to navigate the model	Represents model data in a tabular form, user can scroll through tabular data.
Zoom	N/A	Zooming in and out of the model.	1. Zooming in and out of the model. 2. Pan & Zoom window for navigating a model's canvas.	1. Zooming in and out of the model. 2. Pan & Zoom window for navigating a model's canvas.	N/A	Drop-down zoom menu allows users to select percentage of model being shown.	Drop-down zoom menu allows users to select percentage of model being shown.
Filter	N/A	Hide unsatisfied elements or unselected elements from view.	Elements can be assigned to layers; users can choose to make layers either visible or invisible, filtering elements in the process.	Elements can be assigned to layers; users can choose to make layers either visible or invisible, filtering elements in the process.	N/A	N/A	N/A
Details on demand	Actors' boundaries can be expanded and collapsed to display or hide the elements located within actor boundaries.	Actors' boundaries can be expanded and collapsed to display or hide the elements located within actor boundaries.	N/A	N/A	Expand and collapse tree-hierarchy for element details.	Properties window provides further details about an element that is selected by the user.	When an element is selected from the model, details of the element are displayed in the 'object pane'. Further element details can be added through the object pane.

History		Undo/redo of user actions	Undo/redo of user actions.	Undo/redo of user actions.		Undoing and redoing user actions.	Undoing and redoing user actions.
Extract	N/A	N/A	N/A	1. Automated generation of requirements from the model to alternate formats, such as MS Excel and Word. 2. Extraction of SD model elements to produce SD model from SR model.	1. Generation of alternative processes to the current process	Generation of alternative processes to the current process	N/A
Relate	When an element has been added to the canvas, it can be repositioned by selecting the element with the mouse and dragging it the desired location, allowing elements to be grouped by proximity to one another.	When an element has been added to the canvas, it can be repositioned by selecting the element with the mouse and dragging it the desired location, allowing elements to be grouped by proximity to one another.	1. When an element has been added to the canvas, it can be repositioned by selecting the element with the mouse and dragging it the desired location, allowing elements to be grouped by proximity to one another. 2. Groups of elements can be assigned a 'fill' colour, demonstrating that they are related in some way. 3. Additional Visio shapes can be used as containers, to group elements together.	1. When an element has been added to the canvas, it can be repositioned by selecting the element with the mouse and dragging it the desired location, allowing elements to be grouped by proximity to one another. 2. Groups of elements can be assigned a 'fill' colour, demonstrating that they are related in some way. 3. Additional Visio shapes can be used as containers, to group elements together.	1. These Detailed Interaction Scripts organize information on an activity-level, relating actors, resources, triggering events, and conditions to the activity 2. Tradeoff analysis is used to determine the most viable alternative during evaluation	TAOM4E assigns different default colours to different i* element-types, making it easy to identify different elements.	1. Relates model data using tabular form, in addition to traditional graphical notation. 2. Forward and backward reasoning can be conducted to

Appendix 3: Prototype Evaluation Research Instruments

A – Call for Participation Email

Dear FIS students,

I would like to invite students for participation in a study being conducted in relation to improving i* modelling software. Participation in this study will involve a 60 minute interview during which several low fidelity prototypes will be proposed for evaluation and discussion by participants. Each participant will be paid $20 upon completion of the session.

Eligible participants must have experience with i* modelling. For those interested in participating, please send an email to cocca@fis.utoronto.ca. Interviews will be arranged at the convenience of the participant.

Sincerely,

Chris Cocca

Human Participants Research Consent Form
University of Toronto
Letter of Informed Consent

Date

I, (print name in full) ______________________ acknowledge in signing this consent form that I agree to participate in the research project titled "Towards Improved Tool Support for the i* Modelling Framework" being conducted by Christopher Cocca between September, 2006 and April, 2007.

I understand that the research being conducted relates to the improved tool support for i* modelling, and that my involvement consists in my participation in an interview. I am aware that this session consists of evaluating prototypes for i* modelling software. I understand that information from my interview with the researcher will be studied and may be used in a project report and in future papers, journal articles and books that will be written by the researcher. I understand that I will receive a payment of $20 upon completion of the study.

I grant authorization for the use of the above information with the full understanding that my anonymity and confidentiality will be preserved at all times. I understand that my full name or other identifying information will never be disclosed or referenced in any way in any written or verbal context. I understand that transcripts, both paper and electronic versions, will be secured in the privacy of the researcher's home or office and that any audio or video tapes of my conversations with the researcher will be erased no later than December, 2009. **I also understand that I may ask for audio or video taping to stop at any time I desire, without negative consequences for myself**. I have been explained that researchers do not foresee any risks in my participation in this study. If you have questions about your rights as a research subject, please contact: ethics.review@utoronto.ca or call 946-3273.

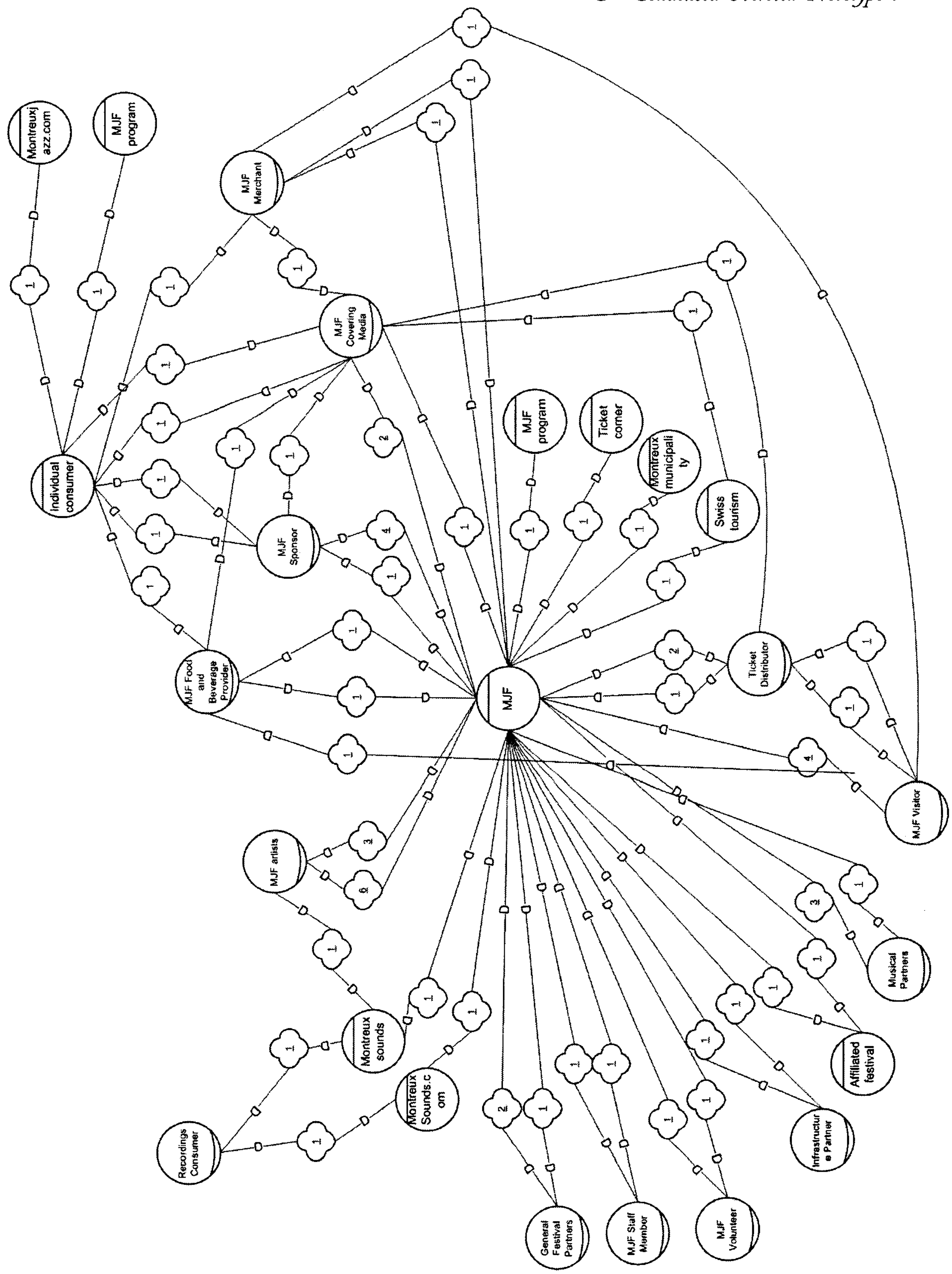
I grant permission to use the following:

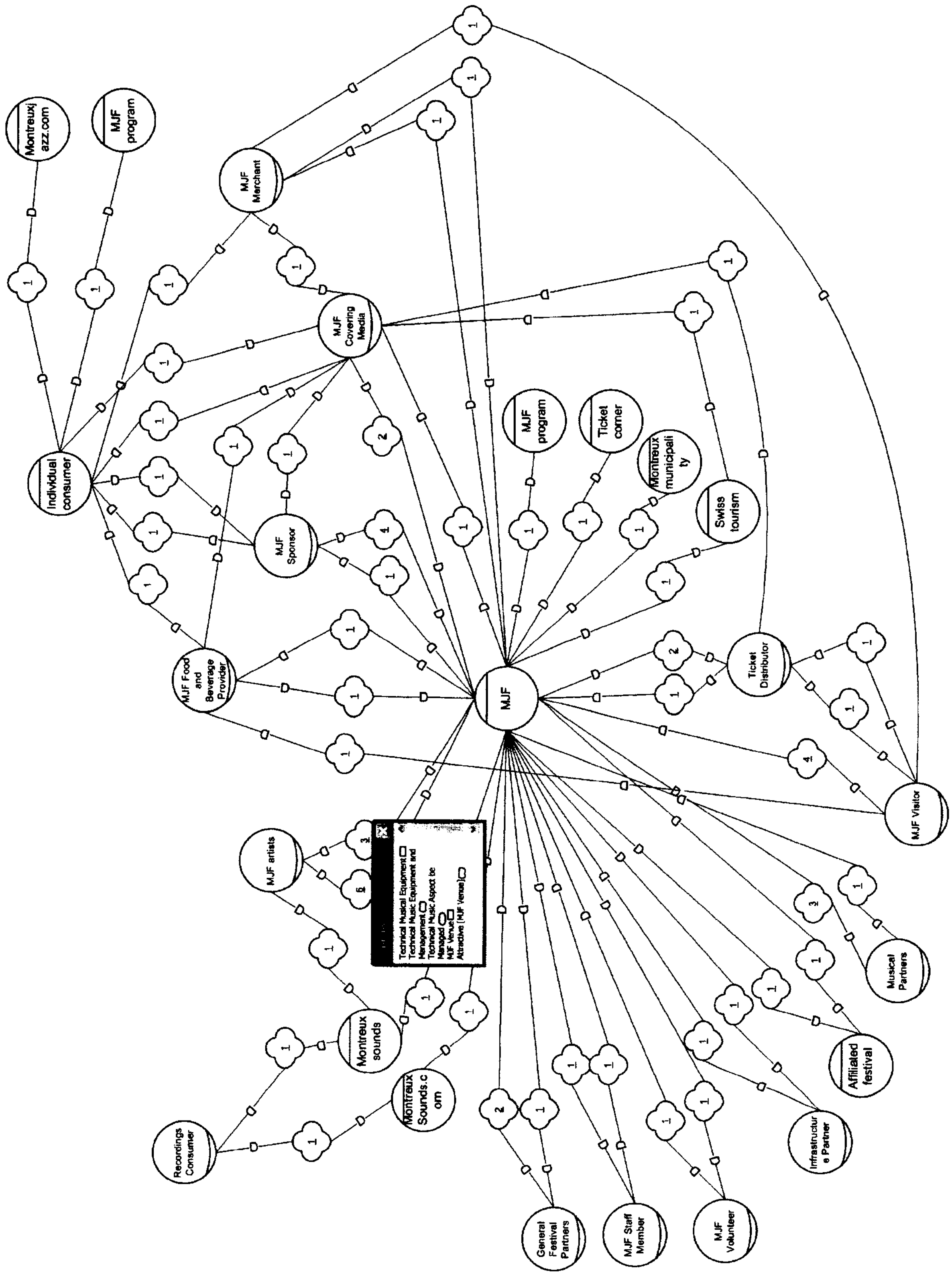
____ **Audio taping**
____ **Video taping**

____ My first name only
or
____ Only a pseudonym

______________________ Signature

______________________ Date


Montreuxj azz.com
MJF program
MJF Merchant
MJF Covering Media
Individual consumer
MJF Sponsor
MJF Food and Beverage Provider
MJF
MJF program
Ticket corner
Montreux municipali ty
Swiss tourism
Ticket Distributor
MJF Visitor
MJF artists
Montreux sounds
Montreux Sounds.c om
Recordings Consumer
General Festival Partners
MJF Staff Member
MJF Volunteer
Infrastructur e Partner
Affiliated festival
Musical Partners


Montreuxjazz.com
MJF program
MJF Merchant
Individual consumer
MJF Covering Media
MJF Sponsor
MJF program
Ticket corner
Montreux municipality
Swiss tourism
MJF Food and Beverage Provider
MJF
Ticket Distributor
MJF Visitor
MJF artists
Musical Partners
Montreux sounds
Montreux Sounds.com
Affiliated festival
Infrastructure Partner
Recordings Consumer
General Festival Partners
MJF Staff Member
MJF Volunteer

depender	dependee	dependums	type
Affiliated Festival	MJF	Permission to use MJF Brand and Franchise	resource
Affiliated Festival	MJF	Promotion [MJF Brand and Franchise]	soft goal
2			
General Festival Partners	MJF	Revenue [Products]	resource
General Festival Partners	MJF	Exclusive Rights	resource
2			
Individual Consumer	MJF Covering Media	News Media	resource
Individual Consumer	Montreuxjazz.com	MJF Merchandise	resource
Individual Consumer	MJF Program	MJF Information	resource
Individual Consumer	MJF Sponsor	Products	resource
4			
Infrastructure Partner	MJF	Payment [Infrastructure]	resource
1			
MJF	MJF Artists	Artists be contracted to MJF	goal
MJF	MJF Program	Information be Provided	goal
MJF	Ticket Corner	Information be Provided	goal
MJF	Ticket Distributor	Tickets be Distributed	goal

MJF	MJF Food and Beverage Provider	Jazz be Used	goal
MJF	Musical Partners	Technical Music Aspects be Managed	goal
MJF	MJF Volunteer	Tasks be Performed	goal
MJF	MJF Staff Member	Tasks be Performed	goal
MJF	Montreux Sounds	Revenue [Recordings]	resource
MJF	MJF Artists	Permission to record concerts	resource
MJF	Ticket Distributor	Revenue [Tickets]	resource
MJF	Montreux Municipality	Free Services	resource
MJF	Swiss Tourism	MJF Advertising	resource
MJF	MJF Sponsor	MJF Advertising	resource
MJF	Covering Media	MJF Advertising	resource
MJF	Ticket Corner	MJF Advertising	resource
MJF	Musical Partners	Technical Music Equipment	resource
MJF	Infrastructure Partner	Infrastructure	resource
MJF	General Festival Partners	Products	resource
MJF	MJF Artists	Good Performance	soft goal
MJF	Affiliated Festival	Promotion [MJF Brand and Franchise]	soft goal
MJF	Musical Partners	Effective Technical Music Equipment and Management	soft goal
MJF	MJF Merchant	Jazz be Used	goal

23

MJF Artists	MJF	Technical Music Aspects be Managed	goal
MJF Artists	Montreux Sounds	Revenue [Recordings]	resource
MJF Artists	MJF	Technical Music Equipment	resource
MJF Artists	MJF	MJF Venue	resource
MJF Artists	MJF	Payment [Concerts]	resource
MJF Artists	MJF	Effective Technical Music Equipment and Management	soft goal
MJF Artists	MJF	Attractive [MJF Venue]	soft goal

7

MJF Covering Media	Individual Consumer	Revenue [News Media]	resource
MJF Covering Media	MJF	Attract People	soft goal
MJF Covering Media	MJF	Contract Stars	soft goal

3

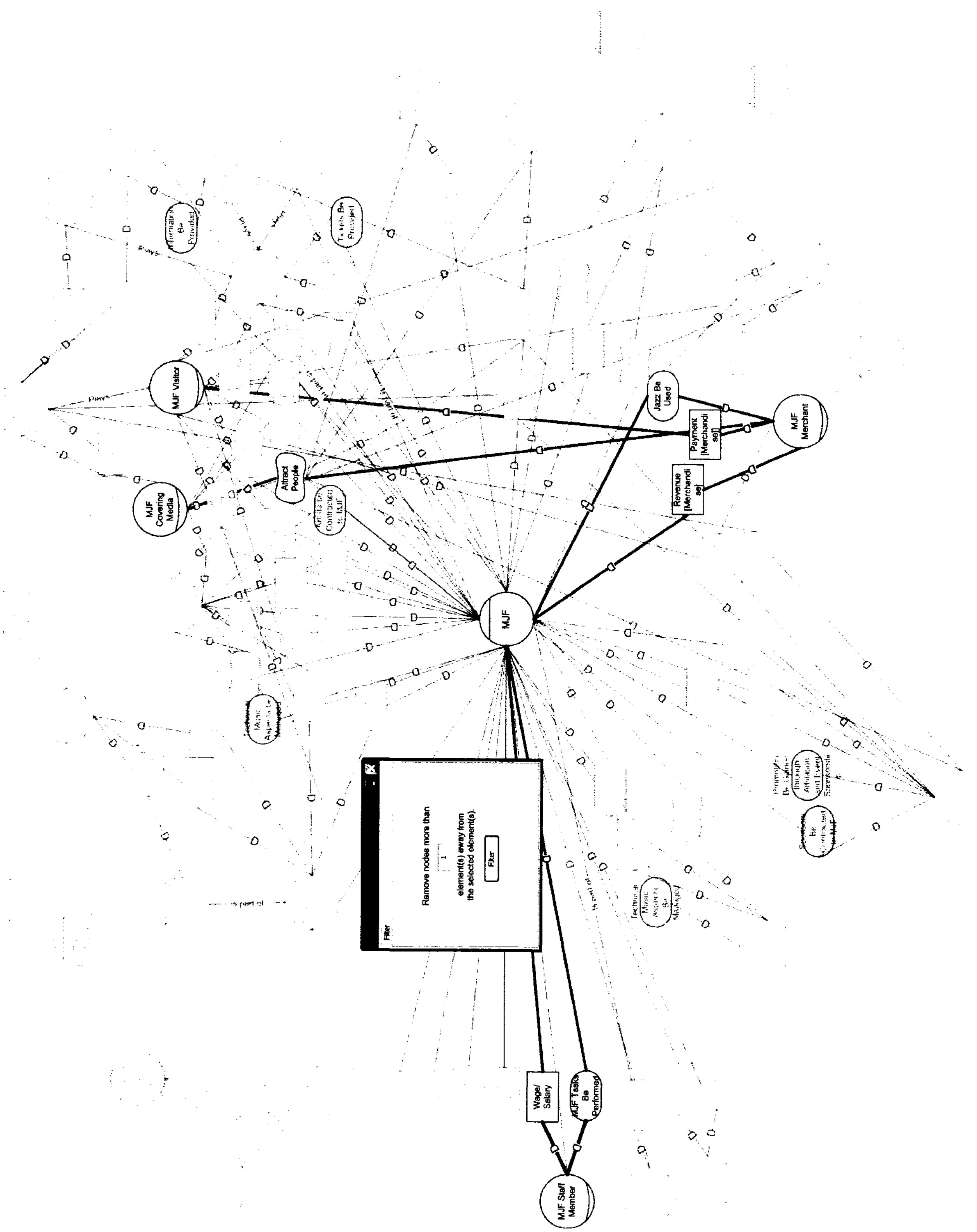
MJF Food and Beverage Provider	MJF	Revenue [Food and Beverage]	resource
MJF Food and Beverage Provider	Individual Consumer	Payment [Food and Beverage]	resource
MJF Food and Beverage Provider	MJF Covering Media	Attract People	soft goal

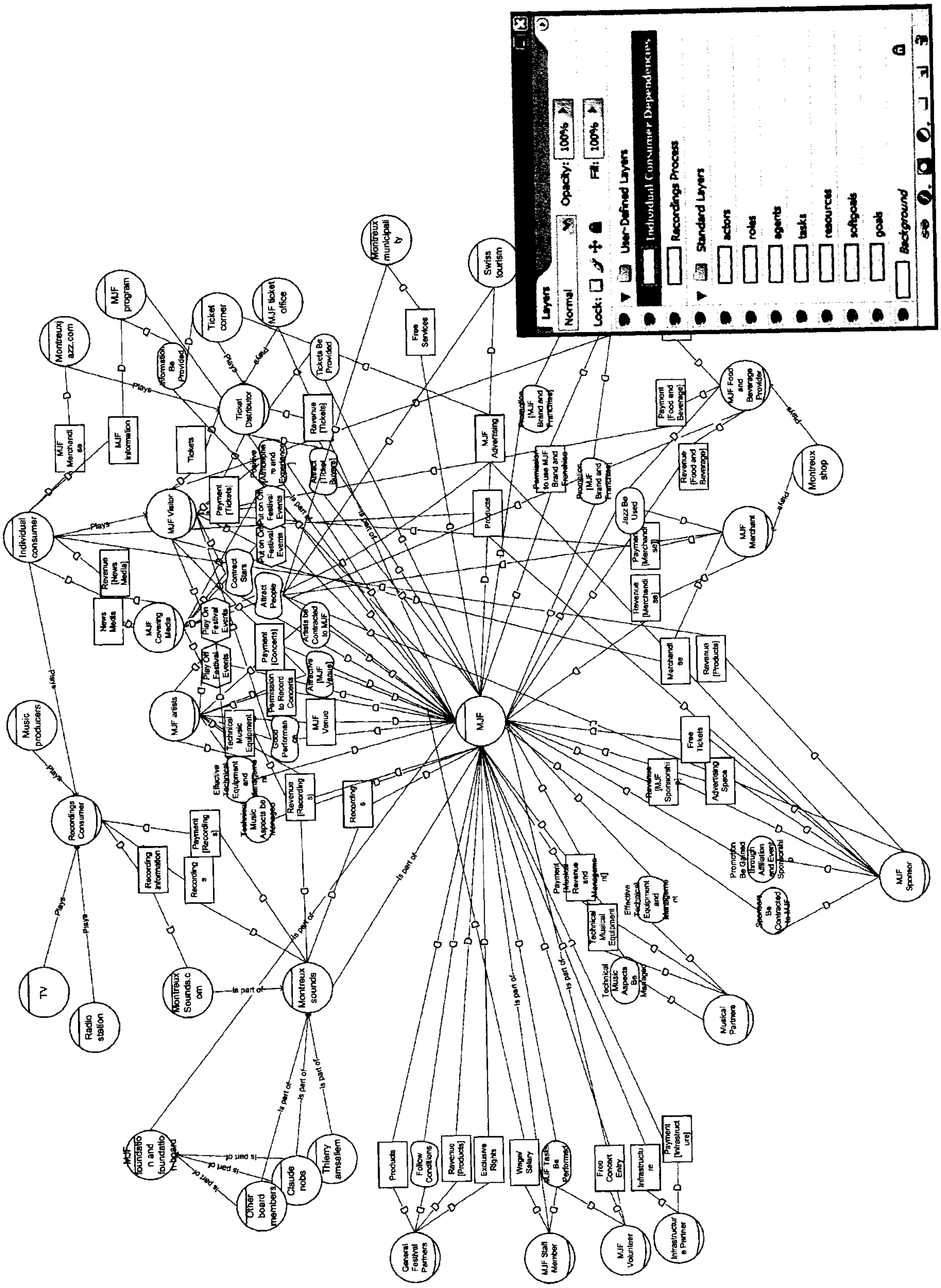
3

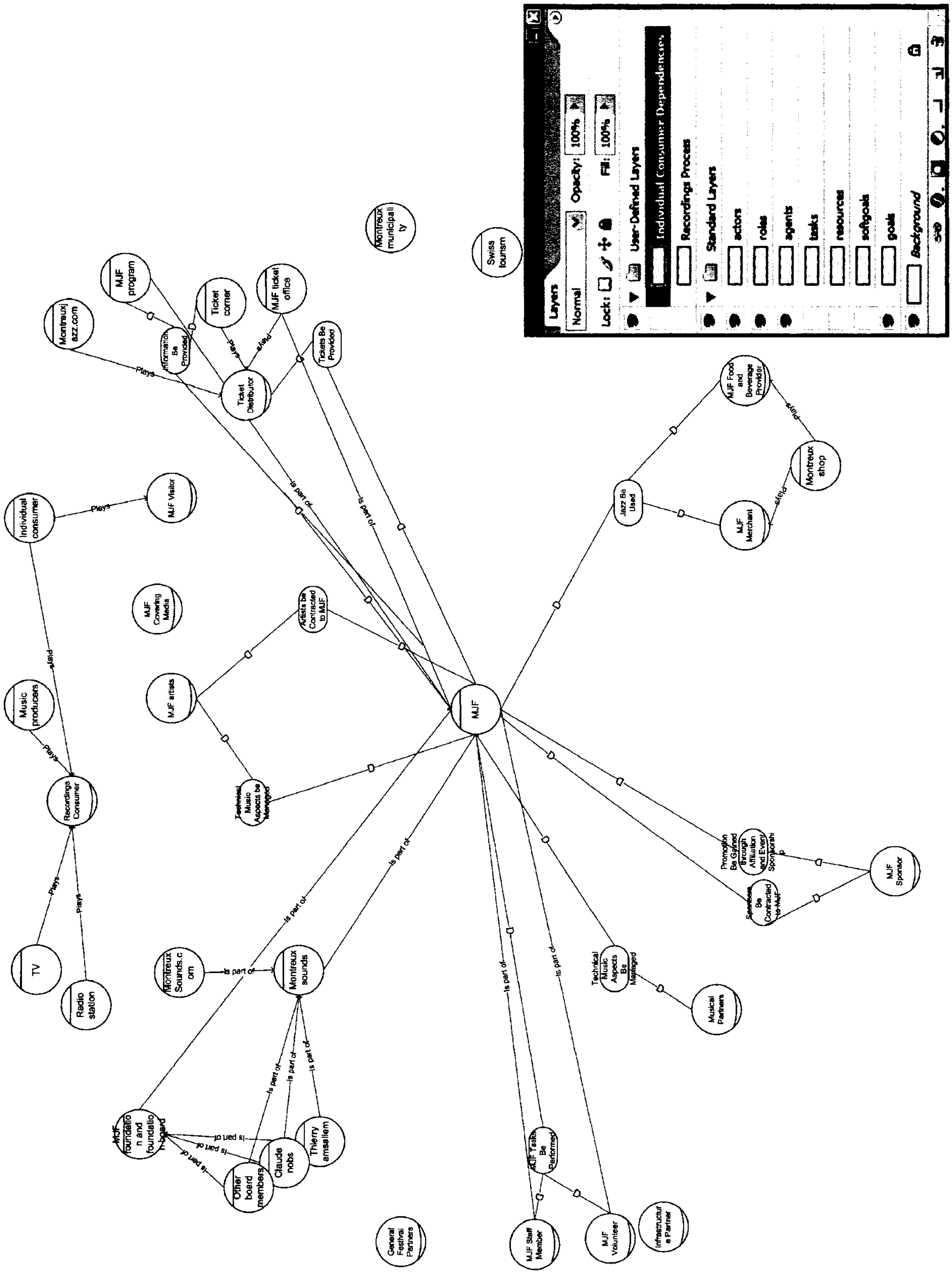
MJF Merchant	MJF	Revenue [Merchandise]	resource
MJF Merchant	Individual Consumer	Payment [Merchandise]	resource

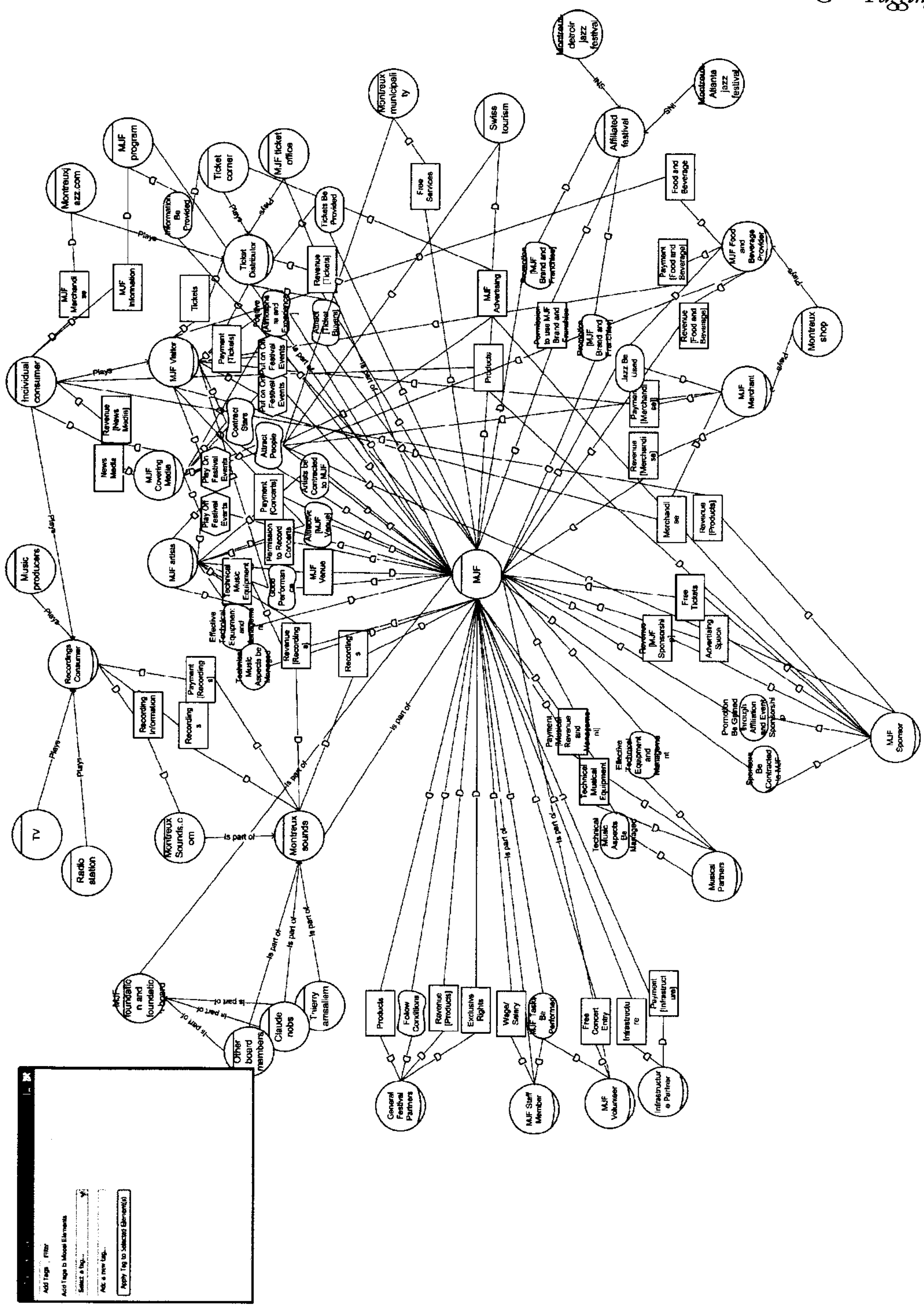
MJF Merchant	MJF Covering Media	Attract People	soft goal
3			
MJF Sponsor	MJF	Promotion Be Gained Through Affiliation and Event Sponsorship	goal
MJF Sponsor	MJF	sponsors be contraced to MJF	goal
MJF Sponsor	MJF	Free Tickets	resource
MJF Sponsor	MJF	Advertising Space	resource
MJF Sponsor	Individual Consumer	Revenue [products]	resource
MJF Sponsor	MJF Covering Media	Attract People	soft goal
6			
MJF Staff Member	MJF	Wage/Salary	resource
1			
MJF Visitor	MJF Merchant	Merchandise	resource
MJF Visitor	MJF Food and Beverage Provider	Food and Beverage	resource
MJF Visitor	Ticket Distributor	Tickets	resource
MJF Visitor	MJF	Attractive [MJF Venue]	soft goal
MJF Visitor	MJF	Good Performance	soft goal
MJF Visitor	MJF	Contract Stars	soft goal
MJF Visitor	MJF	Positive Atmosphere and Experience	soft goal
7			

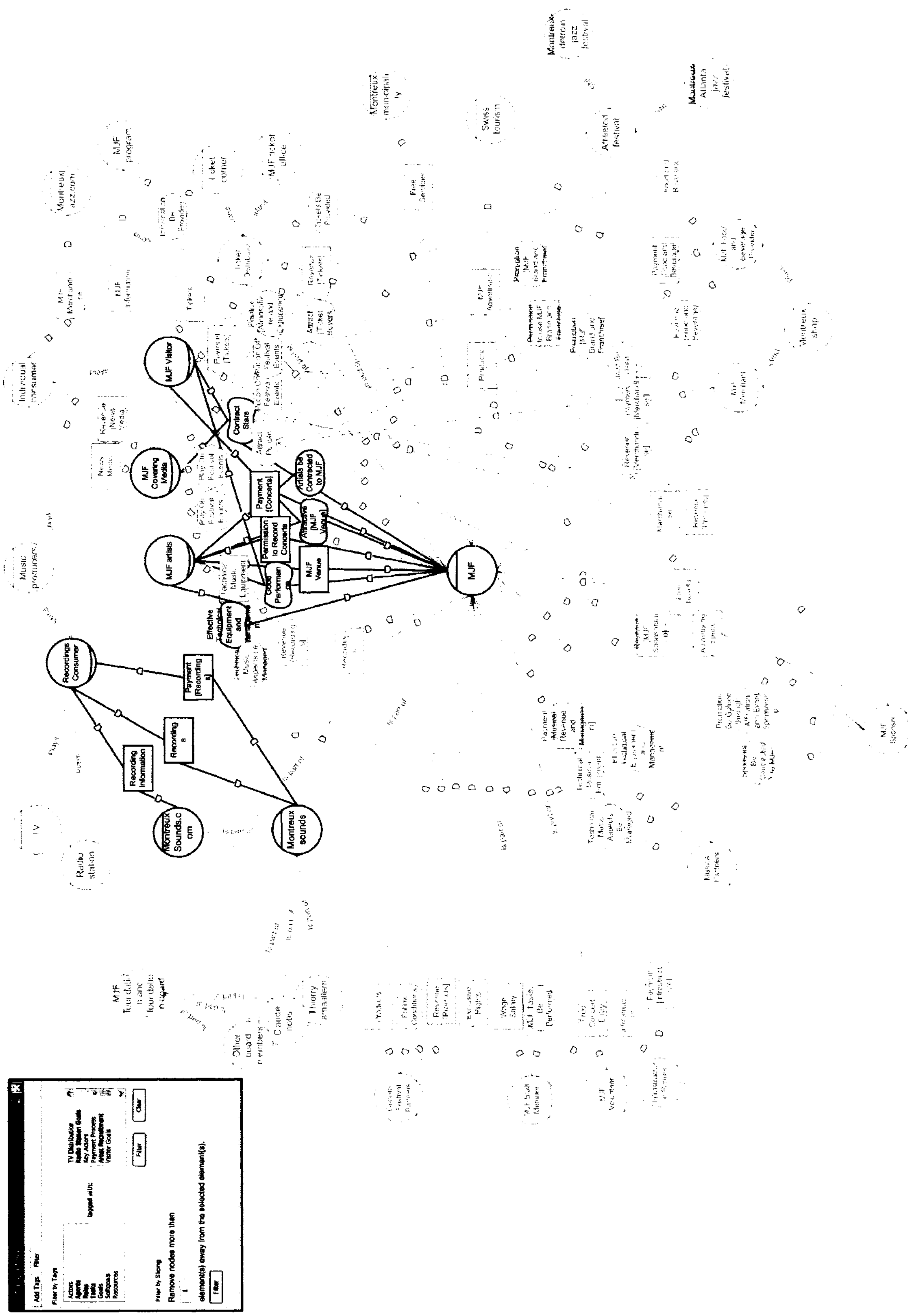
MJF Volunteer	MJF	Free Concert Entry	resource
1			
Montreux Municipality	MJF Covering Media	Attract People	soft goal
1			
Montreux Sounds	Recordings Consumer	Payment [Recordings]	resource
Montreux Sounds	MJF	Recordings	resource
2			
Musical Partners	MJF	Payment [Musical Revenue and Management]	resource
1			
Recordings Consumer	Montreauxsounds.com	Recordings Information	resource
Recordings Consumer	Montreauxsounds.com	Recordings	resource
2			
Swiss Tourism	MJF Covering Media	Attract People	soft goal
1			
Ticket Distributor	MJF Visitor	Payment [Tickets]	resource
Ticket Distributor	MJF	Attract [Ticket buyers]	soft goal
Ticket Distributor	MJF	Follow Conditions	soft goal
2			

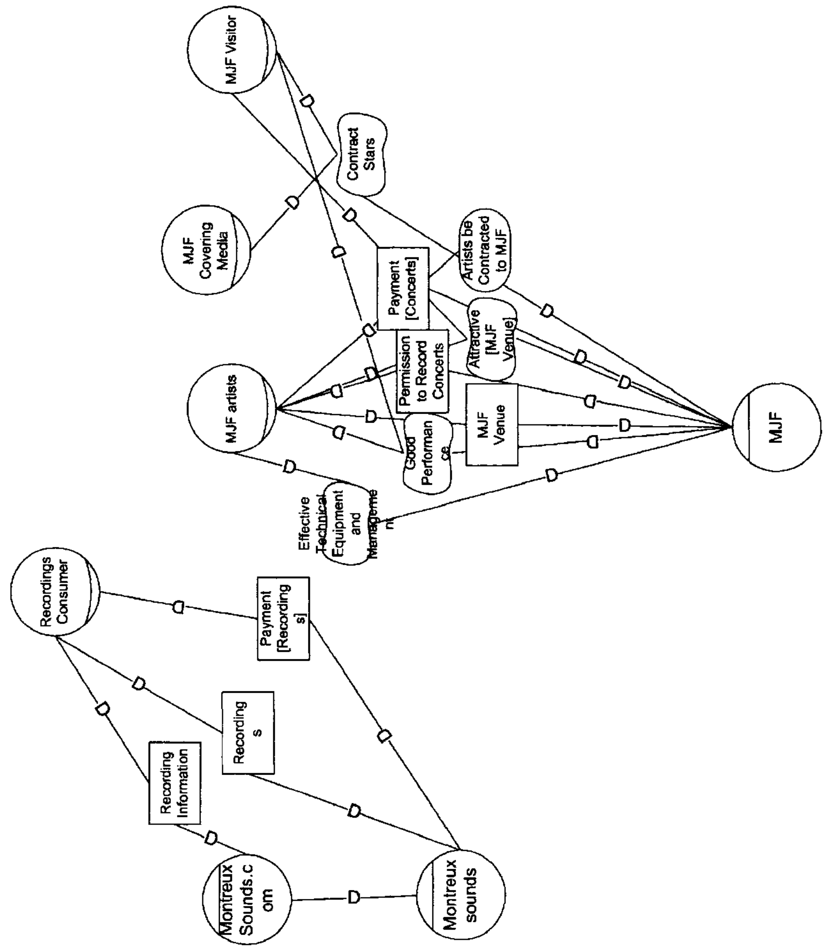

MJF Visitor
MJF Covering Media
Attract People
MJF
Jazz Be Used
Payment [Merchandise]
MJF Merchant
Revenue [Merchandise]
Filter
Remove nodes more than
1
element(s) away from
the selected element(s).
Filter
Wage/ Salary
MJF Tasks Be Performed
MJF Staff Member

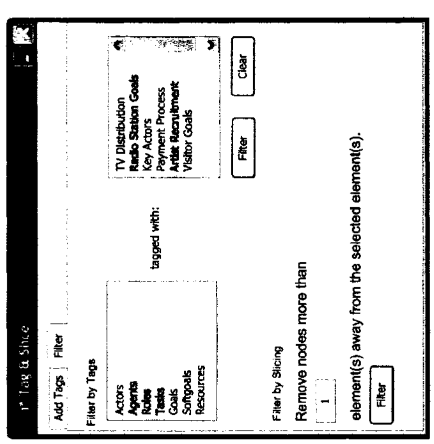

Layers
Normal
Opacity: 100%
Lock:
Fill: 100%
User-Defined Layers
Individual Consumer Dependencies
Recordings Process
Standard Layers
actors
roles
agents
tasks
resources
softgoals
goals
Background
MJF
Individual consumer
MJF Visitor
Montreux jazz.com
MJF program
Ticket corner
MJF ticket office
Ticket Distributor
Montreux municipality
Swiss tourism
Free Services
Tickets Be Provided
Revenue [Tickets]
MJF Advertising
MJF Merchandise
MJF Information
Tickets
Payment [Tickets]
Products
MJF Food and Beverage Provider
Payment [Food and Beverage]
Revenue [Food and Beverage]
Montreux shop
MJF Merchant
Jazz Be Used
Revenue [News Media]
News Media
MJF Covering Media
Contract Stars
Attract People
Artists Be Contracted to MJF
Payment [Concerts]
MJF artists
MJF Venue
Music producers
Recordings Consumer
Recording Information
Recordings
Payment [Recordings]
Revenue [Recordings]
Free Tickets
Advertising Space
Revenue [Products]
MJF Sponsor
Technical Musical Equipment
Musical Partners
TV
Radio station
Montreux Sounds.com
Montreux sounds
Claude nobs
Thierry amsallem
Other board members
Products
Follow Conditions
Revenue [Products]
Exclusive Rights
Wage/ Salary
MJF Tasks Be Performed
Free Concert Entry
Infrastructure
Payment [Infrastructure]
General Festival Partners
MJF Staff Member
MJF Volunteer
Infrastructure Partner
Plays
is part of

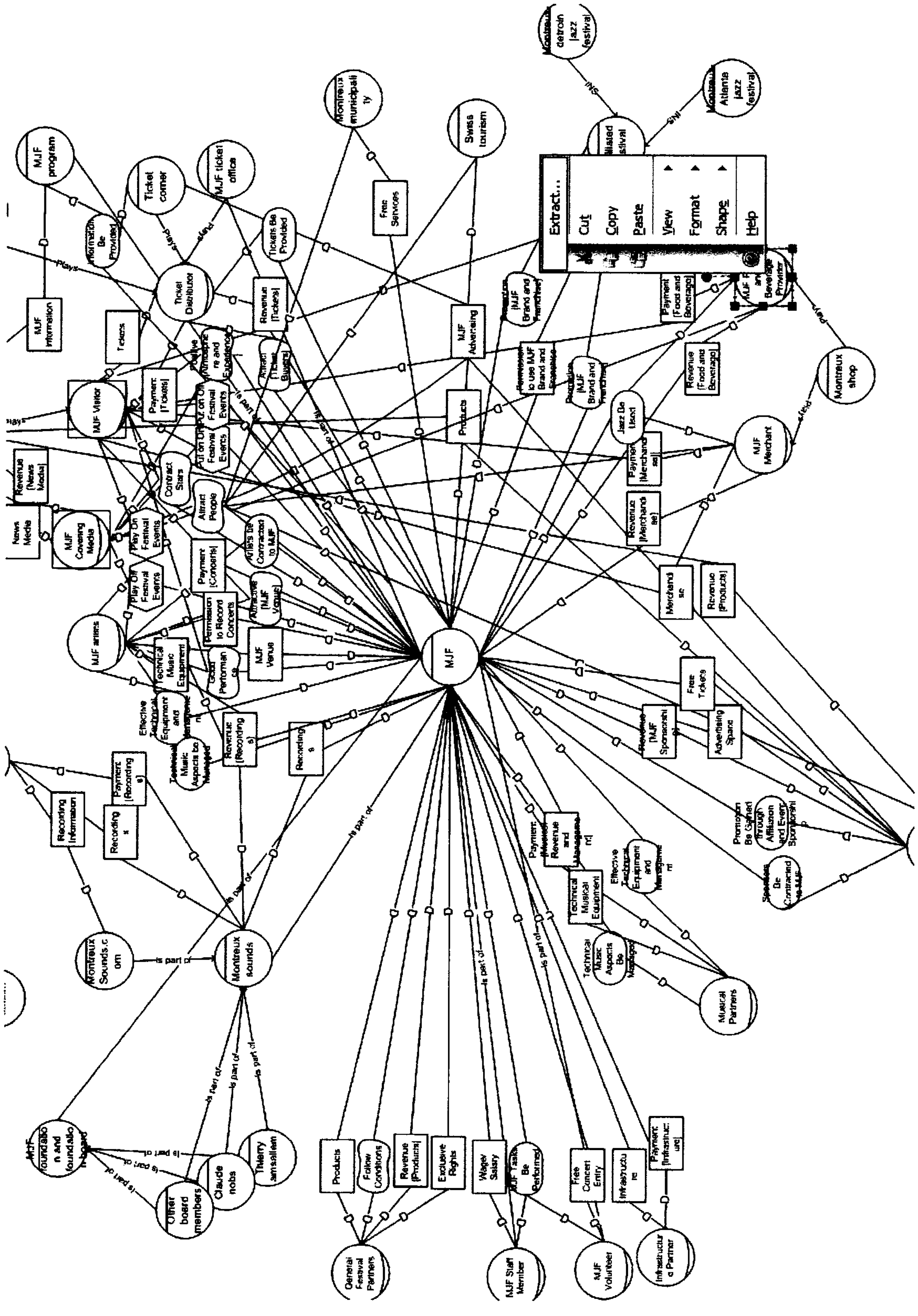

Layers
Normal
Opacity: 100%
Lock:
Fill: 100%
User-Defined Layers
Individual Consumer Dependencies
Recordings Process
Standard Layers
actors
roles
agents
tasks
resources
softgoals
goals
Background
MJF
Ticket Distributor
MJF program
Ticket corner
MJF ticket office
Montreuxjazz.com
Information Be Provided
Tickets Be Provided
Montreux municipality
Swiss tourism
MJF Food and Beverage Provider
Montreux shop
MJF Merchant
Jazz Be Used
Individual consumer
MJF Visitor
MJF Covering Media
Artists be Contracted to MJF
MJF artists
Music producers
Recordings Consumer
Technical Music Aspects be Managed
TV
Radio station
Montreux Sounds.com
Montreux sounds
MJF foundation and foundation board
Other board members
Claude nobs
Thierry amsallem
General Festival Partners
MJF Staff Member
MJF Volunteer
MJF Tasks Be Performed
Infrastructure Partner
Musical Partners
MJF Sponsor
Plays
is part of

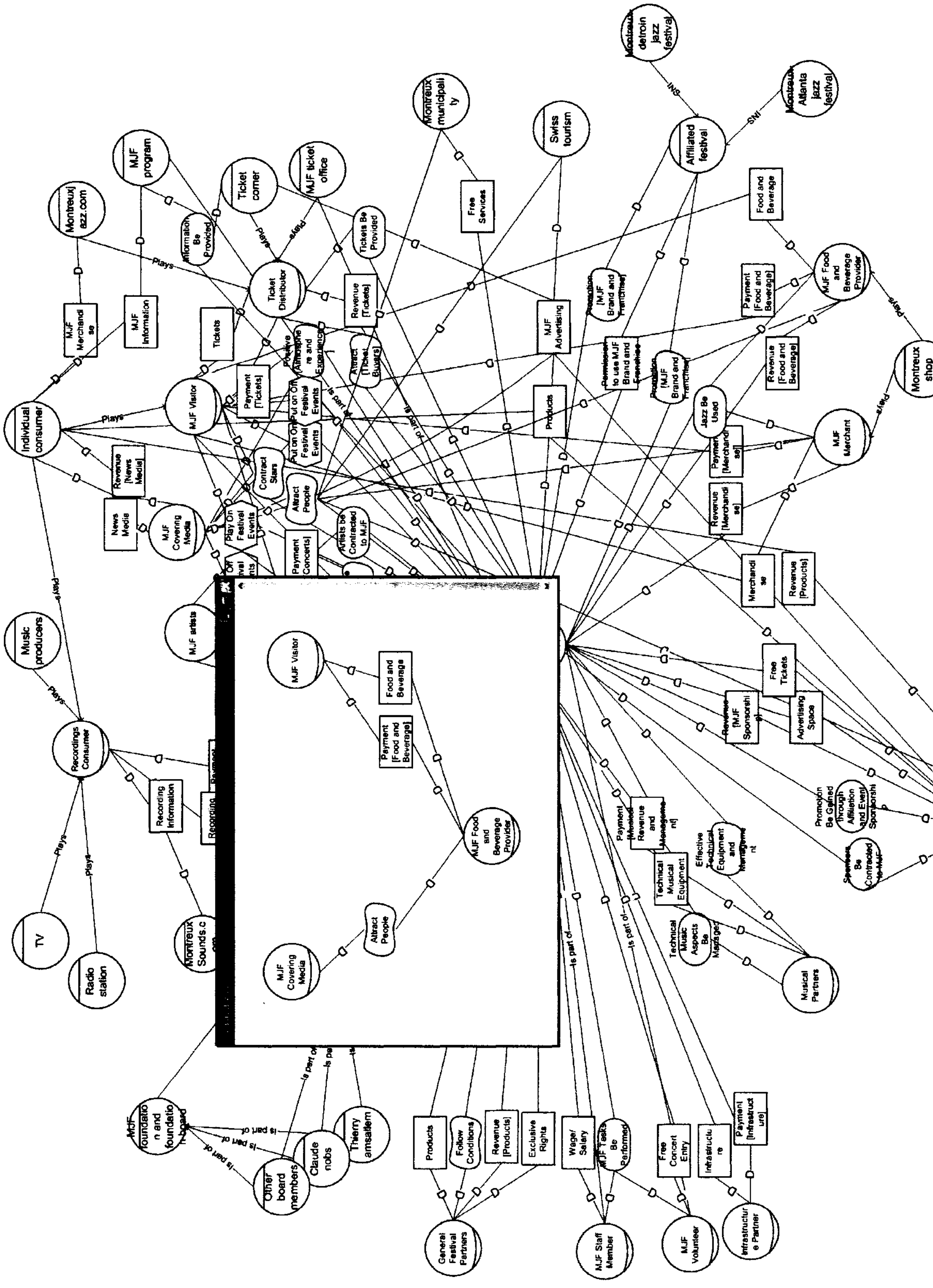

MJF
Montreux sounds
Montreux Sounds.com
MJF Sponsor
Individual consumer
MJF Visitor
MJF artists
Recordings Consumer
Music producers
TV
Radio station
Ticket corner
MJF ticket office
MJF program
Ticket Distributor
Montreux municipality
Swiss tourism
Affiliated festival
MJF Food and Beverage provider
Montreux shop
MJF Merchant
MJF Covering Media
Musical Partners
General Festival Partners
MJF Staff Member
MJF Volunteer
Claude nobs
Thierry amsallem
Other board members
is part of
Plays
Tickets
Products
Exclusive Rights
Free Services
Food and Beverage
MJF Advertising
Advertising Space
Free Tickets
Wage/ Salary
Free Concert Entry
Infrastructure




MJF Visitor
Contract Stars
MJF Covering Media
Artists be Contracted to MJF
Payment [Concerts]
Attractive [MJF Venue]
Permission to Record Concerts
MJF artists
MJF Venue
Good Performance
MJF
Effective Technical Equipment and Management
Recordings Consumer
Payment [Recordings]
Recordings
Recording Information
Montreux Sounds.com
Montreux sounds


Tag & Slice
Add Tags
Filter
Filter by Tags
Actors
Agents
Roles
Tasks
Goals
Softgoals
Resources
tagged with:
TV Distribution
Radio Station Goals
Key Actors
Payment Process
Artist Recruitment
Visitor Goals
Filter
Clear
Filter by Slicing
Remove nodes more than
1
element(s) away from the selected element(s).
Filter


MJF
Montreux sounds
Montreux Sounds.com
MJF artists
MJF Visitor
Ticket Distributor
MJF program
MJF ticket office
Ticket corner
Swiss tourism
Montreux municipality
Free Services
MJF Advertising
Products
MJF Merchant
Montreux shop
Musical Partners
General Festival Partners
MJF Staff Member
MJF Volunteer
Infrastructure Partner
Claude nobs
Thierry amsallem
Other board members
Extract...
Cut
Copy
Paste
View
Format
Shape
Help
is part of


Individual consumer
MJF Visitor
MJF program
Montreuxjazz.com
Ticket corner
MJF ticket office
Ticket Distributor
Montreux municipality
Swiss tourism
Affiliated festival
Montreux detroit jazz festival
Montreux Atlanta jazz festival
MJF Food and Beverage Provider
Montreux shop
MJF Merchant
MJF Sponsor
Musical Partners
Infrastructure Partner
MJF Volunteer
MJF Staff Member
General Festival Partners
Claude nobs
Thierry amsallem
Other board members
MJF foundation and foundation board
Montreux Sounds.com
TV
Radio station
Recordings Consumer
Music producers
MJF artists
MJF Covering Media
News Media
Food and Beverage
Payment [Food and Beverage]
Revenue [Food and Beverage]
Attract People
Tickets
Payment [Tickets]
Revenue [Tickets]
MJF Merchandise
MJF Information
Revenue [News Media]
Free Services
MJF Advertising
Products
Merchandise
Revenue [Products]
Free Tickets
Advertising Space
Recording Information
Exclusive Rights
Wage/ Salary
Free Concert Entry
Payment [Infrastructure]
Plays
SNI
is part of

Prototype 1 – Model Overview: Condensed Graphical

The intent of this model is to provide an overview of the SD i* model. This prototype merges similar dependency links. The number of dependencies between the depender and dependee is displayed for the user. The prototype displays details in regards to the dependencies in a pop-up box that can be accessed by clicking on the hyperlinked number.

Prototype 2 – Model Overview: Textual, Tabular

This prototype represents the model in a tabular format. Model data can be sorted by element type, by element name. Sub-counts can be produced to display which actors contain the most dependencies.

Prototype 3 – Model Filtering using Tagging

Tags can be applied to the model using this prototype. Tags are applied by selecting one or more items and either creating a new tag for the element(s) or by selecting tags from the existing collection. The user can then apply various attributes to the tags, such as colour, transparency and filtering.

Prototype 4 - Model Filtering using Slicing

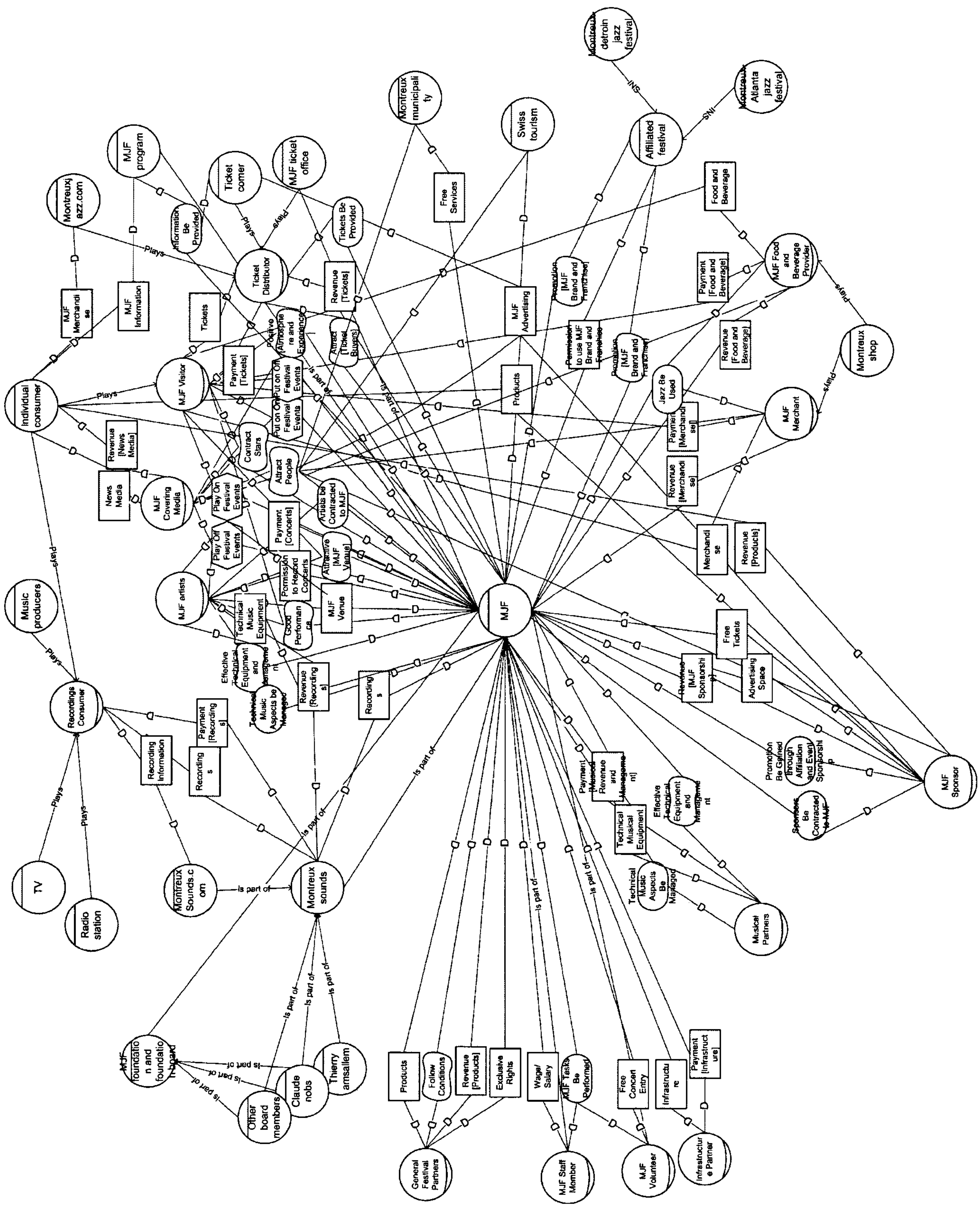
This mechanism allows users to filter elements from the model. The user can select one or more elements and chose to filter all information more than *N* elements away from the selected node.

Prototype 5 – Model Filtering using Layers

This mechanism allows users to assign elements to layers. Layers are stacked upon one another and can be made visible or invisible by the user. Elements can be selected and assigned to one or more layers. Layers can be named by the user. A number of default layers exist for the common i* element types, such as goal, soft goal, resource, task, agent, etc. When a user adds such an element type to the canvas, the element is automatically assigned to the appropriate layer.

Prototype 6 – Extraction/Isolation of Selected Actors

This prototype enables users to select elements from the model and display them in a new window. If one element is selected, all elements directly related to the element are also extracted to the new window. If more than one element is extracted, items shared by the selected elements will be extracted. Multiple windows can be displayed at one time, and each window can be individually saved or printed.


MJF
MJF Visitor
Individual consumer
MJF artists
Recordings Consumer
Music producers
TV
Radio station
Montreux sounds
Montreux Sounds.com
MJF foundation and foundation board
Claude nobs
Thierry amsallem
Other board members
General Festival Partners
MJF Staff Member
MJF Volunteer
Infrastructure Partner
Musical Partners
MJF Sponsor
MJF Merchant
Montreux shop
MJF Food and Beverage Provider
Affiliated festival
Montreux detroin jazz festival
Montreux Atlanta jazz festival
Swiss tourism
Montreux municipality
MJF ticket office
Ticket corner
Ticket Distributor
MJF program
Montreuxjazz.com
MJF Covering Media
News Media
MJF Venue
Plays
Is part of
Free Services
Food and Beverage
Tickets
MJF Information
MJF Merchandise
Products
Merchandise
Revenue [Products]
Free Tickets
Advertising Space
MJF Advertising
Recordings
Payment [Recordings]
Recording Information
Revenue [Recordings]
Payment [Tickets]
Payment [Concerts]
Revenue [Tickets]
Revenue [News Media]
Payment [Food and Beverage]
Revenue [Food and Beverage]
Payment [Merchandise]
Revenue [Merchandise]
Revenue [MJF Sponsorship]
Technical Music Equipment
Technical Musical Equipment
Follow Conditions
Exclusive Rights
Wage/ Salary
Free Concert Entry
Infrastructure
Payment [Infrastructure]
Contract Stars
Attract People
Play On Festival Events
Play Off Festival Events
Jazz Be Used

Task 1

Analyze the relationship between MJF artists and MJF.

Task 2

Compare the relationships of MJF and Ticket Corner with the relationships of MJF and General Festival Partners.

Task 3

Determine the most important actor in the model.

Task 4

Determine all of the goals of MJF

Opening Interview

Please explain your experience with i* modelling.

Please describe any software you have used when working with i* modelling.

Following Each Task

Please select the prototype you believe would *best* assist you in completing this task.

Why do you feel this prototype was the best choice?

How will this prototype help you for this task? What functions of the prototype are especially useful?

This prototype is in its early stages of development; we expect that there are a number of improvements or changes that could be made to make it better. Could you think of ways to better assist this task?

Were there other prototypes you also considered in selecting? Why? Why did you not select this prototype as the best choice?

Exit Interview Question

Were there certain aspects, concepts or features that were not included in any of the prototypes that you feel would make analyzing i* models easier?

References

14th IEEE International Requirements Engineering Conference. (2006). Retrieved April 18, 2007 from http://www.ifi.unizh.ch/req/events/RE06/WS_Tut_DocSym/Workshops.html#W2

Adobe. (2007) Retrieved March 12, 2007 from http://www.adobe.com/

ArcView: The Geographic Information System for Everyone. (2006). Retrieved March 12, 2007 from http://www.esri.com/software/arcview/

Bertolini, D., Novikau, A., Susi, A., & Perini, A. (n.d.). *TAOM4E: an Eclipse ready tool for Agent-Oriented Modeling. Issue on the development process.* Retrieved March 12, 2007 from http://sra.itc.it/tools/taom4e/references.php?action=show&file=TAOM4E_technical_report.pdf

Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J. & Perini, A. (2004). TROPOS: An Agent-Oriented Software Development Methodology. *Journal of Autonomous Agents and Multi-Agent Systems, 8(3)*, 203-236.

Breitkreutz, B.J., Stark, C., Tyers, M. (2003). Osprey: a network visualization system. *Genome Biology, 4(R:22).*

Card, S. K., Mackinlay, J., & Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think.* San Francisco: Morgan Kaufmann.

Chung, K.L., Nixon, B., & Yu, E. (1995). Using non-functional requirements to systematically support change. (1995). *In Proceedings of the Second IEEE International Conference on Requirements Engineering (RE'95).* Los Alamitos, California.

Chung, V., Cocca, C., Samavi, R., Sutherland, J. (2005). *Systems Analysis with Strategic Modelling: FIS 1341 Assignment.* Unpublished Manuscript.

Creswell, J. W. (2003). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches.* 2nd ed. Thousand Oaks, CA: Sage.

The Eclipse Foundation. (2007a). *Eclipse.org Home.* Retrieved March 11, 2007 from http://www.eclipse.org/

The Eclipse Foundation. (2007b). *Eclipse Modeling – EMF Home.* Retrieved March 11, 2007 from http://www.eclipse.org/modeling/emf/?project=emf

The Eclipse Foundation. (2007c). *Eclipse Graphical Editing Framework (GEF).* Retrieved March 11, 2007 from http://www.eclipse.org/gef/plan/gef_plan_3_3.php

Feather, M.S., Cornford, S.L., Kiper, J.D., & Menzies, T. (2006). Experiences using Visualization Techniques to Present Requirements, Risks to Them, and Options for Risk Mitigation. *In Proceedings of the First Annual Requirements Engineering Visualization Conference (REV'06).* Waterloo, ON, pp. 10-19.

Grau, G. (n.d.) *JPrim Tool.* Retrieved March 11, 2007 from http://www.lsi.upc.es/~ggrau/JPRIM/index.html

Grau, G. (2006). *REDEPEND-REACT – Architecture Analysis Tool,* Retrieved July 12, 2006 from http://www.lsi.upc.es/~ggrau/REDEPEND-REACT/

GR-TOOL. (2006). Retrieved July 17, 2006 from http://sesa.dit.unitn.it/goaleditor/.

Horkoff, J. (2006). Using i* for Evaluation. Unpublished master's thesis, University of Toronto, Ontario, Canada. Retrieved May 20, 2006 from: http://www.cs.utoronto.ca/~jenhork/MScThesis/Thesis.pdf

i* Wiki: i* Wiki Home. (2006). *i* Tools.* Retrieved March 11, 2007 from http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=2

Lapouchnian, G. (n.d.). *The Knowledge Management Lab.* Retrieved March 11, 2007 from http://www.cs.toronto.edu/km/

Leica, M. (2005). Scalability Concepts for i* Modelling and Analysis. Unpublished Master's Thesis, University of Toronto, Ontario, Canada. Retrieved July 22, 2006 from http://www.cs.toronto.edu/pub/eric/O/Marcel-Leica-MSc-TH.pdf

Lindo System's index page. (2006). Retrieved March 11, 2007 from http://www.lindo.com/

Lockerbie, J. (2005). *Automating the Pattern-Based Generation of Requirements for I* System Models,* MSc Dissertation, School of Informatics, City University.

Maiden, N.A.M., Jones, S., Ncube, C., & Lockerbie, J. (n.d.). *Using i* in Requirements Projects: Some Experiences and Lessons,* Retrieved March 12, 2007 from http://www.cs.toronto.edu/pub/eric/istarbk/Maiden-istar-ch.pdf

Maiden, N.A.M., Manning, S., Jones, S., & Greenwood, J. (2004). Towards Pattern-Based Generation of Requirements from System Models, *Proceedings of the 10th Anniversary International Workshop on Requirements Engineering: Foundation for Software Quality.* Riga, Latvia.

Maiden, N.A.M. & Rugg, G. (1996). *ACRE:* Selecting methods for Requirements Acquisition, *Software Engineering Journal,* 11, pp. 183-192.

openOME – A Requirements Engineering Tool. (2006). Retrieved March 11, 2007 from http://www.cs.toronto.edu/km/openome/

Oxford University Press. (2004). "software". *Oxford Reference Online.* Retrieved March 12, 2007 from http://www.oxfordreference.com.myaccess.library.utoronto.ca/views/ENTRY.html?s ubvie w=Main&entry=t11.e4896

Ozkaya, I. (2006). Representing Requirement Relationships. *Proceedings of the First Annual Requirements Engineering Visualization Conference (REV'06)* Waterloo, ON, pp. 3-7.

Preece, J. Rogers. Y., & Sharp, H. (2002). *Interaction Design: beyond human-computer interaction.* New York: Wiley.

Roberston, S. (2001). Requirements Trawling: techniques for discovering requirements. *International Journal of Human Computer Studies, 55*, pp. 405-421.

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. Proceedings from VL '96: *The 11th IEEE International Symposium on Visual Languages*, Boulder, CO, pp. 336-343.

Synder, C. (2003). *Paper prototyping: the fast and easy way to design and refine user interfaces.* San Francisco: Morgan Kaufmann.

TAOM4E Homepage. (2006). Retrieved July 12, 2006 from http://sra.itc.it/tools/taom4e/

Taylor, S.J. & Bogdan, R. (1984). *Introduction to Qualitative Research Methods: The Search for Meanings.* New York: Wiley

Tropos. (2007). Retrieved March, 26 2007 from http://troposproject.org

You, J. (2004). Meta-model-driven views to address scalability in i* modeling. Unpublished master's thesis, University of Toronto, Ontario, Canada. Retrieved August 22, 2006 from http://www.cs.toronto.edu/pub/eric/O/JaneYouMSc-TH-v4.4.pdf

Yu, E. & Myloupoulos, J. (1994). Understanding Why in Software Process Modelling, Analysis, and Design. Proceedings from ICSE'94: *16th International. Conference on Software* Engineering, pp. 159-168.

Yu, E. (1995). Modelling strategic relationships for process reengineering. PhD Thesis, Department of Computer Science, University of Toronto, Ontario Canada.

Yu, E. (1997). 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. pp. 226-235.

Yu, E. & Mylopoulous, J. (1994). Towards Modelling Strategic Actor Relationships for Information Systems Development – With Examples from Business Process Reengineering. *Proceedings of the 4th Workshop on Information Technologies and Systems.*

Yu, Y. (2000). *OME3 Home Page.* Retrieved March 11, 2007 from http://www.cs.toronto.edu/km/openome/