SCALABILITY CONCEPTS FOR I* MODELLING AND ANALYSIS

by

Marcel Florin Leica

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

SCALABILITY CONCEPTS FOR I* MODELLING AND ANALYSIS

Marcel Florin Leica

Master of Science

Graduate Department of Computer Science

University of Toronto

2005

Goal and agent oriented modelling techniques represent one of the major accomplishments of the Requirements Engineering research community.

As modelling frameworks cross the boundaries between research phase prototypes and application in complex real life projects, scalability challenges, which could not be easily anticipated through simple example application, surface.

Models, like software products, have their own lifecycle. They go through an initial development phase, are then validated by stakeholders and finally used for reasoning purposes by the analysis team. We argue that each phase entails its own scalability challenges and requires specific mechanisms to address them.

We propose an iterative, “one concern at a time“ analysis approach and present a set of model decomposition concepts based on the i* model topology and the analysis question at hand, to address complex model reasoning challenges. Additionally, we offer our insights and practical experience results in dealing with complex model development and validation difficulties.

Contents

List of figures

List of tables

1.Introduction

1.1 Motivation

Modelling techniques are widely accepted in the Requirements Engineering community as an elicitation aid as well an indispensable tool for organizing information and supporting the reasoning process in the early design phases [1]. Models are useful in guiding the elicitation process, uncovering problems such as conflicting requirements or misunderstandings of the domain [2], but most importantly they provide the modeller with an abstraction of the domain which allows in-depth analysis and predictive reasoning to be performed.

Over the last decades a large number of modelling notations have been developed, addressing various stages of the software engineering process and having various degrees of expressivity.
With notations such as UML [3], now widely recognized as the industrial standard of the software development process, which address mainly the software design phase, it became evident that the early analysis phases could also benefit from the use of modelling support. However, the object-oriented paradigm, which is well suited for the design and development phases, does not provide the adequate abstraction mechanisms to represent uncertainty, intentionality and dependencies which are part of the real world.

With the rise of Requirements Engineering as field in its own right in the Software Engineering area, other paradigms have emerged as more adequate abstractions for the modelling needs of the early elicitation phases. Among those, the most prominent ones are the goal-oriented [4, 5] and agent-oriented paradigms [6]. One of the most popular modelling frameworks for the early requirements engineering stages is i* [7], which embodies the goal and agent oriented paradigms.

The central concept of the i* modelling framework is the intentional actor. While most agent models and languages are intended as abstractions of computational behaviour and interactions, i*'s actors are autonomous, social and intentional [8]. Actors have intentional properties such as goals, beliefs commitments and dependencies. This enables i* models to capture uncertainty in requirements, expose actor vulnerabilities and explore the social dimensions of actor interactions which are more difficult, if not impossible, to represent in the frameworks focused on behaviour and interactions only.

Given the increased reliance on technological solutions and the multitude of dimensions they affect, the "one size fits all" solutions are no longer acceptable. Past failures such as the London Ambulance System or the Arianne 5 examples [9] have left their mark on the way software engineers approach technology related projects. Practice has shown that an increased emphasis on the early phases of the requirements engineering process yields results that are more "fit-for-purpose" and leads to lower development costs.[10]

Modelling notations differ widely in terms of concepts, richness of representational constructs and their degree of support for the reasoning processes. Notations with few high-level representational constructs usually yield simple models that can be used as a bird's eye view of the big picture. The effort required to create models based on such notations is relatively small but the reasoning support of the resulting diagrams is also reduced. Such modelling notations were advocated by early researchers in the Data Management Systems area as the right level of abstraction for representing and reasoning about database schemas. According to [11] "the usefulness of any diagram is inversely proportional to the size of the model depicted".

Psychological research also seems to support this idea since studies using a number of different stimuli have found that the human cognitive capacity is limited to ''seven plus or minus two'' concepts at a time [12, 13].

Studies conducted for developing and testing a quality framework for data models have reached similar conclusions regarding diagram comprehensibility by novice end-users [14, 15].
However the same studies show that "people with higher levels of expertise can handle larger amounts of information because they have more highly developed knowledge schemas" [15], and require fewer working memory elements to understand the same amount of information [16, 17].

During the early phases of the requirements engineering process the focus of the analysis team is to collect as much information as possible about the domain to enable comprehensive reasoning to be performed later. Requirements engineering modelling notations are usually richer than data models in terms of representational constructs to allow a faithful and accurate representation of the information. It comes to us as no surprise that typical requirements engineering models abstracting organizational contexts number hundreds and even thousands of elements. Such diagrams are well beyond the comprehensibility point of novice users and their high complexity pose challenges for their usage by even expert modellers.

Reducing the size of the models to the point of compliance with the ''seven plus or minus two'' rule of thumb has little or no effect on expert user understanding and may even have a negative effect called the expertise reversal effect [18]. Therefore different development guidelines and complexity management approaches should be applied for detailed models intended for comprehensive reasoning than for models intended for presentation to novice users.
The implications of these studies for conceptual modeling are:

- The “seven plus or minus two” items in a diagram, rule of thumb applies to novice users but is not applicable to people with a higher level of expertise.
- Expert modellers need different mechanisms in coping with the model complexity, which is a function of the underlying domain complexity.
- Model presentability constraints should not be applied to models intended for extensive requirements engineering analysis.
- Validation of complex model content by the stakeholders should be performed using intermediary forms of representation and different levels of abstraction.

Our work aims to identify the analysis needs of expert modellers when dealing with increased model complexity and to provide diagram simplification mechanisms which are relevant for the analysis questions that arise in the early stages of the software engineering process.

Given the fact that models can abstract all the information captured in the elicitation phase, they are perfect vehicles for conducting reasoning about business re-engineering, service planning and organizational impact analysis.

When models reach a high degree of complexity however, reasoning on them without a viable analysis strategy is very difficult. When operating on models with high intrinsic complexity, the modeller should have a very clear set of analysis questions that he would like to reason about, based on the information stored in the diagrams. Our experience has shown that complex models do not need to be used in their entirety to support the reasoning process, for most analysis questions.

Our work proposes systematic, break-down mechanisms for complex i* diagrams based on the analysis questions that the modeller is trying to answer. We introduce the model slicing and levelling concepts based on the i* Strategic Rationale diagram topology and ideas originating in aspect oriented programming, program slicing and strategic relevance research.

1.2 Related work

Scalability is a major concern in research fields such as distributed networks, databases, parallel computing, multi-agent systems etc. where precise measurements of various system properties allow metrics and unambiguous definitions for scalability to be devised [20, 21, 22].

However for conceptual modelling, such precise metrics do not exist and therefore an unambiguous definition of scalability cannot be devised. Scalability is most generally defined as: *"how well a solution to a problem works when the size of the problem increases"* [23].

Bearing in mind the general definition, we will attempt to identify the modelling and reasoning challenges that arise when the domain complexity is high and devise solutions for addressing them.

Although there is large number of modelling frameworks available, the topic of scalability is rarely addressed for most of them. The main reason for this is the lack of a complex real-life project to validate the fact that they scale up gracefully from a "toy example" model to a highly complex organizational setting. This explains partially the industry's reluctance to adopt such frameworks into practice and the considerable gap between the research "state of the art" and the industry's "status quo".

The importance of this issue has started being recognized in the modelling research community with the development of a complex exemplar [19] which can serve as a real-life sized validation project for the various modelling frameworks. Such an exemplar is extremely useful because it will allow the opportunity for the community to highlight strengths and weaknesses of various frameworks on a complex setting and will offer fair comparison grounds between various modelling alternatives by separating the solution approaches from the problem.

Many conceptual modelling notations have opted for simplified semantics and enforce tree structures for their resulting diagrams. In KAOS [27] and EKD [28] the high-level goals are refined through and/or decompositions to low-level operations which are assigned to agents. Similarly NFR [49], which has now been incorporated into i*, gradually refines non-functional requirements, represented through softgoals, to well known low-level operationalizations.

Tree structures are convenient for addressing scalability as they allow branch collapsing and expanding at every level of refinement. Even so, the authors of the NFR framework have reported scalability problems when applying the methodology to complex case studies [29].

However in order to capture the social dimension of agent interactions, which covers autonomy, intentionality and sociality, i* has adopted the more generous network-graph diagram structure and richer semantics to allow goal refinement and agent relationships to be represented in a single diagram. Based on their focus KAOS and EKD are said to be more goal oriented than agent oriented [8], while i* goes the farthest in addressing agent modelling issues.

KAOS and EKD offer multiple levels of abstraction, focusing on specific perspectives, and each grouping a set of closely related meta-concepts. Empirical studies have shown contradicting results over the usefulness of the division of models into diagrams representing various levels of abstraction.

A study conducted by Moody [15] indicated that Entity-Relationship models divided into several levels of abstractions (i.e. Levelled Data Models) increased the time required for model comprehension compared to the single levelled classical ER models. However a different study [30] has compared Levelled Data Models to the Clustered Entity Models [11] and Structured Data Models [31] which represent the two predominant paradigms for clustering models: aggregation and generalization. The results indicated that Levelled Data Models were perceived superior in terms of documentation efficiency, clustering consistency and ease of use.

Further studies are required to corroborate either of these findings. However, neither study mentions the effort required to build the different types of models which can represent an important comparison factor as well.

Models serve as means of capturing the domain knowledge in all its complexity in an abstract form and should allow complex reasoning and analysis to be performed on them with a reasonable effort/gain ratio. The intrinsic complexity of the application domain is the driving factor of model complexity.

Modelling notations which are rich in representational constructs such as i* are more prone to scalability problems than others, but their extensive power of expression is needed in order to represent faithfully and accurately the information gathered in the initial elicitation phase.

The complex of nature of today's organizations, the intricacies of client relations with internal and external entities, along with the multifaceted nature of requirements often allows multiple perspectives on the same issue to exist. In such situation one must be aware of the particularities of each viewpoint [24]. Surfacing the inconsistencies between viewpoints allows the analyst to gain a richer understanding of the application domain [25] and to focus its attention on so called "problem areas". By negotiating with the stakeholders over conflicting requirements [26], the analyst can work towards achieving a compromise, thus reducing the risk of products side-effects that are deemed unacceptable by either of the parties.

By encapsulating the rationale of each intentional actor in an individual entity, i* allows multiple and even conflicting viewpoints to coexist in a diagram, enabling extensive reasoning on conflict consequences and mitigation strategies to be performed.

Scalability challenges of i* models have been addressed in [32] through the creation of a set of views based on selection rules formulated in the conceptual modeling language Telos [33].

Most modelling scalability studies however, do not distinguish between the various stages of a model's lifecycle and their associated challenges. Our experience has shown that complex domains raise scalability challenges during:

- The model building phase
- The model validation phase involving stakeholders
- The reasoning phase based on the complete and validated model

The challenges encountered in each of these phases are different and as such require different mitigation strategies. Our work is focused on addressing the scalability challenges encountered in the reasoning phase, while hoping to provide interesting insights and ideas to address those encountered in the first two phases.

1.3 Strategic Requirements Analysis for Kids Help Phone Project

The purpose of the “Strategic Requirements Analysis for Kids Help Phone” project was to analyze how well Internet-based communication channels can be adapted to counselling requirements and to assess the impact of online service deployment as integral part of KHP’s service on various stakeholders from within and outside the organizational context.
The rich organizational context offered us a real-life complex case study, which has enabled us to assess and improve the scalability of the i* modelling framework. All the concepts presented in this work have been developed and used during the course of this project.

1.3.1 Kids Help Phone

Kids Help Phone (KHP) defines itself as “Canada's only national, bilingual, 24-hour, toll-free confidential and anonymous phone and web counselling, referral and information service for children and youth” [34].Its mandate is to provide professional counselling and information resources for kids in need across Canada. Kids contact the organization on issues ranging from severe problems (suicidal tendencies, physical abuse, drug problems, bullying) to relatively minor issues (relationship problems, age related issues, information).

The company maintains a toll-free, 24-hour phone line staffed by professional counsellors and a web bulletin board as the main contact avenues between itself and kids in need of help.
Kids Help Phone was among the pioneers in launching phone-based counselling, in a time when most traditional counsellors thought that the lack of visual cues made the phone an inadequate medium for counselling.

Given the wide range of issues that kids contact the company about, properties such as anonymity, confidentiality and safety of service usage became absolute requirements for all the services that the company offers. The company does not trace calls, does not use call display and does not record any identifying information from its phone users. Furthermore Internet posts that contain identifying information are edited are edited by counsellors before being published for the wider audience.

The phone line is toll-free and staffed by bilingual counsellors that have professional degrees and at least 5 years of experience. By using the phone, kids are able to have support for their problems within minutes. By communicating their problems and venting their emotions kids are pulled out of isolation and through counselling encouraged to seek further help for their problems. The counselling approached used is a non-intrusive and friendly "resolution focus" approach, through which kids are helped in exploring their alternatives and making their own choices.

If kids require further assistance at the end of the call they are directed to local resources in their own community or in extreme situations, such as reports of abuse, the local authorities are notified, if sufficient identifying information has been provided.

Over the past decade the Internet has revolutionized communication, opening numerous communication channels that need to be considered by companies such as KHP. The young generation, which is the company's target audience, is web savvy and very comfortable with communicating over the Internet using VOIP, text messaging, emails or through chat rooms.

Many of today's kids prefer the Internet as means of communication to the phone. Kids who are facing very difficult situations tend to isolate themselves and are sometimes reluctant to talk on the phone openly about their problems and would rather resort to communication channels that give them more privacy.

In light of this, KHP has recognized that it must also change in order to remain relevant and accessible to kids on their terms. The company has taken an initial step by providing the Ask a Counsellor section on their website, which is a bulletin board where kids can post questions and read answers to their own questions, as well as answers to questions posted by other kids.

Currently the service is asynchronous and the delay between the initial post and the reply is usually between 24 and 48 hours. In that period the initial post is reviewed and edited, if inappropriate content is found, a counsellor writes up an answer which is reviewed by a supervisor and then posted on the bulletin board. The internal process of post editing and reviews along with the current layout of the service and the post-reply delay are sufficient to guarantee kids the safety of service usage and anonymity.

However KHP has identified the immediacy of service as a very important requirement for kids and the need for context information and an emotional connection with the kids as a very important requirement for counsellors.

With the post-reply counselling methodology, counsellors sometimes receive too little information to construct a mental picture of what the kid is going through. While on the phone counsellor can ask questions to clarify any uncertainties, the current web setup forces the counsellors to make assumptions about what they think the problem might be. Text based counselling also deprives the counsellors of auditory cues and makes the establishment of an emotional connection harder.

Given the kids' need for immediacy, their preference for Internet communication channels and the counsellors' need to establish a dialogue and to achieve an emotional connection with kids, the solution seems to lie in the real-time online services realm.

Moving towards real-time online services is a big challenge for the company given its limited resources and the challenges that Internet services entail. Given the sensitivity of the issues at stake a careful analysis needs to be conducted in order to determine which variant of online service best serves the needs of counsellors and kids and is financially sustainable by the company.

1.3.2 Research approach

From a research perspective the project is multifaceted as it allowed us to develop a standardized methodology for analyzing new services, it provided us with a large-scale case study for our modelling methodologies and enabled us to combine two very popular requirements engineering methodologies: the goal oriented modelling approach represented by the i* modelling framework [8] and the viewpoints methodology [24].

The information required for conducting the analysis was collected through a series of interviews with stakeholders from all levels of the organizational context. The aim of the initial phase of the project was to get a complete picture of the organizational setting, to understand its internal and external relationships and to capture the knowledge acquired inside i* models, which would allow for thorough analysis of the impact of introducing new technologies.

I* modelling had been used previously for requirements engineering, business process reengineering, organizational impact analysis and software process modelling and its focus and semantics recommended it as the appropriate tool for conducting the analysis.

Given the sheer volume of information, the complexity of the organizational context and the sensitivity of the issues at stake, the resulting models had a high intrinsic complexity.
In order to conduct the analysis efficiently on the resulting models, a series of concepts were developed during the project which represent the focus of this work.

2. Model life-cycle scalability challenges

Requirements engineering models have a lifecycle of their own. Models go through an initial development phase, are validated based on the stakeholder input and finally used for reasoning purposes by the people conducting the analysis.

Most research projects that address modelling scalability usually treat complexity challenges as an undivided problem. Most proposed solutions address the end-model's complexity issues while the challenges of model building and model validation by the stakeholders are almost never mentioned.

Our experience has shown that complex domains raise modelling scalability challenges during:

- The model building phase
- The model validation phase involving stakeholders
- The reasoning phase based on the complete and validated model

The challenges encountered in each of these phases are different and as such require different mitigation strategies. The process of model development is not linear and all the above mentioned phases are repeated a number of times until satisfactory model quality has been achieved.

We will address briefly the i* modelling challenges encountered in the first two phases in the following sections and we will elaborate on the challenges of the third phase in the remainder of this thesis.

2.1 Complex model development challenges and mitigation strategies

Among the modelling frameworks available for Requirements Engineering analysis, i* goes the farthest in addressing agent and goal modelling issues. Its construct richness enables it to capture very complex organizational settings and abstract them in a graphical form.

However as we have argued before, model complexity is a direct function of the underlying domain complexity and modelling notations rich in representational constructs are more prone to scalability problems. When models reach a high level of complexity, applying the standard model development practices becomes increasingly difficult. Without taking additional steps, the modelling activity becomes error-prone, the model development time increases and consequently the resulting overall model quality decreases significantly.

In order to tackle these development challenges the modeller needs to have good modelling tool support available and should resort to a methodology that enables intuitive diagram breakdown. We will briefly sketch the modelling methodology used during the Strategic Requirements Analysis for Kids Help Phone project and some of the tool support requirements we have identified.

2.1.1 Addressing development challenges through methodology

Some of the most powerful defences that modellers have in addressing model development scalability challenges are complexity awareness and planning.

Realizing early on that models will have a high intrinsic complexity, will allow a modeller to adopt a diagram segmentation strategy that will reduce the challenges faced in the model development phase. In the case of the Strategic Requirements Analysis for Kids Help Phone project this issue was evident given the sheer volume of information collected and the complexity of the organizational setting.

Some modelling techniques propose various level of abstraction for model creation. KAOS [27] and EKD [28] have goal centered models to address stakeholder intentions, process models to address dynamic issues and agent models to address agent responsibilities. Such views, while adequate for presentation, do not necessarily facilitate the analysis and model development phases. When thinking about an organizational issue, modellers have difficulties in mentally switching between artificially enforced levels of abstraction. I*'s approach favours the encapsulation of multiple such levels into a single diagram. This approach results in a reduced cognitive load for the modeller and a more streamlined model development process. However, in order to address the complex model development challenges, the modeller needs to come up with his own segmentation strategy.

We argue that model segmentation based on the properties of the domain is more intuitive than a segmentation based on the modelling language's enforced levels of abstraction.

Particularities of the domain often dictate what represents a good modelling unit. Such units can represent an individual stakeholder, a department within the organization or organizations as a whole. According to [35] a good model decomposition should be:

- Complete
- Non-Redundant
- Self Contained
- Unified
- Cognitively Manageable
- Flexible
- Balanced
- Loosely Coupled
- Highly Cohesive

There are a large number of alternative decompositions that can be produced for a particular problem [36]. By applying a decomposition that is inspired by the natural groupings of the application domain and applying the above principles as selection criteria, when multiple such decompositions exist, the modeller should be able to successfully overcome most of the development challenges that complex models raise.

In order to successfully construct complex diagrams, the modeller should have a solid tool support package to aid him in the development phase.

2.1.2 Addressing development challenges through tool support

The official tool support of the i* modelling framework is the Organization Modelling Environment (OME) tool [37]. OME is a modeling tool, "meant to aid in the development and representation of models developed in various modeling frameworks. It provides a graphical view of the models, which are themselves stored in a knowledge base". [38]

OME was built as an extendable modelling environment capable to supporting multiple modeling frameworks. At the time this document was written OME fully supported the i* and NFR frameworks.

Given the fact that OME was developed as a prototype research tool, it did not posses the optimization and scalability features that commercial drawing packages have. Our previous experience with it showed that it was not optimized and stable enough to support the development of large scale models. Thus we regarded the Strategic Requirements Analysis for Kids Help Phone project as an opportunity to collect the requirements for the next generation of tool support for the i* modelling package.

We resorted to Microsoft Office Visio 2003[39] a well known, commercially available general drawing package as the modelling tool for i* modelling. For this purpose we developed a stencil containing all the representational constructs of the i* modelling framework.

Besides the available features of general drawing packages, we have noticed the need for several scalability support features, which are specific to conceptual model development.

We have argued earlier that diagram decomposition enables the modeller to address the model development challenges to a certain extent. However in order to conduct a comprehensive reasoning of all the domain implications, the sub-models need to be merged into a single comprehensive domain model. By applying the model decomposition guidelines listed in Section 2.1.1 the resulting sub-models to be readily mergeable and the merging process should be easily automatable. Thus automated merging support should be available in the next tool support package for i* modelling.

When conflicts or inconsistencies are present in the sub-models, the tool should assist the resolution of those conflicts with computer-supported negotiation [40, 41].
Given the fact that domain information is scattered in a number of diagrams, when applying the model decomposition methodology, tool support features such as a query facility and model change management mechanisms would also be necessary. Without such features, the task of maintaining consistency between the various models falls solely on the modeller leaving room for human error.

When conducting reasoning on the comprehensive merged model, the analyst can employ i*'s qualitative evaluation algorithm to assess the impact of various possible design alternatives on the goals of the organizational stakeholders. The qualitative evaluation algorithm could also use adequate tool support in the propagation of labels across intentional links, which can only be partially automated since resolving conflicts, tradeoffs and making compromises requires human input.

We have listed here a few tool support requirements that would mitigate some of the scalability challenges of complex model building. This list is by no means complete and further research will be required until such time as a complete requirements specification for modelling tool support is devised.

2.2 Complex model validation challenges and mitigation strategies

As we have seen in the previous sections, i* models abstracting complex organizational settings greatly exceed the “cognitive bandwidth”[42] size defined by the Cognitive Load Theory [16,17] and empirically validated for conceptual models in [15].

The understandability of conceptual models is a critical issue in IS development [15]. Poor understanding of conceptual models is deemed as one of the main causes for which stakeholders and end-users fail to accurately verify requirements [43]. Empirical studies have shown that more than 50% of the errors occurring in the systems development phase are the result of inaccurate or incomplete requirements [44].

The large number of delivered systems which do not meet user requirements suggests that stakeholders and end users have significant difficulties in understanding and validating conceptual models [45, 46]. These findings suggest that conceptual models used for reasoning and analysis by expert modellers are not adequate means for validating and communicating requirements to end users.

Although the application domain can be extremely complex, the number of issues or requirements that require stakeholder validation is relatively small since in the majority of cases, 20% of a project's features will provide 80% of user benefits [46].

Since complete model validation, although desirable, is not feasible in most of the cases, we should turn to intermediary means of representation for the critical requirements that require stakeholder validation.

In the Strategic Requirements Analysis for Kids Help Phone project we validated the critical requirements using simplified models focused on a small set of concerns and textual lists of issues.
Our experience with both approaches has been very positive and the input of the stakeholders has been invaluable in enhancing our understanding of the issues important for the company.
Through our practical experience we have found that small simplified models, or textual representations stimulate stakeholder involvement in requirements validation, are more appealing and less intimidating than comprehensive models developed for reasoning. This line of study appears very promising and further investigation of these approaches is needed before a coherent set of guidelines for model validation can be developed.

2.3 Addressing scalability challenges in the reasoning phase

2.3.1 Discussion

We have seen from our previous experience that models abstracting real-life complex environments tend to become overwhelmingly complex. While the difficulties encountered in the model development process can be addressed to some extent by different modelling alternatives along with adequate scalability tool support, the complexity of the resulting models requires further analysis and new concepts to be introduced to enable efficient reasoning on them.

The complete model's size depends primarily on the complexity of the domain which is invariant regardless of the methodology adopted for the model development. Creating sub-models, representing various aspects of the organizational context, reduces the model building difficulties, but at the end of the model development phase all the resulting sub-models should be merged into a single model to facilitate a comprehensive, domain-wide analysis of dependencies and design choice effects. As argued in section 2.1.2 the merging operation can be fully automated, if basic modelling guidelines are followed.

The high density of contribution links inside rationale models and the high number of dependencies between actors, make reasoning on complex models without a viable analysis strategy very difficult. In order to be able to perform the analysis accurately and effectively, the modeller needs to have available ways of simplifying the diagram based on the kinds of questions that are driving the analysis.

A comparison between a model abstracting a real-life situation and a small "toy example" model [7] is presented in Table 2.3-1 to illustrate the difference in complexity between the two.

	KHP Services diagram	Meeting scheduler diagram
Actors	9	3
Softgoals	185	7
Goals	24	5
Tasks	130	15
Resources	9	2
Help links	109	7
Hurt links	36	4
Make links	12	0
Break links	7	0
Means-Ends links	45	5
Decomposition links	54	12
Dependency Links	320	12

Table 2.3-1 Complexity comparison between two service diagrams

We have included this comparison in order to illustrate the extreme difference in complexity between a complex real-life service analysis and a simplistic, manageable diagram. Both diagrams are service related, but given the domain complexity and the sensitivity of the issues at hand, the KHP Services one is considerably more complex. The KHP services diagram that we represented in Table 3-1 represents only one of the 14 diagrams we created in our analysis.

We will illustrate in the following chapters, the fact that although i* models can reach a high level of complexity, which is needed in order to faithfully represent the information gathered in the initial elicitation phase, the analysis does not have to be performed on the big models as a whole. Instead the analysis should be guided by elicitation and design questions and their focus should translate in a need to operate only on segments of the model rather then the whole.

The model-slicing concepts, introduced in Chapters 3 and 4, map perfectly to several analysis questions and they represent powerful tools in dealing with the scalability problems raised by very complex models. We will explore some analysis questions that arise in the early requirements engineering process and illustrate how our concepts can synthesize the model content relevant to those questions, while filtering the rest of the information.

The modelling activity is usually driven by several purposes:

- It exposes misunderstandings or things that are insufficiently explored that arise as pertinent questions during the modelling activity
- The model can help expose conflicting or unfeasible requirements
- The model provides a convenient, cost-effective way of reasoning on the effects of systems before actually building them.

The first point illustrates the gains of the model building activity, while the others illustrate those of the subsequent reasoning process. In order to make the reasoning process easier and cost-effective, we need to define mechanisms of breaking down the complex diagrams into more manageable pieces.

The types of questions that arise in the early stages of requirements engineering analysis are usually WHY, HOW and WHAT questions. Keeping in mind those types of questions, we will define concepts relevant to the reasoning and analysis process that are meant at addressing the scalability problems we have identified.

Before presenting our concepts we begin by presenting some preliminary concepts that our approach draws on.

2.3.2 Model slicing

The high intrinsic complexity of diagrams such as the one presented in Table 2.3-1 makes reasoning on them increasingly difficult. When the complexity passes the comprehensibility threshold, reasoning on the model as a whole becomes virtually impossible and therefore the analysis needs to be guided by a set of questions relevant to the problem at hand.

Modelling scalability issues have been addressed in the past by [30]. This work created a framework for creating presentation views and proposed several single actor simplifications. Our work aims to discover and present several such views that we have found useful both from scalability and analysis perspectives.

We will define concepts that apply to the comprehensive strategic rationale model that includes all the intentional actors of the modelled domain, since we are interested in analysing the effects of operationalizations across the whole spectrum of the domain.

The term and the idea of slicing come from program slicing. Program slicing is a fundamental operation that can aid in solving many software-engineering problems.

The slice of a program with respect to a set of program elements S is a projection of the program that includes only program elements that might affect (either directly or transitively) the values of the variables used at members of S. Slicing allows one to find semantically meaningful decompositions of programs, where the decompositions consist of elements that are not textually contiguous.

Program slicing has applications to program understanding, maintenance, debugging, testing, differencing, specialization, reuse, and merging [47].

While in the case of programs, the process of determining a semantically meaningful slice requires a very complex analysis, conceptual modelling diagrams allow for a much easier implementation of the concept. Related intentional elements are naturally linked by contributions, correlations or dependency links thus making the slicing process easier to implement.

The concept of program slicing has a direct mapping to the concept of top-down model slicing, in which we will create a projection of the model considering only the elements that are contributing (either directly or transitively) towards the element of interest.

Conversely, the bottom-up model slicing projection traces the contributions that an element makes (either directly or transitively) towards the higher-level intentional elements of the domain actors. We will illustrate in the following chapters the usefulness of the slicing concepts from the analysis and scalability perspectives and present their applicability in the KHP project.

3 Bottom-up model slicing

In this chapter we present the bottom-up slicing concept and illustrate its usefulness both from an analysis and scalability perspective. We illustrate the bottom-up slicing methodology with an example based on the Strategic Requirements Analysis for Kids Help Phone study and summarize our findings.

3.1 Analysis justification

Often when conducting early Requirements Engineering analysis, analysts are faced with many design alternatives that need to be considered during the requirements elaboration process.

In order for the reasoning process to succeed, the effects of the considered approaches have to be traced through the entire spectrum of the considered domain. which entails a high analysis overhead.

In a typical i*organizational context diagram, client goals are usually represented by high level intentional elements such as goal and softgoals. Conversely, new services or design alternatives represent low level operationalizations whose effects have to be thoroughly analyzed before a design decision is made.

The bottom-up model slice will start from a low-level operationalization and trace its effects all they way to the affected high-level client goals. The elements that are not affected by the operationalization will be filtered out. The resulting view is considerably simpler and allows an easier application of the evaluation algorithm. When the design space has a high number of alternatives, the modeller can perform the reasoning based on model slices, which are considerably smaller, and reduce the design space to a manageable size by eliminating unsatisfactory alternatives.

The alternative goal refinement analysis technique, advocated by [48] as an appropriate tool for exploring alternative system proposals, provides the right level of abstraction for involving decision makers in validating and completing the alternative pool.

Performing a comparative study of various design alternatives is useful because it will yield interesting insights regarding the strengths and deficiencies of each approach. Because of their weaknesses some alternatives may be ruled out from the beginning, while other approaches, although significantly better, may need to be supplemented to successfully address the problem. Instead of implementing a single alternative, analysis may reveal that several complimentary alternatives are needed in order to successfully address the problem.

While the type of solutions and their corresponding design alternatives may vary dramatically in different domains, the underlying idea of the comparative analysis is applicable in all of them.

Since a goal oriented modelling technique, such as i*, offers an excellent medium for capturing the available domain information in a graphic form, the comparative analysis of design alternatives can and should use the resulting models as a basis for conducting the reasoning. However, when models reach a high level of complexity, using the entire model for the reasoning becomes neither feasible nor needed.

The bottom-up model slicing concept offers a very useful abstraction mechanism that can be used to help answer several pertinent questions that arise in the analysis process, by providing a method of filtering out elements that are not affected by a certain design decision. In order to achieve a manageable level of complexity for the models we will have to sacrifice completeness to a certain extent. However, as we will show in the following sections, comparative analysis does not require complete models is order to produce accurate results.

We illustrate in Table 3.1-1 the mapping between several analysis questions and concepts derived from the bottom-up model slicing concept.

Analysis question	Associated concept
1. WHAT are the all the objectives from the domain spectrum affected by certain design alternative?	1. Bottom-up model slice having the desired alternative as starting point
2. HOW does the chosen alternative affect the satisfaction of the specified objectives of the client?	2. Bottom-up model slice with evaluation
3. WHY does an alternative affect a certain need/desire?	3. A path through the model slice from the operationalization of the alternative to the element of interest

Table 3.1-1 Correspondence between comparative analysis questions and presented concepts

We will analyze each question into more detail and illustrate how the bottom-up model slicing concept can be used by the modeller as basis for the reasoning process related with that particular question.

3.1.1 Analysis question 1 – The WHAT question

This question is one that arises early in the reasoning process of a particular design alternative. By asking the question as phrased in Table 3.1-1, the analyst is interested in the scope of this choice from the domain spectrum.

Based on the scope of the alternative and of the relevance of the affected elements, the designer can prioritize the alternatives accordingly. Localized alternatives that affect a smaller number or less relevant factors may be considered lower priority than others.
We will conduct further analysis on the scope of design alternatives in Chapter 5.

When the domain complexity is high, the diagrams abstracting it become very complex. In order to conduct a comprehensive analysis the modeller is faced with the very difficult task of reasoning about possible alternatives and tracing their effects across a very complex diagram.

Getting a fairly accurate idea of the scope of a design choice, on a diagram containing several alternatives (i.e. >3 alternatives) and having an intrinsic complexity comparable to the ones we created for the KHP project, is almost impossible.

However, applying the bottom-up slicing methodology, which can be fully automated, can help answer this early requirements engineering question and thus aid the analyst in his effort.

The bottom-up slicing process can start from any element ranging from a low level operationalization to a mid level intentional element, depending on the type of analysis required and the particularities of the domain.

We have illustrated, among the side effects of the increase in complexity, the fact that errors can occur in the resulting diagrams. Errors by omission are also frequently encountered in large diagrams. By singling out a design alternative and reducing the size and complexity of the diagram such errors become more obvious and are easily corrected.

3.1.2 Analysis question 2 – The HOW question

The second question illustrated in Table 3.1-1 arises from the need to assess the effects of several design alternatives on the various client needs and objectives. An answer to this question provides an excellent basis for alternative comparisons and can be obtained by performing a qualitative evaluation of the effects of those alternatives on the client goals.
The i* modelling framework possesses a qualitative reasoning algorithm that was adapted from the NFR framework [49].

The evaluation algorithm allows the user to obtain qualitative results for the high level intentional elements, by assigning labels to low level intentional elements (leaf nodes) and tracing the effects of those elements through the propagation of their corresponding labels.
Chung and others describe the NFR evaluation algorithm as "useful in selecting among alternatives. In the presence of competing alternatives, developers can use the procedure to see what impact a particular selection has on the satisficing of their softgoals"[49].

The i* evaluation algorithm is still an active research topic. Our aim is not to pursue evaluation research questions but rather to give a brief overview of the "status quo" of the algorithm and to present the elements used by our concepts.
We illustrate in Table 3.1.2-1 the labels used by i*'s evaluation procedure, along with their corresponding definitions. The definitions have been collected and adapted from [49] and [55].

I* Evaluation Label	Symbol	Definition
Satisficed	✓	**Satisficed:** An element that is considered sufficiently satisfied.
Denied	✗	**Denied:** An element with unsatisfactory achievement.
Weakly satisficed	✓•	**Weakly Satisficed:** An element with inconclusive positive support for his ancestors.
Weakly denied	✗•	**Weakly Denied:** An element with inconclusive negative support for his ancestors.
Conflict	↯	**Conflict:** An element that can be satisficeable and deniable. (Receives contradictory contributions from descendents).
Undecided	•?	**Undetermined:** An element that is neither satisficeable nor deniable.

Table 3.1.2-1 The evaluation labels of i*'s qualitative reasoning algorithm

The evaluation algorithm proceeds in a bottom-up fashion by labelling leaf nodes and propagating labels upwards. Leaf nodes representing operationalizations are labelled accordingly by the modeller.

Based on the labels and contribution types, their effects propagate towards the high level goals. The algorithm requires human judgement on deciding whether elements receiving contradictory contributions can be considered Satisficed, Denied or anything in between. Human judgement is also required in assessing the relative importance of contributing elements. We will provide further details related to label propagation in Chapter 5. Since the evaluation algorithm requires both human judgment and thorough knowledge of the domain, it cannot be fully automated.

The i* evaluation algorithm provides a convenient way of reasoning using a standardized approach between alternatives. This approach confers credibility to the comparison process and minimizes modeller cognitive bias in the analysis.

In the i* modelling framework, goals – defined as decision points by Yu and others [50], represent a way of indicating choice among alternatives. The underlying tasks that operationalize those goals can be viewed as design choices which are considered for implementation. This allows the modeller to embed a space of design alternatives of considerable size into a single i* diagram.

When dealing with very large Strategic Rationale diagrams, the ability to reason on the diagram and to perform an effective analysis is negatively affected by their intrinsic complexity. Finding the relevant elements and tracing their contributions in order to apply the evaluation algorithm is very difficult and error-prone.

By applying the bottom-up slicing algorithm on the comprehensive diagram the analyst is provided with a greatly simplified version of the diagram including only the elements relevant to the design choice he is interested in. Subsequently the evaluation algorithm can be applied on the model slice which is considerably smaller.

The evaluation algorithm was conceived as a way of conducting a qualitative analysis on a complete model. Its results that can indicate the achievement or denial of client objectives based on the selected design alternatives. However by performing a slicing procedure on the model we are sacrificing its completeness in order to reduce it to more manageable size. The elements that are filtered out by the slicing algorithm are not affected by the considered design alternative and as a result their satisfaction can be considered constant with regards to the choice criteria. However, they can have a significant impact on the satisfaction of client objectives and should not be omitted in later analysis phases.

Thus the results of the evaluation algorithm on a model slice cannot be considered absolute or complete. They can be viewed as partial results, indicating the effects of various design alternatives on the client goals. By comparing the effects of various design alternatives the analyst can reduce the design space considerably and decide which alternatives warrant further analysis. We will illustrate in

Chapter 4 the top-down slicing concept, which focuses the analysis of design alternatives on one objective at a time but preserves the model completeness with respect to that objective, as a way to conduct a more comprehensive analysis on the remaining design alternatives.

3.1.3 Analysis question 3 – The WHY question

The third question illustrated in Table 3.1-1 is another fundamental question that arises in the early Requirements Engineering process. The importance of understanding the WHY questions in improving or redesigning a process is highlighted in [51]. WHY questions are important not only in the elicitation stages of the Requirements Engineering process but also in the subsequent analysis and design phases.

The question, as stated in Table 3.1-1, is concerned with tracing the contribution between an operationalization and a high level goal. Sometimes the contributions between intentional elements are not obvious or are very indirect requiring further modeller validation.

When being presented with a bottom-up model slice the modeller might notice intentional elements whose relation with the chosen design alternative is not obvious. This could either reveal a non-obvious property of the domain, be caused by an error in the model, or it could indicate that further simplifications by abstraction can be performed on the model. We will analyze the last alternative into more detail in a Chapter 5.

In order to assess which of three above described situations is occurring, the modeller can create a path view of the bottom-up model slice linking the two intentional elements. The path view provides further abstraction, by showing only the contribution chain that links the design alternative to the intentional element of interest. If the contribution is considered to be correct, this would indicate that a less obvious property of the domain has been revealed by the analysis. In the other cases the model might have to be corrected or further simplifications could be applied.

We have noticed in our practical experience that errors in the models become more obvious in the model slices and thus their usage allows the analyst to obtain higher quality models.

Once again the bottom-up model slice proves to be a useful concept, helping the modeller clarify his uncertainties of the domain as well as improving the quality of the underlying models. Given its high complexity reduction potential and wide analysis usages, the bottom-up slicing concept appears as a powerful analysis and scalability tool.

3.2 Bottom-up model slicing methodology

3.2.1 Ancestor vs. Descendent discussion

In order to present the bottom-up slicing methodology we will have to first define some preliminary concepts.

Actors are linked through dependencies in i*. Actors are intentional but in most cases they cannot achieve all their goals independently. By depending on others for services and resources, actors create a collaborative network where each of them is trying to fulfill its own goals. On the other hand dependencies can be viewed as vulnerabilities, where a failure to provide a service or resource by an actor can have negative consequences on several actors that depend on it.

Figure 3.2.1-1 illustrates the dependency link ontology of the i* modelling framework.

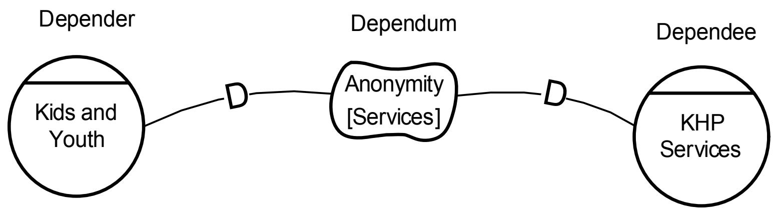


Figure 3.2.1-1 The i* dependency ontology

According to [53] an agent/actor that depends on another agent/actor for something (*dependum*) is called *depender*. Conversely, the agent/actor being depended upon is called *dependee*. We will propagate the effects of the originating model slice element through dependencies, in order to asses the consequences of that particular alternative an all the actors from the domain spectrum that are affected by it.

Two other very important concepts for our slicing methodologies are the ancestor and descendent concepts.

The contribution links in i* are directed, each having a source intentional element and a destination intentional element. As defined in [32] we will consider the source intentional element of a contribution link to be a direct descendent of the destination intentional element. Conversely, the destination intentional element of a contribution link is an ancestor of the source intentional element.

We will illustrate the two concepts with an example containing various types of links that are defined in the i* modelling notation.

The diagram in Figure 3.2.1-2 presents a simplified view of the Counselling Information Provider role from the counselling model developed for the KHP project.

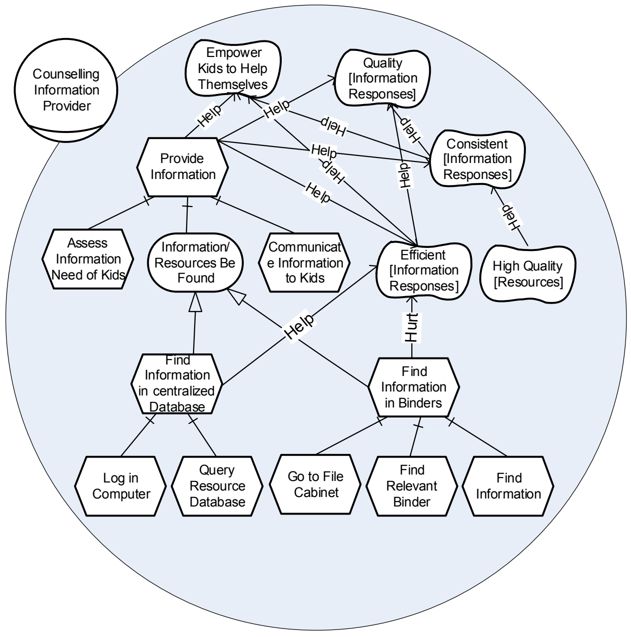

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Consistent [Information Responses]
Provide Information
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
Help
Hurt

Figure 3.2.1-2 Counselling Information Provider Role illustrating descendent – ancestor relationships

Counsellors have to provide various kinds of information to the kids who are calling like: legal information such as legal age to smoke, get married etc , drug information or contact information for local specialized resources such non profit agencies, shelters etc.

One of the goals of the counsellors and of the organization is to empower kids to help themselves by providing them with information relevant for their needs.

Given the diversity of information kids require and the fact that KHP is a national organization, counsellors currently have to use various information sources such as information binders, the Internet and a small electronic library created by the company.

In the Counselling Information Provider Role, the *Find Information in centralized database* and *Find Information in Binders* tasks are direct descendents of the *Information/Resources be Found* goal according to the descendent definition mentioned above. Also, conversely we can say that the *Information/Resources be Found* goal is a transitive ancestor of the *Log in Computer*, *Find Information* and the rest of the bottom-level tasks.

The high level softgoals *Empower Kids to Help Themselves* and *Quality [Information Responses]* are the highest level ancestors, for this intentional actor, of the above mentioned bottom level tasks.

The bottom-up model slicing methodology will trace all the ancestors, direct or transitive, for a certain intentional element.

However, since we are interested in the effects of the chosen intentional elements across the whole spectrum of the domain we will not limit the slicing algorithm to a single intentional actor. A domain wide analysis is important, because usually design decisions affect several intentional actors and their goals.

Our algorithm will find all the ancestors of an intentional element belonging to the source actor, trace the ancestors across dependencies (from dependee to depender) to other intentional actors and then continue transitively the ancestor finding process in those actors. The result is a domain view containing all the elements affected by the design alternative, which can be used for the kinds of analysis illustrated in Chapter 3.1.

3.2.2 Algorithm implementation

Unlike other goal oriented modelling methodologies such as KAOS [54], or NFR[49] that enforce tree-like structures for goal refinement diagrams, i* models have a more complex network graph-like structure. This can be attributed to the richer semantics of i* and to its focus on intentionality analysis.

The i* concepts, such as actors, intentional elements and links are represented internally in the Organization Modelling Environment (OME) tool using the Telos conceptual modeling language.
The OME tool represents the official modelling support of the i* modelling framework. However given its scalability and reliability problems we have resorted to the use of a commercial drawing tool for the construction of our models.

Since the representational constructs of the future i* modelling tool are yet to be determined we will use a simplified, class based description of i* concepts. Our aim is not to redefine i*'s concepts or to propose a different meta-model but rather to provide a simplified, more neutral, graph-based representation of i* concepts that allows us to easily implement our algorithms.
Given the fact that the structure of i* diagrams is very similar to that of network graphs with a direct correspondence between intentional elements (i.e. softgoals, goals, tasks, resources) and the node concept in graphs and between intentional links and graph edges we will use a concept representation that accommodates those similarities and allows for a simple implementation of our algorithms.
Since i* diagrams frequently present circularities we will have to address them in our algorithms to prevent infinite loops.

We present the base classes corresponding to the graph node and edge concepts in Figures 3.2.2-1 and 3.2.2-2

```
/**The class representing i*'s intentional elements*/
class Node {
      //The actor to which the intentional element belongs to
      Actor parent;
      //Type of the intentional element (i.e. softgoal, goal, etc)
      String type;
      //Element id (unique identifier)
      int id;
      //Vector of links TO the current shape
      Edge[] incoming_links;
      //Vector of links FROM the current shape
      Edge[] outgoing_links;
      //Layer of the shape – used for the model levelling concept
      int level;
}
```

Figure 3.2.2-1 The Node class representing i* intentional elements.

Nodes correspond to i* intentional elements such as softgoals, goals, tasks or resources. An intentional element in our representation has a parent, a type, a unique identifier, a level, and two arrays representing its incoming and outgoing links. This representation is well suited for applying our slicing algorithms.

An intentional link in our representation has a parent, a type, a unique identifier, a source node and a destination node. Using the node and edge concepts we defined, we will represent an SR i* diagram as an array of nodes interconnected by edges.

```
      /**The class representing i*'s intentional links*/
      class Edge {
            //The actor to which the intentional element belongs to
            Actor parent;
            //Type of the link (i.e. Dependency, Help, Hurt etc)
            String type;
            //Link id (unique identifier)
            int id;
            //Source intentional element identifier (i.e. descendent)
            int source_id;
            //Destination intentional element (i.e. ancestor)
            int destination_id;
}
```

Figure 3.2.2-2 The Edge class representing i* intentional links.

For simplicity we will not present the Actor class and its various specializations (i.e. role, agent, position, etc.) since the type of actor is not important for the slicing algorithm.

We will use a non-recursive breadth-first traversal of the i* model in order to generate a bottom-up slice. The algorithm implementation along with its main helper method are illustrated in Figure 3.2.2-3 and 3.2.2-4.

```
 /**Method that implements the bottom-up slicing algorithm
   *@param The slice starting element
**/
void create_bottom_up_slice(Node startElement){
Queue myQueue = new Queue();
myQueue.enqueue(startElement);
//remove first element in queue and add all his descendents
       while (!queue.isEmpty()) {
              Node temp = myQueue.dequeue();
              //mark element as visited;
              temp.setVisited();
              //add the direct ancestors and dependers of temp to the
       queue
              addAncestorsAndDependers(temp, myQueue);
       }

}
```

Figure 3.2.2-3 The bottom-up slicing algorithm

The slicing method uses an additional Queue structure to store the direct ancestors of the current element and to ensure a breadth first traversal. The method references two helper methods: setVisited() and addAncestorsAndDependers().

Our approach is to enqueue the slice originating element, find its direct ancestors and dependers, enqueue them and then continue transitively the ancestor and depender finding process. We have chosen a non recursive approach to implement the bottom-up slicing algorithm. Since i* diagrams often present circularities we will avoid infinite loops by flagging elements that have been visited or enqueued to be visited.

Since the setVisited() method is trivial we only will illustrate the implementation of the addAncestorsAndDependers() method which is presented in Figure 3.2.2-4

```
/** Adds the ancestors and dependers of the specified Node to the queue
*@param The current node
*@param The ancestor queue
**/
void addAncestorsAndDependers(Node temp, Vector queue) {
       //get the outgoing links for the temp Node
       Edge[] outgoing_links = temp.getOutgoing_Links();
       //find the destination node of each outgoing link
       for (int i = 0; i < outgoing_links.length; i++) {
              //if we haven't been down this path before
              if (! outgoing_links[i].wasVisited()) {
                     //mark link as visited
                     outgoing_links[i].setVisited();
                     //get the ID of the node at the destination end of current
link
                     int dest_id = outgoing_links[i].getDestination();
                     Node destination = getNode(diagram,dest_id);
                     //if destination node was not visited and is not in the
queue
                     if
((!destination.wasVisited())&&(!destination.wasEnqueued())){
                            //add the destination node to the queue
                            queue.enqueue(destination);
                            //mark the node as enqueued
                            destination.setEnqueued();
                     }
              }
       }
}
```

Figure 3.2.2-4 Method that enqueues direct ancestors and dependers

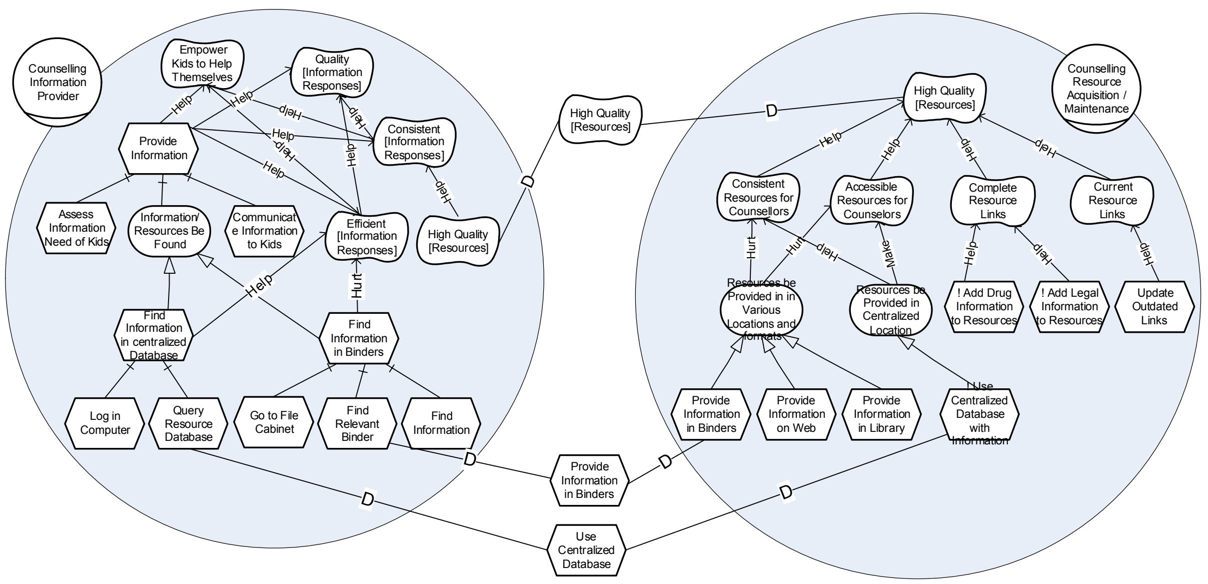
As we can see the bottom-up slicing concept has a straight forward implementation. This coupled with its wide analysis usages and complexity reducing potential makes it a very useful scalability concept.

3.3 Comparative analysis example using bottom-up slicing

We will illustrate the bottom-up model slicing algorithm with a simple example containing only two intentional actors. Although the bottom-up slicing concept is intended for diagrams far more complex, we cannot illustrate its application on our complex KHP diagrams given the presentability constraints of our current work. The example we are using is based on our KHP models and is sufficient to convey the gist of the bottom-up slicing concept.

The diagram presented in Figure 3.3-1 features the Counselling Resource Acquisition/Maintenance role besides the Counselling Information Provider role, whose internal rationale we discussed in section 3.2.1.

The diagram is a simplified version of the comprehensive KHP Strategic Rationale models of the two counselling roles. All dependencies to/from other intentional actors have been removed for simplicity.



Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Consistent [Information Responses]
Provide Information
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
Help
Hurt
D
High Quality [Resources]
Provide Information in Binders
Use Centralized Database
Counselling Resource Acquisition / Maintenance
High Quality [Resources]
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information
Make

Figure 3.3-1 SR model for the Counselling Information Provider and Counselling Resource Acquisition/maintenance roles

Since we have already described the rationale of the Counselling Information Provider role, we will focus on describing the rationale of the Counselling Resource Acquisition/ maintenance role in this section.

To be able to provide consistent and efficient information responses to the kids who call, counsellors need high quality resources available to them. According to the counsellors, the resources should be consistent, complete, current and accessible. The Counselling Resource Acquisition/Maintenance role encompasses the activities that counsellors have to perform in order to keep the resources in the above mentioned parameters.

In order to keep the resources complete, the counsellors would like to have drug and legal information added to the existing resources. Updating outdated information links helps keep the resources current, while providing them in a centralized location and standardized format helps keep the resources consistent and accessible. Currently, some information is stored electronically and some in information binders and in the library. The information stored in binders consists of printed sheets containing information from various sources: web pages, flyers, guides etc.

3.3.1 Creating the model slices for the design alternatives

To illustrate the bottom-up slicing algorithm and the types of analysis we can perform on the resulting view, we will compare the effects of using a centralized database for the information against the effects of using information binders as resources. While this is not the most critical decision that KHP faces, the choice of information resources for counsellors is important and is adequate to illustrate our comparative analysis methodology.

In order to perform a comparative analysis between the two alternatives we will generate the bottom-up model slices corresponding to the *!Use Centralized Database for Information* and *Provide Information in Binders* tasks, apply the evaluation algorithm and compare the results obtained on them. The two bottom level tasks represent two viable operationalizations of providing information resources to counsellors.

Although the evaluation results obtained on the bottom-up model slices are not absolute, they should indicate more positive or negative effects for one of the two design alternatives. When the design space is considerably larger, the comparative analysis methodology can also be used to reduce the number of design alternatives that need to be considered for a more comprehensive analysis. When the design space has been reduced to a reasonable size, the modeller can perform a thorough analysis of the remaining alternatives and asses their effects on the client goals using the top-down slicing concept presented in Chapter 4, which addresses scalability by focusing the analysis on a single client objective at a time.

The diagram in Figure 3.3.1-1 illustrates the bottom-up model slice corresponding to the *!Use Centralized Database for Information* task.

The greyed-out intentional elements and links represent the elements that were filtered out by the slicing algorithm as unaffected by the current design choice. We opted to keep those elements in the diagram to visually illustrate the complexity ratio between the complete diagram and the bottom-up model slice.

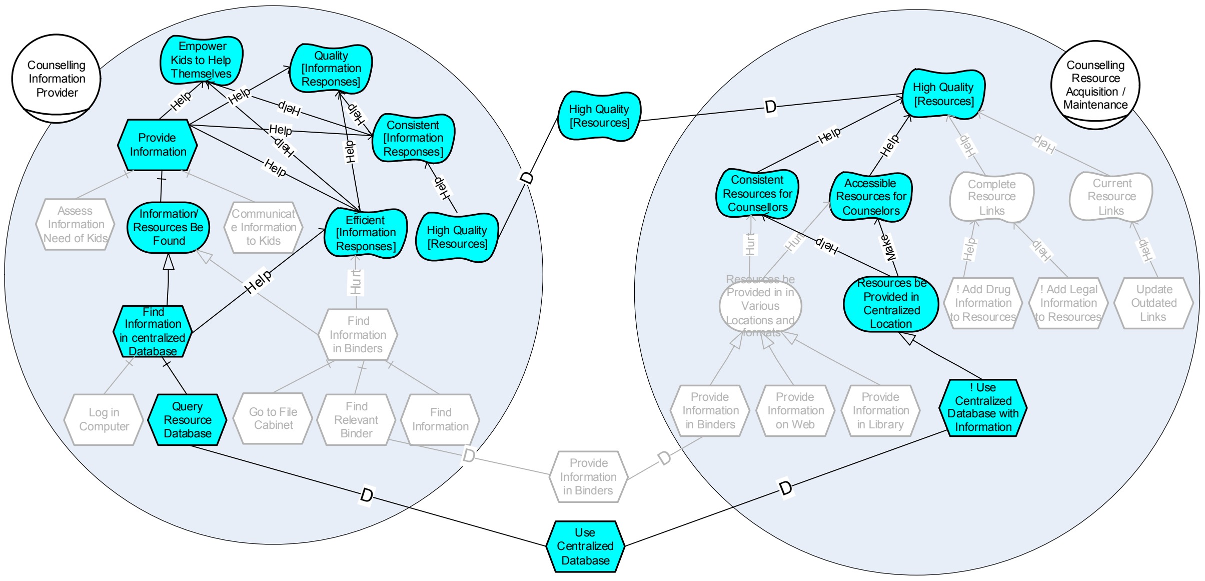

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Provide Information
Consistent [Information Responses]
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
Provide Information in Binders
Use Centralized Database
High Quality [Resources]
Counselling Resource Acquisition / Maintenance
High Quality [Resources]
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information
Help
Hurt
Make
D

Figure 3.3.1-1 Bottom-up model slice for the *!Use Centralized Database with Information* task

We can observe that the result of choosing an alternative in one role has effects on the high level goals of the other. In the case of our complete diagrams the effects of one major technology choice would spread out into several intentional actors. This justifies tracing the effects of a design decision on the goals of all the actors from the domain spectrum that are affected by it.

In some cases, the effects of a design decision on the goals of other actors appear to be very inconclusive or very indirectly related with the specified purpose of that decision. This can indicate that further simplification of the diagrams is possible and we will address this situation in detail in Chapter 5.

We can also note that there can be several paths of propagation for the effects of a design decision. In the example in Figure 3.3.1-1 the effects of choosing a centralized database approach propagate upwards through a means-ends link in the Counselling Resource Acquisition/Maintenance and through dependencies to the Counselling Information Provider role.

Similarly, we will generate the bottom-up model slice for the *Provide Information in Binders* task from the Counselling Resource Acquisition/Maintenance role. The corresponding diagram is presented in Figure 3.3.1-2.

As we can easily see the complexity of the resulting diagram after applying the slicing methodology has greatly decreased, making the process of reasoning on it much more manageable. The algorithm filtered out 51.5% of the number of intentional elements and 51.3% of the intentional links, while still keeping a coherent view of the diagram, containing only the elements relevant for the type of analysis we are conducting.

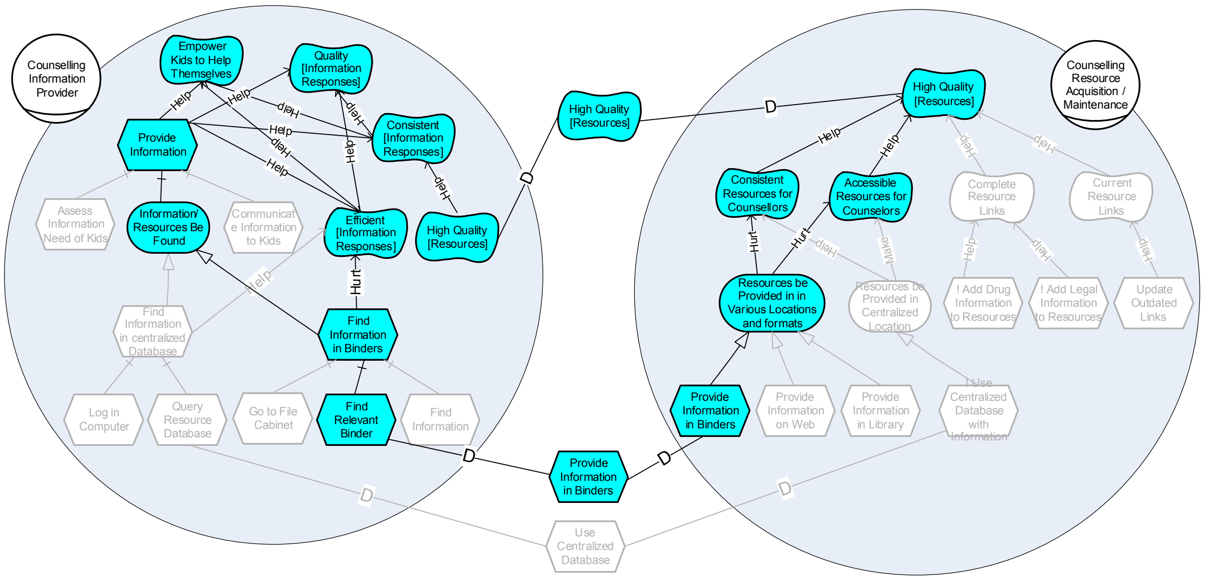

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Provide Information
Consistent [Information Responses]
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
High Quality [Resources]
Provide Information in Binders
Use Centralized Database
Counselling Resource Acquisition / Maintenance
High Quality [Resources]
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information
Help
Hurt
Make
D

Figure 3.3.1-2 Bottom-up model slice for the *Provide Information in Binders* task

We have to take into account the fact that the considered example has a level of complexity nowhere near our original comprehensive diagrams. In the case of complex diagrams, the algorithm can filter out entire intentional actors that are unaffected by the chosen alternative. In the case of our comprehensive KHP diagrams, the bottom-up slicing algorithm typically filtered around 75% of the intentional elements thus greatly reducing complexity. However, the filtering effectiveness of the bottom-up slicing algorithms depends on the domain properties and modelling approach.

3.3.2 Evaluating the effects of the design alternatives

Given the simplicity of the example we considered for illustration purposes we can see that the two alternatives are similar in scope. Since we cannot use the scope criteria as basis for choosing among alternatives, we will conduct a qualitative analysis on the resulting model slices using i*'s evaluation algorithm. An overview of i*'s link types can be found in section 5.1 and a summary of the label propagation rules can be found in Table 5.2-1.

The result of the evaluation algorithm applied on the slice corresponding to the *!Use Centralized Database for Information* task is presented in Figure 3.3.2-1.

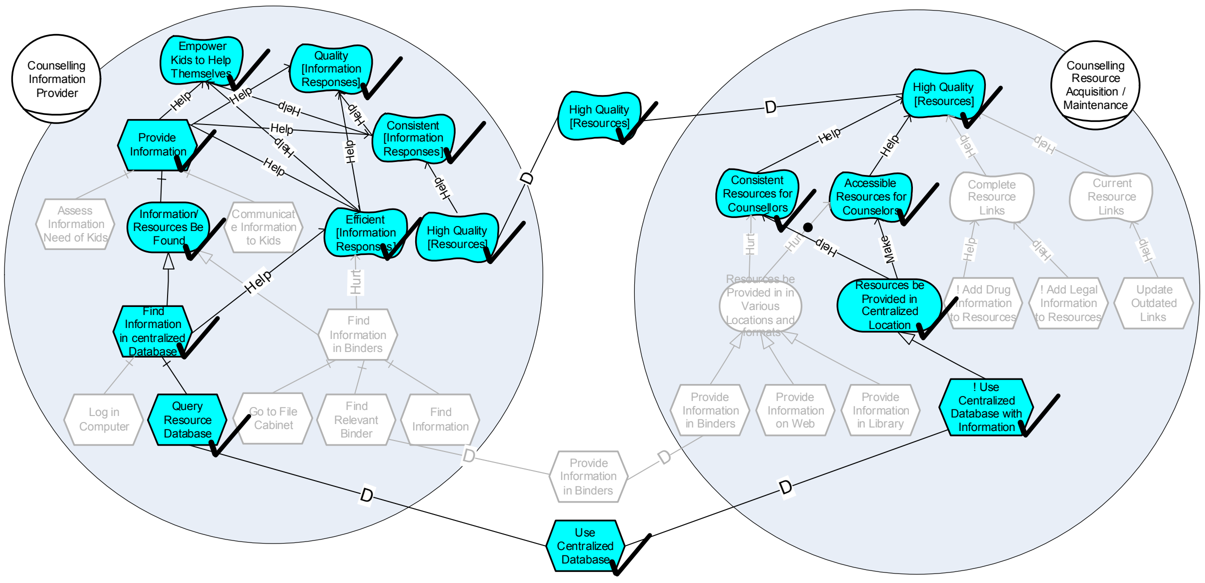

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Provide Information
Consistent [Information Responses]
Assess Information Need of Kids
Information Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
High Quality [Resources]
Provide Information in Binders
Use Centralized Database
Counselling Resource Acquisition / Maintenance
High Quality [Resources]
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Resources be Provided in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information
Help
Hurt
Make
D

Figure 3.3.2-1 Bottom-up model slice for the *!Use Centralized Database with Information* task with evaluation

The evaluation algorithm proceeds in a bottom-up fashion from the chosen alternative to the higher level goals of the relevant actors. We start by labelling the *Use Centralized Database for Information* task with satisficed and propagate the label upwards. We will elaborate more on the propagation rules of the labelling algorithm in Chapter 5.

We have to keep in mind that the evaluation results, although comparable, are not complete. We can see from the diagrams presented in Figure 3.3.1-1 and 3.3.1-2 that high-level softgoals such as *High Quality [Resources]* and *Consistent [Information Responses]* are receiving contributions from other elements that were filtered out. We can consider those elements as constants with regards to the comparison criteria. However, in order to conduct a complete analysis on those intentional elements we can use the top-down model slice concept which is described in Chapter 4.

As we can see, choosing the centralized source of information approach results in the satisficing the *Accessible Resources for Counsellors*, *High Quality [Resources]* , *Consistent [Information Responses], Efficient [Information Responses]* and *Quality [Information Responses]* softgoals.
Thus implementing a centralized information database contributes positively towards the quality of the resources available to counsellors and to the consistency of the information responses that counsellors provide.
We will use evaluation results of the high-level softgoals as a basis for the comparison between alternatives.
The results of applying the evaluation algorithm for the second alternative are presented in Figure 3.3.2-2.

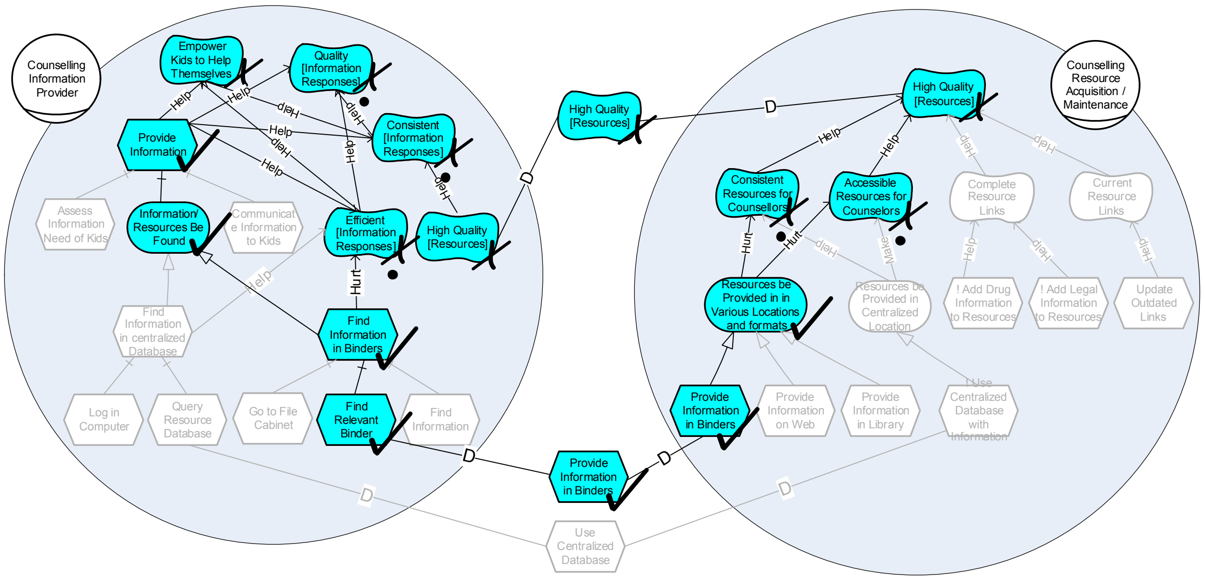

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Consistent [Information Responses]
Provide Information
Efficient [Information Responses]
High Quality [Resources]
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
Help
Hurt
D
High Quality [Resources]
Provide Information in Binders
Use Centralized Database
Counselling Resource Acquisition / Maintenance
High Quality [Resources]
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information
Make

Figure 3.3.2-2 Bottom-up model slice for the *Provide Information in Binders* task with evaluation

For this alternative we can see that the labels of the high level softgoals are different from those resulting from the previous alternative.

Providing resources in information binders results in a partial denial of the *Consistent Resources for Counsellors* and *Accessible resources for Counsellors* softgoals. This translates into a denial of the *High Quality [Resources]* softgoal which propagates into a partial denial of the *Consistent [Information Responses]* softgoal.
Thus we can conclude that providing resource information to counsellors in binders affects negatively the quality of the available resources and the consistency of their information responses.
By comparing the two alternatives we can conclude that the centralized information source alternative is significantly better than the information binder approach.

Based on this kind of analysis we can recommend to KHP to implement a centralized source of information as a replacement for the current way of providing resources to counsellors.

3.3.3 Validation of the comparative analysis results

We will validate the comparative analysis conclusions by evaluating the complete model for the two possible design decisions. The evaluation of the whole model is possible here given the small scale size of the considered example and is performed here only to illustrate the validity of our comparative analysis results and methodology. However performing several evaluations on our complete models would not be feasible. Our experience has shown that on such models, the slicing algorithm typically filters out between 70%-80% of the diagram content for different alternatives, thus dramatically reducing the diagram complexity.

Given that in the evaluation of the complete models there can be several evaluation patterns based on the elements that were not included in the model slice, we will use several scenarios in order to thoroughly validate our analysis conclusions.

Since the elements that don't belong to either model slice are independent of the choice criteria we will have to separately assign them with labels.

In order to assess the influence of those elements on the high level intentional goals we will consider two extreme scenarios:

- In the first scenario we will consider that information resources are kept current and complete by updating outdated links and adding drug and legal information to the resources. We will name this *the optimistic scenario.*
- In the second scenario we will consider that no steps are being taken to ensure that resources are kept complete and current. We will name this *the pessimistic scenario*.

The optimistic scenario

The first scenario is the most desirable one and the one more likely to happen since the KHP Company is taking steps towards improving the quality of the resources that are available to counsellors. However, the pessimistic scenario is also interesting for analysis since it will take us at the opposite end of the spectrum which will enable us to better assess the impact of elements filtered by the slicing algorithm and to perform a thorough validation of the comparison results obtained using the slicing methodology.

We will conduct the evaluation process for the two information providing alternatives for both of the scenarios.
The diagram in Figure 3.3.3-1 illustrates the evaluation results for the centralized information resource alternative in the optimistic scenario.

In order to evaluate the effects of selecting the centralized information source approach we will label the *!Use Centralized Database with Information* task with satisficed and the *Provide Information in Binders*, *Provide Information on Web* and *Provide Information in Library* tasks with denied. Since we are conducting the evaluation for the optimistic scenario we will label the *Add Drug Information to Resources, Add Legal Information to Resources* and *Update Outdated Links* tasks with satisficed.

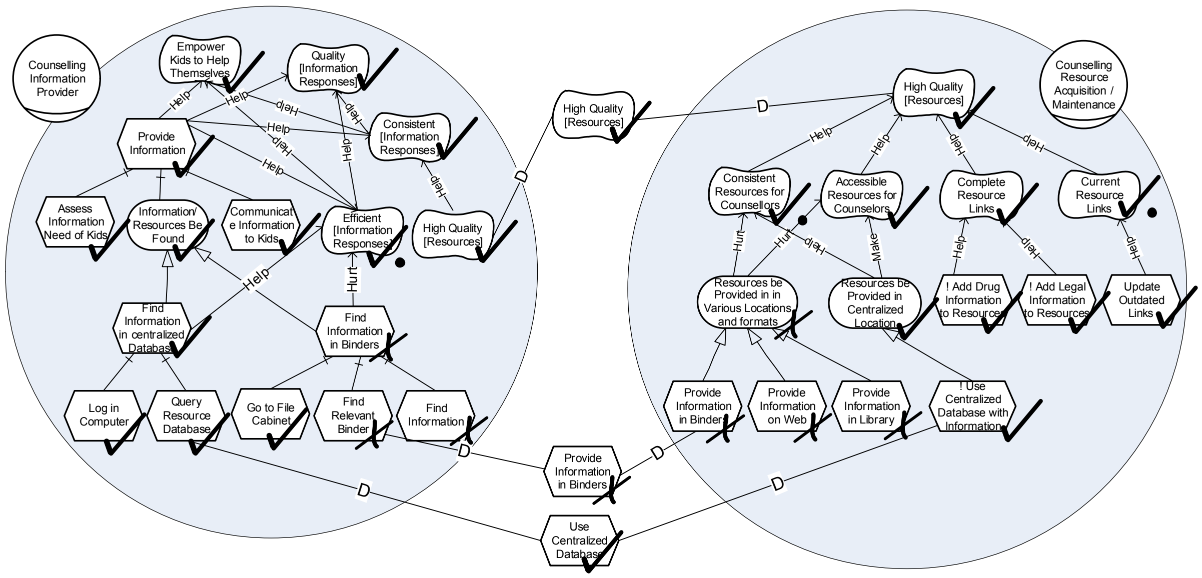

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Provide Information
Consistent [Information Responses]
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
Help
Hurt
D
High Quality [Resources]
Provide Information in Binders
Use Centralized Database
Counselling Resource Acquisition / Maintenance
High Quality [Resources]
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Make
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binder
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information

Figure 3.3.3-1 Model evaluation for centralized information source alternative in the optimistic scenario

By comparing the evaluation results for the high-level softgoals with those corresponding to the bottom-up model slice presented in Figure 3.3.2-1, we can see that the evaluation labels for the high level softgoals in both intentional actors are identical.
Thus in the example we considered, the model slice corresponding to the centralized source of information technology, although significantly simpler, is sufficient to conduct an analysis on the satisficing of the high-level softgoals.

This result, although stronger than the initial stated purpose of the bottom-up slicing methodology, is not generalizable. We cannot assume that in all situations contributions to high-level intentional elements that are filtered by the slicing algorithm will not have the potential to change the evaluation labels for those softgoals in some scenario.

Our aim is to use the results obtained on the complete models in both scenarios to validate the results of our comparative analysis performed on the model slices.

We can conclude from the analysis that using the centralized information source alternative along with taking steps to ensure that the information is current and complete results in the provision of consistent, quality information responses in an efficient manner by the counsellors.

The evaluation results of the information binder alternative conducted on the complete model for the optimistic scenario are presented in Figure 3.3.3-2.

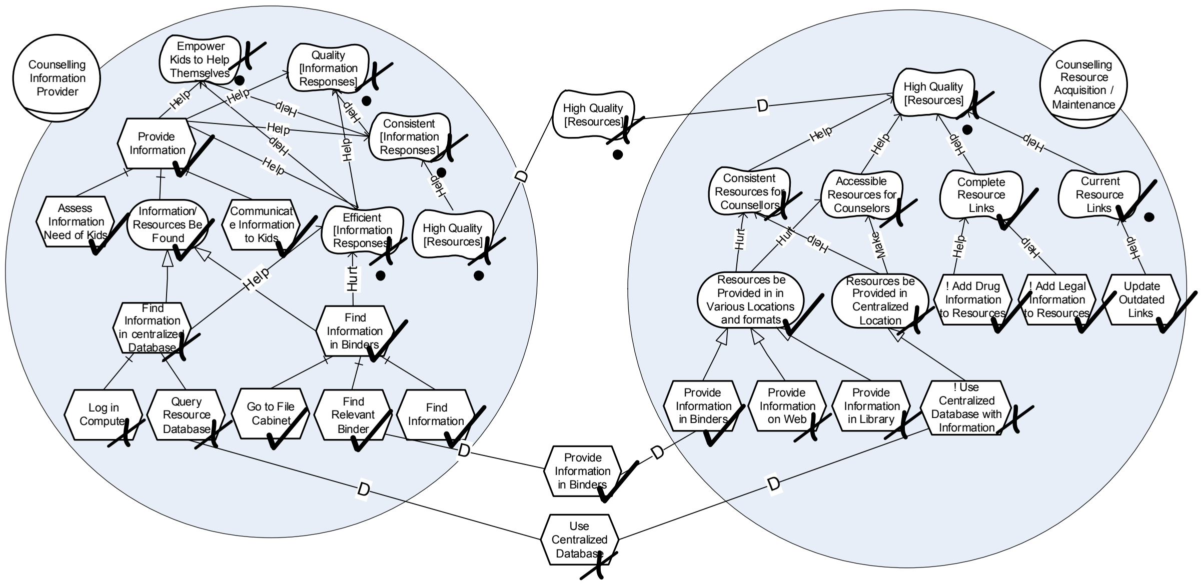

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Consistent [Information Responses]
Provide Information
Help
Hurt
High Quality [Resources]
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
D
Provide Information in Binders
Use Centralized Database
Counselling Resource Acquisition / Maintenance
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Make
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information

Figure 3.3.3-2 Model evaluation for information binder alternative in the optimistic scenario

In order to focus the analysis on the information binder alternative we will have to single out its effects by labelling the *Provide Information in Binders* task with satisficed and the *!Use Centralized Database with Information, Provide Information on Web* and *Provide Information in Library* tasks with denied. By propagating the labels upwards we will obtain the evaluation results for the high-level softgoals.

By comparing the evaluation results of the high-level softgoals with those obtained using the bottom-up slicing methodology, presented in Figure 3.3.2-2, we can see that the results are again identical. We can conclude from the analysis that providing the information coming from various sources and in various formats affects negatively the efficiency and consistency of counsellor information responses, even when steps for keeping that information complete and accurate are taken. We can observe that in the optimistic scenario the analysis conclusions based on the model slices were validated on the complete model for both of the alternatives. More important the comparison result between the two alternatives, indicating that the centralized information source approach was more desirable, was validated by the comprehensive analysis.

The pessimistic scenario

In the pessimistic scenario we will consider that no steps are being taken to ensure that the information resources are either complete or accurate.

Thus we will consider that the elements contributing to the *Complete Resource Links* and *Current Resource links* softgoals are all unsatisficed and we will label them accordingly.

In order to complete the evaluation we will propagate the labels of the leaf nodes upwards towards the top level intentional elements.

The evaluation results for the centralized information approach in the pessimistic scenario are presented in Figure 3.3.3-3

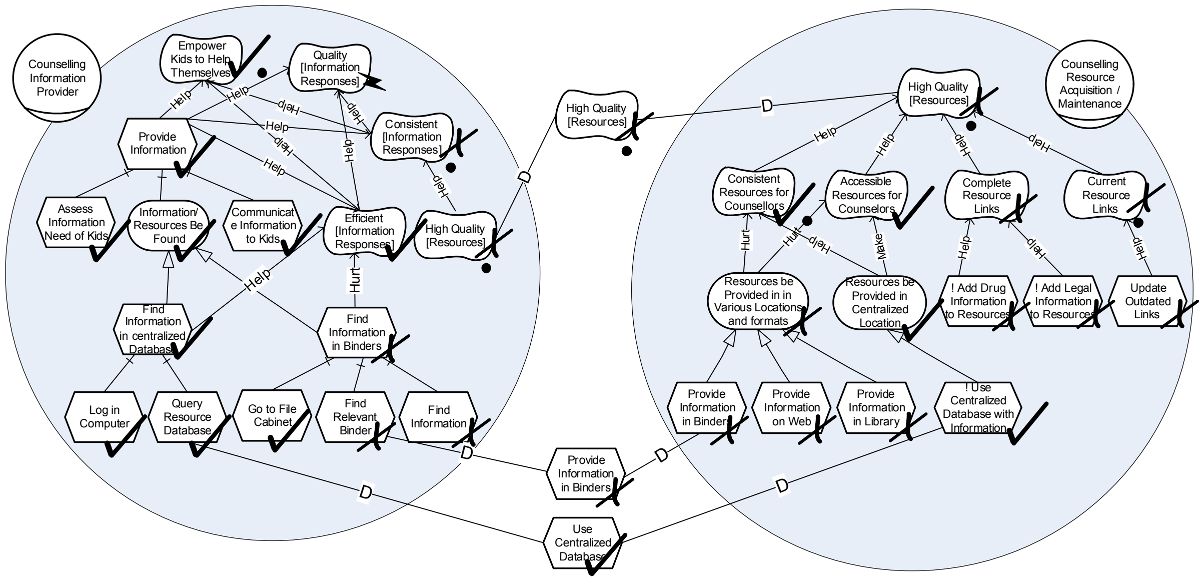

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Provide Information
Consistent [Information Responses]
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
Help
Hurt
D
Provide Information in Binders
Use Centralized Database
High Quality [Resources]
Counselling Resource Acquisition / Maintenance
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Make
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information

Figure 3.3.3-3 Model evaluation for centralized information source alternative in the pessimistic scenario

As we can see the resulting labels for some high-level intentional softgoals are different than the ones we obtained on the model slice for this design alternative (Figure 3.3.2-1).

In the alternative that resources are neither complete nor current we cannot consider them to be of high quality even though they might be accessible and consistent.

Thus we have labelled the *High Quality[Resources]*softgoal in the Counselling Resource Acquisition/Maintenance with weakly denied. The low quality of the resources affects negatively the consistency of the information responses. Although implementing this alternative does improve the efficiency of the responses, in order to have high quality responses additional steps to ensure the completeness and accuracy need to be taken.

As we anticipated a bottom-up model slice is not the appropriate tool to conduct a thorough analysis of the high-level intentional elements. However the purpose of the bottom-up model slice is to function as an aide in comparative analysis between different alternatives. The evaluation result obtained on bottom-up model slices can be used as a comparison basis between alternatives. In order to conduct thorough analysis for an intentional element the modeller can resort to the top-down model slicing technology which is presented in Chapter 4.

We will conduct the evaluation for the information binder methodology in the pessimistic scenario as well, to assess whether the comparative analysis results based on the model slices hold.
The fact that in this scenario no steps are being taken to ensure completeness and accuracy of resources combined with the choice of providing information in binders which affects negatively accessibility and consistency leads to a very negative result for the *High Quality[Resources]* softgoal. This result combined with the negative impact this alternative has on the efficiency of information responses leads to a strong denial of all high-level softgoals. (Figure 3.3.3-4)

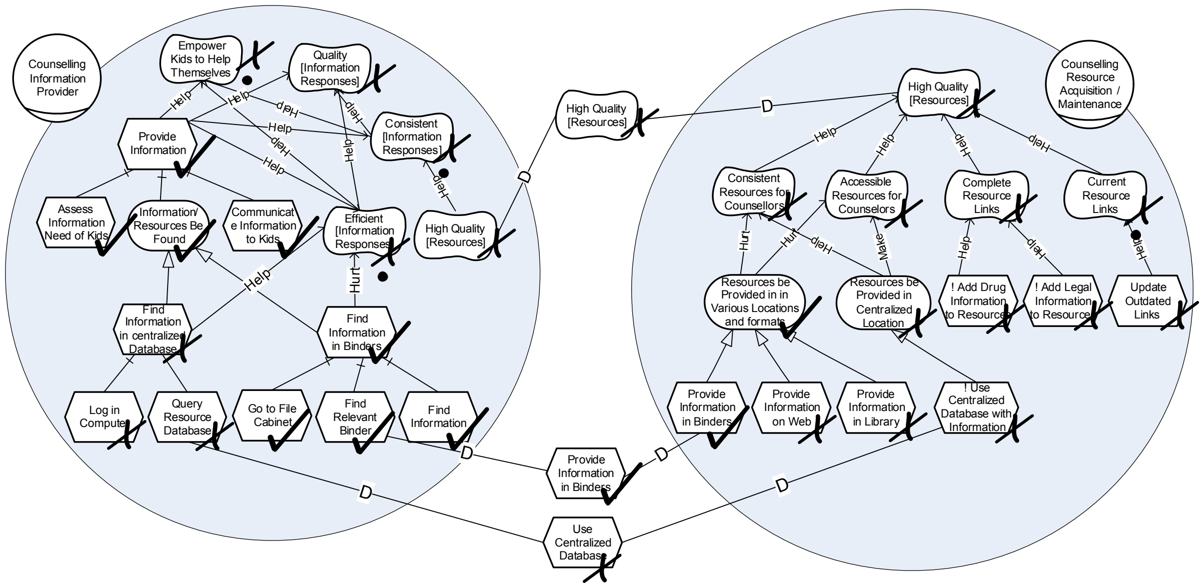

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Consistent [Information Responses]
Provide Information
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
Help
Hurt
D
Provide Information in Binders
Use Centralized Database
High Quality [Resources]
Counselling Resource Acquisition / Maintenance
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Make
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information

Figure 3.3.3-4 Model evaluation for information binder alternative in the pessimistic scenario

We can conclude that employing this alternative and not taking any steps to ensure that resources are kept complete and accurate will result in inconsistent and inefficient information responses, thus reducing the company's ability to provide kids with information relevant to their needs.

By comparing the evaluation results based on the complete model for this alternative (Figure 3.3.3-4) with those obtained for the centralized information source alternative (Figure 3.3.3-3) we can observe that the latter ensures a better satisficing of the high-level softgoals than the former.

We can therefore conclude that the centralized information source alternative is more desirable than the information binder alternative. We have seen that the centralized information approach is superior to the information binder approach in both of the scenarios we have considered.

We have illustrated that the comparative analysis conclusions, drawn based on the bottom-up model slices, are correct and consistent with the findings based on the complete models.

We were able to conduct a full analysis of the complete models here because the models we use to explain the concepts are very simple. However attempting an analysis based on the evaluation of the complete model for models with complexity similar to those presented in Table 3.1 is exponentially more difficult.

3.4 Practical application of the bottom-up slicing concept – the KHP case study

In the previous sections we have argued that the bottom-up model slice concept provides a very useful simplification of very complex i* models with respect to pertinent analysis questions that arise in the Requirements Engineering process and is well suited for conducting a comparative analysis of design alternatives.

We will illustrate the above mentioned assertions with facts from the KHP project, where this concept was first devised and used.

The aim of our analysis in the KHP project was to determine possible new technologies that could be employed by the company as means of providing counselling.

From the first set of interviews, several technologies arose as possible alternatives to supplementing current services and all the interviewees provided us with details from various perspectives that completed the picture. The choice of possible technologies had some important constraints that arose from the organizational context:

- The chosen technology(ies) should enable counsellors to acquire as many cues and context information as possible
- The chosen technology(ies) should enable the use of current counselling approaches and should allow effective counselling to be performed
- The chosen technology(ies) should not incur a high training overhead on the counsellors

- The chosen technology(ies) should guarantee the safety, anonymity and confidentiality of service users and providers
- The chosen technology(ies) should minimize the potential for counselling liability problems for the company

During the interviews, counsellors voiced concerns over the use of web as a counselling medium. Some were concerned that the web would deprive of much needed cues and context information, making it "too dry" of a medium for counselling, while others were concerned with the fact that written counselling may appear too direct and could increase the company's liability issues. There was also concern among counsellors that some counselling alternatives might foster further isolation of the kids with problems.

Given the sensitivity of the issues at stake, a thorough in-depth analysis was required before a decision, on which approaches should be pursued, could be made. Choosing a new counselling service also has serious impacts on various departments within the organization. A careful analysis of those effects needed to be conducted before a decision could be reached.

The high volume of information encompassed in the interviews was translated into models using the methodologies described in Chapter 2. Our emphasis on detail in the modelling stage rendered models with a high intrinsic complexity.

In order to perform a thorough, in-depth analysis of various technologies we had to identify and evaluate the effects of each technology alternative across the whole spectrum of the domain.
Since we were not constrained to choosing only one new technology, our analysis would determine whether a single technology can fulfill all the given constraints or whether several technologies had to be implemented.

The number of possible technology alternatives that arose from our initial elicitation phase, supplemented with technologies that we considered as viable alternatives lead to a design space of considerable size to be considered. By performing a slice-based comparative analysis between technologies we managed to reduce the set of viable technologies to a significantly smaller one.

We employed the use of the bottom-up model slicing technique to conduct our analysis on various technology alternatives. The advantage of employing the slicing technique was that it provided us with a view consisting only of the elements that were relevant to that particular technique. The resulting diagram was far less complex and more manageable than our original one.

We present in Table 3.4-1 a detailed comparison between the model slice corresponding to the bulletin board with delayed moderation technology and the whole rationale diagram based on the number of intentional elements and corresponding links.

	KHP SR diagram	Bottom-up model slice
Actors	61	18
Softgoals	783	270
Goals	160	6
Tasks	530	34
Resources	57	5
Help links	880	223
Hurt links	99	23
Make links	25	2
Break links	9	4
Means-Ends links	150	6
Decomposition links	303	12
Dependency Links	875	186

Table 3.4-1 Complexity comparison between the comprehensive KHP SR model and bottom-up slice for bulletin board with delayed moderation technology

As we can easily see from Table 3.4-1 the model slice is considerably smaller than the comprehensive model. In Table 3.4-1 we through actor we denote the actor class with all of its specializations: agent, role, position, etc.

We applied i*'s evaluation algorithm on the resulting slice, which provided us with qualitative results that we used as a basis for comparison between technology alternatives.

However evaluation results obtained for high level goals cannot be considered complete. High-level goals may be receiving contribution from various intentional elements which can be either related to other alternatives or independent of the choice criteria.

Those contributions would be filtered out by the bottom-up slicing algorithm as unrelated with the chosen alternative.

The evaluation results should be viewed as partial results for high-level intentional elements and as means for conducting comparisons between alternatives. In order to conduct a full evaluation of a client goal, the modeller can employ the top-down model slice concept which is described in detail in Chapter 4. Using the top-down model slicing concept the modeller can get a comprehensive picture of all the elements that contribute to that specific client goal.

3.5 Conclusion

We have illustrated that in order to conduct a comparative analysis one does not need to use all the information encapsulated in an i* diagram. The bottom-up slicing algorithm decreased dramatically the complexity of the diagram by filtering out the elements that were not related with the analysis question at hand (which represented approximately 50% of the information in the diagram in this case). Conducting the reasoning process on the significantly simpler model slice proved to be much easier than reasoning on the entire diagram.

We have validated the analysis results obtained on the bottom-up model slices against those obtained by reasoning on the complete model and we have indicated the need for the top-down model slice concept for analysis.

4. Top-down model slicing

We have seen in the previous chapter that the bottom-up slicing concept is very useful in conducting comparative analysis and reducing the complexity of large diagrams. When conducting early requirements engineering analysis, other valid analysis questions and concerns may arise.

We will show in this chapter several additional analysis questions that we have encountered in our experience and illustrate how the top-down model slicing concept can help us successfully answer those questions while reducing the complexity of the underlying models.

4.1 Analysis justification

When conducting a model based requirements engineering analysis, there are different types reasoning that can be conducted. In the previous chapter we have illustrated how the comparative analysis technique can be performed while employing the bottom-up model slice concept to reduce the complexity of the underlying models.

Once the set of viable solutions has been explored and the less satisfactory solutions have been eliminated, the modeller can focus the analysis on assessing whether the organizational objectives are satisfied with the current set of alternatives and the implications of those alternatives on various stakeholders.

This second step is as important as the comparative analysis one, since it allows the modeller to conduct a comprehensive analysis of the high-level objectives, including the cross-cutting effects of various alternatives and effects of elements that are independent from the alternative selection criteria. We will refer to this analysis as Goal Oriented Analysis (GOA) for the remainder of this document.

The GOA approach allows us to focus the analysis on one objective at a time and to conduct a comprehensive analysis of whether the current set of design choices ensures the satisfaction of that objective. By conducting this analysis iteratively for all the specified objectives, we can conclude on whether the current design choices will satisfice the specified organizational objectives.

Depending on the nature of the domain, organizational objectives, expressed by hard goals or softgoals in i*, can have a very limited number of elements contributing to them or can be very broad in scope. When the number of elements that are contributing to the satisfaction of the objective is fairly small, the relative importance of each element with regards to the objective satisfaction can be seen as rather high. Conversely, with a goal that is very broad in scope, the individual importance of loosely connected elements decreases. This type of reasoning however is very simplistic and insufficient for determining the relative importance of element contributions.

Prioritizing among contributions is an operation that involves human judgement that can be aided to a certain extent by the scalability concepts we propose. We will discuss the model levelling concept in Chapter 5 as an aid in prioritizing contributions and assessing the directness of the contribution for intentional elements.

When the models abstracting the application domain reach a certain level of complexity, determining the scope of the goal and the relative importance of the contributors becomes increasingly difficult. Since we have argued that in such cases the analysis should be driven by a set of pertinent questions we will illustrate how the top-down model slice concept can help answer questions such as the ones mentioned above, while significantly decreasing the complexity of the underlying models.

We will define the top-down model slice concept as a way of reducing the complexity of the diagrams and its associated scalability problems, while keeping a coherent view containing all the relevant information needed or the analysis.

While the bottom-up model slicing concept started from a low level design decision and traced its effects on the various client objectives, the top-down starts from a high level client objective and traces downwards all the design decisions that affect that particular client goal. In the case of the bottom-up model slice the evaluation results for various client goals were incomplete and could only indicate positive or negative influences on them by various design decisions.

By performing an evaluation of the top-down model slice, the analyst can obtain absolute and complete results for client goals that are considered of interest. Although the accuracy of the results depends highly on the quality of the model and of the evaluation, the analyst should be able to predict with relative high certainty whether client objective can be satisfactorily achieved with a set of design decisions. The evaluation results obtained on top-down model slices are stronger than those obtained on bottom-up model slices because of the completeness of the slice with respect to a certain client goal.

Besides its scalability usages, the top-down model slicing concepts has a great analysis value because it can help the analyst answer some fundamental questions that arise in the requirements engineering process.

Table 4.1-1 summarizes some of the pertinent GOA questions that can be answered using the top down model slicing concept.

Analysis question	Associated concept
1. WHAT are the design decisions that affect a specified objective of the client?	1. Top-down model slice
2. HOW do the current design decisions affect a specified objective of the client?	2. Top-down model slice with evaluation
3. WHICH is the relative importance of design decisions for the objective?	3.Top down model slice with evaluation
4. WHY does a design decision affect a high level objective?	4. A path through the model slice from the operationalization of the design decision to the high level objective

Table 4.1-1 Correspondence between Goal Oriented Analysis questions and presented concepts

As we can see from Table 4.1-1 the covered analysis questions are pivotal for the early RE analysis process – the WHAT, HOW, WHICH and WHY questions. The questions are similar to the bottom-up concepts but here they address a different perspective. While in the bottom-up slicing case the focus was on reasoning about a low level operationalization representing a design decision, here the focus is on the satisfaction of client objectives based on all the design decisions that are considered.

We will briefly address each of these questions and illustrate how the top-down model slice concept, which provides a simplified view of the diagram, can help the modeller in conducting the reasoning process.

4.1.1 Analysis question 1 – The WHAT question

The first analysis question aims at identifying the scope of the design decisions affecting a certain client objective. Usually in the analysis process the obvious effects of design decisions are easily observable by the modeller.

However, side effects of those decisions are less evident and ignoring or overlooking them can have serious consequences on the subsequent implementation and deployment of the service. By enforcing a focus on detail through successive goal refinement, the i* modeling framework facilitates the process of inferring less obvious side effects of design alternatives.

As we have argued earlier, i* allows the embedding of a design alternative space of considerable size into a single diagram. The ability of the modeller to assess the scope of the objective and to conduct efficient reasoning decreases with the increase in complexity. By employing the use of the top-down model slice concept the modeller can work on a very simplified view of the diagram containing only the elements that are relevant for the analysis question at hand.

The top down slicing process will start from the element of interest and trace downwards all the operationalizations that affect it either directly of transitively. The start point of the algorithm can be a high or mid-level softgoal or goal. The decrease in complexity depends on the particularities of the domain and on the level of the starting element.

4.1.2 Analysis question 2 – The HOW question

After obtaining the comprehensive picture of the design decisions that affect a client goal, the modeller needs to apply a qualitative reasoning methodology in order to determine if the goal is satisfied or not with the current design setup. Goal oriented modelling is highly advocated as a way of guiding the elicitation process and helping the requirements engineer to ask the stakeholders the most relevant analysis questions. However, given the effort required for building such complex models, we may want to use them as a tool for predicting the effects of design decisions, before proceeding to the implementation.

In order to perform this reasoning process, the modeller can resort to the evaluation algorithm described in Chapter 3. The evaluation procedure that i* provides, offers us qualitative results for the achievement of client goals. Applying this standardized analysis approach on various design setups can also help us minimize the cognitive bias towards one design alternative or another, conferring more objectivity to the overall result.

By using the evaluation algorithm on the top-down model slice rather than on the whole model, we can reduce significantly the effort required to conduct the reasoning. The evaluation result for the starting element of the top-down model slice should be identical to that obtained on the comprehensive model, since all the elements that contribute to it either directly or transitively will be included in the slice. Thus any elements contained in the comprehensive model but not in the top-down slice can be considered irrelevant to the analysis question at hand.

The evaluation results for this concept are stronger than those obtained on the bottom-up model slicing concept. However given the trade-off between accuracy of results and filtering effectiveness we expect the bottom-up slicing algorithm to be more effective in reducing the complexity of the diagrams. We will not attempt a comparison between the two concepts since they are addressing different steps and questions in the analysis process. The effectiveness of the slicing algorithms can also be affected by the characteristics of the domain.

4.1.3 Analysis question 3 – The WHICH question

When determining the scope of the design choices that affect a client goal, the modeller can get a clearer picture of the relative importance of those choices towards the satisfaction of that goal. Establishing the relative importance of alternatives is a cognitive process where human judgement is required. However the analyst can be aided in making that judgement by the results of the qualitative evaluation process as well as the perceived scope of the solution pool for the client goal. As illustrated in the two previous sections, the top-down model slicing concept is a valuable aid in determining the scope of current design choices affecting the client objective and conducting the qualitative reasoning process. We will expand our analysis on this issue in Chapter 5 and present the model levelling concept which can help the modeller perceive how strongly are the intentional elements of the model slice connected to the slice originating point.

4.1.4 Analysis question 4 – The WHY question

The last question in Table 4.1-1 is concerned with the cause-effect traceability in the analysis.

This question is an opportunity for the modeller to verify model correctness, by tracing the contribution path linking the two intentional elements whose relationship is well known, as well as analyze less obvious side effects of design choices that can be inferred from the models. The path through the model represents the contribution chain linking the two intentional elements of interest.

There can be multiple paths linking two intentional elements and the concept is applicable to both types of model slices. This type of analysis can reveal less obvious properties of the domain as well as inconsistencies or errors in the model. If the contribution chain between the two elements is considered valid but less important than others than further simplification of the models can be achieved. We will discuss this situation in detail in Chapter 5.

4.2 Top-down model slicing methodology

We will describe the methodology and illustrate its usefulness for conducting analysis with a few examples. As with the previous methodology, since we are interested in a domain-wide analysis of client goals, we will not restrict the slicing algorithm to a single intentional actor.

Our algorithm will find all the descendents of an intentional element belonging to the source actor, trace the descendents across dependencies (from depender to dependee) to other intentional actors and then continue transitively the descendent finding process in those actors. The result is a domain view, containing all the descendents of the element of interest, which can be used for the kinds of analysis illustrated in Section 4.1

4.2.1 Algorithm implementation

We will use a non-recursive breadth-first traversal of the i* model similar to the one presented in Chapter 3 in order to generate a top-down slice. For representation we will use the Node and Edge classes defined in Chapter 3.

The algorithm implementation along with its main helper method are illustrated in Figure 4.2.1-1

```
/**Method that implements the top-down slicing algorithm*/
*@param The slice starting element
**/
void create_top_down_slice(Node startElement){
Queue myQueue = new Queue();
myQueue.enqueue(startElement);
//remove first element in queue and add all his direct descendents
      while (!queue.isEmpty()) {
             Shape temp = myQueue.dequeue();
             //mark element as visited;
             temp.setVisited();
             //add the direct descendents and dependees of temp to the queue
             addDescendentsandDependees(temp, queue);
             }
      }
```

Figure 4.2.1-1 The top-down slicing algorithm

The slicing method uses an additional Queue structure to store the direct descendents and dependees of the current element and to ensure a breadth first traversal. The method references two helper methods: setVisited() and addDescendentsAndDependees(). Our approach is to enqueue the slice originating element, find its direct descendents and dependees, enqueue them and then continue transitively the descendent and dependee finding process.

```
/** Adds the descendents and dependees of the specified Node to the queue
*@param The current node
*@param The descendent queue
**/
void addDescendentsAndDependees(Node temp, Vector queue) {
        //get the incoming links for the temp Node
        Edge[] incoming_links = temp.getIncoming_Links();
        //find the source node of each incoming link
        for (int i = 0; i < incoming_links.length; i++) {
                //if we haven't been down this path before
                if (! incoming_links[i].wasVisited()) {
                        //mark link as visited
                        incoming_links[i].setVisited();
                        int source_id = incoming_links[i].getSource();
                        Node source = getNode(diagram,source_id);
                        //if source node was not visited and is not in the queue
                        if (!source.wasVisited())&&(!source.wasEnqueued())){
                                //add the source node to the queue
                                queue.enqueue(source);
                                //mark the node as enqueued
                                source.setEnqueued();
                        }
                }
        }
}
```

Figure 4.2.1-2 Method that enqueues direct descendents and dependees

Both slicing algorithms have straight forward implementation with our representation. Given their simplicity, their wide analysis usages and scalability potential we would recommend their implementation in any tool support packages for goal-oriented modelling techniques.

4.3 Goal oriented analysis example using top down model slicing

We will illustrate the top-down model slicing methodology using an example (Figure 4.3-1) that contains several intentional actors. The example is based on the KHP project and presents two counselling roles along with other intentional actors from the organizational context. We will focus our analysis on the relationship between the Counselling Training and the Counselling Public/Internal Relations roles and for the purpose of simplicity we will only present the Strategic Rationale models for those roles. The rest of the intentional actors will be kept at Strategic Dependency Level.

The Counselling Training role contains all the activities related to training that counsellors perform. New counsellors receive training during the probation period from more senior counsellors, in order to understand the technological side of counselling as well as effective approaches for phone and web counselling. All counsellors receive periodical training from counselling operational and clinical supervisors and there are also workshops directed at familiarizing counsellors with new web technologies. In order to be more effective in promoting the counselling activity and the KHP organization, counsellors would like to receive training to improve their public speaking skills.

The Counselling Public/Internal Relation role comprises all the activities that counsellors perform to promote the counselling activity internally and the KHP organization in general.
In order to promote counselling internally and to have a say in the decisions of the company, counsellors are unionized and want to participate in board meetings. The counsellors need to develop their public speaking skills in order to speak about their job at fundraisers and Student Ambassador Conferences, as well as to maintain a proactive relationship with the media.

The company's experience has showed that individual donors but also corporate sponsors respond very well to counselling stories illustrating the nature of their activity and therefore an enhanced external "counsellor visibility" would be beneficial both for company image as well as fundraising revenues.

With all this in mind, we will focus our analysis on the things that counsellors need to do in order to achieve internal recognition. Internal recognition is very important for counsellor morale given the emotionally demanding nature of their work.

We illustrate in Figure 4.3-1 the setting we have described earlier.

As we can see, there are several actors in the diagram and the inner complexity of the two roles that are presented at SR level is slightly higher than that of our previous examples. However, the diagrams represent a simplified version of our original models and their intrinsic complexity is not comparable to that of our comprehensive diagrams.

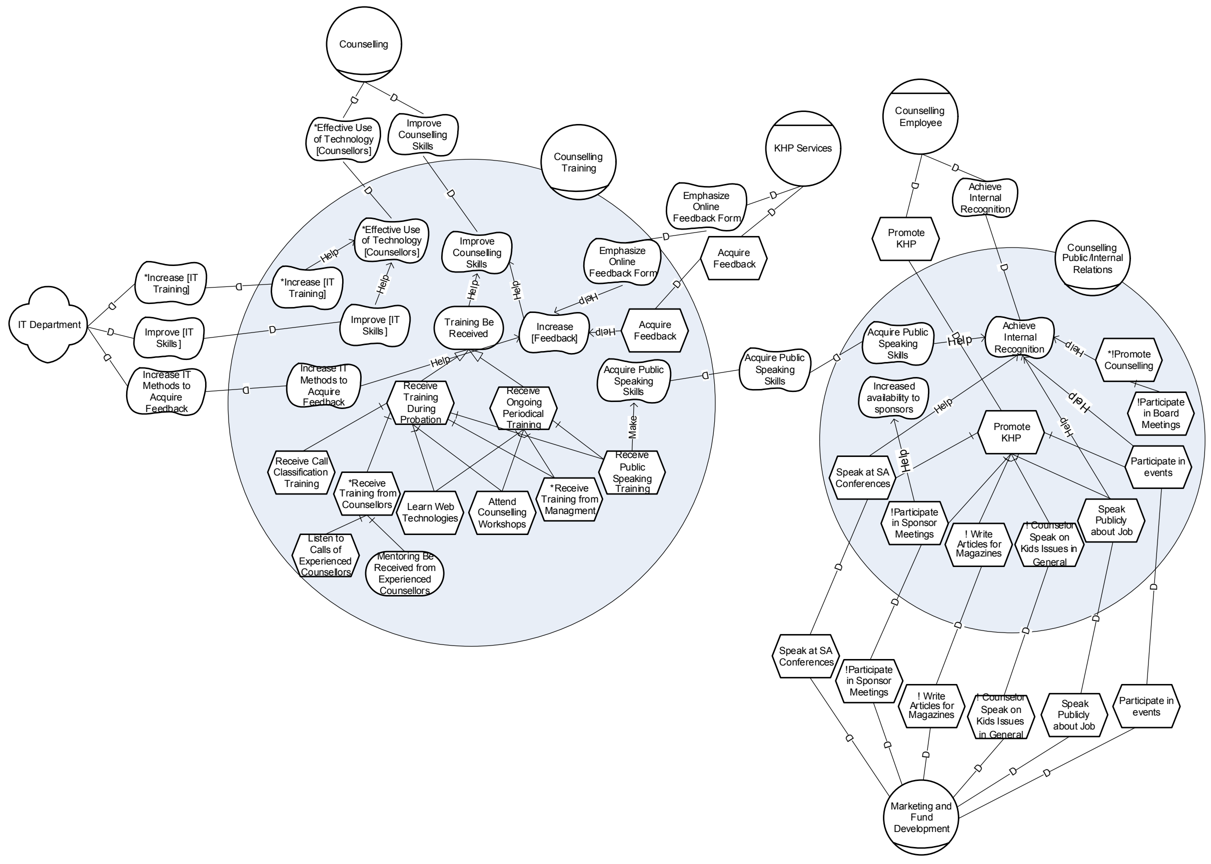

Counselling
*Effective Use of Technology [Counsellors]
Improve Counselling Skills
Counselling Training
KHP Services
Counselling Employee
Emphasize Online Feedback Form
Acquire Feedback
Achieve Internal Recognition
Promote KHP
Counselling Public/Internal Relations
IT Department
*Increase [IT Training]
Improve [IT Skills]
Increase IT Methods to Acquire Feedback
Help
Training Be Received
Increase [Feedback]
Acquire Public Speaking Skills
Receive Training During Probation
Receive Ongoing Periodical Training
Make
Receive Public Speaking Training
Receive Call Classification Training
*Receive Training from Counsellors
Learn Web Technologies
Attend Counselling Workshops
*Receive Training from Managment
Listen to Calls of Experienced Counsellors
Mentoring Be Received from Experienced Counsellors
Increased availability to sponsors
*!Promote Counselling
!Participate in Board Meetings
Participate in events
Speak at SA Conferences
!Participate in Sponsor Meetings
! Write Articles for Magazines
! Counselor Speak on Kids Issues in General
Speak Publicly about Job
Marketing and Fund Development

Figure 4.3-1 The SR model for the Counselling Training and Counselling Public/Internal Relations roles

The intentional elements whose name begins with an ! represent tasks that are not being performed currently, but are considered desirable and viable alternatives by certain segments in the organization. We will focus our analysis on the following question:

- How can counsellors achieve internal recognition in the organization?

Being recognized internally is very important for the counselling morale and for the quality of counselling. Various actors from the organization have expressed the desire to involve counsellors more in the decision making process and to expand their availability to sponsors and media. All the suggested changes will have an impact on the counsellor's perception of recognition from the organization.

4.3.1 Creating the top-down model slice

We will analyze all the elements that affect the internal recognition perception by using the top down model slice concept, having the *Achieve Internal Recognition* softgoal as a starting point.

The result of applying the top-down model slicing concept on the diagram are presented in Figure 4.3.1-1

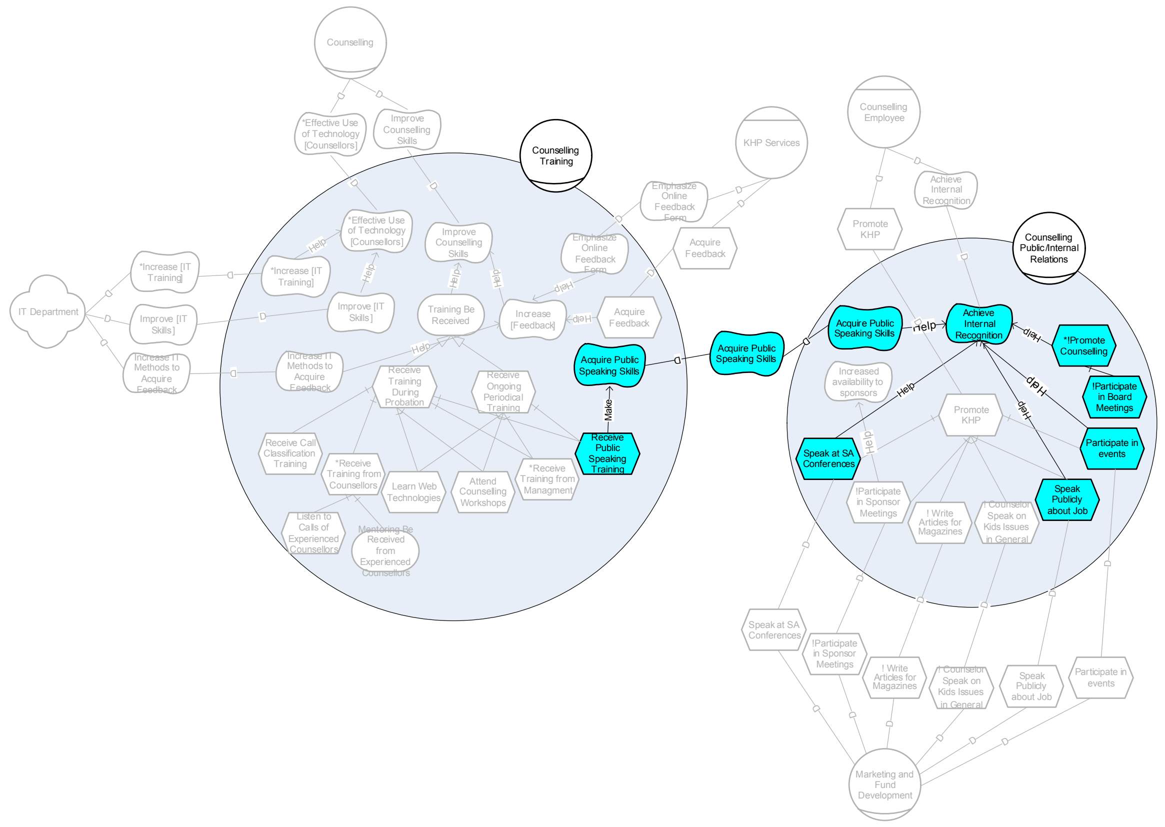

Counselling Training
Acquire Public Speaking Skills
Make
Receive Public Speaking Training
Acquire Public Speaking Skills
Counselling Public/Internal Relations
Acquire Public Speaking Skills
Help
Achieve Internal Recognition
*!Promote Counselling
!Participate in Board Meetings
Participate in events
Speak Publicly about Job
Speak at SA Conferences
Help

Figure 4.3.1-1 The top-down model slice for the *Achieve Internal Recognition* softgoal

The algorithm proceeds downwards from the *Achieve Internal Recognition* softgoal and adds all the elements that contribute either directly or transitively to it.

The greyed-out intentional elements and actors represent elements that have been filtered out by the slicing algorithm because they are irrelevant to the analysis question at hand. As we can easily see the irrelevant elements, outnumber by far those that have relevance for our analysis question. More important, we can see that the algorithm filtered out intentional actors that did not contain any intentional elements that contributed to the *Achieve Internal Recognition* softgoal. Although these actors were presented here at Strategic Dependency level for simplicity purposes, their Strategic Rationale models have been created and their intrinsic complexity was high. Thus, applying this concept on the comprehensive Strategic rationale model would result in filtering out the rationale for several actors, greatly reducing complexity.

The diagram presented in Figure4.3.1-1 also included the intentional elements that were filtered out by the slicing algorithm in order to illustrate visually the ratio between elements relevant to the analysis question and elements that are irrelevant. Given the ratio between the relevant elements for the analysis and those filtered out, we present in Figure 4.3.1-2 the diagram containing only the elements that belong in the top-down model slice for the *Achieve Internal Recognition* softgoal.

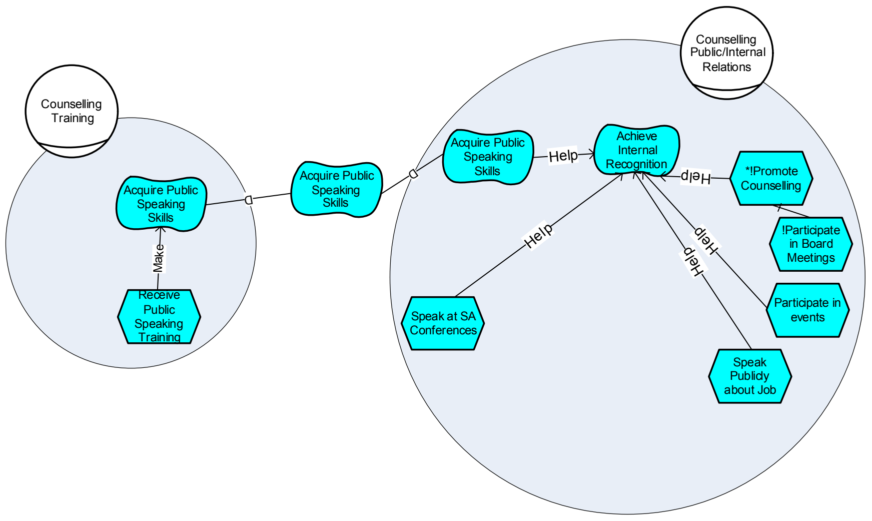

Counselling Training
Acquire Public Speaking Skills
Make
Receive Public Speaking Training
Acquire Public Speaking Skills
Counselling Public/Internal Relations
Acquire Public Speaking Skills
Help
Achieve Internal Recognition
Help
*!Promote Counselling
!Participate in Board Meetings
Help
Help
Help
Participate in events
Speak at SA Conferences
Speak Publicly about Job

Figure 4.3.1-2 The *Achieve Internal Recognition* top-down model slice without filtered elements

We will conduct our subsequent reasoning on the simplified diagram presented in Figure 4.3.1-2.

4.3.2 Scenario analysis on top-down model slice

In real-life situations there is often more then one way to solve a problem. Solutions can range from "perfect, regardless of the cost" to "good enough" when cost/benefit ratios are of concern. In i* models, these issues translate to the existence of several ways in which a goal can be achieved and to the degree of satisficing for that particular goal.

In order to select the adequate alternative, one has to analyze various scenarios leading to the solutions. Such a scenario analysis can be useful in determining the relative importance of the elements contributing to the achievement of the goal, as well as selecting the best scenario within the given constraints. We will illustrate this kind of analysis on two scenarios:

- "Internal recognition is crucial for our counsellors. We want to achieve it and higher costs are not a problem"
- "We want to improve the internal recognition of our counsellors. However we do not want to invest in training facilities at this point"

Both of the scenarios are fictional and are used to illustrate the scenario based analysis on top down model slices. The two quotes describing the scenarios do not come from our interviews with the organization but rather represent a brief description of two different fictional viewpoints.

We will illustrate this kind of analysis in the example presented in Figure 4.3.2-1. For simplicity we have removed the elements filtered out by the slicing algorithm.The results of conducting the evaluation on the model slice in the first scenario are presented in Figure 4.3.2-1

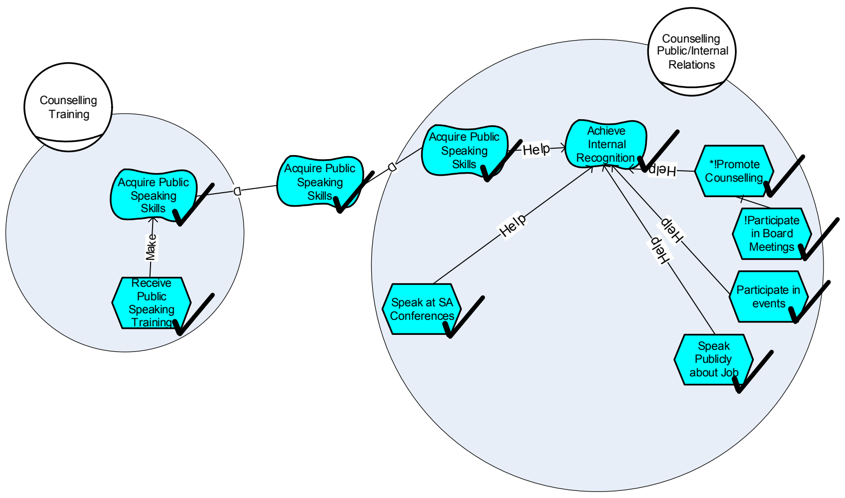

Counselling Training
Acquire Public Speaking Skills
Make
Receive Public Speaking Training
Acquire Public Speaking Skills
Counselling Public/Internal Relations
Acquire Public Speaking Skills
Help
Achieve Internal Recognition
Help
*!Promote Counselling
!Participate in Board Meetings
Help
Help
Participate in events
Help
Speak at SA Conferences
Speak Publicly about Job

Figure 4.3.2-1 Evaluation results for model slice in first scenario

The first scenario is one that is not encountered frequently in practice. Almost in all cases, the cost and resources that need to be allocated are of concern to those that need to provide it. Our interviewees from the organizational context provided us with the options for improving the internal recognition of counsellors that are represented in Figure 4.3.2-1. However if the cost and resources needed were of no concern, probably several other possibilities for achieving internal recognition of counsellors could be considered.

We indicate the fact that resources are of no concern in this scenario by labelling all bottom level elements with satisficed. This indicates that needed training resources will be allocated and that counsellors can take time off from counselling to participate in conferences and events.
The evaluation results indicate that in this scenario the *Achieve Internal Recognition* softgoal is strongly satisficed. As expected, the outcome of this scenario is highly positive but such scenarios are rarely feasible in real-life situations.

In the second scenario we will consider that required training resources cannot be allocated at this point. We will indicate this by labelling the *Receive Public Speaking Training* task from the Counselling Training role with denied. We will propagate the effects of the leaf labels and analyze the outcome for the *Achieve Internal Recognition* softgoal.
The results of the evaluation procedure for model slice in this scenario are presented in Figure 4.3.2-2

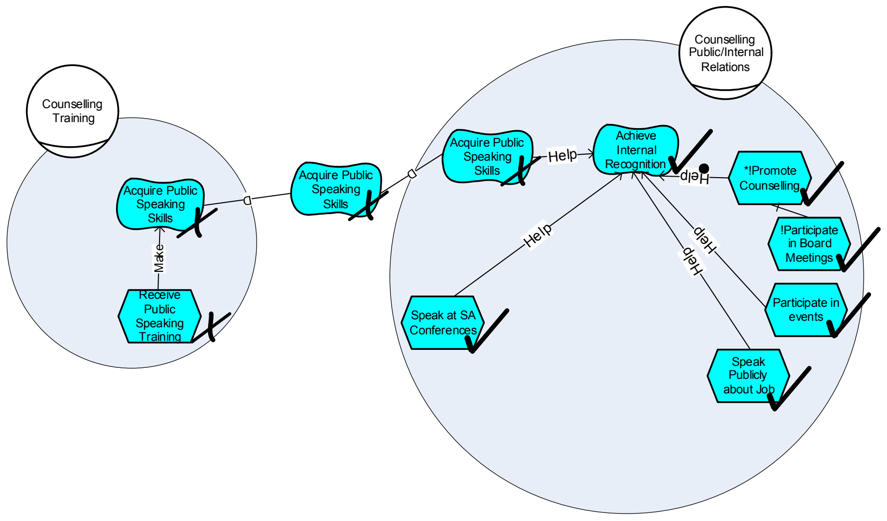

Counselling Training
Acquire Public Speaking Skills
Make
Receive Public Speaking Training
Acquire Public Speaking Skills
Counselling Public/Internal Relations
Acquire Public Speaking Skills
Help
Achieve Internal Recognition
Help
*!Promote Counselling
!Participate in Board Meetings
Help
Help
Participate in events
Help
Speak at SA Conferences
Speak Publicly about Job

Figure 4.3.2-2 Evaluation results for model slice in first scenario

As we can observe from the diagram in Figure 4.3.2-2 the fact that training resources are not available affects negatively the counsellor's public speaking skills. Without training, counsellors may be more reluctant to speak publicly about their job or at Student Ambassador Conferences. However most counsellors do feel comfortable about public speaking and the overall result for the Achieve Internal Recognition is a positive one. In this scenario we can conclude that public speaking training is not essential in achieving internal recognition. By satisficing the other bottom level tasks, a satisfactory achievement of the internal recognition softgoal can be obtained. By conducting an analysis on various combinations of bottom level task combinations, the modeller can get a better feel for the relative importance of operationalizations and design a solution within the given constraints.

4.3.3 Validation of Goal Oriented Analysis results

In terms of completeness this concept is stronger than the bottom-up slicing one. Although having a comprehensive rationale model is useful and needed in order to conduct a thorough analysis, an all-at-once analysis for all the issues that are of concern is neither feasible nor recommended. The top-down model slice concept offers a way of focusing the analysis on one issue at a time, while providing the relevant information for that issue in a more manageable diagram. The goal-oriented analysis approach is advocated by [52] as a way of refining and clarifying requirements incrementally. By driving the modeller to focus the analysis on a single goal at a time, the slicing concept allows for better resource management and prioritization of concerns that are of interest.
One of the strong points of the top-down model slicing concept is its completeness with respect to the analyzed issue. The result of applying i*'s qualitative evaluation algorithm on the starting point of the top-down model slice is identical to the one obtained on the comprehensive model for the same softgoal. Thus the irrelevant elements, although needed for the general context and for other analysis questions, can be safely discarded for the current analysis question.

By applying this methodology, the modeller doesn't ever have to reason on the comprehensive diagram as a whole. Instead model slices can be created for all the elements of concern. If the resulting model slice is still considered to be rather complex, the modeller can use the model levelling concept presented in Chapter 5 to further simplify the slice or he can employ the top-down slicing concept on lower level descendents of the element of interest and aggregate the results subsequently.

The number of elements filtered by the top-down slicing algorithm can vary depending on the particularities of the domain and the level of the intentional element. Our experience with KHP has shown that when applied for mid-level goals or softgoals the algorithm filtered at least 50% of the intentional elements. For our considered example, the elements that belonged to the *Achieve Internal Recognition* model slice represented about 30% out of the whole rationale for the two actors. Moreover 71% of the intentional actors were filtered by the algorithm since they were not related with the analysis question at hand.

4.4 Practical application of the top-down slicing concept – the KHP case study

The top-down model slice is a concept that resulted from our KHP project. After reducing the size of the design space by using the bottom-up model slicing concept, we needed a way to thoroughly evaluate the effects of the remaining alternatives on the key stakeholder objectives.

We set up a prioritization exercise with various stakeholders from the organizational context, in which they were asked to rank the various objectives we identified in our initial elicitation phase according to their importance for the organization. Involving stakeholders in the design is a practice strongly advocated by the participatory design community. By working themselves on developing the solution, stakeholders get a sense of involvement and ownership over the resulting solutions. This minimizes the proverbial hostility towards change and ensures a smooth integration of new technological solutions in the clients' work environment.

We used the prioritization exercise as a starting point for our subsequent analysis phase. For each of the important goals of the organization, we constructed a top down model slice on which we applied i*'s qualitative evaluation algorithm. Given the fact that top-down model slices were significantly simpler than our comprehensive model, the effort required to conduct the evaluation and reasoning was significantly decreased.

4.5 Conclusion

We have illustrated the top-down model slicing concept as a useful tool in conducting goal and agent oriented analysis and overcoming some of the scalability problems we presented. We have identified the types of analysis that the concept supports and provided examples to illustrate them.

5. Model Levelling

As we have previously seen, scalability challenges can be successfully addressed by the top-down and bottom-up slicing concepts. When diagrams reach a certain level of complexity, they become indispensable tools for conducting the analysis. From our experience with the KHP project the two slicing concept are sufficient to reduce the complexity of the underlying models to a manageable size for analysis. However, one can envision more complex, or wider domain scopes that would lead to even more complex diagrams. In such cases the modeller would have to resort to some of our proposed tool support solutions for the model building process, but in these situations the resulting model slices could have a relative high complexity themselves. We have argued throughout this work that i*'s successive refinement algorithm can help us infer less obvious properties of the domain.

However when diagrams get to the level of complexity we have mentioned, the inference process, if taken too far, can lead to results that have a lower analysis value.
The two model slicing concepts that we presented so far are defined as views abstracting only the elements that are relevant for an analysis question. From our practical experience we have observed that elements that are linked to the slice originating point through a long chain of weak links tend to be less relevant for the analysis than the ones who are linked more directly.

Elements whose contribution to/from others is less significant can give us grounds for further simplification for our model slices based on the relative cohesion between them and the slice focal point. We will define the strong and weak links concepts from an evaluation perspective and then we will introduce the model levelling concept as a way of analyzing the cohesion between the intentional elements in the model slice, that allows the modeller to perform further simplifications of the underlying model if needed.

5.1 The link types of the i* modelling framework

The i* modelling framework has a very rich semantics in both intentional element and link types. Such rich semantics are needed for representing accurately complex organizational systems. The strength of the various contribution and dependency links is different and we will attempt to achieve a categorization based on it, for the purpose of our analysis. We will use the label propagation rules of i*'s qualitative evaluation algorithm as a basis for selecting the category for the various types of links. The i* links are divided into five categories: contribution links, correlation links, decompositional links, actor relation links and dependencies. Figures 5.1-1 to 5.1-2 illustrate the above mentioned link categories. However for the type of analysis we are conducting we will need a different categorization approach.

The contribution and correlation links are very similar in terms of semantics and strength levels and therefore our categorization will apply identically to both.

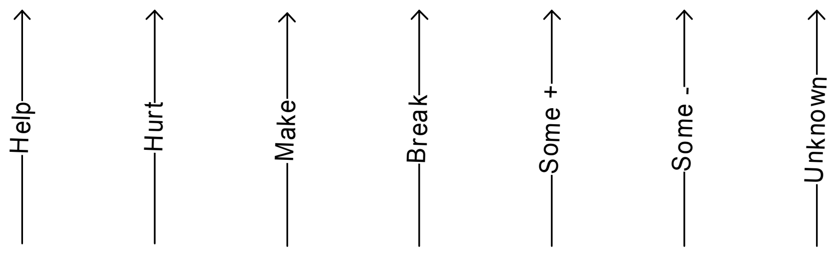


Figure 5.1-1 The contribution links in the i* modelling framework

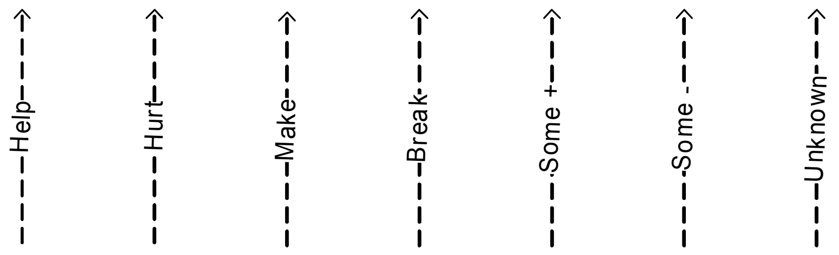


Figure 5.1-2 The correlation links in the i* modelling framework

The decompositional links represent a special category of intentional links in terms of evaluation. When performing the evaluation for an ancestor node that has several incoming decompositional links, one has to evaluate the satisficing of all its descendents before assigning a label to the parent node. Figure 5.1-3 illustrates the decompositional links that are part of the i* modelling framework.

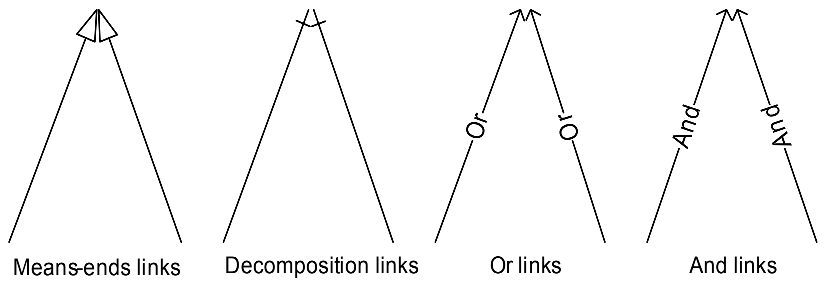


Figure 5.1-3 The decompositional links in the i* modelling framework

The actor relation links are used for further refining of intentional actor types. Their use facilitates concepts such as: generalization, specialization, encapsulation and inheritance. The actor relation links are not used for evaluation purposes and therefore will not be included in our subsequent analysis. Figure 5.1-4 illustrates the actor relation links of the i* modelling framework.

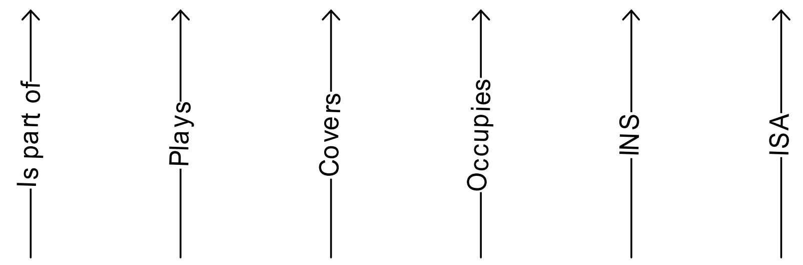


Figure 5.1-4 The actor relation links in the i* modelling framework

5.2 Strong and Weak links

The richness of the i* notation, allows the modeller to express several levels of confidence for the effects that an intentional element has on the others. For example if a modeller links two elements through a make contribution link, he expresses confidence in the fact that the achievement of the first element leads to the achievement of the other. Similarly, when considering task decompositions the modeller knows exactly which steps need to be taken in order to satisfice the parent task. Since in some cases there is no clear cut criteria for achieving some goals, or several things can contribute positively towards achieving them, the modeller can express those effects by using help or hurt contribution links or correlation links if the result is deemed to be a side-effect.

A first step in creating a strength-based categorization of links has been taken by the development of the i* qualitative evaluation algorithm. We will use its label propagation rules as a basis for our categorization. The authors of the NFR and i* frameworks have defined the following strength based ordering of evaluation labels [49], which we will use for our subsequent definitions:

✓ = ✗ ≥ ✓• = ✗•

In defining a strong link we have to distinguish between contribution and decompositional links given their different evaluation patterns. For contribution links the resulting evaluation label depends only on the originating node label and the contribution type, while for decompositional links the result depends on the labels of all its descendents. The first situation is illustrated in Figure 5.2-1

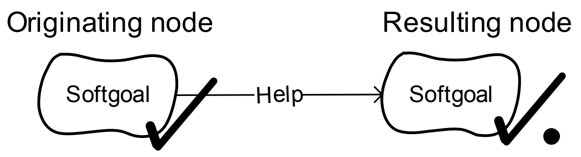


Figure 5.2-1 Propagation of labels through contribution links

Strong contribution link:

Definition: A contribution link L is a strong link if it propagates the originating node label O either unchanged or negated, where O ∈ { ✓, ✗ }.

Weak contribution link:

Definition: A contribution link L is a weak link if it lowers the strength of the originating node label for all the values of O, where O ∈ { ✓, ✗ }.

Since there are no degrees of weakness for elements labelled with ✓• or ✗• we have reduced the possible set of values for O to ✓ and ✗.

Applying the above mentioned definitions and the propagation rules of i*'s qualitative evaluation algorithm, we will create a categorization of intentional links.

Table 5.2-1 presents a summary of the label propagation rules and was adapted from [55]

Originating Node Label	Contribution Type						
	Make	Break	Help	Hurt	Some+	Some-	Unknown
	Resulting label						
✓	✓	✗	✓•	✗•	✓•	✗•	?
✓•	✓•	✗•	✓•	✗•	✓•	✗•	?
✗•	✗•	✓•	✗•	✓•	✗•	✓•	?
✗	✗	✓•	✗•	✓•	✗•	✓•	?

Table 5.2-1 The label propagation rules of i*'s evaluation algorithm

By analyzing the cells in Table 5.2-1 we can conclude that the *Make* and *Break* links are strong while the *Help, Hurt, Some+, Some-* and *Unknown* links are weak.

Decompositional links

In the case of decompositional links, the resulting node label depends on the labels of all its descendents. In the case of the *Means-Ends* and *Or* links the resulting node label is the maximum label of all its descendents, while for *Decomposition* and *And* links the resulting node label is the minimum label of all its descendents. The authors of the NFR framework define the ordering of evaluation labels in [49]. Adapting their ordering according to the evaluation labels of the i* modeling framework results in:

✓ ≥ ✓• ≥ ?• = ⚡ ≥ ✗• ≥ ✗ [55]

We will illustrate the labelling results with examples:

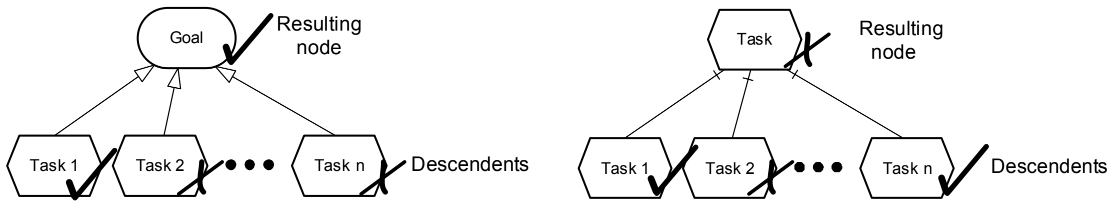


Figure 5.2-2 Propagation of labels through decompositional links

As we can see in Figure 5.2-2 the resulting label of the *Goal* element represents the maximum label of all its descendents, while the resulting label of the *Task* element represents the minimum label of all its descendents.

Thus we can perceive the decompositional links both as strong and weak links in the setting we have created so far. In our examples there is one link that propagates its originating node label while the rest could be considered irrelevant for the result. An unequivocal definition would have to take into account all descendent labels and can only be expressed for satisficeability (in the case of *Means-Ends* and *Or* links) or deniability (in the case of *Decomposition* and *And* links).

Since our concern is with assessing how strong the connection between the slice focal point and the rest of the intentional elements is, we will consider that all decompositional links are strong. We are using this approach since the results of our analysis will be used for further filtering of the diagrams we prefer taking an inclusive approach that preserves more elements in the diagram.

Dependency links

Dependency links can be considered strong links since they are propagating the originating node label unchanged. Thus we can apply the strong links definition, which we presented for contribution links, for dependencies. Figure 5.2-3 illustrates the label propagation procedure through dependencies.

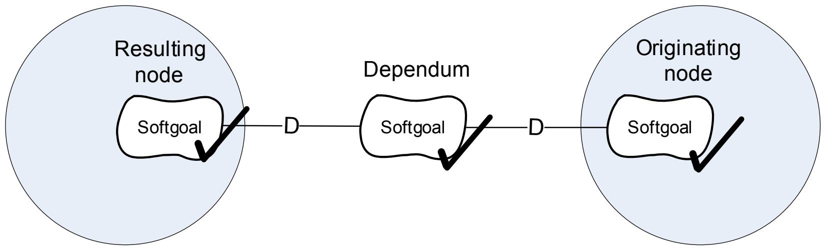


Figure 5.2-3 Propagation of labels through dependencies

We conclude our analysis on strong and weak links by summarizing our classification results in Table 5.2.-2. We will use Table 5.2-2 as a basis

Link	**Strength**
Help	Weak
Hurt	Weak
Make	Strong
Break	Strong
Some +	Weak
Some -	Weak
Unknown	Weak
And	Strong
Or	Strong
Means-ends	Strong
Decomposition	Strong
Dependency	Strong

Table 5.2-2 The strength based classification of i* intentional links

We will use this categorization as a basis for defining the model levelling concept as a useful concept in analyzing the cohesion of the model and application domain as well and reducing the complexity of the model slice.

5.3 The model levelling concept

Both of the model slicing concepts we have presented start from a modeller focal point of interest and trace through their descendents or ancestors in order to obtain a simplified view of the model containing only the relevant information for the analysis question at hand.

The purpose of the model slice is to include all the elements that are affected by a design decision (bottom-up model slice) or all the elements that affect a certain client goal (top-down model slice). However not all of the elements inside a model slice are equally relevant to the analysis question at hand. The notion of analysis relevance can be broken down into strategic relevance and directness. The strategic relevance component allows a prioritization between client objectives and concerns. However, a clear prioritization of goals based on their strategic relevance may not be achieved without a significant involvement of the client in the design process.[56]

The directness property is important when the mechanism used for creating the analysis view is inference or transitivity. When we construct model slices we consider that the effects of operationalizations propagate towards high-level goals through several levels of indirectness, depending on the domain properties.

As we have seen in the previous section, links can be categorized into weak and strong links from the evaluation perspective. From our experience when intentional elements are connected through a long chain of weak links to a design decision, the effects of that decision on them are inconclusive or tenuous and the validity of their evaluation results is questionable. In the evaluation procedure each weak link in the contribution chain is viewed as one that lowers the strength of the contribution to/from the starting intentional element. The evaluation algorithm only allows two levels of strength in its labelling. (i.e. satisficed and weakly satisficed, denied and weakly denied.). Thus the algorithm does not reflect how long the evaluation chain is and how "far" is the design choice conceptually from the other elements it affects.

By performing a levelling of the model slice we can study the cohesion of the intentional elements in the model slice as well as allow the modeller to further simplify the resulting model slice by performing a level based filtering.

The first part of the analysis can be used to verify the directness of the effects to/from the element of interest. We will label the slice originating node with level 1 and then label its descendents/ancestors with levels 2 to n. The higher the level of an intentional element, the less direct its connection to the slice originating intentional element will be.

The purpose of the model levelling process is to offer a way to analyze the cohesion of the intentional elements in the diagram and to safeguard the reasoning process against questionable evaluation results, coming from an inference procedure taken too far. If the number of levels in a model slice is relatively small, then the elements inside the slice are strongly connected among themselves. This will indicate a strong cohesion of the elements and the need to carefully consider each of them when conducting the analysis. Conversely, a high number of levels will indicate that some intentional elements are very indirectly related to the analysis question at hand and that further simplification of the model slice is possible.

Depending on the properties of the domain and the intrinsic complexity of the models, the two slicing concepts we have introduced can be sufficient to reduce the model complexity for analysis purposes. By applying the model levelling concept the requirements engineer simplify the model slices even further, by performing level-based simplifications of the diagrams if needed.

5.4 Model Levelling methodology

We will describe the levelling methodology and illustrate its usefulness for conducting analysis with a few examples. The model levelling algorithm can be applied on the two type of slices presented in Chapters 3 and 4. The algorithm allows the modeller to visually identify the level of each intentional element and to perform level based simplifications of the slice if needed.

5.4.1 Algorithm implementation

We will employ a breadth first approach in order to assign levels to the intentional elements. The algorithm should be performed on an existing slice obtained using one of the slicing algorithms presented in Chapters 3 and 4. Since there are 2 types of model slices we will use two distinct helper methods to deal with each of them. These methods are presented in Figures 5.4.1-2 and 5.4.1-3.

We will use the same representation based on the Node and Edge classes that we described in Chapters 3 and 4. The levelling method takes in a vector parameter which represents the slice on which the algorithm will be performed and returns a vector containing the model slice along with its corresponding levels. We present the implementation of the model levelling algorithm in Figure 5.4.1-1.

```
/**Method that implements the model leveling algorithm*/
*@param The model slice on which the leveling is performed
**/
Vector levelSlice(Vector Slice)
             Queue myQueue = new Queue();
             //initialize the starting point of the slice with level 1
             slice_start_point.setLevel(1);
             //enqueue the starting point
             myQueue.enqueue(slice_start_point);
             while (!myQueue.isEmpty()){
             //remove first element in queue and level his direct
             //descendents or ancestors
                    Shape temp = myQueue.dequeue();
                    if (sliceType == bottom_up)
                           levelDirectAncestors(temp,Slice,myQueue);
                    elseif(sliceType == top_down)
                           levelDirectDescendents(temp,Slice,myQueue);
             }
             //return resulting slice with levelling
             return Slice;
       }
```

Figure 5.4.1-1The slice levelling algorithm

```
/**Method that levels the direct ancestors of the current element based on
*their link types*/
*@param temp The current element
*@param bottom-up_slice The slice we are operating on
*@param Queue The element queue for breadth first traversal
**/
void levelAncestors(Shape temp,Vector bottom_up_slice, Queue myQueue) {
       //get the outgoing links of temp
       Vector outgoing_links = temp.getOutgoing_Links();
       //set level for direct ancestors of temp
       for (int i = 0; i < outgoing_links.size(); i++) {
             Link link = outgoing_links.elementAt(i);
             Shape ancestor = link.getDestination();
             //compute level for the direct ancestor
             int level = temp.getLevel();
             if (link.getType() in weak_links)
                    level++;
             //if the ancestor was not visited or we found a
             //more direct path to it
             if((ancestor.getLevel()==Shape.default_level)
             ||(ancestor.getLevel()>level)) {
                   //set the level of the ancestor
                   ancestor.setLevel(level);
                   //enqueue the ancestor;
                   queue.add(0, ancestor);
             }
       }
}
```

Figure 5.4.1-2 Helper method that computes the level for the direct ancestors of a node

The method in Figure 5.4.1-2 computes the level for the direct ancestors of a node. Since diagrams often contain circularities and sometimes there are several paths linking the originating slice node to other intentional elements we will take an inclusive approach in assigning the level values. If there are several paths to an intentional element its assigned level will correspond to the path with the highest degree of directness.

```
/**Method that levels the direct descendents of the current element based on
*their link types*/
*@param temp The current element
*@param bottom-up_slice The slice we are operating on
*@param Queue The element queue for breadth first traversal
**/
void levelDescendents(Shape temp,Vector slice, Queue myQueue) {
             //get the incoming links of temp
             Vector incoming_links = temp.getIncoming_Links();
             //set level for direct descendents of temp
             for (int i = 0; i < Incoming_links.size(); i++) {
                   Link link = Incoming_links.elementAt(i);
                   Shape descendent = link.getSource();
                   //compute level for the direct descendent
                   int level = temp.getLevel();
                   if (link.getType() in weak_links)
                         level++;
                   //if the descendent was not visited or we found a
                   //more direct path to it
                   if((descendent.getLevel()==Shape.default_level)
                   ||(descendent.getLevel()>level)) {
                         //set the level of the descendent
                         descendent.setLevel(level);
                         //enqueue the descendent;
                         queue.add(0, descencent);
```

```
                            }
                    }
}
```

Figure 5.4.1-3 Helper method that computes the level for the direct descendents of a node

5.5 Model levelling examples

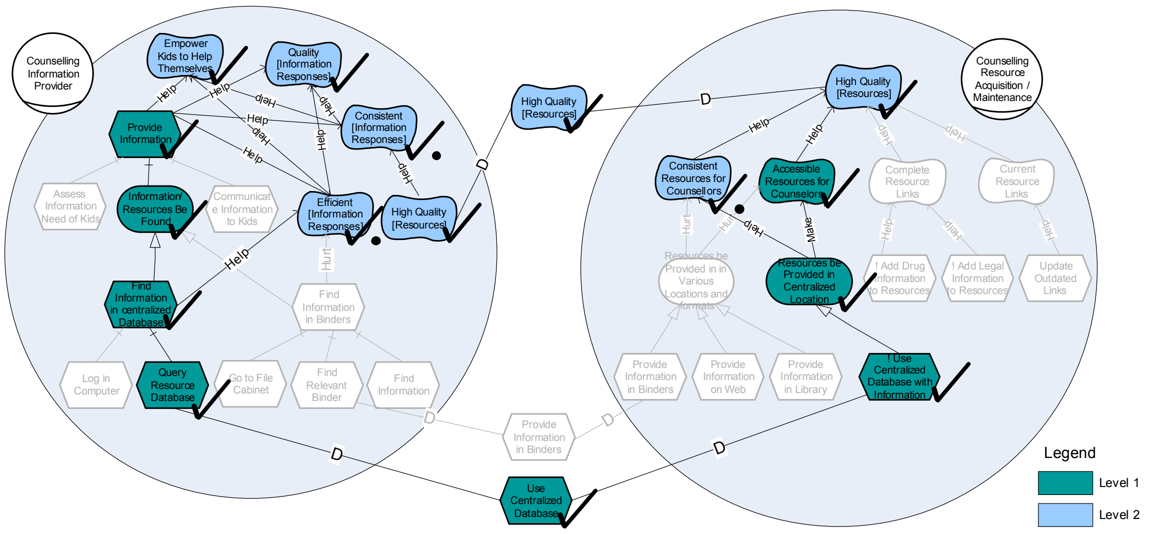
5.5.1 Cohesion analysis using the model levelling concept

We will illustrate the model levelling algorithm on the example we used to illustrate the bottom-up slicing technology in Chapter 3. Since those models were small and simplified, given the presentability size constraints, we expect the analysis to indicate a high level of cohesion between intentional elements and a small number of levels.

These models are sufficient to illustrate how the model levelling algorithm works but its application is only warranted on complex models such as the ones we have generated for the KHP project. On such models the levelling algorithm can yield interesting insights about the degree of cohesion of intentional elements as well as filtering alternatives for the diagrams.

We will apply the model levelling algorithm on the bottom-up model slices for the centralized information source and the physical binder alternatives. The explanations of the internal rationale of the two actors are presented in Chapter 3.

The models in Figure 5.5.1-1 and 5.5.1-2 illustrate the results of the levelling algorithm on the two model slices.



Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Consistent [Information Responses]
Provide Information
Assess Information Need of Kids
Information Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
Help
Hurt
D
High Quality [Resources]
Provide Information in Binders
Use Centralized Database
Counselling Resource Acquisition / Maintenance
High Quality [Resources]
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information
Make
Legend
Level 1
Level 2

Figure 5.5.1-1 Bottom-up model slice with levelling for the *!Use Centralized Database with Information* task

As expected the results indicate a high cohesion of the intentional elements in the two diagrams since we only have two levels. This illustrates the fact that there is a very strong connection between the choice of information resource provided to the counsellors and the quality of their responses to the kids' information inquiries.

The levelling starts at the *!Use Centralized Database with Information* and *Provide Information in Binders* tasks which are assigned level 1 and proceeds by assigning levels to the other intentional elements based on the type of link through which they are connected.

This analysis indicates that the choice of information resource is very important and should be carefully considered proposing changes to the available logistics of the counsellors.

Such a high degree of cohesion indicates that a level-based filtering is not recommendable here since it would eliminate elements that are very directly affected by this design choice.

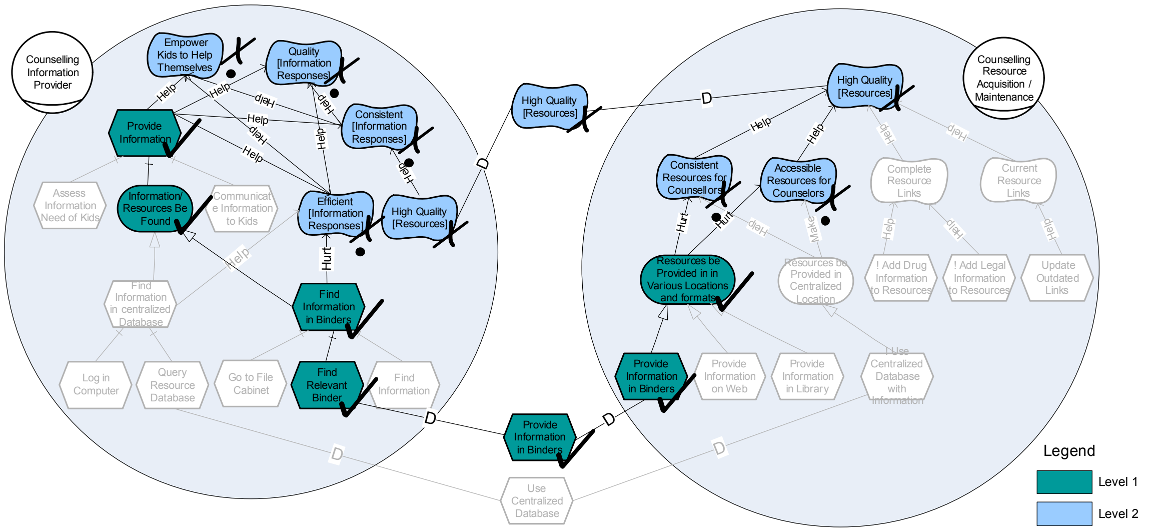

Counselling Information Provider
Empower Kids to Help Themselves
Quality [Information Responses]
Consistent [Information Responses]
Provide Information
Assess Information Need of Kids
Information/ Resources Be Found
Communicate Information to Kids
Efficient [Information Responses]
High Quality [Resources]
Find Information in centralized Database
Find Information in Binders
Log in Computer
Query Resource Database
Go to File Cabinet
Find Relevant Binder
Find Information
High Quality [Resources]
Provide Information in Binders
Use Centralized Database
Counselling Resource Acquisition / Maintenance
High Quality [Resources]
Consistent Resources for Counsellors
Accessible Resources for Counselors
Complete Resource Links
Current Resource Links
Resources be Provided in in Various Locations and formats
Resources be Provided in Centralized Location
! Add Drug Information to Resources
! Add Legal Information to Resources
Update Outdated Links
Provide Information in Binders
Provide Information on Web
Provide Information in Library
! Use Centralized Database with Information
Help
Hurt
Make
D
Legend
Level 1
Level 2

Figure 5.5.1-2 Bottom-up model slice with levelling for the *Provide Information in Binders* task

We can see by comparing the two models that the information alternatives and their corresponding levels are similar in scope. In particular the high-level softgoals of the two intentional actors are labelled identically for the two design alternatives with one exception.

As we can see in the model in Figure 5.5.1-1 that the *Accessible Resources for Counsellors* is labelled with level 1 while in the model from Figure 5.5.1-2 it is labelled with level 2.

The expressed purpose of a centralized information resource for counsellors is to ensure better accessibility to information. Rather then searching for information stored in various places (i.e. library, binders, web, etc) the counsellors could access instantaneously the information they needed if such a centralized system was available. Such a system would also improve the consistency of the information resources but it cannot guarantee it. In order to ensure the consistency of the information resources additional steps need to be taken. The strong connection between this alternative and accessibility is highlighted in the model levelling results through the assignment of level 1 (indicating a strong level of relevance for the analysis question) to the *Accessible Resources for Counsellors* softgoal.

In the second model slice, corresponding to the information binder alternative, we do not have a similar situation. The information binder alternative is not one that was engineered to ensure either accessibility or consistency of the information resources. It was employed because it was affordable and coupled with other information resource alternatives it had positive effects on the quality of information resources. Thus this alternative has a very direct connection to reducing the costs and a less direct connection to the accessibility and consistency of resources. This is illustrated in the model levelling results presented in Figure 5.5.1-2 by the fact that the *Accessible Resources for Counsellors* and the *Consistent Resources for Counsellors* softgoals are assigned level 2. The strong connection of this alternative to minimizing expenses is shown by the labelling of the *Reduce Costs* softgoal from the KHP actor, which is not shown here for simplicity purposes, with level 1.

Thus we can see that the model levelling concept, which labels automatically intentional elements based on the above described criteria, has enabled us to analyze the cohesion of the model slices and to acknowledge the importance of the information resource design decision for the quality of information responses. The obtained results are intuitive and consistent with our understanding of the organizational environment.

5.5.2 Slice filtering using the model levelling concept

When reasoning over a domain with high intrinsic complexity, such as the one of KHP, the modeller faces an inherent danger of trusting the model and evaluation results as absolutely correct, 100% of the time. The model should be viewed more as an elicitation and reasoning validation tool than an oracle and results that do not appear intuitive should always undergo a reality check. We have argued previously that the bottom-up model slice concept provides us with a convenient view that represents the scope of a certain design alternative.

We have also argued that the i* framework favours the effect inference process which can reveal interesting and less obvious properties of the domain. When the inference process is carried too far the evaluations results seem to indicate effects that are tenuous at best. We will illustrate such a case with an example from our project.

One of the technologies that have been considered for usage as a web counselling alternative is a web bulletin board with delayed moderation. This would allow kids to post and reply to messages and the contents would be verified periodically by counsellors to remove inappropriate content. This alternative has the advantage of allowing kids to interact with each other and socialize but presents the grave danger of potential abuse of the service by pedophiles and the receiving of bad advices from other persons.

In order to compare this technology with other alternatives we have created a bottom-up slice for the *Kids Use Bulletin Board with Delayed Moderation* task from the KHP Web Services agent. Upon inspection of the resulting slice we have seen that the effects of this design choice tend to propagate to other intentional actors and that the effects vary in relevance for different intentional elements. This has led us to produce the model levelling classification.

In this particular instance we have noticed that the choice of service seems to affect some goals inside the corporate sponsors of the company. One such example is the *[Positive Image to Employees]* softgoal from the Corporate Sponsor actor. Companies that sponsor charities are focused on maintaining a positive image to their employees. We will trace the effects of the *Kids Use Bulletin Board with Delayed Moderation* task from the KHP Web Services agent on the *[Positive Image to Employees]* softgoal from the Corporate Sponsor actor by using the path concept as described in Chapter 3. The path and its corresponding evaluation results are extracted from the bottom-up model slice for the Bulletin Board with Delayed Moderation technology which cannot be included here given the size constraints. We illustrate the resulting path in Figure 5.5.2-1

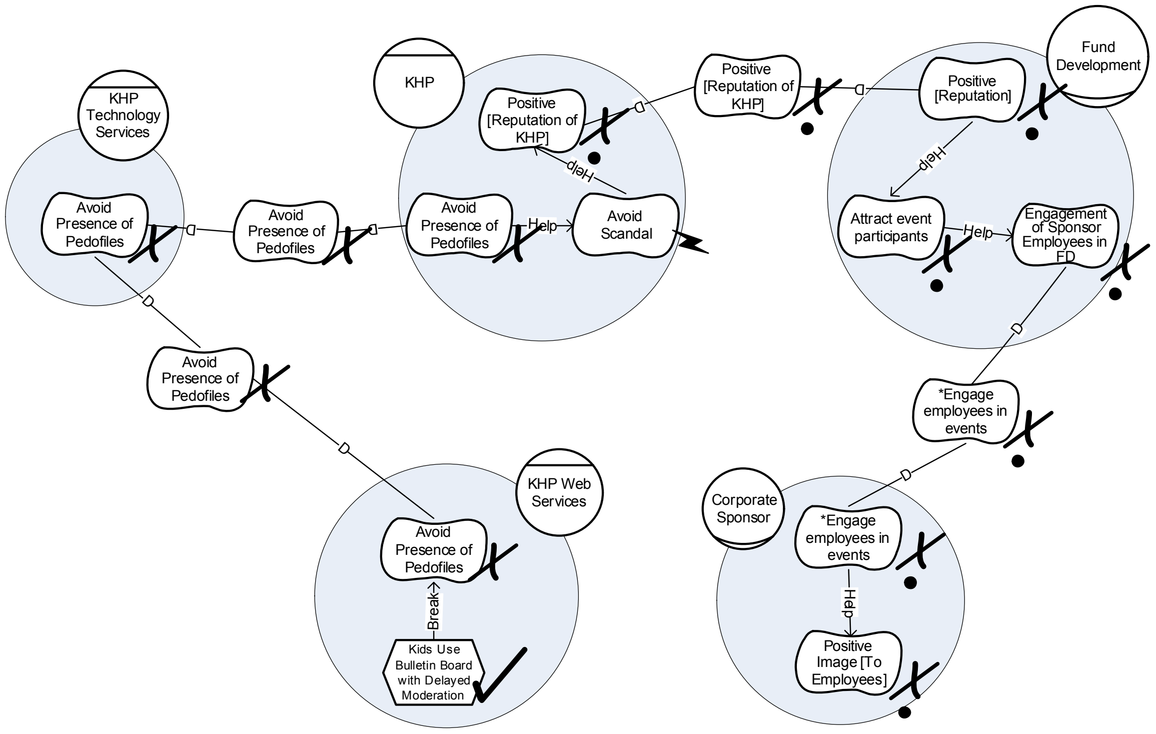

KHP Technology Services
Avoid Presence of Pedofiles
Avoid Presence of Pedofiles
KHP
Positive [Reputation of KHP]
Help
Avoid Presence of Pedofiles
Help
Avoid Scandal
Positive [Reputation of KHP]
Positive [Reputation]
Fund Development
Help
Attract event participants
Help
Engagement of Sponsor Employees in FD
Avoid Presence of Pedofiles
*Engage employees in events
KHP Web Services
Avoid Presence of Pedofiles
Break
Kids Use Bulletin Board with Delayed Moderation
Corporate Sponsor
*Engage employees in events
Help
Positive Image [To Employees]

Figure 5.5.2-1 Path through the model slice to illustrate tenuous inference results

As we have mentioned before, this technology does not provide effective mechanisms to avoid the presence of the pedophiles as potential abusers of the system and threats to the safety of kids. Thus choosing this alternative results in a denial of the *Avoid Presence of Pedophiles* softgoal.

This softgoal propagates upwards from the KHP Web Services agent to the KHP agent through the KHP Technology Services agent. In the KHP agent the denial of the *Avoid Presence of Pedophiles* softgoal results in a negative effect on the *Avoid Scandal* softgoal which in turn affects negatively the *Positive [Reputation of KHP]* softgoal. Following the propagation of effects all the way to the Corporate Sponsor role leads us to conclude that the choice of this service by KHP will affect negatively the image that the employees of corporate sponsors have about their own company. This conclusion suggests that KHP can influence the internal opinion of corporate sponsor employees based on the service choices that it makes. This assertion is tenuous at best and does not constitute valid criteria for choosing a service by KHP.

When conducting the reasoning process on a model slice, the modeller has to take into account the directness of the relation between the design alternative and various client goals. In order to do this the requirements engineer can employ the model levelling concept, which illustrates graphically how directly intentional elements are affected by a certain design decision. If the nature of the relation between some intentional elements and the slice originating node is very indirect, then a level based filtering can be applied on the model slice.
Since printability on a regular size page sheet is one of the challenges that arise with the increase in model and slice complexity we are unable to show the model levelling concept applied on a regular size model slice here. We will illustrate the levelling and filtering algorithms on the path example illustrated in Figure 5.5.2-1. This is sufficient to convey the general ideas of the levelling concepts although they were not conceived for application on model paths.

The purpose of the model-slicing techniques is to provide a model view that contains the elements that are affected by a design decision (bottom-up model slice) or that have an effect on client goals (top-down model slice). However, depending on the domain peculiarities and modelling style, the analyst can be faced with situations in which some elements in the resulting slices are deemed to fall outside the scope of the analysis questions driving the slicing process. In such cases the modeller can decide after applying the levelling algorithm and conducting a careful analysis of the resulting levels that level-based filtering may be applied. The act of deciding which levels should be filtered out is a highly cognitive task that involves human judgment and thorough knowledge of the domain. This step cannot and should not be automated. Also, the configuration and number of levels in a model depends heavily on the modelling style and domain properties. The model levelling concept should be viewed as an aid in the simplification process but not as a deciding factor. Other approaches based on cognitive evaluation of the relative importance of intentional elements could also be used for simplification purposes.

We presented the path example in Figure 5.5.2-1 as a way of illustrating the fact that sometimes the inference process can be carried too far when constructing complex models and their corresponding slices. The effect of the chosen technology on some softgoals of the other intentional actors was questionable. We will apply the model levelling algorithm in order to analyze the cohesion of the intentional elements within the path and to analyze whether level based filtering would be recommendable. The results of the levelling algorithm are presented in Figure 5.5.2-2.

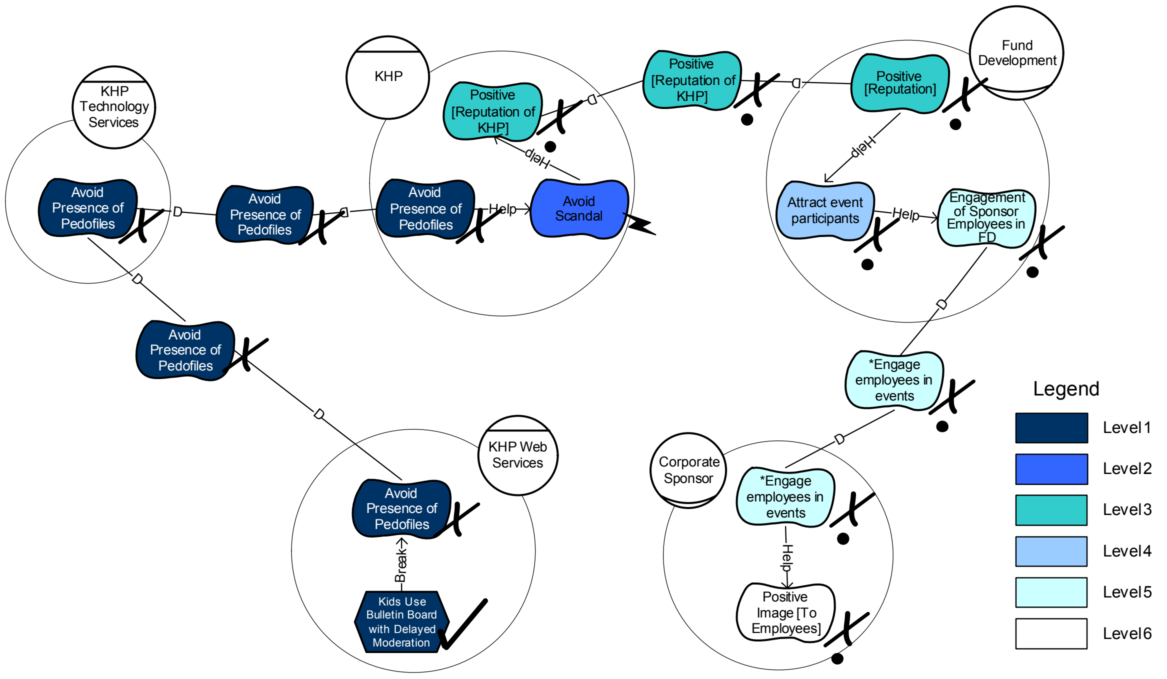

KHP Technology Services
Avoid Presence of Pedofiles
Avoid Presence of Pedofiles
KHP
Positive [Reputation of KHP]
Help
Avoid Presence of Pedofiles
Help
Avoid Scandal
Positive [Reputation of KHP]
Positive [Reputation]
Fund Development
Help
Attract event participants
Help
Engagement of Sponsor Employees in FD
Avoid Presence of Pedofiles
*Engage employees in events
Legend
Level1
Level2
Level3
Level4
Level5
Level6
KHP Web Services
Avoid Presence of Pedofiles
Break
Kids Use Bulletin Board with Delayed Moderation
Corporate Sponsor
*Engage employees in events
Help
Positive Image [To Employees]

Figure 5.5.2-2 Path example with levelling

As we can see in Figure 5.5.2-2 the number of levels is considerably bigger in an example based on our complex models than in the examples illustrated in Figure 5.5.1-1 and 5.5.1-2. The typical number of levels in our individual technology slices ranged between 6-10 levels. This indicated potential for further simplification of our model slices which would make the reasoning process easier. By analyzing the intentional elements we can observe that elements from level 4 and above drift away from the area of influence of the design technology choice. The effect of the technology choice on fundraising event participation, and corporate sponsor employee engagement and perception is tenuous and its use as choice criteria for a technology alternative would be less valid.

On the other hand, we can see that this technology has a direct impact on the protection of kids from pedophiles and a transitive one on the reputation of the organization.

These effects are direct and we can certainly agree that if this technology was chosen and pedophiles would use the service to get to kids the company's reputation would be affected. After these considerations we can conclude that we can filter out intentional elements labelled with level 4 and bigger from the diagram presented in Figure 5.5.2-2. The results of applying the filtering process are presented in Figure 5.5.2-3.

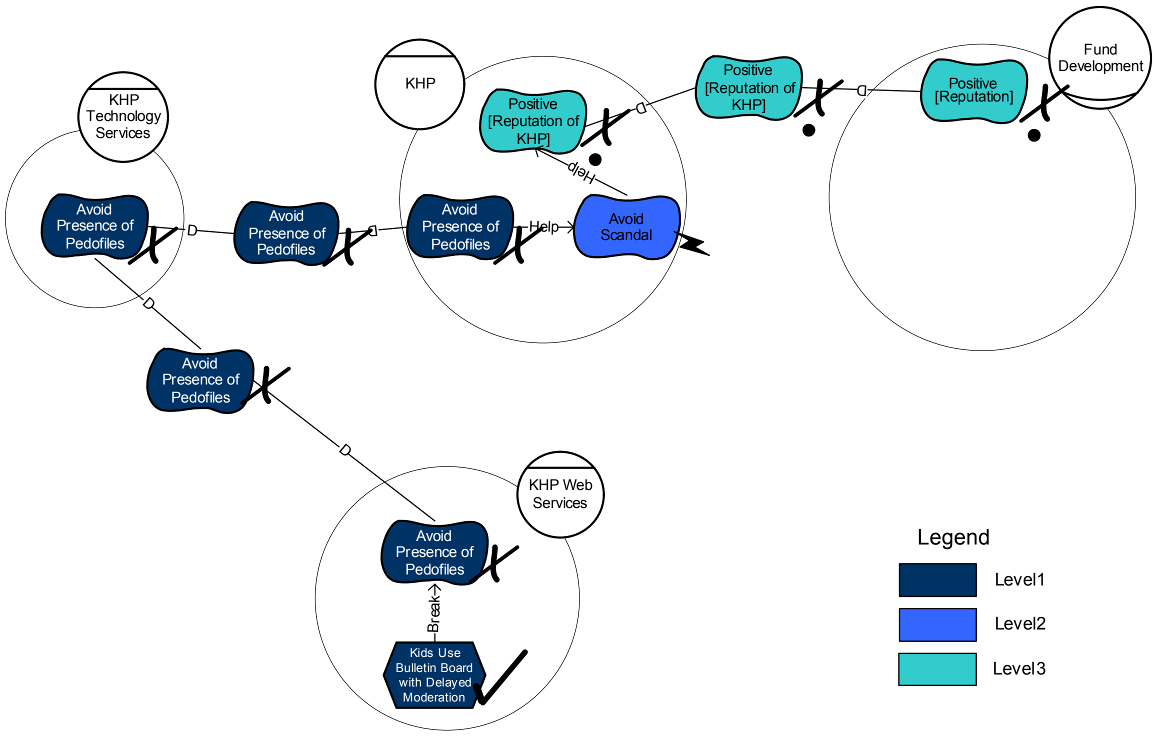

KHP Technology Services
KHP
Fund Development
Positive [Reputation of KHP]
Positive [Reputation of KHP]
Positive [Reputation]
Avoid Presence of Pedofiles
Avoid Presence of Pedofiles
Avoid Presence of Pedofiles
Avoid Scandal
Help
Help
D
Avoid Presence of Pedofiles
KHP Web Services
Avoid Presence of Pedofiles
Break
Kids Use Bulletin Board with Delayed Moderation
Legend
Level1
Level2
Level3

Figure 5.5.2-3 Filtered path example with levelling

As we can see this concept does not have the same model complexity reducing properties as the model slicing concepts. This result is expected given the fact that model slices are significantly simpler than our comprehensive models. Even though the number of level has been reduced to half the number of filtered elements is relatively small. This indicates that a high percentage of the elements in the model slices are relevant to the design alternative chosen and that these elements should be taken into consideration when conducting the analysis.

5.6 Conclusions

We have illustrated the fact that intentional elements appearing in i* model slices have different degrees of relevance for the analysis question that drives the slicing process.

While the model slicing techniques reduce the complexity of comprehensive models by filtering out elements that are irrelevant for the analysis question at hand, there is still a potential for simplifying the resulting slices based on the relative relevance of the remaining elements for that particular question. The model levelling concept proposes a prioritization of the elements of a model slice based on the directness of their effects on/from the slice originating element.

A high cohesion between the intentional elements of a model slice, which translates in a small number of levels for that slice, indicates that all the remaining elements are very important and should be considered carefully for the analysis question at hand. Conversely, a slice that has a high number of levels indicates the potential for further simplification through a level based filtering.

During our practical experience on the KHP project we have observed that the model levelling algorithm can also help improve the quality of the underlying models. Differences in the assigned levels for elements whose importance was deemed to be similar indicated inconsistencies the models such as missing contributions or disproportionate refinements of various client goals.

Such errors or inconsistencies were very difficult to observe while operating on the complex comprehensive models.

Thus we have illustrated the use of the model levelling concept as an aid in conducting the analysis on model slices and a potential tool for further reducing the diagram complexity and improving overall model quality.

6. Contributions and future work

6.1 Contributions

This work represents an important step towards addressing the modelling scalability issues that large scale applications pose.

Model development, like software development has a lifecycle of its own. In our current work we identify three major phases in a model's lifecycle that present scalability challenges. Given the widely different nature of scalability challenges particular to each phase, a "one size fits all" solution is neither recommendable nor needed.

We argue that each phase should be addressed separately and customized solutions addressing specific scalability challenges should be designed for all of them.

One of the main results of this research is the development of a set of concepts that address the reasoning scalability challenges that very complex diagrams raise. Our approach introduces systematic model breakdown mechanisms based on the topology of i* diagrams and the analysis questions that the modeller is trying to answer.

The analysis results and proposed concepts in our current work have been developed and validated against a very complex real-life project. This increases our confidence in the validity and applicability of our research in successfully addressing scalability problems of very complex organizational settings.

The model slicing concepts are suited for a wide variety of analysis questions as illustrated in Chapters 3 and 4 and they provide a convenient, fully automated abstraction mechanism for diagram decomposition and filtering. We propose an iterative analysis approach, focused on one concern at a time and illustrate the applicability of this methodology to comparative and goal-oriented analysis. Besides allowing for a better resource management and prioritization of concerns, this approach provides the selection criteria for deciding which analysis questions warrant further investigation.

As illustrated in Chapters 3 and 4, typically each analysis question requires the information and reasoning support of only a fraction of the overall comprehensive model. The two slicing methodologies automatically extract the portion of the model relevant for the analysis question, while filtering the rest.

Besides the topological model structure criteria we propose the directness and analysis relevance criteria for addressing scalability. Anchored in the qualitative evaluation research, the model levelling concept offers a convenient way to assess the relevance of various concerns for analysis questions and to safeguard the reasoning process against questionable evaluation results.

The proposed concepts streamline the model supported analysis process, advancing i* in a more practical and easy to use stage.

Besides the analysis scalability support concepts that we developed, we have highlighted the importance of modelling methodology and adequate tool support in addressing complexity related challenges.

The handful of guidelines, live examples and concepts offered in this work, have hopefully provided a rich enough set of methodologies to advance the modelling of large-scale applications.

6.2 Future directions

Our work continues research on the modelling scalability issues done in [32] with an emphasis on the challenges encountered in the analysis phase.

We highlight the importance of modelling methodology and adequate tool support in addressing scalability challenges. In the methodology realm further work is required before a clear and complete set of modelling guidelines can be developed.

During the course of the Strategic Requirements Analysis for Kids Help Phone project we have collected a series of requirements that will be integrated in the next generation of tool support for the i* modelling framework. A prototype tool was developed during the course of the project to facilitate and validate the usage of slicing mechanisms. Integration of our scalability concepts with the current tool support research developments [57] is required to create a consistent and unified effort for tool support development.

We have seen throughout this thesis that partitioning models according to the link direction alone, can result in significant model scale-downs.

However, domain knowledge may contribute to generic guidelines from a different dimension. Applications coming from closely related domains may possess similar characteristics that can be generalized and reused. By coupling knowledge catalogues with a consistent naming convention for intentional elements we may be able to provide a deeper level of analysis in assessing strategic relevance and context based diagram simplifications. This line of future direction is considered important given the fact that the distributed nature of the i* framework is appropriate for modeling applications which are very rich in domain knowledge.

In the original i* development work [53], Yu provides three degrees of dependency strength: open (uncommitted), committed, and critical. Our model levelling concept may be adapted to provide analysis support for actor commitment and vulnerability.

Another set of promising concept comes from the aspect-oriented programming research community. Concern documentation, cross-cutting or scattered concern identification may provide useful ideas and mechanisms for addressing modelling scalability.

Bibliography

[1] Nuseibeh, B.A., Easterbrook, S.M., “Requirements Engineering: A Roadmap”, In A. C. W. Finkelstein (ed): “The Future of Software Engineering“ , IEEE Computer Society Press., 2000.

[2] Easterbrook, S.M., Lecture slides - CSC 2106: Requirements Engineering, 2004.

[3] Rumbaugh, J., Jacobson, I., Booch, G., “The Unified Modeling Language Reference Manual”, Addison-Wesley, 1998.

[4] J. Mylopoulos, J., Chung, L., Yu, E., “From Object-Oriented to Goal-Oriented Requirements Analysis”, Communications of the ACM, 42(1), January 1999, pp. 31-37.

[5] Lamsweerde, A.v., “Goal-Oriented Requirements Engineering: A Guided Tour”, Proceedings, 5th IEEE International Symposium on Requirements Engineering (RE'01), Toronto, August, 2001, pp. 249-263.

[6] Yu, E., “Agent Orientation as a Modelling Paradigm”, Wirtschaftsinformatik 43(2), April 2001, pp. 123-132.

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

[8] Yu, E., “Agent-Oriented Modelling: Software Versus the World”, Agent-Oriented Software Engineering AOSE-2001 Workshop Proceedings, Springer Verlag, 2001, pp. 206-225.

[9] Vliet, H.v., “Software Engineering: Principles and Practice” , Willey, 2000.

[10] Boehm, B., “Software Risk Management”, IEEE, 1989.

[11] Feldman, P., Miller, P., “Entity Model Clustering: Structuring a Data Model by Abstractions.” The Computer Journal, 29(4), 1986, pp. 348-360 \.

[12] Baddeley, A.D., “The Magical Number Seven: Still Magic After All These Years?” Psychol. Rev. 101 (2), 1994, pp. 353–356.

[13] Miller, G.A., “The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity For Processing Information”, Psychol. Rev. 63, 1956, pp. 81–97.

[14] Moody, D.L., Shanks, G., “Improving the Quality of Data Models: Empirical Validation of a Quality Management Framework”, Journal of Information Systems, 28(6), 2003, pp. 619 – 650.

[15] Moody, D.L., Benczúr, A., Demetrovics, J., Gottlob, G. (Eds.): “Cognitive Load Effects on End User Understanding of Conceptual Models: An Experimental Analysis”, ADBIS, LNCS 3255, 2004, pp. 129–143.

[16] Sweller, J., “Cognitive Load During Problem Solving: Effects on Learning”,Cognitive Science, 12, 1988, pp. 257-285.

[17] Sweller, J., “Cognitive Load Theory, Learning Difficulty and Instructional Design”, Learning and Cognition, 4, 1994.

[18] Kalyuga, S., Ayres, P., Chang, S.K., Sweller, J., “The Expertise Reversal Effect”, Educational Psychologist, 38 (1), 2003, pp. 23-31.

[19] Yu, E., Cysneiros, L.M., “Agent-Oriented Methodologies-Towards a Challenge Exemplar” in Proc of the 4 Intl. Bi-Conference Workshop on AOIS , Toronto, 2002.

[20] Rana, O. F., Stout, K., “What is scalability in multi-agent systems?”, Proceedings of the 4th International Conference on Intelligent Agents, June 2000 , ACM, pp. 56-63.

[21] Woodside M.,”Scalability Metrics and Analysis of Mobile Agent Systems”, Proceedings of Workshop "Infrastructure for Scalable Multi-Agent Systems" at Agents 2000, Barcelona, 2000.

[22] Brataas, G., Hughes, P., “Exploring architectural scalability”, WOSP, California, USA, 2004, pp. 125-129.

[23] Dictionary of Computing at http://wombat.doc.ic.ac.uk, Last viewed Dec 2004.

[24] Menzies, T., Easterbrook, S.M., Nuseibeh, B.A., Waugh, S. "An Empirical Investigation of Multiple Viewpoint Reasoning in Requirements Engineering", Proceedings, Fourth IEEE International Symposium on Requirements Engineering (RE'99), Limerick, Ireland, June 7-11, 1999.

[25] Nuseibeh, B.A., Easterbrook, S.M., Russo, A., "Making Inconsistency Respectable in Software Development ", Journal of Systems and Software, 58 (2), 2001, pp. 171-180.

[26] Easterbrook, S.M. "Negotiation and the Role of the Requirements Specification", In P. Quintas (ed) Social Dimensions of Systems Engineering: People, Processes, Policies and Software Development, London: Ellis Horwood, 1993, pp. 144-164.

[27] Lamsweerde, A.v., "Goal-Oriented Requirements Engineering: from System Objectives to UML Models to Precise Software Specifications," Tutorial Presented at ICSE'03, Portland, May 2003, 159 p.

[28] Bubenko, J.A., Persson A, Stirna J. ,"User Guide of the Knowledge management Approach Using Enterprise Knowledge Patterns", Stockholm (Sweden), Department of Computer and Systems Science, Royal Institute of Technology, 52 p.

[29] Chung, L., Nixon, B., "Dealing with non-functional requirements: three experimental studies of a process-oriented approach.", Proceedings of the 17th international conference on software engineering, Seattle, WA, April 1995, pp. 24-28.

[30] Moody, D.L., "Comparative Evaluation of Large Data Model Representation Methods: The Analyst's Perspective", Proceedings of the 21st International Conference on Conceptual Modeling, Tampere, Finland, 2002, pp. 214-231.

[31] Simsion, G.C., "A Structured Approach To Data Modelling", The Australian Computer Journal, August 1989.

[32] You, Z., "Using Meta-Model-Driven Views to Address Scalability in i* Models", M. Sc. thesis, 2004, University of Toronto, 238p.

[33] Koubarakis, M., Mylopoulos, J., Stanley, M., Borgida, A.," Telos: Features and Formalization.", Department of Computer Science, University of Toronto. Report nr KRR-TR-89-4, Feb. 1989, 84 p.

[34] Kids Help Phone website at http://www.kidshelpphone.ca/beingthereforkids/about-khp/ourservices/counsel/khp_counsel.html, Last visited Jan 2004.

[35] Moody, D.L., Flitman, A., "A Methodology for Clustering Entity Relationship Models - A Human Information Processing Approach", Proceedings of the 18th International Conference on Conceptual Modeling, Paris, France, 1999, pp. 114-130.

[36] Klir, G.J., "Architecture of Systems Problem Solving", Plenum Press, New York., 1985.

[37] Organization Modelling Environment (OME), [Tool]. Knowledge Management Lab, Bell University Labs, University of Toronto. Available at: http://www.cs.toronto.edu/km/ome/, Last viewed Jan 2005.

[38] OME 3 architecture [Online Documentation], Knowledge Management Lab, Bell University Labs, University of Toronto. Available at: http://www.cs.toronto.edu/km/ome/docs/architecture/architecture.html, Last viewed Jan 2005.

[39]Microsoft Office Visio 2003 information page, Available at http://www.microsoft.com/office/visio/prodinfo/default.mspx, Last viewed Jan 2005.

[40] Easterbrook, S.M., "Resolving Requirements Conflicts with Computer-Supported Negotiation.", In M. Jirotka & J. Goguen (eds.) Requirements Engineering: Social and Technical Issues, 1994, pp. 41-65, London: Academic Press.

[41] Easterbrook, S.M., "Handling Conflict Between Domain Descriptions With Computer Supported Negotiation", Knowledge Acquisition: An International Journal, Vol. 3, No 4, 1991, pp.255-289.

[42] Moody, D.L., “Dealing with Complexity: A Practical Method for Representing Large Entity Relationship Models” (PhD Thesis), Melbourne, Australia: Department Of Information Systems, University of Melbourne, 2001.

[43] Lindland, O.I., Sindre, G., Solvberg, A., “Understanding Quality In Conceptual Modelling”, IEEE Software, 11 (2), March 1994, pp. 42-49.

[44] Lauesen, S., Vinter, O., Preventing Requirement Defects, In Proceedings of the Sixth International Workshop on Requirements Engineering: Foundation for SoftwareQuality (REFSQ'2000), Stockholm, Sweden, June 5-6 2000.

[45] Standish Group, “The CHAOS Report into Project Failure”, The Standish Group International Inc., 1995, Available at: http://www.standishgroup.com/sample_research/chaos_1994_1.php, Last viewed Jan 2005.

[46] Standish Group, Unfinished Voyages, The Standish Group International Inc., 1996 Available at http://www.standishgroup.com/sample_research/unfinished_voyages_1.php, Last viewed Jan 2005.

[47] Horwitz, S., Reps, T., Binkley, D., “Interprocedural slicing using dependence graphs.”, In Proceedings of the ACM SIGPLAN 88 Conference on Programming Language Design and Implementation, (Atlanta, GA, June 22-24, 1988), ACM SIGPLAN Notices 23, 7, 1988, pp. 35-46.

[48] Lamsweerde, A.v., “Goal-Oriented Requirements Engineering: A Guided Tour”, Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE'01), Toronto, August, 2001, pp. 249-263.

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

[50] Liu, L., Yu, E., Mylopoulos, J., Security and Privacy Requirements Analysis within a Social Setting. International Conference on Requirements Engineering (RE’03), Monterey, California, September 2003.

[51] Yu, E., Mylopoulos, J.,"Understanding "Why" in Software Process Modelling", Analysis, and Design, Proceedings of 16th International Conference on Software Engineering, May 16-21, 1994, Sorrento, Italy, pp. 159-168.

[52] Yu, E., Mylopoulos, J., "Why Goal-Oriented Requirements Engineering", Proceedings of the 4th International Workshop on Requirements Engineering: Foundations of Software Quality (8-9 June 1998, Pisa, Italy). E. Dubois, A.L. Opdahl, K. Pohl, eds. Presses Universitaires de Namur, 1998. pp. 15-22.

[53] Yu, E., "Modelling Strategic Relationships for Processing Reengineering" [dissertation], Toronto (ON): Department of Computer science, 1994, University of Toronto, 124 p.

[54] Dardenne, A., Lamsweerde, A.v., Fickas, S.,"Goal-Directed Requirements Acquisition", in The Science of Computer Programming 20, 1993.

[55] Horkoff, J., "I* evaluation tutorial", July 2004, pp. 5-11.

[56] Luke, R., and others, "The Promise and Perils of a Participatory Approach to Developing an Open Source Community Learning Network", Participatory Design Conference 2004. Vol. 1: Artful Integration: Interweaving Media, Materials and Practices, New York, The Association for Computing Machinery, 2004, pp. 11-19.

[57] Open OME development project, Available at http://sourceforge.net/projects/openome, Last viewed Jan 2005.