Evaluating Goal-Oriented Analysis in the Domain of Enterprise Information Systems

Malihe Tabatabaie, Fiona A.C. Polack, and Richard F. Paige[1]

Department of Computer Science, University of York, YO10 5DD, York, UK
{malihetb, fiona, paige}@cs.york.ac.uk

Abstract. Enterprise information systems (EIS) development raises specific challenges related to the scale and complexity of the business processes that must be supported. Techniques for managing the challenges of scale and complexity are often based on *modelling,* and are targeted at different parts of the EIS development process. This paper presents the results of an evaluation (based on EIS for health care) of a number of goal-oriented analysis approaches in the domain of EIS, and concludes that existing approaches lack process guidance and, in particular, guidance on the evaluation of the results of goal analysis. We present an analysis of the problem of evaluation, and proposes a scenario-based approach to evaluate the results of goal analysis.

1 Introduction

This paper summarises the results of assessing goal-oriented analysis approaches in the context of *Enterprise Information Systems (EIS)* development, and identifies a scenario-based evaluation approach that can be applied to the results of goal-oriented analysis. The aim of the assessment is to investigate which approach is more suitable for our example of EIS. We identify some limitations and strengths of each approach, as a preliminary stage to develop a process model for goal analysis of EIS (see [26]).

The term EIS is widely used in academia, industry and business markets. Moreover, other related terms are widely used, such as small and medium enterprises, enterprise portals, enterprise solutions [5], enterprise architecture [22], enterprise services [20] etc. Our study on the definition and domain of enterprise and EIS [25, 24] illustrates the common uses of this term in relation to businesses. We observe in particular that a key characteristic of EIS is having multiple *linked business processes* that create complexity [24]. Developing this type of system is costly, and failure in IT projects generally have a substantial effect on the whole of society. One of the examples of EIS that affects the UK is its national health system (NHS) IT project. Our study of this type of EIS has allowed us to categorise some of the common challenges of EIS [6]:

- lack of clear links between EIS and the organisation's key priorities;
- lack of agreed measures of success;
- lack of effective engagement with stakeholders;
- lack of understanding of and contact with the supply industry at senior levels in the organisation;
- lack of ability to visualise complex software systems.

An enterprise is a collection of small or medium-sized businesses (SMEs) and partners that operates as a single organisation. An EIS is a software system that integrates the business processes of organisation(s) to improve their functioning [24, P. 24] [25]. The challenges of developing and deploying EIS vary across enterprises, with many challenges related to scale and complexity. Some factors that create complexity in EIS are [25]:

- size and growth of information technology (IT), information systems (IS) and the organisation itself;
- the interactions between different IT and IS;
- the involvement of many different parties in the constructions and use of IT and IS; and,
- the ever-increasing rate of organisational and social change.

It is these factors that give rise to the failures of communication and understanding identified as challenges [6] above. Techniques that can help us directly address these challenges in communication may help to reduce the risk of developing and deploying EIS.

Communication difficulties are typically apparent and critical to overcome in the early stages of the information systems engineering lifecycle, i.e., when trying to identify clear requirements, policies, objectives, and regulatory constraints. We thus propose a *goal-oriented* approach to facilitate communication and understanding in the development of EIS. At this stage we cannot expect business people to design their own software systems but we can provide the ability for them to identify and structure their goals; similar actions could be taken on the part of the software systems developers as well. Through a structured process of analysis and modelling, EIS developers are led towards an understanding of the priorities of the enterprise organisation and its organisational goals. Analysis requires identification of a sufficiently-complete sets of goals, by engagement with the different group of stakeholders. Engagement with senior staff can help to create understanding between developers and organisational seniors. Structuring the goals of EIS leads towards a vision of the EIS-to-be and systems-as-is, and can address the challenge of EIS structural visualisation.

The need to identify enterprise-level goals led us to review well-known *goal-oriented approaches (GOA)* to requirements elicitation and engineering. Some of the GOAs that we reviewed have been applied to examples of EIS. However, they have all been applied by experts in the approach itself (instead of domain experts); as well, these approaches generally are not associated with a clear process of how to extract and refine the goals and goal structures, which is a critical concern for EIS where overlapping, contradictory, and possibly inconsistent goals may lead to enterprise-wide problems. The reasons for this may be because there are various types of EIS with very different requirements; it may therefore not be possible to develop a generic process that can satisfy all of them. Approaches tailored for specific EIS domains may be more profitable. Nevertheless, when there is no process the approach is likely to be dependent on experts, which limits the usage of the approach.

To address these difficulties, we are developing and implementing a process based on application of KAOS heuristics, and best practices taken from other goal-oriented

approaches. We have also tailored this process specifically to the kinds of EIS considered in our experiments so far: that is, EIS to support large governmental enterprises in the domain of health care. We have not yet applied our process to other kinds of EIS in other domains.

The paper is structured as follows. Section 2 presents an example of an EIS for managing stroke care across England. The example is used as the basis for our assessment of goal-oriented analysis approaches. Section 3 presents a general review of goal-oriented approaches. This leads to a more detailed review of the four most well-known goal-oriented approaches, and summarises their shortcomings and strengths, thus allowing us to propose improvements for EIS analysis. Section 4 explains the challenges of evaluating the results of a goal-oriented analysis, and proposes a scenario-based approach that is sufficient for the health care EIS domain in which we are working, and which deals with some of the shortcomings from Section 3. We then conclude with pointers for future work.

2 EIS Domain: Stroke Care

We have been basing our recent research on a specific example of EIS in the health care domain. The example is for an EIS for managing patient care, particularly for care of people who have had a stroke. This section introduces the example and justifies why it is a suitable EIS example; the next section presents an assessment of goal-oriented approaches that we have applied to capturing and analysing the goals of enterprises in this domain.

"Stroke has been described as 'an earthquake in the brain'. The shock waves of stroke can leave a lasting and profound impact on how people move, see, speak, feel or understand their world" [27]. Stroke is the third biggest killer in the UK [27]. Stroke Care is not only about supporting people who have just had a stroke, but also "helping people of all ages establish and maintain a lifestyle that promotes good health and minimises the risk of ever having a stroke" [27].

Stroke care is management of medical and care procedures related to stroke within organisations such as NHS in UK. Stroke care is based on different business and organisational processes relating to governmental and other organisations as well as hospitals. There are different views over the goals and functionalities of stroke care system from different groups of stakeholders, for instance, politician, health specialists, IT specialists, and different groups of society such as patient and their family. This organisational and medical system requires a strong and clear connection between different business processes and partners. Therefore, stroke care is an example of EIS. There are different documents and proposals for this system that show the large group of society members and organisations who are interested in achieving the goals of Stroke Care. After reviewing a group of the documents, "National Stroke Care" [27] from UK Department of Health [27] became a primary resource in this research: this report lists and describes non-functional requirements and soft goals, and strategies of stroke care management. The other key sources of our information about Stroke Care are:

- Interview with IT expert;
- An IT proposal from a local health team [18];

– Public documents of an American version of stroke care systems [21];
– Public documents of electronic patient record systems in Denmark [8];

In some aspects, because of lack of available documents, realistic assumptions have been made.

3 Goal-Oriented Approaches

In this section we review goal-oriented approaches to requirements engineering, and then present an assessment of different leading approaches when applied to EIS for stroke care.

There are at least 15 distinct GOAs [10] from areas of computer science such as artificial intelligence [9], software assessment [29], requirements engineering [17, 3], safety argumentation [13, 14].

Considering the definition of EIS, the aim of this research, and the internal characteristics of each GOA, four approaches related to requirements engineering and argumentation were identified for detailed analysis: GSN [13], KAOS [17], GBRAM [2], and i^* [3]. KAOS and GBRAM have approaches for defining and refining the goals of an organisation; KAOS is the better-documented approach, and has been used in many successful projects. i^* is a well-known and successful approach that has been applied to health care [1]; it claims to encourage the involvement of stakeholders in requirement analysis, and to help the developers to achieve a deep understanding of the domain. GSN is widely used for presenting the structure of arguments (for example in relation to safety certification), and has been applied in research contexts to requirements analysis [7]. GSN structures an argument using goals, for presentation to stakeholders; in this way it can be a bridge between developers and stakeholders. GSN is used in practice in the domain of safety cases and has been used in many large projects.

The following critique of these methods is based on their literature and on applying each of the approaches to an example of EIS development (Section 2).

3.1 Critique of GSN [23]

Whilst GSN provides a powerful structuring notation, it lacks a supporting process. GSN is typically used to summarise the evidence of safety from traditional safety analysis techniques. GSN does not provide a goal identification method, which compares unfavourably with other approaches: for example, GBRAM goal identification uses instance scenarios [2].

The basic GSN guidelines that exist [15, 13] have no requirement to document sources of goals. Despite its use to structure an argument over evidence, GSN does not enforce traceability between goals and sources (though there are examples of acceptable types of goal reference [15], and textual references are often included in element labels). This hampers the use of GSN for goal analysis and complicates the process of applying changes to the goal structure, even in the context of safety-case maintenance for which it is designed [28]. There are patterns and suggestions to address the maintenance problem but no mechanism of validation or enforcement to clarify the source of the goals structure [30, 14].

Because of its origins in safety-case argumentation, the GSN notation includes useful features such as modularity, but cannot easily express contradiction or priority of goals. In EIS, it is invariably the case that the enterprise assigns priorities to its goals, and it is generally the case that some identified goals will conflict (excellence vs cost-saving is a common instance). Similarly, because of its origins, each GSN requirement is expressed as a claim over evidence and the goals are derived from requirements – in most analytical uses of GOA [11, 2, 17], goals are expressed before identifying the requirements and are a source for identifying the requirements.

3.2 Critique of i^*

The motivation of i^* is to understand why a business process is the way it is, rather than just to describe the requirements of a business process. i^* supports process modelling and reengineering [31]. i^* was chosen for review because of: its focus on understanding of the business environment [31] and the domain; its encouragement of stakeholders involvement in requirements analysis; its visual virtual notation for communication between stakeholders; and, finally, its successful use in health care systems [1]. These positive attributes in addition to the tool support influence our eventual process.

However, like [1], we find that i^* requires too much detailed information early in the design process. Whilst the level of detail may exist in business-specific IS development, it is not appropriate for analysis of enterprise goals. This problem applies in some part to the other GOAs, but is particularly problematic in i^*. The main question before using i^* is whether the developers have the required amount of detailed information for allocating the dependency relationships and whether they need the detailed level of goal structure that includes implementation information.

3.3 Critique of GBRAM

[2] presents GBRAM as an approach for analysing, identifying, and classifying the goals, agents and stakeholders without providing detailed instruction. The detail level of GBRAM is more appropriate than that of i^*: goals are a "logical mechanism for identifying, organizing and justifying software requirements" [2]. GBRAM provides a top-down approach to refining and structuring the goals; it addresses documenting and identifying goals, as well as the issue of knowing when the goals are adequately specified. The disadvantage of GBRAM is its lack of a generic process for analysis: GBRAM is helpful in starting the collecting and refining of goals, and influences our eventual process, but the lack of documentation does not allow us to use the method as-is for EIS.

3.4 Critique of KAOS

The KAOS is an tool-supported approach to goal-based requirement engineering that has a comparatively well-defined process, with many conceptual similarities to GBRAM. It has both a top-down approach and a bottom-up approach to identifying and refining goals. In this sense, it is more detailed and complete than GBRAM; KAOS' approach

is also at an appropriate level for EIS, and in early stages does not seem to require the sort of detailed information of the organisation in analysis that i^* seeks. The rigorous approach to goal definition in KAOS can cause difficulties for stakeholders who are outside IT or IS development, but adds its own value in respect of concrete definition. KAOS methodology is actively evolving and is well-documented; it is applied in many industrial cases. As in other approaches, the process of applying KAOS is not well-defined even though examples illustrates that it exist at some level [17, P. 502], but the advantage of documentation makes it the basis for our EIS process.

In conclusion, firstly, we find a general lack of clear process guidance for GOAs, which makes the approaches heavily reliant on the expertise of practitioners. It is arguable that there is insufficient world-wide expertise to support the needs of EIS development, and thus more guidance in the application of approaches is necessary. Secondly, a key limitation of all the reviewed GOAs is the lack of evaluation techniques. Evaluation, again, relies heavily on expertise. Whilst this can work in practice, more systematic consideration of evaluation is needed. KAOS suggests evaluation using bottom-up and top-down approaches to evaluate the goal models created - asking a "why question" from child goals to their parents and vice versa [17]. We are interested in ways of evaluate the results of goal-oriented approaches.

4 Evaluating a Goal-Oriented Analysis

A goal-oriented analysis must be evaluated to ensure that its results are valid and appropriate. In particular, how we can ensure that the collected goals and the goal structure is sufficient to allow EIS developers to start with the next phase? This section addresses how to evaluate the results of a GOA in the domain of our health care EIS.

The four goal-oriented approaches discuses above are heavily reliant on expert review. It is clear that expert review has an essential role in evaluating the final results in this area of research, however we believe that other evaluation options could allow the developers to evaluate the analysis results, complementing and focussing expert review.

In GSN there are four steps to check the results:

- Step 1 - Argument comprehension check
- Step 2 - Well-formedness checks
- Step 3 - Expressive sufficiency checks
- Step 4 - Argument criticism and defeat

The checklist approach and criteria based approach are beneficial in general cases and the obvious type of problems that can happen frequently in different projects, but may miss special issues that can happen in some cases [12]. To cover special cases we propose to use scenario-based evaluation. Scenarios are used for collecting data in requirement engineering and goal-oriented approaches, nevertheless to the best of our knowledge, not in the evaluation phase of GOA. However, the technique is very successful in software architecture domain [4]. Scenarios are easy to create and understand, they are simple to use and inexpensive, hence they are suitable to use in many fields [4]. Scenarios have the ability to define the goals of actors [19] thus they can be easily adapted for goal-oriented approaches. They can be detailed and narrowed down

as much as required. Scenarios "can be made deliberately incomplete to help developers cope with uncertainty" [19].

All these characteristics lead us to use this approach for evaluation in addition to other possible techniques such as expert review. For this research *a scenario is a short statement describing the goals and expectations of one of the stakeholders from the system.*

The review of current categories of scenarios illustrates two different groups that could be applied for goal-oriented approaches: one from software architecture point of view [4], and one from the goal-model point of view [12].

The goal-model point of view defines different roles for scenarios: *Introduction*, *planing*, *automation*, *operationalisation*, *evaluation* [12, P. 312].

An architectural point of view is given by the ATAM approach [4], which presents the following scenario types:

Use case scenario: "Use case scenario describes a user's intended interaction with the completed running system". "use case scenario expresses a specific stakeholder's desires" [4, P. 54]. This scenario could be used in our case of scenarios for goal-model evaluation.

Growth scenario: "Represent typical anticipated changes to a system" [4, P. 54]. Changes and evolution is one of the features in EIS. Scenarios that represent the functionalities of this type of systems should cover this aspect and feature of EIS.

Exploratory scenarios: "the goal of these scenarios is to expose the limits or boundary conditions of the current design, exposing possibly implicit assumptions" [4, P. 54]. This type of scenario also could be used for goal model.

To investigate which categories are suitable for evaluating the results of EIS goal models, we generated some scenarios that could cover some of the primary functionalities of the Stroke Care system. After generating the scenarios, we reviewed the criteria to evaluate which set of scenarios are most appropriate. The results illustrate that categories in [12] (*Introduction*, *planing*, *automation*, *operationalisation*, *evaluation*) are detailed and focused on different stages of software development such as planing, automation, and operation. However, at this stage it is more practical to generate scenarios for the categories given in ATAM: some of the categories in [12] cannot be generated at the early stages of development, and the categories are not always helpful. Therefore, allocating the scenarios to ATAM categories are more applicable in the Stroke Care example. We tailored the description of ATAM categories to match with the research domain as follows:

- *Use case scenario* to describe the desires and goals of stakeholders from the running system.
- *growth scenario* to describe the anticipated goals if the system changes in future, or the scenarios that describe the possibility of goals that change in future.
- *exploratory scenarios* that describe the limits and boundary goals.

We define the requirements for the the scenarios in GOA and the style of presenting the scenarios. As the results of our study on approaches to use scenarios such as [16] the following process is extracted:

- Select a few high priority scenarios from one group each time
- Present each scenario as a sequence diagram
- Check if the agents are captured in the diagram
- Check if the goals are captured in the diagram
- Generate requirements if possible
- Generate expectations if possible
- Generate operations if possible

By applying this evaluation process to scenarios related to Stroke Care we addressed the following evaluation criteria:

- *Understandable*: The goals are understandable because at least one scenario justifies the condition under which the goal (s) could exist.
- *Correct*: The correctness of goal structure is checked against the scenarios.
- *Right level of detail*: evaluate the level of detail for goals by creating a sensible and understandable scenario out of the documented goal (s)
- *Unambiguous*: Detect ambiguity by presenting the ambiguous scenarios for goals.
- *Redundancy*: Address the redundancy if no rational scenario could justify the existence of goal.
- *Organised*: It address the structure and organisation of goal-structure by creating scenarios that includes several goals.

5 Conclusion and Future Work

Goal-oriented approaches to modelling and analysis can facilitate the development of EIS, particularly for health care. Facilitation comes from better structuring and explicit modelling of goals and points of contention amongst stakeholders. The review of goal-oriented approaches illustrates three different approaches that have been used in examples similar to our EIS for health care. A detailed review using an EIS example was used to analyse which approach(es) could beneficially be applied. The analysis illustrated that KAOS methodology is the most convenient approach for our example. However, this technique same as other goal-oriented approaches has limitations on evaluating the results. Therefore, we proposed and tested scenario-based evaluation to evaluate the results of goals. It is possible that stakeholders can understand the scenarios related to the goals more easily than the goal structure; hence, this technique not only can help the developers to do the evaluation but also helps to engage the experts with the process of evaluating more systematic.

Work is continuing on the testing and documentation of scenario-based evaluation. More work is needed in testing scenario-based evaluation and identifying and applying possible appropriate evaluation techniques.

References

1. Yuan An, Prudence Dalrymple, Michelle Rogers, Patricia Gerrity, Jennifer Horkoff, and Eric Yu. Collaborative social modeling for designing a patient wellness tracking system in a nurse-managed healthcare center. In *International Conference on Design Science Research in Information Systems and Technology*. ACM, 2009.

2. Annie I. Antón. Goal-based requirements analysis. In *IEEE International Conference on Requirements Engineering*, pages 136–144, 1996.
3. Jaelson Castro, Manuel Kolp, and John Mylopoulos. Towards requirements-driven information systems engineering: the tropos project. *Information Systems*, 27(6):365–389, 2002.
4. Paul Clements, Rick Kazman, and Mark Klein. *Evaluating Software Architectures: Methods and Case Studies*. Addison-Wesley, 2002.
5. Google. Enterprise Solutions [online]. [Accessed 3 March 2010] Available at: www.google.co.uk/enterprise/public_search.html, March 2010.
6. RAE-BCS Working Group. The challenges of complex IT projects. Technical report, The Royal Academy of Engineering, 2004.
7. Ibrahim Habli, Weihang Wu, Katrina Attwood, and Tim Kelly. Extending Argumentation to Goal-Oriented Requirements Engineering . In *Advances in Conceptual Modeling - Foundations and Applications: ER Workshops*, volume 4802 of *LNCS*, pages 306–316. Springer, 2007.
8. Eben Harrell. In Denmark's Electronic Health Records Program, a Lesson for the U.S. [online]. [Accessed 16 Sep 2009] Available at: www.time.com, April 2009.
9. Jun Hong. Goal recognition through goal graph analysis. *Journal of Artificial Intelligence Research*, 15:1–30, 2001.
10. Evangelia Kavakli and Pericles Loucopoulos. Goal driven requirements engineering: Analysis and critique of current methods. In John Krogstie, Terry A. Halpin, and Keng Siau, editors, *Information modeling methods and methodologies*, pages 102–124. IGI, 2005.
11. Evangelia Kavakli, Pericles Loucopoulos, and Despina Filippidou. Using scenarios to systematically support goal-directed elaboration for information system requirements. In *IEEE Symposium and Workshop on Engineering of Computer Based Systems*, pages 308–314. IEEE Computer Society, 1996.
12. Evangelia Kavakli, Pericles Loucopoulos, and Despina Filippidou. Using scenarios to systematically support goal-directed elaboration for information system requirements. In *IEEE Symposium and Workshop on Engineering of Computer Based Systems*, pages 308–314. IEEE Computer Society, 1996.
13. T. P. Kelly. *Arguing Safety – A systematic approach to managing Safety Cases*. PhD thesis, University of York, Department of Computer Science, 1998.
14. Tim Kelly. A systematic approach to safety case management. In *SAE 2004 World Congress*. Society for Automotive Engineers, 2004.
15. Timothy Kelly. A six-step Method for Developing Arguments in the Goal Structuring Notation (GSN). Technical report, York Software Engineering, Flixborough, UK, 1998.
16. Philippe B. Kruchten. The 4+1 view model of architecture. *IEEE Software*, pages 42–50, 1994.
17. Axel Van Lamsweerde. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley, 2009.
18. Leeds Team. Team notes: Research strategy on information systems for stroke care. Technical report, University of Leeds, August 2008.
19. Mary Beth Rosson and John M. Carroll. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Academic Press, 2002.
20. SAP. DEFINE ENTERPRISE SERVICES [online]. [Accessed 3 March 2010] Available at: www.sdn.sap.com/irj/sdn/define-es, March 2010.
21. Lee Schwamm, Arthur Pancioli, Joe Acker, Larry Goldstein, Richard Zorowitz, Timothy Shephard, Peter Moyer, Mark Gorman, Claiborne Johnston, Pamela Duncan, Phil Gorelick, Jeffery Frank, Steven Stranne, Renee Smith, William Federspiel, Katie Horton, Ellen Magnis, and Robert Adams. Recommendations for the establishment of stroke systems of care. *American Stroke Association*, 36:690–703, 2005.

22. SPARX SYSTEMS. Advanced Modeling & Design Platform [online]. [Accessed 3 March 2010] Available at:www.sparxsystems.com/, March 2010.
23. Malihe Tabatabaie. Applying gsn to stroke care. Technical report, University of York, 2009.
24. Malihe Tabatabaie, Richard Paige, and Christopher Kimble. Exploring the boundaries of enterprise information systems. In *Second York Doctoral Symposium on Computing*, volume YCS-2008-434, pages 27–34. Department of Computer Science, University of York, 2008.
25. Malihe Tabatabaie, Richard Paige, and Christopher Kimble. Exploring enterprise information systems. In Maria Manuela Cruz-Cunha, editor, *SOCIAL, MANAGERIAL, AND ORGANIZATIONAL DIMENSIONS OF ENTERPRISE INFORMATION SYSTEMS*, pages 415–432. IGI, 2010.
26. Malihe Tabatabaie, Fiona A.C. Polack, and Richard Paige. Kaos-b: Analysing eis architecture using kaos. 2010.
27. DH Stroke Policy Team. National Stroke Strategy. Department of Health: www.dh.gov.uk/en/Publicationsandstatistics/Publications/, 2007.
28. Michael Warren. Introducing systematic reuse into military ATC safety case management. Master's thesis, University of York, UK, 2004.
29. David M. Weiss, David Bennett, John Y. Payseur, Pat Tendick, and Ping Zhang. Goal-oriented software assessment. In *International Conference on Software Engineering*, pages 221–231. ACM, 2002.
30. Weihang Wu and Tim Kelly. Combining Bayesian belief networks and the Goal Structuring Notation to support architectural reasoning about safety. In *SAFECOMP*, volume 4680 of *LNCS*, pages 172–186. Springer, 2007.
31. E. Yu and J. Mylopoulos. Towards modelling strategic actor relationships for information systems development – with examples from business process reengineering. In *Workshop on Information Technologies and Systems*, pages 21–28, 1994.