Acta Ciencia Indica, Vol. XLII M, No. 2, 181 (2016)

GOAL PROGRAMMING MODEL FOR GENERATING AND EVALUATING ALTERNATIVES IN REQUIREMENT ANALYSIS

D. SARATHSENREDDY

Department of Mathematics, Anurag College of Engineering, Aushapur, Hyderabad (Telangana), India

AND

G. RAVINDER BABU

Department of Computer Science and Engineering, Trinity College of Engineering and Technology, Karimnagar (Telangana), India

RECEIVED : 22 June, 2016

Determining user requirements and generating alternative system solutions to meet these requirements are two critical steps in the requirement analysis phase of the system development life cycle. Much of the MIS research in the requirements analysis phase has been devoted to the topic of requirement determination and its verification. Alternative generation and evaluation is left, to a significant degree, to the judgment and expertise of an analyst. This paper proposes a multiple criteria decision making (MCDM) approach for generating and valuating alternatives when the user requirements are expressed in terms of certain operational criteria such as time, cost, risk, etc. These alternatives form the basis for the user to make the necessary trade-offs.

KEYWORDS : Generating, Evaluating Alternatives, Cash Management System, analysis.

INTRODUCTION

The last fifteen years have seen an increased emphasis on an early phase in Requirements Engineering (RE) when the focus is on stakeholders and their goals, rather than the system-to-be. *i** [20] is a modelling framework that supports modelling and analysis during this phase. According to *i** and an associated requirements analysis process (Troops [2]), one begins requirements acquisition by identifying stakeholders ("actors") and their goals. These goals are decomposed and delegated to other actors, thereby creating networks of delegations. The process ends when all initial ("root") goals can be fulfilled if all actors deliver on their delegations. One or more of these actors represent the system-to-be. The functional requirements for the system to- be are determined by all delegations to these system actors.

Exploring the space of alternative actor dependency networks is a difficult design task. This is so because such networks represent complex socio-technical systems where organizational, human and system actors depend on each other to fulfill root-level goals. Moreover, there are no generic criteria to guide the design process by determining whether a solution is good-enough, or even optimal. Our long-term objective is to develop such criteria and use them through tools and systematic design processes.

The purpose of this paper is to propose a framework for the automatic selection and evaluation of design alternatives. The framework supports both the generation and evaluation of alternatives. Specifically, the framework adopts multi-agent planning techniques and uses off-the shelf planning tools. Alternatives are evaluated with respect to individual interests of system actors (*i.e*. their own goals). Ideas from Game Theory [14] are used to determine whether an alternative is an equilibrium. In particular, an alternative is in equilibrium if no actor can do better with respect to its own goals by adopting a different strategy for delegating and accepting delegations. When combined together, these two steps support the designer/requirements engineer in selecting alternatives that are in equilibrium with respect to the local strategies of each actor. An early version of this idea is used in [3] to propose a framework to generate alternative designs for secure systems. This paper goes further by describing a prototype tool that generates alternatives, presents some experimental results, and also proposes evaluation techniques for alternatives based on game-theoretic notions. The process of the best alternative selection consists of the following steps:

1. Identify system and human actors, goals and their properties. Define goal decompositions and dependency relationships among actors.

2. For each actor identify criteria to evaluate alternatives.

3. Automatically explore the space of alternatives" on the upper level" to identify assignments of coarse-grained goals to actors.

4. Separately for each actor, automatically explore the alternative ways to satisfy the goals the actor was assigned at step 3. According to above identified evaluation criteria, select "the best" alternative for each actor. During this step, alternative refinements of coarse-grained goals and delegation dependencies among actors are explored.

5. Evaluate the combined solution consisting of alternatives identified at step 4. In case it does not satisfy one or several system actors (*e.g*. they are overloaded with respect to others), return to step 4 to search for another alternative. Ideally, the process stops after a number of iterations when the socio-technical structure is optimized enough to comply with the individual interests of the system actors. If no satisfactory alternatives can be generated at some step, the designer should return to steps 1 or 2, and revise either the initial structure, or the evaluation criteria.

DATA OF THE PROBLEM

This study was carried out in a MNC company located in Hyderabad during a feasibility study for the automation of its cash management system. The treasury department of the company is responsible for managing the firm's cash reserves, cash flow, and investments. Payments are made by the utility customers at many geographically dispersed district banks and this deposit information is transmitted to the treasury department the next day. The treasury used this information to make appropriate deposits and wire transfers among its member banks and the main bank, and to decide on certain short-term investments. The company, at the time of the study, incurred significant opportunity costs in managing its cash because of the lag between customer payments and investment decisions.

The primary objective in developing the new system was to obtain timely date on deposits to accelerate investment opportunities. However, management was also concerned with maximizing the success in undertaking this information systems project and keeping the overall cost of the project below the estimated opportunity costs incurred in one year. One other objective was to achieve the automation of the district ledger subsystem.

The project team formed to complete this study included: two operation clerks and a cash manager from the utility, and two graduate students and a professor from a university. Initial data about the system were obtained using interview techniques, and data flow diagrams were used to document the system and obtain feedback. The project meetings were held weekly in the early stages of the project and monthly in the later stages. The entire study took about nine months to complete and was presented to the management team which included the treasurer and two district office managers. While no MIS professionals were directly involved in the project team in the preliminary stages of the study (partly to get an unbiased view), a number of MIS application managers were actively consulted in the later stages, especially to help estimate the cost / time parameters.

An analysis of the current system resulted in the identification of the following six subsystems within the treasury department that relate to the task of cash management:

1. SS1 - Cash Reporting Subsystem
2. SS2 - Transfer Selection Subsystem
3. SS3 - Deposit Transfer Subsystem
4. SS4 - Wire Transfer Subsystem
5. SS5 - Bank Ledger Subsystem
6. SS6 - District Ledger Subsystem

The information shows the interaction among these subsystems in terms of the data they share and the files they control. The four objectives of the system were identified as: time, cost, risk and district ledger updating. The alternative designs of the 6 subsystems were identified as: current, *i.e.* manual; and altered, i.e. automated. Thus, a total of 64 (2^6) different configurations were possible. Using the methodology described earlier, several design alternatives were generated for management consideration. The required information is given below.

Table 1. Process Sequence

GOAL 1 : G1 ≤ 3.5 HOURS			
SS1 →	SS2 →	SS5 →	SS3
19.0/0.5	0.75/0.25	1.50/0.5	1.25/0.75
GOAL 2 : G2 ≤ 3.5 HOURS			
SS1 →	SS2 →	SS5 →	SS4
19.0/0.5	0.75/0.25	1.50/0.5	1.25/1.00
LEGEND: PROCESS TIMES UNDER CURRENT/ALTERED STATE			

GOAL PROGRAMMING MODEL

Goal 1: Satisfy Timeliness Requirement. The time-related goals called for reducing the processing times associated with deposits and wire transfers to no more than 3.5 hours. The paths that affect these goals and the estimates of the time spent in each subsystem on the path under the current and proposed designs and shown in Table 1. The time estimates under the current system were provided by the treasury department personnel and the estimates under the automated design were obtained from corporate systems personnel. The resulting time goals can be expressed as follows:

$$19X_{11} + 0.5X_{12} + 0.75X_{21} + 0.25X_{22} + 1.25X_{31} + 0.75X_{32} + 1.5X_{51} + 0.50X_{52} \leq 3.5$$

$$19X_{11} + 0.5X_{12} + 0.75X_{21} + 0.25X_{22} + 1.25X_{41} + 1.00X_{42} + 1.5X_{51} + 0.50X_{52} \leq 3.5$$

Table 2. Estimated Cost Parameters in Rupees

Development Cost					Operation Cost					
Cost Parameter	**Hardware**	**Software**	**Conversa-tion**	**Installa-tion**	**Training**	**Data Entry**	**Output**	**Sub-total**	**Discounted Operation Cost**	**Total**
C1	49000	5000	0	7000	10000	0	100	100	379.1	71379
C2	700	1500	200	4000	1050	6000	200	800	3032.8	10483
C3	700	500	0	600	200	200	50	250	947.8	2948
C4	700	700	200	1000	200	200	50	250	947.8	3748
C5	700	1000	800	1000	600	600	50	650	2464.2	6564
C6	700	500	800	1000	200	300	0	300	1137.3	4337
OR12	0	100	0	0	250	200	50	250	947.8	1298
OR16	0	100	0	0	250	200	50	250	947.8	1298
OR23	0	100	0	0	250	200	50	250	947.8	1298
OR24	0	100	0	0	250	200	50	250	947.8	1298
OR25	0	400	0	0	900	800	100	900	3411.9	4712

Table 3. Risk Assessment

Criterion	**Score Representing Likelihood of Success**						
	Weight	**SS1**	**SS2**	**SS3**	**SS4**	**SS5**	**SS6**
New hardware	0.20	30	05	05	05	05	05
New software	0.25	35	10	0	0	10	0
Familiarity of the organization with similar systems	0.20	50	20	20	20	20	20
Experience level of user/designer with similar systems	0.35	50	05	05	05	05	05
Total weighted score representing likelihood of failure		42.25	9.25	6.75	6.75	9.25	6.75

Goal 2: Minimization of Information System Cost. This objective minimizes the cost of altering the current design of subsystems to the new design. This cost consists of the one time investment cost and the recurring operating costs. The operating cost component included only the cost of data entry and output as they both result in direct cash outlay. The cost of running the computer (CPU costs), storage, and maintenance were ignored since these were treated as fixed overhead by the organization and were not allocated to each user application. Under a different cost allocation scheme, some of these costs may be explicitly included in the cost expression. Table 2 represents the cost of changing the design of each subsystem to the new automated design and the cost of its operation. The above costs were estimated under the assumption that the design of all other subsystems will remain unchanged.

In addition, the table also shows the possible cost savings (*e.g.*, OR_{12}) due to the overlapping use of resources by interacting subsystems (*e.g.*, 1 and 2). The target value of development cost was Rs. 50,000. The resulting cost expression is:

$$71379X_{12} + 10483X_{22} + 2948X_{32} + 3748X_{42} + 6564X_{52} + 4337X_{62} - 1298X_{12}X_{22}$$
$$- 1298X_{12}X_{62} - 1298X_{22}X_{32} - 1298X_{22}X_{42} - 4712X_{22}X_{52} \le 50{,}000.$$

Goal 3 : Minimize Risk of Failure. The application development managers were consulted to obtain the likelihood of failure of each subsystem under automation. They evaluated the likelihood of failure by assigning scores for each subsystem against a number of criteria (see Table 3).

Using the criteria weights assigned by them, the final likelihood of failure score for each subsystem was calculated. Assuming that the risk of failure to remain under the current design is zero, the risk objective can be expressed as:

$$42.25X_{12} + 9.25X_{22} + 6.75X_{32} + 6.75X_{42} + 9.25X_{52} + 6.75X_{62} \le 0.$$

Goal 4 : Automate District Ledger Subsystem. Management indicated a desire for automating the district ledger subsystem (SS6). However, this goal is of low priority and is pursued only after all other objectives and goals are achieved to the degree desired. This objective is represented as:

$$X_{62} = 1$$

Structural Constraints

Since only one design of each subsystem can be implemented, structural constraints were added. For example, the structural constraint for subsystem 1 would be:

$$X_{11} + X_{12} = 1$$

In addition to these constraints, other translation constraints that translate the product terms $X_{12}X_{22}$, $X_{12}X_{62}$, $X_{22}X_{32}$, $X_{22}X_{42}$, and $X_{22}X_{52}$ of the cost equation to the appropriate linearized variable $Y_{12,\,22},\ldots\ldots\ldots, Y_{22,\,52}$ were also added to the model.

Management Preference Structure

The above objectives and goals are conflicting in nature. Improving the timeliness of the information tends to increase system costs and may increase the likelihood of failure as more changes are introduced into the organization. To make these tradeoffs, management provided the following preference structure:

Table 4

Priority	Goal
1.	Minimize process time deviation
2.	Minimize total cost deviation
3.	Minimize the total score representing the likelihood of failure
4.	Automate the subsystem that updates the district ledger (SS6)

RESULT AND DISCUSSION

The solution will be obtained by using the QSB^+ computer software may be interpreted as follows. In this particular case, solution P_1 dominated P_2 and P_3 dominated P_4. Solution P_3 is associated with least risk and cost; however, for additional risk and cost, P_1 allows for the automation of subsystem 6 as well. Note, however, that both the solutions P_1 and P_3 cost 27 and 30 thousand dollars more than what was budgeted. If this additional expenditure is too large, the user may wish to run the problem again with a higher priority assigned to the cost objectives.

Table 5. Alternative Planning Strategies and Their Achievement Levels

System Portfolio	Subsystems to be Automated	Goals / objectives Achieved
1	Cash reporting subsystem (SS1) Bank Ledger Subsystem (SS5) District ledger subsystem (SS6)	Time Goal – achieved Cost objective – underachieved by Rs.30,982 Risk objective – underachieved by 58.25 Computerize SS6 – Yes
2	Cash reporting subsystem (SS1) Deposit transfer subsystem (SS3) Bank ledger subsystem (SS5)	Time Goal – achieved Cost objective – underachieved by Rs.30,980 Risk objective – underachieved by 58.25 Computerize SS6 – No
3	Cash reporting subsystem (SS1) Bank ledger subsystem (SS5)	Time Goal – achieved Cost objective – underachieved by Rs.27,943 Risk objective – underachieved by 51.50 Computerize SS6 – No
4	Cash reporting subsystem (SS1) Transfer selection subsystem (SS2)	Time Goal – achieved Cost objective – underachieved by Rs.30,510 Risk objective – underachieved by 51.50 Computerize SS6 – No

Additional alternative solutions can be generated by specifying a larger value of Z. These alternative solutions allow for trade-offs and rethinking on the part of the user before acceptance of a solution for implementation. Also, many subjective factors that often cannot be formulated mathematically in the model can be considered effectively at this stage in narrowing down the user's choice to one alternative.

In this particular case, the company used other subjective information and selected the alternative P_2, which called for the automation of subsystems 1, 3 and 5. One of the main reasons for this choice was that the bank ledger system was considered much more critical as it maintains bank balances that are constantly changing due to transfers made among banks. A real-time maintenance of bank ledgers provides quick information on bank balances before and after transfers. If this requirement was known earlier in the investigative step, it could have been expressed as a constraint in the problem (the Bank Ledger system is a necessary prerequisite to the Transfer System). However, it is not often feasible for management to a priori state these prerequisites. A methodology such as this can thus enhance evaluation of alternatives using information that is either subjective or difficult to extract from the user in the investigative stage.

References

1. Bresciani, P., Giorgini, F., Giunchiglia, J., Mylopoulos and Perini, A., *TROPOS : An Agent-Oriented Software Development Methodology*, **8(3)**, 203–236, ENCES (2004).

2. Fox, M. and Long, D., Pddl 2.1 : An extension to pddl for expressing temporal planning mains. *J. Artif. Intell. Res. (JAIR)*, **20**, 61–124 (2003).
3. van Lamsweerde, A., Requirements engineering in the year 00 : a research perspective, Pages 5–19, ACM (2000).
4. Peer, J., Web Service Composition as AI Planning – a Survey, *Technical Report*, University of St. Gallen, (2005).
5. Russell, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition (2002).
6. Tardos, E., Network games, In *Proceedings of the Annual ACM Symposium on Theory of Computing* (2004).
7. Bryl, V., Massacci, F., Mylopoulos, J. and Zannone, N., Designing secure systems through planning, In *CAiSE* (2006).
8. Castillo, L., Fdez-Olivares, J. and Gonzlez, A., Integrating hierarchical and conditional lanning techniques into a software design process for automated manufacturing. In *ICAPS* (2003), *Workshop on Planning under Uncertainty and Incomplete Information*, pages 28–39 (2003).
9. Edelkamp, S. and Hoffmann, J., Pddl2.2 : The language for the classical part of the 4th international planning competition, Technical Report 195, University of Freiburg, (2004).
10. Gans, G., Jarke, M., Kethers, S. and Lakemeyer, G., *Modeling the impact of trust and distrust in agent networks*, pages 45–58 (2001).
11. Letier, E. and van Lamsweerde, A., Reasoning about partial goal satisfaction for requirements and design engineering, *SIGSOFT Softw. Eng. Notes*, **29(6)**, 53–62 (2004).
12. Gans, G., Jarke, M., Kethers, S. and Lakemeyer, G., *Modeling the impact of trust and distrust in agent networks*, pages 45–58 (2001).
13. Kaiya, H., Horai, H. and Saeki, M. (2002) 'AGORA: Attributed Goal-Oriented Requirements Analysis Method', *IEEE Joint International Conference on Requirements Engineering*.
14. Bruno, Gonzalez-Baixauli, Julio, C. S., Mylopoulos, J., 'Visual Variability Analysis for Goal Models', *12th IEEE International Requirements Engineering Conference (RE'04)*, Kyoto, Japan, pp.198-207 (2004).
15. Emmanuel, L. and van Lamsweerde, Axel, 'Reasoning about Partial Goal Satisfaction for Requirements and Design Engineering', Proceedings of the 12th ACM SIGSOFT Twelfth International Symposium on Foundations of Software Engineering, pp. 53–62 (2004).
16. van Lasweerde, Axel, 'Reasoning About Alternative Requirement Options', Conceptual Modeling: Foundations and Applications, *Lecture Notes in Computer Science*, Volume **5600**, pp. 380-397 (2009).
17. Paolo, G., Mylopoulos, J., Eleonora, N. and Roberto, S., 'Reasoning with Goal Models', *Proceedings of the 21st International Conference on Conceptual Modeling*, pp. 167-181 (2002).
18. Ivan, J. Jureta, *et al*, Clear Justification of Modeling Decisions for Goal-Oriented, *Requirements Engineering*, Volume **13**, Issue , pp 87-115 (2008).
19. Naeem, Ur, Rehman, *et al*, 'Comparative study of Goal-Oriented Requirements Engineering', *4th International Conference on New Trends in Information Science and Service Science (NISS)*, pp. 248–253 (2010).
20. Subhas, M., Vinod, K. and Uma, K., 'Goal-oriented or scenario-based requirements engineering technique - what should a practitioner select?', *Canadian Conference on Electrical and Computer Engineering*, pp 2288–2292 (2005).
21. Jennifer, Horkoff and Eric, Yu, 'Analyzing Goal Models – Different Approaches and How to Choose Among Them', SAC'11, TaiC hung, Taiwan (2011).

❑