Multi Agent Cooperative Reasoning Laboratory

Kansas State University
234 Nichols Hall
Manhattan, KS 66506-2302

Phone: (785) 532-6350
Fax: (785) 532-7353
http://macr.cis.ksu.edu/

Technical Report

A Value-Based Goal Model

by

Scott A. DeLoach and Xinming Ou

MACR-TR-2011-01

May 19, 2011

TABLE OF CONTENTS

A Value-Based Goal Model

Scott A. DeLoach and Simon Ou

Department of Computing & Information Sciences, Kansas State University

1 Introduction

The software being produced today is at least an order of magnitude more complex than that being developed a decade ago. Businesses today are demanding applications that operate autonomously, adapt in response to dynamic environments, and interact with other distributed applications in order to provide wide-ranging solutions [1]. To respond appropriately in today's complex environments, software needs to be aware of what it is doing and why in order to take the appropriate steps to achieve its business objectives. There are several instances of these kinds of systems including information systems, service-oriented systems, wireless sensor networks, and multi-robot systems [2]. In each of these types of systems, a critical aspect of adaptivity is understanding *why* those tasks need to be performed.

A central feature of these systems is that they are closely tied to their environment and adapt in response to changes in that environment. Such systems are termed *dynamically adaptive systems*[4]. Recently, it has been suggested that *requirements reflection*, or reasoning over system requirements available as runtime objects, is key to developing dynamically adaptive systems [3, 4]. Specifically, [4] proposes three key challenges to requirements reflection: a runtime representation of the requirements, synchronization between the requirements and the architecture, and dealing with uncertainty. During the last several years, we have developed an approach for capturing requirements as system goals, resulting in the Goal Model for Dynamic Systems (GMoDS) [5]. GMoDS allows system goals to be created as needed and reside in a number of states. However, all goals are *achievement* goals and are treated as having the same intrinsic value to the system.

While GMoDS is useful for highly dynamic systems, there are systems that have a known set of goals that represent the standard operating procedure that should be maintained. Thus, goals of achievement are not suitable and, as such, GMoDS would not appropriate. In such systems, we want to *maintain* a certain set of goals and make trade-offs on which goals to maintain (or not) when all goals cannot be fully maintained. Such situations include when we want to ensure that a set of services are provided on a continual basis, with some services being more important than others. The objective is to provide information to the system for making informed decision (when required) about what services were more important to the system and which services should be maintained at all costs.

2 VALUE-BASED GOAL MODEL (VGM)

Formally, the VGM is a tree whose nodes are value-based goals (G) rooted at goal g0 . Typically, goal g0 represents either the overall operational (business) goal of the network or the overall goal of system security. Goals are typically defined as some desired state of the world, and this is true of value-based goals as well. However, value-based goals are not traditional *achievement* goals whose state must be attained by the system, but instead are *maintenance* goals whose state must be preserved by the system. Thus instead of achieving goals, the objective of a value-based system is to maintain the current set of goals or the value of the system.

The key to determining the overall value of a VGM tree is to know which goals are currently maintained. Thus, we define a set called *maintained* that captures the current set of goals maintained by the system. The maintained set is computed by first determining the leaf goals in the maintained set and then computing the parent goals that are in the maintained set. A formal definition of the maintained set is given once the various types of goals are defined.

2.1 VGM Value

The value of a VGM is based on the current set of goals in the maintained set. Therefore, we define a function, value, to compute the current value of a VGM with a specified maintained set. This value function can be used to determine the current value of the VGM as well as future values of the VGM based on actions that modify the maintained set.

$$value(vgm, maintained) = currentValue(vgm.g_0)$$

One key fact that must be stated to understand the definitions below is that the currentValue of any goal is not in the maintained set is zero.

$$g \notin maintained \Rightarrow currentValue(g) = 0$$

2.2 Value Goal

The root goal, g0, of a value-based goal model represents the overall value of the system. Goal g0 is always a value goal, which is decomposed into a set of sub-goals that each are assigned a maxValue. (The units associated with this maxValue are application specific.) The root value goal is decomposed into one or more of the following types of goals: Composition, AND, OR, or Leaf. For example, in the VGM shown in Figure 1, sub-goal g1 has a maxValue of $10, g2 has a maxValue of $25 and g3 has a maxValue of $50. Thus, the overall system would have a value of $85 if all three sub-goals are maintained. However, if due to system failure sub-goal g1 could no longer be maintained, the value of the system would be decreased to $75. Thus, the maximum value of a VGM rooted at goal g is

$$root(g) \Rightarrow maxValue(g) = \sum_{(g,g') \in subgoal} maxValue(g')$$

while the current value of the VGM rooted at goal g is

$$root(g) \Rightarrow currentValue(g) = \sum_{(g,g') \in subgoal} currentValue(g')$$

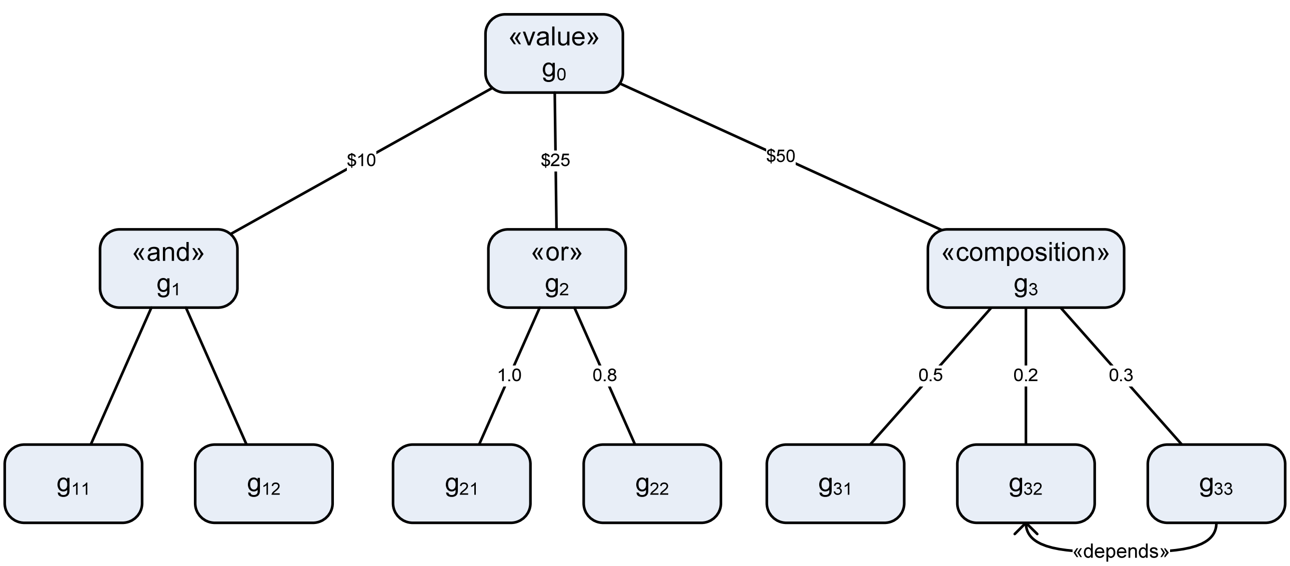

Figure 1. Operational Goal Model

2.3 Composition Goals

If a goal is a Composition goal, all of its sub-goals contribute a percentage to its value. Thus, each sub-goal of a Composition goal has an associated contribution value and the contributions of all sub-goals of a Composition goal must equal 1.0. Formally, this is state as

$$composition(g) \wedge (g, g') \in subgoal \Rightarrow 0 \le contribution(g') \le 1$$

and

$$composition(g) \Rightarrow \left(\sum_{(g,g') \in subgoal} contribution(g') \right) = 1$$

The current value of a Composition goal is the sum of the sub-goal contributions that are currently maintained. Thus the current value of a Composition goal g is

$$composition(g) \Rightarrow currentValue(g) = maxValue(g) * \sum_{\substack{(g,g') \in subgoal \\ \wedge\, g' \in maintained}} contribution(g')$$

Thus in Figure 1, each sub-goal of g3 has a contribution to g3's overall value. If all sub-goals are maintained, the overall contribution is 1.0 and the value of g3 remains at $50. However, if sub-goal g31 is not maintained, then the overall contribution is only 0.5 (0.2 from g32 + 0.3 from g33) and the value of g3 drops to $25.

The maxValue of each sub-goal can also be computed as its contribution multiplied by its parent's value. So, for goal g33, its individual value is 0.3 * $50 = $15.

$$composition(g) \wedge (g, g') \in subgoal \Rightarrow maxValue(g') = maxValue(g) * contribution(g')$$

2.4 AND Goals

An AND goal denotes the case when all sub-goals goals are required to be maintained in order for the parent goal to be maintained thus contributing its full value. In Figure 1, for goal g1 to have its $10 value, both goals g11 and g12 must be maintained. If either or both of them are not maintained, then the value of goal g1 is 0. However, in more complex cases, the current values of the sub-goals may be maintained, but not at the maxValue level. Thus, we define the current value of an AND goal to be the minimum currentValue of all its sub-goals (assuming all are maintained). Thus the current value of an AND goal g is defined as (note that if a sub-goal is not maintained its current value is zero)

$$and(g) \wedge \bigwedge_{(g,g') \in subgoal} g' \in \ maintained \Rightarrow$$
$$currentValue(g) = \ \min\left(\{currentValue(g')|(g,g') \in subgoal\}\right)$$

We define the individual maxValues of each sub-goal of an AND goal to be equal to the value of its parent since the failure to maintain any one of the sub-goals will reduce the value of the parent to zero. Formally, this is stated as

$$and(g) \wedge (g, g') \in subgoal \Rightarrow maxValue(g') = maxValue(g)$$

2.5 OR Goals

An OR goal is similar to an AND goal as its value is based on a Boolean operator, in this case logical OR. Thus, if any sub-goal of an OR goal is maintained, then the OR goal itself is maintained. However, unlike the AND goal, each sub-goal has a contribution value associated with it. So, in Figure 1, if g21 is not maintained but goal g22 is maintained, then the value of g2 becomes (0.8 * $25) or $20. The notion of an OR goal is that there are may be multiple ways to maintain a specific goal, although some may be better than others. The maximum values of the sub-goals g21 and g22 are their contribution multiplied by the maximum value of their parent g2, or $25 and $20 respectively. Formally, this is stated as

$$or(g) \wedge (g, g') \in subgoal \Rightarrow maxValue(g') = maxValue(g) * contribution(g')$$

The current value of an OR goal is the maximum current value of its maintained sub-goals. Thus the current value of an OR goal is defined as

$$or(g) \Rightarrow currentValue(g) = \max\left(\{currentValue(g')|(g,g') \in subgoal\}\right)$$

2.6 Leaf Goals

Leaf goals have no sub-goals and contribute to the overall value of the goal tree based on their parent's type. In actuality, only Leaf goals are actively maintained by the system. As the system maintains (or fails to maintain) Leaf goals, the overall value is aggregated based on parent goal types until a final value for the system is arrived at. Using the description above, it should be noted that the value of a system is not simply the value of all its Leaf goals and thus care should be taken when using the values of Leaf goals independently of their parent goals. In many cases, the value of a Leaf goal (that are sub-goals of AND/OR goals directly or indirectly) can only be computed in light of a specific configuration. For instance, the value of g21 is $25 while the value of g22 is $20; however, since their parent goal is an OR goal, only one of them needs to be maintained. Thus, if g21 is currently being maintained, then performing work to maintain g22 does not increase in overall system value. Likewise, even though goal g11 and g12 both have a maximum value of $10, unless both g11 and g12 are maintained, the value contributed to the system individually is $0. Therefore, the true value of a goal can only be determined in light of the current system operation, which is captured in the maintained set.

In our example, if all Leaf goals are maintained by the system, then the value of g0 is at its maximum, or $85. However, if only Leaf goals g12, g22, g31, and g33 are maintained, the value of g0 decreases to ($10 * 0) + (0.8 * $25) + ((0.5 + 0.3) * $50) or $60.

2.7 Depends

In addition, a depends relation may exist between goals. If goal g1 requires that goal g2 be maintained in order for it to be maintained, we say g1 depends on g2. The depends relation is not used directly in calculating the value of a given tree but is used to determine which Leaf goals are currently in the maintained set. Thus, in the example in the preceding paragraph, since g33 depends on g32, if g32 is not maintained, then g33 cannot be considered maintained and thus the value of g0 decreases even further to ($10 * 0) + (0.8 * $25) + (0.5 * $50) or $45.

3 Maintained Set

The value of the maintained set is computed bottom up starting with the Leaf goals. The maintenance of Leaf goals is dependent on their state being maintained by the system. For a Leaf goal g to be in the maintained set, two conditions must be met: (1) the state of g must be maintained by the system (we use maintained(g) to denote this), and (2) all the goals that g depends on must be in the maintained set. This is stated formally as

$$leaf(g) \Rightarrow$$

$$(g \in maintained \;\Leftrightarrow maintained(g) \wedge (\forall\, (g, g') \in \; depends \wedge \; g' \in maintained))$$

The maintenance of parent goals (Value, Composition, AND, OR) is based on their logical conditions.

$$root(g) \vee composition(g) \vee or(g) \Rightarrow$$
$$(g \in maintained \Leftrightarrow (\exists\, (g, g') \in subgoal \wedge g' \in maintained))$$

$$and(g) \Rightarrow (g \in maintained \Leftrightarrow (\forall\, (g, g') \in subgoal \wedge g' \in maintained))$$

4 EXAMPLE

An example of a VGM for an online web system is shown in Figure 2 that includes a web server that supports online-shopping business and additional computers that are used by employees for their daily work. In our example, the system's goals are decomposed into two main goals: supporting a web store (Webstore) and work access (User Work Access). Since the Webstore goal constitutes the majority of our business income, we have estimated that if this goal is 100% operational, we will generate approximately $50,000 per hour from it. Likewise, for each hour it is not operational, our business will lose $50,000. The User Work Access goal represents the ability of the employees to access the system in order to maintain it and perform their daily work. The estimated value of this access is approximately $3,000 per hour. The Webstore goal is decomposed into a set of subgoals, where each subgoal is weighted to express its contribution to its parent goal. The Browse goal represents the ability of consumers to browse our website looking for goods to purchase. This constitutes a major aspect of the webstore and thus we have assigned a 60% value to the overall Webstore goal. The next goal, Shopping Cart, is obviously important for customers who actually purchase items from the website. While customers can browse the website, without the ability to add items to their shopping cart, they cannot actually buy those items. Since there is still functionality to the webstore without the shopping cart, we give it a 20% contribution. Finally, the Place Order goal represents the actual transaction where

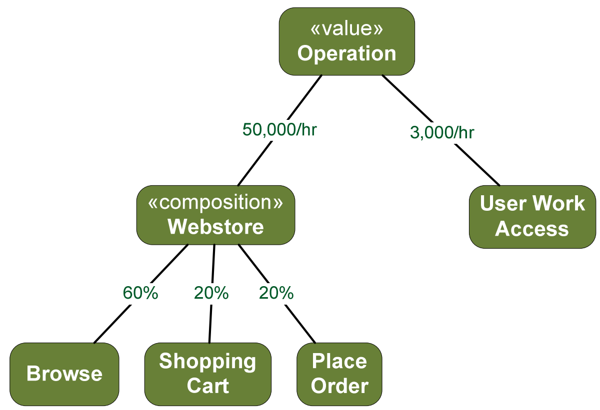

Figure 2. Operational VGM

the customer buys what is in their shopping cart. While customers can shop and add items to their cart, without the ability to check out and purchase their items, the system is not complete. Thus, we assign a 20% contribution to this goal.

Based on the values assigned to the VGM, we can now perform computations to determine the effect of maintaining or not maintaining certain goals to the overall system value.

5 CONCLUSIONS

This paper has proposed a VGM that is useful for situations where a static set of goals should be maintained by a system. The VGM formulation allows that system to make tradeoffs based on the importance of various system goals based on a variety of factors.

6 REFERENCES

1. Kube, C.R., and Zhang, H. 1993. Collective robotics: from social insects to robots. Adaptive Behavior, 2, 2, 189–218.

2. Parker, L.E. 2008. Distributed intelligence: overview of the field and its application in multi-robot systems. *J. of Physical Agents*, 2, 1, 5–14.

3. Bencomo, N., Whittle, J., Sawyer, P., Finkelstein, A. and Letier, E. 2010. Requirements reflection: requirements as runtime entities. *Proc. of the 32nd ACM/IEEE Intl. Conf on Software Engineering - Vol 2* (ICSE '10), ACM, 199-202.

4. Sawyer, P., Bencomo, N., Whittle, J., Letier, E, Finkelstein, A. 2010. Requirements-Aware Systems: A Research Agenda for RE for Self-adaptive Systems. *Proc of 18th IEEE Intl Requirements Engineering Conf*. 95-103, IEEE.

5. DeLoach, S.A. and Miller, M. 2010. A goal model for adaptive complex systems. *Int J. of Computational Intelligence: Theory and Practice*. 5, 2.