Formalization of Business Goal Models Based on Category Theory: A Postprint

Authors: Li Zonghua, Li Bixin

Date: 2019-04-01T00:00:00+00:00

Abstract

Goal-oriented requirement language (GRL) models focus on to-be-determined requirements and are widely applied to initial requirement modeling of business systems, where the correctness of these models affects the development quality of business systems. Given that the formalization of business goal models can verify model correctness, this paper proposes a method for formalizing GRL models using category theory. First, based on the GRL meta-model structure, the morphism mechanism in category theory is applied to formally describe the relationships between nodes in GRL models, such as goal-to-goal, goal-to-task, and task-to-task relationships. Then, by adding initial and terminal objects to the category model, adjacent sequences are designed to represent the causal relationships among the implementation of multiple goals and tasks. Finally, the structural properties of correctness for business goal model systems are designed. Experiments applying the Web Payment system demonstrate that the formalized business category model can verify the correctness of GRL models and improve the quality of goal modeling.

Full Text

Preamble

Vol. 37 No. 5
Application Research of Computers
ChinaXiv Partner Journal

Formalizing Business Goal Models Based on Category Theory

Li Zonghua[1,2], Li Bixin[2]
(1. School of Computer Science & Technology, Huaiyin Normal University, Huaian, Jiangsu 223300, China;

2. School of Computer Science & Engineering, Southeast University, Nanjing, Jiangsu 211189, China)

Abstract: The Goal-Oriented Requirement Language (GRL) focuses on capturing tentative requirements and has been widely applied to initial requirements modeling for business systems. The correctness of GRL models significantly impacts the development quality of these systems. Since formalizing business goal models can verify their correctness, this paper proposes a method for formalizing GRL models using category theory. First, based on the GRL meta-model structure, we apply the morphism mechanism from category theory to formally describe relationships between goal nodes, between goal and task nodes, and between task nodes. Then, by adding initial and terminal objects to the category model, we design neighborhood sequences to represent the causal relationships among multiple goals and task implementations. Finally, we design structural properties for verifying the correctness of business goal model systems. A case study using the Web Payment system demonstrates that the formalized business category model can verify GRL model correctness and improve the quality of goal modeling.

Keywords: GRL model; category theory; model formalization; model correctness verification

---

0 Introduction

Capturing clear and explicit business goals is crucial for business systems. Consequently, the appropriateness, completeness, clarity, and consistency of business goal models significantly influence software quality [1]. Goal-Oriented Requirements Engineering (GORE) views business systems and their environments as collections of active components that can constrain their behavior to ensure mandatory task enforcement [2,3]. The Goal-Oriented Requirement Language (GRL), developed by the Osis team, is a language that supports goal-oriented modeling [4,5]. It focuses on tentative requirements analysis for capturing business system goals, developing alternative solutions, and determining goal contribution values [2,7]. GRL models can effectively define initial system problems and have been widely applied to initial requirements modeling for business systems [1,6,8].

GRL allows business analysts and developers to describe system goals, tasks, softgoals, and other information using graphical notations. However, these notational descriptions [4] do not include complete goal model semantics [9], making GRL model correctness difficult to verify [10]. Since goal models' constraint relationships can handle interdependencies between business goals, Popova et al. [11] proposed a comprehensive formal framework for goal-oriented approaches that uses predicate language to formalize hard goals, soft goals, goal constraints, goal decomposition, and goal refinement. Giachetti et al. [12] applied model-driven methods to design specific Object Constraint Language (OCL) rules for verifying goal requirements models. Mendonça et al. [13] proposed a context-based formal method for a goal-oriented dependency analysis framework to help experts evaluate design-time and runtime dependability in different contexts. Diamantini et al. [14] designed an ontology-based goal modeling method. While these formal methods can effectively formalize goal model elements and verify correctness, they are somewhat inadequate in characterizing relationships between goal nodes.

Category theory uses abstract methods to describe mathematical structures and represent relationships between structures [15]. It formalizes specific categories using "objects" and "morphisms" and can be used to discover and verify connections between structures in different domains [16]. In software engineering, researchers have used category theory to formally analyze software architectures [17~19]. Therefore, this paper leverages category theory' s advantages in describing structures to formalize business goal models, using "morphisms" to characterize relationships between business goal nodes and thereby verify the correctness of goal models.

1 GRL Model

GRL can identify stakeholders (actors) and their intentions (goals, tasks, and softgoals), as well as graphically distinguish functional requirements (goals, tasks) from non-functional requirements (softgoals) [20]. Its meta-model structure is shown in [Figure 1: see original paper]. GRL' s focus lies in extracting initial business goals—identifying each participant' s business goals from the perspectives of different end-users and business collaborators, then analyzing the relationships among these business goals [21].

The basic modeling elements of GRL are shown in [Figure 2: see original paper]. GRL consists of three categories: intentional elements, intentional relationships, and actors. Intentional elements include goals, tasks, softgoals, and resources. Intentional relationships include decomposition, contribution, correlation, and dependency. These elements and relationships enable models that can answer questions such as: Why are specific behaviors, information, and structures selected to describe system requirements? What alternatives should be considered? What criteria should be used to evaluate alternatives? And what reasons justify selecting one alternative? Therefore, a GRL model can be considered as comprising actors, the set of goals or tasks that actors need to accomplish, and linking elements.

2 Formalization of Business Goal Models

Since any goal model system must have initial and terminal goal objects, the key theoretical contribution of this paper lies in identifying initial and terminal objects within the category model. The formal description is as follows: Suppose an initial object I exists in category $\mathbf{C}$, where every object in $\mathbf{C}$ except I must be the codomain of some specific morphism, while I is only the domain of some morphism. That is, there exists an object X in **C** such that there is a unique morphism from I to X. Therefore, object I can be considered the initial goal object of category **C**.

Suppose a terminal object T exists in category **C**, where every object in **C** except T must be the domain of some specific morphism, while T is only the codomain of some morphism. That is, there exists an object X in **C** such that there is a unique morphism from X to T. Therefore, object T can be considered the terminal goal object of category **C*. Identifying initial and terminal objects facilitates complete formal description of business goal models.

Definition 1. Based on category theory, a system' s business goal graph can be defined as a function where: E is the set of edges, V is the set of nodes. Let , where represents the source node of e, and represents the target node of e as an ordered pair, i.e., . Let denote the set of arrow characters (arrow set), and denote the set of node characters (object set). Then a business goal graph can be defined as a system [22], i.e., , where:

- A represents actors, i.e., participant objects in the goal system, corresponding to business participants in the business system.
- represents the set of morphisms, where represents morphism type "Decomposition," indicating decomposition of a business goal node into several business task nodes; represents morphism type "Contribution," indicating that a task node contributes to other task nodes or goal nodes; represents morphism type "Dependency," indicating that a task node has a dependency relationship with other task nodes or goal nodes.
- represents the set of nodes, corresponding to business services or business activities in the business system, where represents the node name and represents the node type. indicates node type "goal," representing functional goals in the business system, corresponding to business service concepts; indicates node type "task," corresponding to business task concepts; indicates node type "softgoal," describing non-functional requirements in the business system; indicates node type "resource," representing resources needed to achieve business goals. These nodes typically belong to a participant component, so each actor corresponds to a node set.
- represents the source mapping.
- represents the target mapping.
- represents the mapping of arrow character sets.
- represents the mapping of node character sets.

Therefore, a goal model system can be represented as . Several clarifications:

a) If and are both empty sets, then G is called an empty discrete goal model system.

b) If is an empty set, then G is called an empty goal model system.

c) If and are both finite sets, then G is called a finite goal model system.

d) If exists as an ordered pair and the ordered pair is a finite set, then G is called a local goal model system.

e) If exists as a unique ordered pair, then G is called a simple goal model system.

f) For any morphism , the source node of morphism is , and the target node of morphism is .

Based on the above clarifications:

Theorem 1. Let be a goal model system, and .
Proof. If morphism exists···

Definition 2. For any goal model system G, if morphism exists, and G also contains morphism , then morphisms and are identical with respect to .

Definition 3. For any goal model system , if the following conditions hold: , then is called a sub-goal model system of , denoted as . If , then is called a proper sub-goal model system of . If , then is called a wide sub-goal model system of . If for all , then is called a complete sub-goal model system of .

Thus, a sub-goal model system shares the same semantics as its parent goal model system and represents a goal node within the parent system. This sub-goal model system is independent and reusable.

Definition 4. For any goal model system , if nodes and satisfy the following conditions: , then a temporary morphism must be added between nodes and , indicating that nodes and have a bi-morphism property. This represents that nodes connected by dependency relationships in the goal model system belong to different actors, and the execution results of target node tasks must return to source node tasks.

This definition is illustrated in [Figure 3: see original paper], showing that nodes A, B, and C belong to different actors. Since dependency relationship exists between node A and node B, node B' s execution results must return to node A. Similarly, dependency relationship exists between node C and node B, so node C' s execution results must return to node B. According to Theorem 1, nodes A and C also have a dependency relationship, thus morphism exists, and node C' s execution results must ultimately return to node A.

Definition 5. For any goal model system , if node satisfies the following conditions: , then is called the initial object of goal model system G. If node satisfies: , then is called the terminal object of goal model system G.

Definition 6. For any goal model system , if the following conditions are satisfied: , then there exists a goal neighborhood sequence representing an action sequence of a task in the goal model system. Simultaneously, there exists a neighborhood sequence set representing the set of all nodes in this sequence.

3 Structural Properties of GRL Model Correctness

Popova et al. [11] define goal model correctness as verifying whether conflicts exist between business goals. Giachetti et al. [12] consider goal model correctness as verifying whether business goals and resources are associated with actors and whether business tasks are associated with entities or resources. This paper defines goal model correctness as formally verifying that: no isolated goal nodes exist in the model, no closed-loop goal neighborhood sequences exist, and intentional relationships connect business goals.

Definition 6. For any goal model system , the following conditions must hold: [The definition appears to be repeated here in the original text; we maintain the structure.]

Theorem 2. A correct goal model system is connected.
Proof. First, we prove that no nodes in the goal model system are isolated. Then we prove that any node can be reached from the start object and can reach the terminal object. This proves the model is connected.

a) If for every , then is the initial object of goal model system G.

b) If for every , then is the terminal object of goal model system G.

c) If is both an initial and terminal object, then is a zero object.

Theorem 3. A correct goal model system has no cycles.
Proof. For any goal model system , applying Definition 4, G has initial and terminal objects. Applying Definition 6, all nodes in G are in the neighborhood sequence set and satisfy Theorem 2. Therefore, G is cycle-free, meaning each neighborhood sequence represents a system function. Otherwise, the system would lack a clear start or terminal object, indicating a closed-loop situation where at least one infinitely looping neighborhood sequence exists, which indicates problems in the business system' s goal model.

Theorem 4. A correct goal model system satisfies the "goal-assignment" relationship.
Proof. For any goal model system , where represents a business goal in the system, this goal is decomposed into three sub-goals . In the goal system, represents the global goal, while represent local tasks. Morphism exists···

Business goal models capture initial requirements of business systems by dividing functional modules into a series of task steps. Decomposed goals represent global goals. During goal decomposition, sub-goals must not conflict—implementation of one sub-goal cannot cause failure of another (i.e., sub-goals must not have dependencies that invalidate each other' s preconditions or postconditions). Additionally, sub-goals must not conflict with global goals: global goal realization requires all sub-goal realizations, and if any sub-goal fails, the global goal cannot be achieved. Therefore, node elements in the goal model system satisfy the "goal-assignment" relationship.

Theorem 5. A correct sub-goal model system should share the same semantics as its parent goal model system, not exceed the parent' s resource usage boundaries, and have common softgoals.

Proof. For any goal model system , if , then is a sub-goal model system of G sharing the same semantics. Let represent all resource node sets and softgoal node sets in goal model system G. If , then the sub-goal model system *ng1*' s resources do not exceed the parent system' s boundaries and share common softgoals.

4 Case Study: Web Payment System

4.1 Modeling Environment

The running example in this paper comes from the Web Payment business system in the W3C standard document [23]. The scenario is described as follows: First, a customer browses products on an online shopping site. The customer then selects products and adds them to the shopping cart. Once the customer places an order, they must log into their account and provide their bank account or other payment information (Alipay, Google Wallet, Apple Pay, etc.) to the online shopping site, which also requests verification codes. The online shopping site submits the user' s payment information to the financial company, and both the customer and online shopping site receive payment credentials from the financial company. Finally, the online shopping site sends a digital receipt to the customer, and its shipping department delivers the product to the customer.

4.2 Business Goal Modeling

Business goal modeling is completed through the following activities: (a) Extract and analyze nouns and verbs from natural language descriptions of business requirements to identify actors and functional intentional elements; (b) Classify related intentional elements under actors, showing that each element is enacted by a specific actor; (c) Establish intentional relationships between elements. Through scenario analysis of Web Payment and application of these three GRL modeling activities, the initial goal model of the Web Payment system is shown in [Figure 4: see original paper]. The intentional relationships indicate that whether customers receive good online shopping services depends entirely on services provided by the online shopping site system. For example, a dependency relationship exists between the goal "gain goods options" and the task "provide goods options."Interaction between the customer and financial company depends on services provided by the online shopping site.

4.3 Formalization of Business Goal Models

Based on Definition 1, the formalized GRL model for the Web Payment system has node information listed in . Therefore, the Web Payment business system' s goal model can be defined as a graph category:

where represents arrows for decomposition relationships, represents arrows for dependency relationships, and represents arrows for contribution relationships.

The node character set is:

where the actor set A is:

The source mapping s and target mapping t are:

The arrow character set is:

Based on these definitions, [Figure 5: see original paper] shows the formalized graph category model of the Web Payment system' s GRL model.

4.4 GRL Model Correctness Verification

[Figure 5: see original paper] clearly shows that causal relationships in functional aspects can be characterized through intentional relationships. According to Definition 6, node *ng1* is the initial object of the Web Payment system goal model, while nodes *nt1*, *nt2*, *nt3*, *nt4*, *nt5*, *nt6*, *nt8*, *nt9*, *nt10*, and *nt11* are terminal objects. Multiple morphism sets exist, one of which is:

The entire formalized goal model contains multiple neighborhood sequences, including:

These neighborhood sequences describe the implementation processes of business goals such as product searching, ordering, and payment. Therefore, we must use the formalized model to verify: whether isolated nodes exist, whether any neighborhood sequences contain cycles, and whether the "assignment" relationship holds between global goals and subtasks.

We verify the correctness structural properties defined in Section 3:

a) **Isolated node verification**: The above neighborhood sequences cover all nodes in the model. The model' s neighborhood set is:

This shows that all nodes in the model are non-isolated, meaning the goal model system is connected.

b) **Cycle verification in neighborhood sequences**: Applying Theorem 3, the goal model system contains no infinitely looping neighborhood sequences, indicating the system is cycle-free.

c) **Global goal and subtask assignment verification**: Applying Theorem 4, the system contains four global goals: "gain goods options," "online payment," "provide goods options," and "offer payment service." Each global goal is decomposed into several subtasks without dependencies between them. Only when all subtasks are completed can the global goal be achieved. For example, the goal "offer payment service" requires completion of five subtasks: "offer payment instrument," "initiate payment authorization," "authorize transfer," "complete transfer," and "send digital receipt." If any subtask fails, the goal "offer payment service" is not achieved, indicating an exception in payment service. Therefore, the goal model satisfies the "goal-assignment" relationship.

Based on these three correctness structural properties, we can verify that the goal model shown in [Figure 4: see original paper] is correct.

From a semantic perspective, neighborhood sequence Neighbor-S1 (shown with thick arrows in [Figure 5: see original paper]) represents a complete online shopping business process on the Web Payment system. Due to the defined bi-morphism property between goal-type nodes, neighborhood sequence Neighbor-S4 represents the function where users re-query products when dissatisfied with search results from the online shopping site. Neighbor-S5 indicates exceptions during payment or user-initiated payment abandonment. While other causal sequences like Neighbor-S2 and Neighbor-S3 exist, domain experts and business analysts can use their expertise to analyze their semantics and determine these sequences are incorrect. Therefore, all correct neighborhood sequences in [Figure 5: see original paper] can cover all functional features of the Web Payment business system.

5 Conclusion

This paper addresses GRL goal model correctness verification by extending traditional category theory and designing a graph category approach to formalize GRL models. We define structural properties for verifying goal model correctness. This method formalizes GRL model syntax and semantics, designs formal structures for business goal systems, defines bi-morphism properties and goal system neighborhood sequences, and designs structural properties including connectivity, cycle-freedom, and "goal-assignment" relationships for verifying GRL model correctness.

The Web Payment case study demonstrates that the designed morphism sets can comprehensively describe relationships between nodes in goal models. By formally defining initial and terminal objects of goal systems, the designed neighborhood sequences can completely describe business functions in goal models. Compared with Popova et al.' s predicate method [11] and Diamantini et al.' s ontology-based goal model [14], our formalization method offers advantages in characterizing relationships between goal model nodes.

However, the current formalization is manually generated. To ensure every node in the goal model system can be traversed, the neighborhood sequences contain numerous invalid sequences. Therefore, future work must address automatic execution of formalization and further filtering of effective system sequences to improve formalization efficiency.

References

[1] Jureta I J, Faulkner S, Schobbens P Y. Clear justification of modeling decisions for goal-oriented requirements engineering [J]. Requirements Engineering,

2008, 13(2): 87-115.
[2] Van Lamsweerde A. Goal-oriented requirements engineering: a guided tour [C]//Proc of the 5th IEEE International Symposium on Requirements Engineering. Piscataway, NJ: IEEE Press, 2001: 249-262.
[3] Bernardo M, Inverardi P. Formal Methods for software architectures [C]//Lecture Notes in Computer Science. 2003.
[4] GRL [EB/OL]. http://www.cs.toronto.edu/km/GRL/.
[5] Amyot D. Introduction to the user requirements notation: learning by example [J]. Computer Networks, 2003, 42: 285-301.
[6] Mussbacher G, Ghanavati S, Amyot D. Modeling and analysis of URN goals and scenarios with jUCMNav [C]//Proc of the 17th IEEE International Requirements Engineering Conference. Washington DC: IEEE Computer Society, 2009: 383-384.
[7] Meland P H, Paja E, Gjære E A, et al. Threat analysis in goal-oriented security requirements modelling [J]. International Journal of Secure Software Engineering, 2014, 5(2): 1-19.
[8] Sabatucci L, Cossentino M, Susi A. A goal-oriented approach for representing and using design patterns [J]. Journal of Systems and Software, 2015, 110(12): 136-154.
[9] Laibinis L, Pereverzeva I, Troubitsyna E. Formal reasoning about resilient goal-oriented multi-agent systems [J]. Science of Computer Programming, 2017, 148(11): 66-87.
[10] Zhang Xuan, Wang Xu, Li Tong, et al. Correctness control and detection in aspect-oriented business process modeling [J]. Chinese Journal of Computers, 2018, 41(3): 521-544.
[11] Popova V, Sharpanskykh A. Formal modelling of organisational goals based on performance indicators [J]. Data & Knowledge Engineering, 2011, 70: 335-364.
[12] Giachetti G, Marín B, López L, et al. Verifying goal-oriented specifications used in model-driven development processes [J]. Information Systems, 2017, 64(3): 41-62.
[13] Mendonça D F, Rodrigues G N, Ali R, et al. GODA: a goal-oriented requirements engineering framework for runtime dependability analysis [J]. Information and Software Technology, 2016, 80(12): 245-264.
[14] Diamantini C, Freddi A, Longhi S, et al. A goal-oriented, ontology-based methodology to support the design of AAL environments [J]. Expert Systems with Applications, 2016, 64(12): [page numbers].
[15] Jia Wei, Hua Qingyi, Zhang Minjun, et al. User interface pattern language based on category theory [J]. Journal of Computer-Aided Design & Computer Graphics, 2017, 29(1): 79-89.
[16] Barr M, Wells C. Category theory for computing science [M]. Upper Saddle River: Prentice-Hall, 2012.
[17] Fiadeiro J L. Categories for software engineering [M]. Berlin: Springer, [year].
[18] Zhu M, Grogono P, Ormandjieva O, et al. Using category theory and flow analysis for modeling and verifying properties of communications in

the process-oriented language Erasmus [C]//Proc of the 7th Conference on Computer Science and Software Engineering. 2014: 1-24.
[19] Ormandjieva O, Bentahar J, Huang J, et al. Modelling multi-agent systems with category theory [J]. Procedia Computer Science, 2015, 52: [page numbers].
[20] Mylopoulos J, Chung L, Yu E. From object-oriented to goal-oriented requirements analysis [J]. Communications of the ACM, 1999, 42(1): [page numbers].
[21] Liu Chun, Huang Ranran, Han Daojun. Goal oriented approach for analyzing event model of cyber-physical systems [J]. Computer Science, 2017, 44(4): 100-103.
[22] Li Zonghua, Zhou Xxiaofeng, Gu Aihua, et al. A complete approach for CIM modelling and model formalizing [J]. Information and Software Technology, 2015, 65(C): 39-55.
[23] W3C, Web Payments Use Cases 1.0 [EB/OL]. [2015-07-30]. https://www.w3.org/TR/web-payments-use-cases/.

Note: Figure translations are in progress. See original paper for figures.