Human Behavior, Goals and Model-Driven Software Engineering for Assistive Systems

Judith Michael[1], Bernhard Rumpe[1], Simon Varga[1]

Abstract: Assistive systems might reason about human behavior and specific actions to be able to assist human activities in everyday life or working situations. It is a challenge to create an adaptive, unobtrusive system with high accuracy of supporting actions. Previous work assumes that either a concrete goal is preset for a whole support application, or is chosen from a finite set of goals by the user or is calculated over a finite set of goals by heuristic algorithms. This novel directions paper discusses ideas to reduce the solution space for assistive systems by using observations of human behavior together with domain-specific and general knowledge. We discuss especially challenges for creating goal models, how to combine them with other existing models, and how to use them in model-based software engineering approaches with automatic code generation. A concrete realization of these ideas enables a variety of design decisions regarding modeling languages, the interplay of different languages, and how to handle goals at run-time.

Keywords: Goal Modeling; Assistive Systems; Behavior Goals; Model-Driven Software Engineering

1 Introduction and Challenges

Motivation. To be able to assist human activities in everyday life or working situations, systems need to know facets of human behavior and predict what steps will happen next [AMM14, Mi18]. Research in the field of assisted living [Mi18] discovered three methods to handle human task goals: (1) A system has predefined goals, *e.g.,* in case of fall detection systems, a fallen person has the goal to get help. Thus, the whole application has one goal predefined. (2) A person selects a goal, *e.g.,* assistance in cooking a certain recipe on a tablet, or navigation to the nearest bus station on a smartphone. (3) Heuristic algorithms calculate goals, *e.g.,* based on the frequency of preceding behavior, and a predefined list of goals. All of these approaches have disadvantages: Method (1) lacks the option to allow for dynamic addition and adaption of goals, as the whole system needs to be changed. Method (2) is challenging for users who strive for an unobtrusive system: They might have to choose regularly between a variety of options what behavior the system should support and thus, interact too often with the system. Method (3) can be error-prone dependent on the data sets used for calculating the heuristics. If you want to create an adaptive, unobtrusive system with high accuracy of supporting actions, precise support is *challenging* for existing approaches.

[1] Software Engineering, RWTH Aachen, Germany, www.se-rwth.de, {michael,rumpe,varga}@se-rwth.de

Goal Modeling for Assistive Systems. In this paper, we discuss ideas for a fourth method: To combine goal modeling, observations of human behavior as well as domain-specific and general knowledge about it could reduce the solution space for assistive actions. Thus, we need (1) to investigate required concepts in existing goal modeling languages and either develop a Domain-Specific Language (DSL) or integrate the concepts into an existing DSL to be able to create goal models, (2) investigate the relations to models created with other DSLs, especially for Model-Driven Software Engineering (MDSE) approaches with automatic code synthesis and code generators, and (3) identify goals from observation.

Outline. Section 2 discusses the concept goal, related concepts, goal modeling languages, and shows an example. Section 3 shows major challenges for goal modeling, goal model relations in MDSE approaches, and the identification of goals. The last section concludes.

2 Background and Example

Research interprets the concept *goal* in various ways depending on the perspective. This work uses the term *goal* as a synonym for the *goals of human behavior* or *action goals*. In philosophy, teleology (from greek telos, “end,” and logos, “reason”) assumes that activities or processes always have a reference to some purpose, end, or goal [2]. Following [EW02], goals are the „desired end states people try to attain through the cognitive, affective, and biochemical regulation of their behavior“. [No11] describes a goal as „a set of stable states that are part of the proper subset of the set of stable states“. Most definitions include states, different types of goals, a relationship between a goal, a person, and the behavior of a person: Concepts which can be found in a variety of goal modeling languages.

Goal Modeling is an important aspect in several research directions of Informatics, *e.g.,* Requirements Engineering (RE) [An96, RS05, KL05, HML15, HY16, Be19], Software Engineering [Ho15], Information Systems [SS13], Enterprise Modeling [OFK15, BF16, Kr08], Human-Computer Interaction [Ka01, Mc03], Artificial Intelligence [Br05], Security [EYZ09], Robotics [He15] using a generative approach, or Computational Linguistics [Ca88]. Several *goal modeling languages* exist, *e.g.,* i* [YM93] or KAOS [DvLF93] for RE, for robotics and model-driven engineering [He15], approaches using goal concepts, *e.g.,* Belief-Desire-Intention (BDI) [Br05] from agent-based systems, and more specific goal modeling languages, *e.g.,* for multi-level modeling [OFK15] or i* extensions [Go18]. Their common concepts are, *e.g.,* related subjects, activities, and relations between goals presented as a directed graph [Cv14]. Moreover, they distinct between different types of goals, *e.g.,* soft-goals [MCN92] and hard-goals where the satisfaction can be verified [An96], or achieve, maintain and optimize goals [DvLF93]. Considering the relation between goals and actions, there exist approaches to map goal models to BPMN processes [Kr14] or goal-oriented process mining [GA20].

To summarize, there exists a variety of goal modeling languages but systematic considerations about the use of goal models for MDSE and code generation of assistive systems is missing.

[2] Encyclopaedia Britannica: Teleology, 2016. `https://www.britannica.com/topic/teleology`, last access: 10.5.2020

Example for a goal and its relations. The goal „coffee is prepared“ describes the final state of the action „prepare coffee“. This goal is further described by a related person, *i.e.,* the assisted person in an assistive system, attributes and their assigned values, *e.g.,* the cup is filled and the person has the cup in her hand, and it is marked as a soft goal which could be interrupted by another action such as „answer the phone call“ or „open the door“.

An assistive system needs to know about the context of actions [MS17] to provide meaningful assistance. Dependent on the context of the action „prepare coffee“, *e.g.,* the person cooking coffee is a barista in a coffee shop or someone is cooking for her or himself at home, the preceding actions and, thus, preceding goals differ. The barista serves the coffee to another person who adds milk and sugar, stirs the coffee and drinks it. The person which cooks coffee for herself picks up the newspaper in front of the door, sits down at the dining table, starts reading the newspaper and drinks the coffee.

3 Modeling and Using Behavior Goals

Handling goal models reduces the solution space for assistance situations of assistive systems by using observations of human behavior together with domain-specific and general knowledge. To handle goals in model-driven systems the following three aspects are important: (A) to create goal models, (B) to understand their relationship with models created in other modeling languages and (C) to compare actions and goals in the goal model, to be able to identify the correct action to be supported. Solutions for these three aspects are *challenging* and allow for a *variety of design decisions* but to solve each of them would allow to process goal models in generative approaches for assistive systems.

We discuss the following three steps for systems that rely on MDSE methods. This means that they already use a combination of different models and modeling languages to define the resulting system. They use structure and behavior models to define, *e.g.,* relevant data structures in Class Diagrams (CDs), and represent human behavior in Business Process Model and Notation (BPMN) models or UML Activity Diagrams (ADs).

(A) Creation of goal models. Clearly, it is a *challenge* to create goal models from scratch during observation of actions. Thus, at least a part of these models should be defined at the set-up of an assistive system and could be extended by the assisted person during run-time. To be able to define goal models, we would (1) either have to engineer a new DSL to describe goals or (2) include the relevant concepts in an already existing structure model, *e.g.,* domain CDs, to be extendable at run-time. In both of these solutions, it is necessary to conceptualize the term goal and to identify relevant related concepts.

(B) Relation of goal models to models in other modeling languages. If we want to include support functionalities in systems created with MDSE methods, *e.g.,* [Ad18, Ge20, Ad20], the generator(s) need to know about the relationship between models created with different DSLs, especially those modeling *structure* and *behavior*.

Figure 1 shows the relationships between models from different DSLs. They have to be defined for a concrete application in the model engineering activity during design time.

1. A structure model, using *e.g.,* UML CD during development time or UML Object Diagrams during run-time, describes the relevant concepts of the domain and their relations. Using the aforementioned example this could be, *e.g.,* some cups in a cupboard, an assisted person in the kitchen, the coffee machine on a board, or a package of milk in the fridge.
2. Behavior models, using *e.g.,* BPMN or UML ADs, are created in a next step. This includes processes an assisted person can do, *e.g.,* prepare coffee, drink coffee, bake a cake, or take some medicine before eating breakfast.
   a) Behavior models have connections to structure models (relation (a) in Figure 1), *e.g.,* a „cup“ (instance from class X) is used in action A „prepare coffee“.
3. In a third step, the goal model is added, using *e.g.,* the new Goal DSL. Goals are defined as states, *e.g.,* „coffee is prepared“, „coffee is consumed“, „a cake is baked“, or „medicine is taken“. There exist relationships between goals, *e.g.,* „coffee is consumed“ is only done after „coffee is prepared“ or „coffee is bought“.
   b) A goal includes additional information about objects and their states (relation (b) in Figure 1), *e.g.,* the goal G „coffee is prepared“ has a connection to the „cup“ (instance from class X) to be able to know when it is full.
   c) Which relationships exist between goals and behavioral models depends on the concepts included in a *Goal DSL.* As described in [MM13], *activity theory* [Le78] could be used to further investigate these relationships. [Le78] states that actions (concrete steps) are directed towards certain conscious goals. Thus, we could assume that a goal is related with an action (relation (c) in Figure 1), *e.g.,* action A „prepare coffee“ has the goal G „coffee is prepared“.

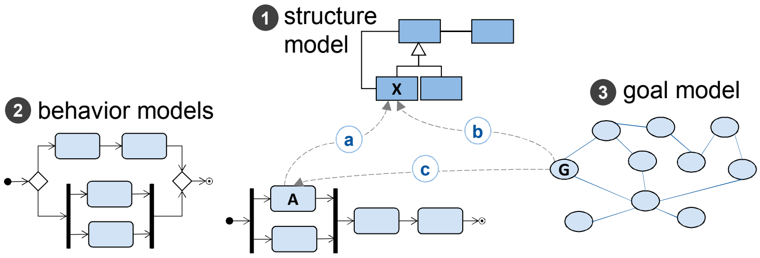

Fig. 1: Relationship between structure model, actions in behavior models, and goals.

Goal models need information about resources, their states, and actions which leads to the *challenge* keeping models in different DSLs consistent. Using a *Goal DSL* might require to include relations between goal models and models in other DSLs directly, *e.g.,* by referencing elements of other models, or using external tagging languages [Gr15]. This makes it easier to handle consistency between models.

(C) Identification of goals from observation. The idea of using human behavioral goals to assist arose from the analysis of cognitive processes. Usually, if someone wants to identify the goal of the activity of another person she is observing, she uses (1) her perception, and combines it with (2) domain-specific knowledge and (3) general knowledge. By taking an observation of real-life as an *example*, it is easy to see, that such goal identification is done by humans automatically several times a day: We observe that somebody puts coffee into a coffee cup. We have domain-specific context information, *e.g.,* in which room the person is, what time of the day it is, and we have general information about humans, *e.g.,* that they drink and eat regularly or that they have a job and work. Using this information, we can conclude that a person's goal is to drink the coffee or to serve it to another person, depending on the context. An assistive system could use these types of information as well:

- To handle *perception*, human actions can be detected by using sensory data (one or more events) at a certain point in time (see (1) observation in Figure 2). Using this data, *e.g.,* more complex activities of human behavior could be detected by activity recognition systems and they could be identified in existing process models (action A in observation and behavior models in Figure 2).
- *Domain-Specific Knowledge* about a certain domain includes information about the environment, spatial, personal, and further behavioral information. This could be represented using context modeling approaches [MS17] and modeled in structure models (see (1) in Figure 1) as well as behavior models (see (2) in Figure 1).
- *General Knowledge* such as common human behavior or physical properties can be additionally modeled in structure and behavior models. This knowledge can be either described in additional models or added to the domain-specific models. Until now, it is not very common to include general knowledge in MDSE. Thus, further investigations are required to gain knowledge about how these models are related to domain-specific models and what strategy to follow in the sense of extendability.

It is possible to manage the necessary information through different kinds of models. Nevertheless, it is still a *challenge* for a system to automatically carry out the last step, the identification of a support action based on observations.

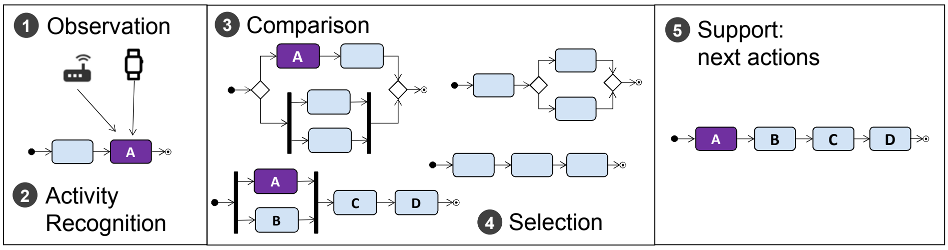

Fig. 2: Identification of an action and its goal from observation.

Figure 2 shows a possible process for identifying support actions in *assistive systems*. Using the terminology from [Le78], actions could have specific goals a person wants to achieve. For a successful support process, assistive systems must pass several steps.

1. **Observation.** The assistive system observes human behavior and gets sensor data (simple events from sensor data in observations), *e.g.,* the light sensor detects light in the kitchen or the motion sensor detects somebody in front of the coffee machine. Knowledge about the sensor's location or its relation to a particular item is usually integrated through hand-written code or by modeling.
2. **Activity recognition.** The simple events are analyzed and complex events or actions are build (action A in Figure 2) [RAM16], *e.g.,* someone is detected in front of the coffee machine (simple event 1) and the start button of the button machine is pressed (simple event 2) means somebody makes coffee (complex event or action).
3. **Comparison.** The system compares a specific activity with existing knowledge, *e.g.,* in behavior models (Figure 2, dark marked actions in observation and behavior models). This comparison results in several candidates in different behavior models.
4. **Selection.** A reasoning component selects the most fitting process based on context information (Figure 1, relation (a)), possible goals according to current states of objects (Figure 1, relation (c)) and information in case bases or knowledge bases about former successful support [AP94]. In our example, this would be *e.g.,* the process including action A, B, C, D in Figure 2.
5. **Support.** The assistive system is now able to guide the person showing the next possible actions to perform if she/he hesitates or behaves unexpectedly [SM20], *e.g.,* actions B, C, and D as next steps.

To sum up, there are different variants to realize these goal modeling, relations to DSLs and goals, and the identification of goals from observation. Goal modeling and a *Goal DSL* might be one way to improve the precision of support activities. However, concrete solutions for these ideas need further investigation.

4 Conclusion

There exist several approaches for robots, self-adaptive systems, or agent-based systems that use goal modeling for influencing the behavior of such systems and robots [Br05, He15]. Nevertheless, to model task goals for usage in MDSE approaches to create assistive systems is an open field that needs further investigation. This paper does not claim to solve all behavior goal modeling challenges, but we hope that it serves as a stimulus for discussions.

References

[Ad18] Adam, Kai; Netz, Lukas; Varga, Simon; Michael, Judith; Rumpe, Bernhard; Heuser, Patricia; Letmathe, Peter: Model-Based Generation of Enterprise Information Systems. In: Enterprise Modeling and Information Systems Architectures (EMISA'18). volume 2097. CEUR-WS.org, pp. 75–79, 2018.

[Ad20] Adam, Kai; Michael, Judith; Netz, Lukas; Rumpe, Bernhard; Varga, Simon: Enterprise Information Systems in Academia and Practice: Lessons learned from a MBSE Project. In: 40 Years EMISA (EMISA'19). LNI P-304. GI, pp. 59–66, 2020.

[AMM14] Al Machot, Fadi; Mayr, Heinrich C.; Michael, Judith: Behavior Modeling and Reasoning for Ambient Support: HCM-L Modeler. In: Proc. Int. Conf. on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE 2014). LNAI, 2014.

[An96] Antón, Annie I.: Goal-based requirements analysis. In: Proc. Second Int. Conf. on Requirements Engineering. IEEE, pp. 136–144, 1996.

[AP94] Aamodt, Agnar; Plaza, Enric: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI COMMUNICATIONS, 7(1):39–59, 1994.

[Be19] Bernabé, César Henrique; Silva Souza, Vítor E.; Almeida Falbo, Ricardo de; Guizzardi, Renata S. S.; Silva, Carla: GORO 2.0: Evolving an Ontology for Goal-Oriented Requirements Engineering. In: Advances in Conceptual Modeling, LNCS. Springer, 2019.

[BF16] Bock, Alexander; Frank, Ulrich: MEMO GoalML: A Context-Enriched Modeling Language to Support Reflective Organizational Goal Planning and Decision Processes. In: Conceptual Modeling, volume 9974 of LNCS, pp. 515–529. Springer Int., 2016.

[Br05] Braubach, Lars; Pokahr, Alexander; Moldt, Daniel; Lamersdorf, Winfried: Goal Representation for BDI Agent Systems. In: Programming Multi-Agent Systems. Springer, pp. 44–65, 2005.

[Ca88] Carberry, S: Modeling the user's plans and goals. Computational Ling., 14:23–37, 1988.

[Cv14] Cailliau, A.; van Lamsweerde, A.: Integrating exception handling in goal models. In: 22nd Int. Requirements Engineering Conf. (RE). IEEE, pp. 43–52, 2014.

[DvLF93] Dardenne, Anne; van Lamsweerde, Axel; Fickas, Stephen: Goal-directed requirements acquisition. Science of Computer Programming, 20(1):3 – 50, 1993.

[EW02] Eccles, Jacquelynne S.; Wigfield, Allan: Motivational beliefs, values, and goals. Annual review of psychology, 53:109–132, 2002.

[EYZ09] Elahi, Golnaz; Yu, Eric; Zannone, Nicola: A Modeling Ontology for Integrating Vulnerabilities into Security Requirements Conceptual Foundations. In: Conceptual Modeling (ER'09). Springer Berlin Heidelberg, pp. 99–114, 2009.

[GA20] Ghasemi, Mahdi; Amyot, Daniel: From event logs to goals: a systematic literature review of goal-oriented process mining. Requirements Engineering, 25(1):67–93, 2020.

[Ge20] Gerasimov, Arkadii; Heuser, Patricia; Keteniß, Holger; Letmathe, Peter; Michael, Judith; Netz, Lukas; Rumpe, Bernhard; Varga, Simon: Generated Enterprise Information Systems: MDSE for Maintainable Co-Development of Frontend and Backend. In: Modellierung 2020 Short, Workshop and Tools & Demo Papers. CEUR-WS.org, pp. 22–30, 2020.

[Go18] Gonçalves, Enyo; Castro, Jaelson; Araújo, João; Heineck, Tiago: A Systematic Literature Review of iStar extensions. Journal of Systems and Software, 137:1 – 33, 2018.

[Gr15] Greifenberg, Timo; Look, Markus; Roidl, Sebastian; Rumpe, Bernhard: Engineering Tagging Languages for DSLs. In: Conf. on Model Driven Engineering Languages and Systems (MODELS'15). ACM/IEEE, 2015.

[He15] Heim, Robert; Mir Seyed Nazari, Pedram; Ringert, Jan Oliver; Rumpe, Bernhard; Wortmann, Andreas: Modeling Robot and World Interfaces for Reusable Tasks. In: Intelligent Robots and Systems Conference (IROS'15). IEEE, pp. 1793–1798, 2015.

[HML15] Horkoff, Jennifer; Maiden, Neil; Lockerbie, James: Creativity and Goal Modeling for Software Requirements Engineering. In: Proc. ACM SIGCHI Conference on Creativity and Cognition. C&C '15. ACM, pp. 165–168, 2015.

[Ho15] Horkoff, Jennifer; Li, Tong; Li, Feng-Lin; Salnitri, Mattia; Cardoso, Evellin; Giorgini, Paolo; Mylopoulos, John: Using Goal Models Downstream. International Journal of Information System Modeling and Design, 6(2):1–42, 2015.

[HY16] Horkoff, Jennifer; Yu, Eric: Interactive Goal Model Analysis for Early Requirements Engineering. Requirements Engineering, 21(1):29–61, 2016.

[Ka01] Kaptelinin, Victor: Activity theory: implications for human-computer interaction. In: Context and consciousness. Activity theory and human-computer interaction. MIT Press, pp. 103–116, 2001.

[KL05] Kavakli, Evangelia; Loucopoulos, Pericles: Goal Modeling in Requirements Engineering. In: Information Modeling Methods and Methodologies, pp. 102–124. IGI Global, 2005.

[Kr08] Krogstie, John: Using EEML for Combined Goal and Process Oriented Modeling: Evaluation through a Case Study. In: Proc. 13th Int. WS on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD'08) at CAiSE'08. pp. 112–129, 2008.

[Kr14] Kraiem, Naoufel; Kaffela, Houda; Dimassi, Jamil; Al Khanjar, Zuhoor: Mapping from MAP Models to BPMN Processes. Journal of Software Engineering, 8(4):252–264, 2014.

[Le78] Leont'ev, Aleksej N.: Activity, Consciousness, and Personality. Prentice-Hall, Englewood Cliffs, NJ, 1978.

[Mc03] McCrickard, D. Scott; Chewar, C. M.; Somervell, Jacob P.; Ndiwalana, Ali: A model for notification systems evaluation—assessing user goals for multitasking activity. ACM Transactions on Computer-Human Interaction (TOCHI), 10(4):312–338, 2003.

[MCN92] Mylopoulos, J.; Chung, L.; Nixon, B.: Representing and using nonfunctional requirements: a process-oriented approach. IEEE Trans. on Software Engineering, 18(6):483–497, 1992.

[Mi18] Michael, Judith; Steinberger, Claudia; Shekhovtsov, Vladimir A.; Al Machot, Fadi; Ranasinghe, Suneth; Morak, Gert: The HBMS Story - Past and Future of an Active Assistance Approach. EMISA Journal, 13:345–370, 2018.

[MM13] Michael, Judith; Mayr, Heinrich C.: Conceptual Modeling for Ambient Assistance. In: Conceptual Modeling - ER 2013. volume 8217 of LNCS. Springer, pp. 403–413, 2013.

[MS17] Michael, Judith; Steinberger, Claudia: Context Modeling for Active Assistance. In: Proc. of the ER Forum and Demo Track 2017. CEUR-WS.org, pp. 221–234, 2017.

[No11] Nogueira Santos, Fabiana Jack; Sampaio do Prado Leite, Julio Cesar; Cappelli, Claudia; Batista, Thaís Vasconcelos; Santoro, Flávia Maria: Using Goals to Identify Aspects in Business Process Models. In: Int. Workshop on Early Aspects. ACM, pp. 19–23, 2011.

[OFK15] Overbeek, Sietse; Frank, Ulrich; Köhling, Christian: A language for multi-perspective goal modelling: Challenges, requirements and solutions. Computer Standards & Interfaces, 38:1–16, 2015.

[RAM16] Ranasinghe, Suneth; Al Machot, Fadi; Mayr, Heinrich C: A review on applications of activity recognition systems with regard to performance and evaluation. International Journal of Distributed Sensor Networks, 12(8), 2016.

[RS05] Rolland, Colette; Salinesi, Camille: Modeling Goals and Reasoning with Them. In: Engineering and Managing Software Requirements, pp. 189–217. Springer, 2005.

[SM20] Steinberger, Claudia; Michael, Judith: Using Semantic Markup to Boost Context Awareness for Assistive Systems. In: Smart Assisted Living: Toward An Open Smart-Home Infrastructure. Springer Int. Pub., pp. 227–246, 2020.

[SS13] Sutcliffe, Alistair; Sawyer, Pete: Modeling Personalized Adaptive Systems. In: Advanced Information Systems Engineering. Springer, pp. 178–192, 2013.

[YM93] Yu, Eric S. K.; Mylopoulos, John: An Actor Dependency Model of Organizational Work: With Application to Business Process Reengineering. In: Proc. of the Conference on Organizational Computing Systems. COCS '93. ACM, pp. 258–268, 1993.