# Configuration Model for Networked Collaborative Cyber-physical Systems

Jennifer Brings, Marian Daun
Paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen, Essen, Germany
{jennifer.brings, marian.daun}@paluno.uni-due.de

***ABSTRACT:*** *Networking can assist to build Collaborative cyber-physical systems. These networks can achieve goals that are unachievable to individual systems. Due to their dynamic natures such networks must handle a variety of configurations which influence the goals that can be achievable. In this paper we study the use of an approach combining a goal and a special configuration model to reason about permissible and unwanted network configurations. The first level assessment outcome from an industrial case study specified that the approach can help discover problematic network configurations for networks of collaborative cyber-physical systems.*

**Keywords:** Goal Modelling, Collaborative Cyber-physical Systems, iStar

**Received:** 8 June 2019, Revised 22 September 2019, Accepted 1 October 2019

**DOI:** 10.6025/jisr/2019/10/4/124-133

## 1. Introduction

Goal modeling, goal-oriented requirements engineering, and goal-based analysis [1] are commonly used to elicit, document, and analyze requirements. Particularly goal models are used to identify conflicts between different stakeholders [2]. However, in previous work we have shown that goal models are also a good fit to foster analysis of collaborative cyber-physical systems [3]. In this situation, multiple systems collaborate with each other to a) achieve some greater good of the so formed overall system network and b) to maximize their individual goal fulfillment. The achievable goals of the system network as well as of the partaking individual systems, thus, greatly depend on their context [4] and in particular on the configuration of the system network [5]. Therefore, it is necessary to reason about the goals to be achieved in each possible configuration and to reason about the configurations in which certain goals can be achieved.

Based on the related work we developed a general solution approach, which generates certain views onto the goal model. In this paper, we report evaluation results from applying this overall approach to the industrial case example of autonomous transport robots taken from the automotive domain.

Therefore, this paper is structured as follows: Section 2 introduces the concrete problem statement and the industrial case example, based on this the related work is discussed in Section 3. Resulting from the related work the general approach is introduced and applied to the case example and results are shown in Section 4. Finally, Section 5 concludes the paper.

## 2. Background

### 2.1 Problem Statement

Collaborative cyber-physical systems can form networks which can achieve more than the individual systems by themselves [6, 7]. For, example vehicles equipped with collaborative adaptive cruise control systems can form platoons which enable all following vehicles to reduce their fuel consumption because the platooning functionality allows them to safely reduce the distance needed to the vehicle ahead. These networks are of a dynamic nature as they can change their configurations during runtime [8]. Vehicles may form a new platoon or join and leave existing platoons at any time. Goal modeling has proven useful to document and reason about the goals of such networks, its individual collaborative cyber-physical systems, and their dependencies. However, goal models are typically used to document goals of individual systems (which might consist of other systems as, e.g., is the case for systems of systems) but not for documenting goals of a network of collaborative cyber-physical systems that consists of a plethora of systems collaborating, particularly considering that many of these systems are of similar types [9].

To allow for proper documentation of all possible network configurations, we introduced dynamicity constraint models in [5]. Dynamicity constraint models are based on cardinality-based feature models [10] and allow the definition of what constitutes a valid network configuration. For example, a valid platoon must consist of at least two vehicles and each platoon needs exactly one leading vehicle. While dynamicity constraint models capture the morphological variability of the system network, the goal model still needs to account for the various goals the overall system network and each individual network want to achieve. However, this largely depends on the concrete network configuration. For example, the goal of a platoon to split platoons into two should only be achievable if there are at least four vehicles in a platoon.

To better reason about the capabilities of the different configurations of such a network, it is necessary to be able to answer the following questions:

1. Which goals can be achieved by a given configuration?

2. Which goals can be achieved by all configurations in a set of given configurations?

3. Which goals can be achieved by at least one configuration from a set of given configurations?

### 2.2 Case Example

We illustrate the problem situation and later on in this paper the initial solution idea using an industrial case example from the automation domain. Smart factories are designed to adapt themselves to different production orders [11]. Therefore, smart factories need to decide whether a certain production order is producible and how this should be conducted. To this end, autonomous transport robots are used to allow for flexible material flows between different production sites. A fleet of transport robots is tasked with transporting goods from one or more pick-up stations to one or more drop-off stations. A simplified dynamicity constraint model, a fragment of the respective goal model and the trace links between the dynamicity constraint model and the goal model are shown in Figure 1.

The network consists of one to 20 robots which can be of three different types, one to five pick-up stations, and one to five drop-off stations. Various goals of the fleet are linked to the dynamicity constraint model. For example, the *Low Maintenance Costs* softgoal excludes any robots of type III, as those are more expensive to maintain than the other types. However, only robots of type III can adjust the height at which they pick up and drop off goods and are thus needed to achieve the *Pick-Ups/Drop-Offs at Different Heights* goal. For a more detailed discussion of the case example please refer to [5].

### 2.3 Illustration of Problem Situation

Coming back to the questions to be reasoned about from Section 1, for example, we would like to be able to answer questions such as which goals can be achieved by a fleet of one robot of type III, one pick-up station of type I, and one drop of station of type II. Other questions we seek answers to are questions like what goals can all configurations achieve in a fleet of one or two robots, but not more than one of type II or III each, one pick-up station of type II, and one drop-off station of type II and what goals can at least one configuration of such a fleet achieve. Table 1 relates the examples given to the kind of questions they are and provides rationales why it is important to be able to answer such questions.

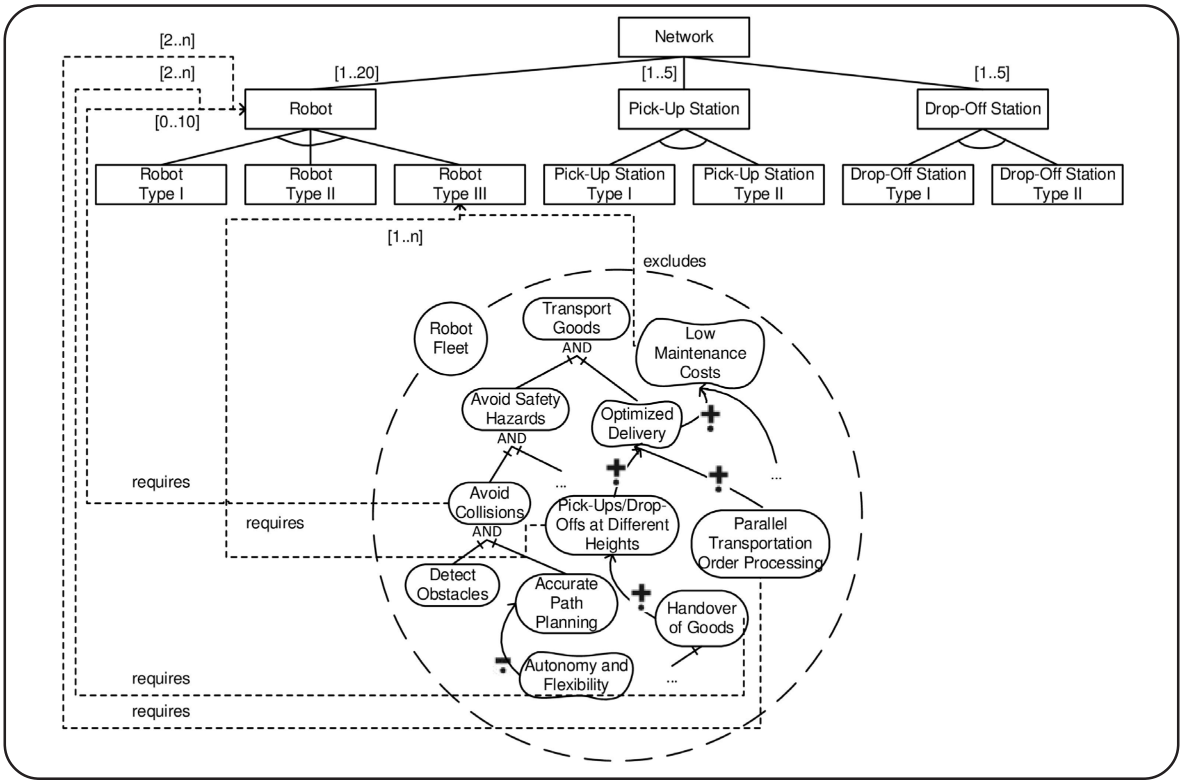

Figure 1. Goal model and dynamicity constraint model for a transport robot fleet

| **Question** | **Example** | **Rationale** |
|---|---|---|
| Which goals can be achieved by a given configuration? | Which goals can be achieved by a fleet of one robot of type III, one pick-up station of type I, and one drop-off station of type II? | It is important to investigate the different configurations and the achievable goals for each configuration. This allows, for instance in identifying network configurations that are unwanted (i.e. that do not sufficiently achieve necessary goals) and, thus, need to be prevented from forming. |
| Which goals can be achieved by all configurations in a set of given configurations? | Which goals can all configurations achieve in a fleet of one or two robots, but not more than one of type II or III each, one pick-up station of type II, and one drop-off station of type II? | For engineers it is important to reason about commonalities between configurations. It is important to investigate which goals will be achieved any time and which only under certain circumstances. |
| Which goals can be achieved by at least one configuration from a set of given configurations? | Which goals can at least one configuration achieve in a fleet of one or two robots, but not more than one of type II or III each, one pick-up station of type II, and one drop-off station of type II? | For engineers it is important to investigate which goals are achievable by which configuration. For instance, it must be ensured that goals are achievable by at least one configuration. |

Table 1. Analytical questions for goal modeling of networks of collaborative cyber-physical system

## 3. Related Work

This section reviews the related work w.r.t. the problem situation and the questions to be answered outlined in Section 2. In particular, approaches exist that combine goal and variability modeling, approaches that derive feature models from goal models, approaches reasoning about goal configurations and goal-modeling approaches considering specific contexts and configurations.

### 3.1 Approaches for using Goal Models for Documenting Variability

Several approaches exist for creating goal models that contain information about variability. Goal models allow for the decomposition of goals into subgoals. These approaches assume that OR decompositions indicate variability. As most goal modeling languages only define one or two types of OR decomposition (inclusive OR and exclusive OR), extensions have been developed to allow for documenting more complex relationships such as entirely optional goals, selection of a certain number of different goals etc. To address this issue. Borba and Silva [12] propose an extension to i* that distinguishes seven different kinds of means-end links. Silva et al. [13] presents an aspect- oriented i* approach to model common and variable features of a product line using a goal model. Variable goals, tasks, resources etc. are assigned to aspects according to the feature model that describes the product line. Semmak et al. [14] [15] extend the KAOS method to support the generation of domain goal models that document all goals of an entire product line. Liu et al. [16], [17] extend the GRL modeling language with feature modeling concepts. Santos et al. [18] presents an approach for using goal models to document the variability of BPMN process models. While these approaches succeed at documenting product line variability in goal models, they are not able to cope with some of the characteristics of networks of collaborative cyber-physical systems such as the possible existence of multiple systems of the same type in such a network and the need to alter the number of permissible systems in such a network according to selected goals.

### 3.2 Approaches for Deriving Feature Models from Goal Models

Approaches also exist for deriving feature models describing a product line from existing goal models. Yu et al. [19] present an approach for deriving feature models from goal models. AND-decompositions in the goal models are mapped to mandatory features. OR-Decompositions are mapped to OR features. A special OR-decomposition is defined that can be mapped to alternative features. The approach is extended in [20] to distinguish two kinds of "true" OR-decompositions. When subgoals can be selected at design time because the selection is based on quality criteria they are mapped to OR features. Otherwise they are mapped to mandatory features. Asadi et al. [21] present an approach for creating feature models based on i* goal models. Goal and feature models are linked by annotating features with Boolean variables that correspond to goals from the goal model. This way the goals can be used as rationales for feature selection.

### 3.3 Approaches for finding Optimal Configurations

Introducing variability into goal models or linking goal and feature models is usually done to find optimal configurations for product lines. Gonzales-Baixauli et al. [22] present an approach to select the most appropriate variant based on softgoals. Two AND/OR –trees are created manually, one containing all goals and tasks of the system, one containing all softgoals. A variant is represented by an OR-path in the goal/task tree. The goals/tasks are manually linked to softgoals using correlation links. Softgoals are assigned priority values. Based on these parameters the tool computes a score for each variant to optimize softgoal fulfilment. António et al. [23] present an approach that combines i* goal modeling and feature models, too. First domain goal models are created, then a feature model of the product line is derived. In application engineering feature and goal models are configured to create models for one product. Silva et al. [24] present a process to identify features and configure feature models. Noorian et al. [25] presents an approach to better select features of product lines according to stakeholder needs. To this end their approach semi-automatically maps elements from goal models and feature models. The approach requires the existence of trace links between the feature model and natural language documents (e.g. interview transcripts) as well as between the goal model and natural language documents. Using natural language processing techniques, the natural language documents are automatically analyzed, and the feature model and the goal model are annotated accordingly, and mappings are identified using integer linear programming. This work is extended in [26] which presents a process for finding the optimal configuration for a product line product bases on stakeholder needs. The feature model, the goal model, and their relations are transformed into an integer linear program which is then used to find the optimal solution. Liu et al. [16], [17] use their extended GRL modeling languages and extend the GRL evaluation algorithm so it can be used to derive valid product configurations.

These approaches aim at finding product configurations that best align with stakeholder goals, often taking the fulfilment level of softgoals into account. The questions we are interested in, however, do not seek an optimal solution but rather aim at detecting configurations that could lead to problems such as the inability to perform certain tasks.

### 3.4 Approaches for finding Configurations for Specific Contexts

Beside regular product lines the relation between goal modeling and variability modeling is also of interest for the development of systems that need to adapt to a dynamic context. In [27] the authors extends the approach from [24] for dynamic software product lines. Changes in the contexts a dynamic software product line can be subject to lead to reconfigurations of the product line to maximize softgoal satisfaction. Ali et al. [28] present a goal modeling approach for systems with variable context. The TROPOS goal model is annotated with facts, which determine when an element from the goal model is valid. Facts are combined into formulae which describe relevant contexts. Consistency checks are performed to ensure contradictory contexts are detected. Conflict analysis is performed to detect conflicting actions and conflicts over resources. Asadi et al. [29] presents an approach that transforms GRL goal models and feature models and their relations into description logic and checks for inconsistencies. Lapouchnian et al. [30], [31] present an approach for parametrized goal models for various contexts. Elements in the goal model are annotated with tags which define in which context an element in valid. Context parameters can be arranged in hierarchies. If a subparameter is assigned to a goal so are the superparameters.

## 4. Case Study Application

From Section 3 results following basic approach, which is detailed in Section 4.1. We evaluated this basic idea using the industrial case example from Section 2.2 and outline the results in the following subsections.

### 4.1 Initial Solution Idea

To give support in answering the questions 1 to 3 from Section 2.1, we propose the generation of dedicated views on the goal model. Figure 2 illustrates our approach. Based on one or a set of configurations and the information from the goal model, the dynamicity constraint model, and their trace links two kinds of views for the goal model can be created. One view containing all elements that are valid for all configurations and one view containing all elements that are valid for at least one of the configurations in this set. Creating such a view for exactly one configuration can be considered a special case where those two views are always identical.

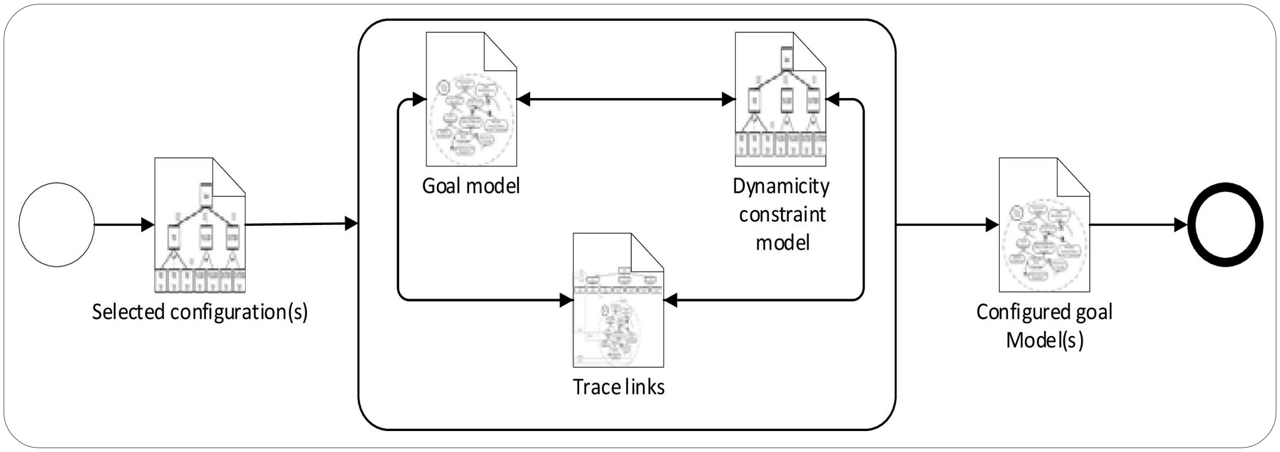

Figure 2. Conceptual overview

While this basic solution idea is based upon the approaches outlined in Section 3, in the following we will focus on assessing the applicability of this idea using the case example from Section 2.2.

### 4.2 Generation of Goal Models based on Configuration Models

Figure 3 illustrates the generation of a goal model based on one given configuration. For illustration purpose a simple network configuration was chosen, consisting of one robot of type III, one pick-up station of type I, and one drop-off station of type II. As a consequence, the softgoal *Low Maintenance Costs* and the goals *Pick-Ups/Drop-Offs at Different Heights* and *Handover of Goods* where goods are exchanged between robots mid-transport cannot be fulfilled and are removed from the goal model to show which goals the given configuration can achieve.

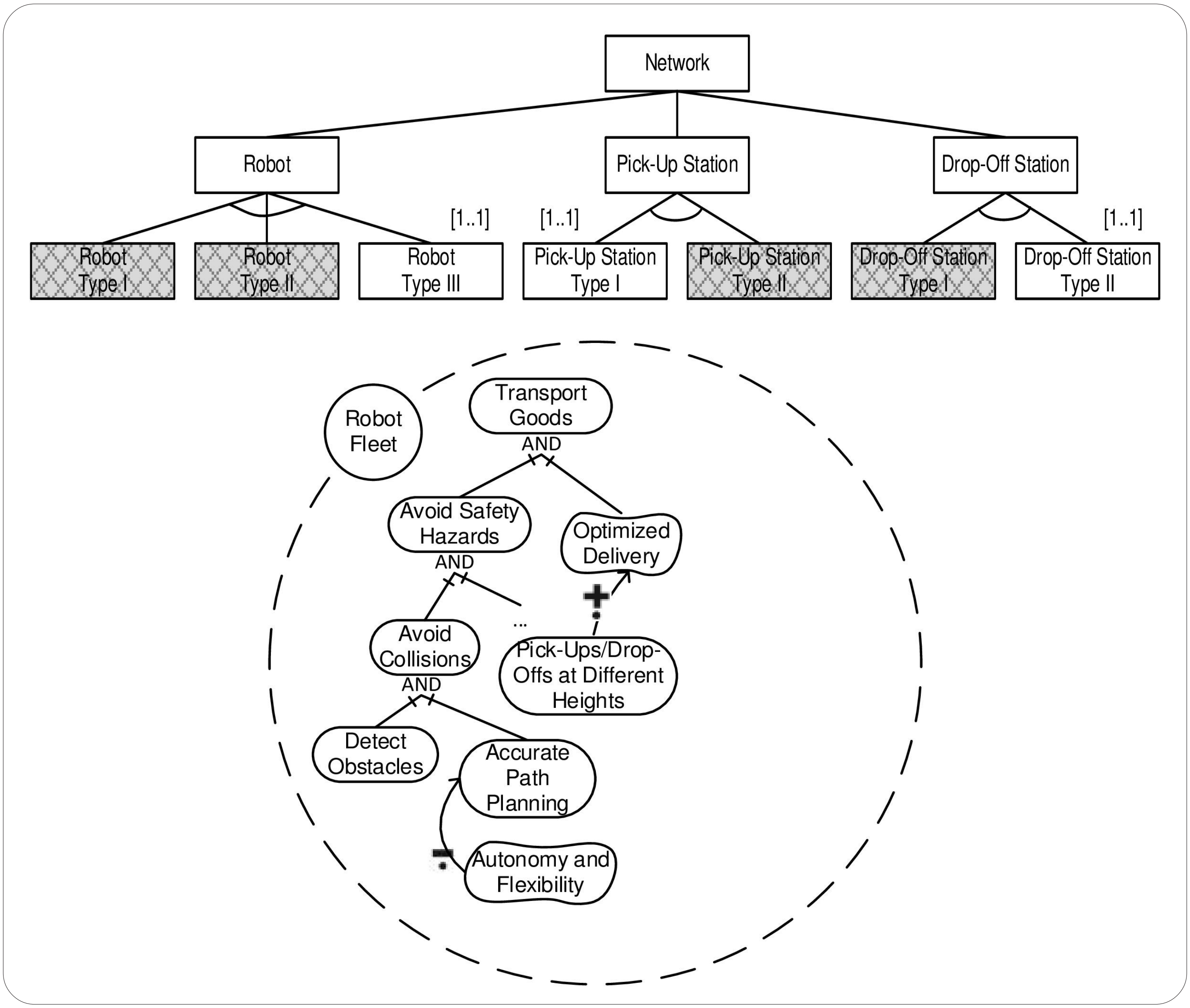

Figure 3. Generation of a goal model based on one specific configuration

Goal models cannot only be generated for one particular configuration but also for a set of configurations. This is of particular interest as these kinds of networks are of a dynamic nature and can change their morphology during runtime. The dynamicity constraint model shown in Figure 4 defines a network that consists of one or two robots, but not more than one of type II or III each, one pick-up station of type II, and one dropoff station of type II. The goal model in Figure 4 shows all goals that can be achieved by all configurations in this set. Again, the softgoal *Low Maintenance Costs* and the goals *Pick-Ups/Drop-Offs at Different Heights* and *Handover of Goods* were removed because having a robot of type III in the network excludes the softgoal *Low Maintenance Costs* and having only one robot in the network excludes the goals *Pick-Ups/Drop-Offs at Different Heights* and *Handover of Goods.*

To explore which goals can be achieved by at least one configuration from a given set of configurations a goal model can be generated accordingly. Figure 5 illustrates this for the set of configurations introduced in Figure 4 Unlike the goal model in Figure 4, this goal model contains the softgoal *Low Maintenance Costs* because if there are no robots of type III in the network this softgoal can be achieved sufficiently. Moreover, the goal *Pick-Ups/Drop-Offs at Different Heights* is included in the goal model because the network may contain a robot of type III and the *Handover of Goods* is included in the goal model because the network may contain up to two robots.

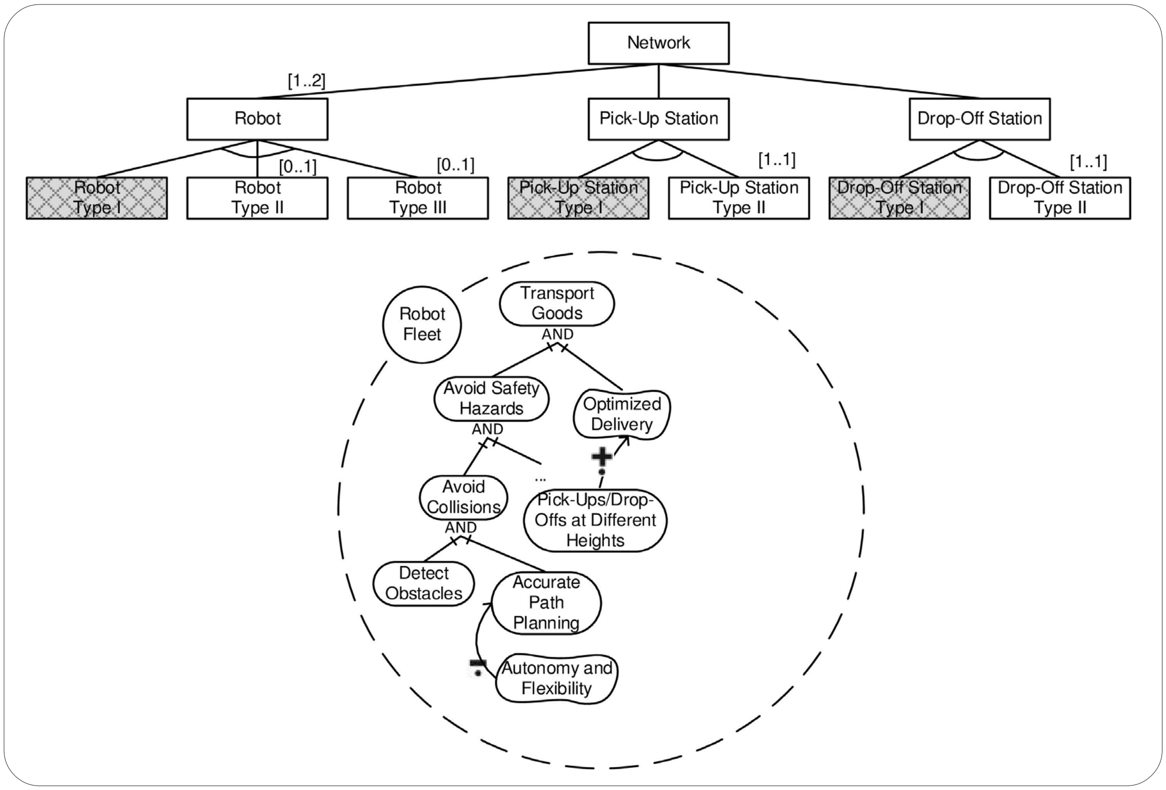

Figure 4. Generation of goal model containing only goals valid in all selected configurations

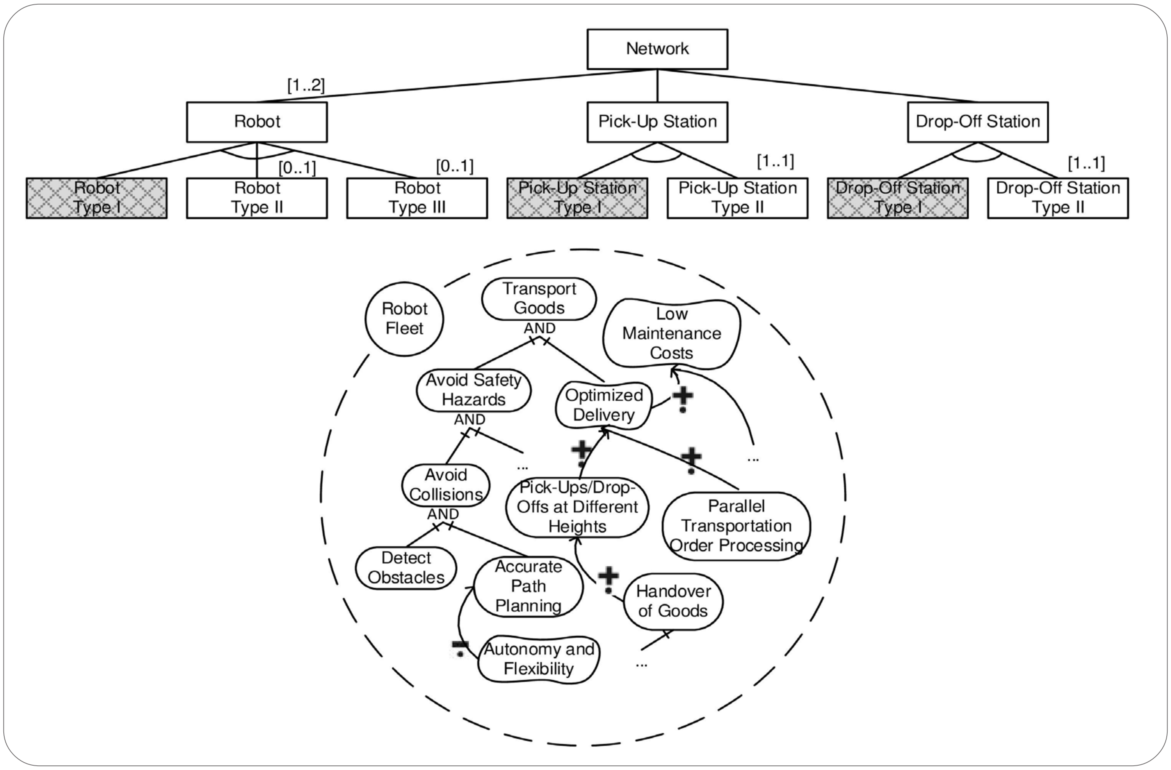

Figure 5. Goal model containing all goals valid in at least one selected configuration

## 5. Lessons Learned

The case study application has shown that the approach can help identify achievable and unachievable goals for different configurations of the system network. This allows for identifying needed and problematic configurations so that developers can deal with these accordingly. For example, developers can shut off functionalities in the individual systems in cases where the current network configuration cannot make use of those functionalities or developers can prevent systems from connecting to a network if that would cause problems.

The approach allows for considering various analytical questions such as the achievability of goals for one particular configuration as well as the achievability of goals for a set of configurations, considering the goals that can be achieved by all of these configurations or at least by one of these configurations. This allows for reasoning about the capabilities of such a network in terms of what can be achieved if the network assumes the needed configuration and what can always be achieved regardless of the current configuration.

To further investigate dependencies between network configurations and goals, it might be useful to consider the opposite direction. I.e. investigating which configurations can achieve a particular goal or which configurations can achieve a set of goals and if there are tradeoffs to achieving a particular or a set of particular goals. I.e., does this prevent the network from achieving other goals?

While the approach considers the dependencies between the network configurations and their goals it might also be worthwhile investigating the dependencies between network configurations and the goals of individual systems as the fulfilment of goals by the network ultimately depends on the fulfilment of goals by the individual systems in the network.

## 6. Conclusion

For the development of collaborative cyber-physical systems, which can dynamically form networks at runtime, it is of importance to investigate which configurations might be problematic. Based on the configuration of the network, the network is able to achieve certain goals or not. Therefore, it is necessary to investigate, which goals can be achieved by which configuration and which configuration can achieve which goals. Furthermore, it is important to identify commonalities in goal fulfilment between different configurations.

Based on contributions from the related work, we developed a solution idea that generates configured goal models showing what goals one configuration or a set of configurations can achieve.

We have shown applicability of the general solution idea using an industrial case example from the automation domain. For the transport robot case example, it was possible to show that our approach provides answers to the questions: Which goals can be achieved by a given configuration? Which goals can be achieved by all configurations in a set of given configurations? Which goals can be achieved by at least one configuration from a set of given configurations?

Future work will deal with the formal definition of the approach considering more intricate relations between the goal and the dynamicity constraint models, such as different kinds of goal decompositions for different configurations, and with more thorough evaluation including prototypical implementation. Also, interesting to investigate will be the configuration of networks based on a selection of goals. Due to dynamic nature of such networks, it might also be interesting to further investigate the impact of unforeseen situations such as systems joining the network that were not anticipated at design time.

## Acknowledgements

This research was partly funded by the German Federal Ministry of Education and Research (grant no. 01IS16043V). We like to thank our industrial partners for their support. Namely, we thank Elham Mirzaei, Martin Neumann, and Jan Stefan Zernickel from InSystems Automation GmbH for their support with the case example.

## References

[1] Horkoff, J., Aydemir, F. B., Cardoso, E., Li, T., Maté, A., Paja, E., Salnitri, M., Piras, L., Mylopoulos, J., Giorgini, P. (2019). Goal-

oriented requirements engineering: an extended systematic mapping study. *Requirements Engineering*. 24, 133–160. https://doi.org/10.1007/s00766-017-0280-z.

[2] Lamsweerde, A. van., Darimont, R., Letier, E. (1998). Managing conflicts in goal-driven requirements engineering. *IEEE Transactions on Software Engineering*. 24, 908–926. https://doi.org/10.1109/32.730542.

[3] Daun, M., Stenkova, V., Krajinski, L., Brings, J., Bandyszak, T., Weyer, T. (2019). Goal Modeling for Collaborative Groups of Cyber-Physical Systems with GRL. *In*: Proceedings of the 32th ACM/SIGAPP Symposium on Applied Computing.

[4] Daun, M., Brings, J., Weyer, T., Tenbergen, B. (2016). Fostering concurrent engineering of cyberphysical systems a proposal for an ontological context framework. *In*: 3rd International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems, EITEC@CPSWeek 2016, Vienna, Austria, April 11, 2016. p. 5–10. *IEEE Computer Society*. https://doi.org/10.1109/EITEC.2016.7503689.

[5] Brings, J., Daun, M., Bandyszak, T., Stricker, V., Weyer, T., Mirzaei, E., Neumann, M., Zernickel, J. S. (2019). Model-based documentation of dynamicity constraints for collaborative cyber-physical system architectures: Findings from an industrial case study. *Journal of Systems Architecture*. 97, 153–167. https://doi.org/10.1016/j.sysarc.2019.02.012.

[6] Mosterman, P. J., Zander, J. (2016). Cyber-physical systems challenges: a needs analysis for collaborating embedded software systems. *Software and Systems Modeling*. 15, 5–16. https://doi.org/10.1007/s10270-015-0469-x.

[7] Tenbergen, B., Daun, M., Obe, P. A., Brings, J. (2018). View-Centric Context Modeling to Foster the Engineering of Cyber-Physical System Networks. *In*: IEEE International Conference on Software Architecture, ICSA 2018, Seattle, WA, USA, April 30 - May 4, 2018. p. 206–216. *IEEE*. https://doi.org/10.1109/ICSA.2018.00030.

[8] Broy, M., Schmidt, A. (2014). Challenges in Engineering Cyber-Physical Systems. *Computer*, 47, 70–72. https://doi.org/10.1109/MC.2014.30.

[9] Daun, M., Brings, J., Bandyszak, T., Bohn, P., Weyer, T. (2015). Collaborating Multiple System Instances of Smart Cyber-physical Systems: A Problem Situation, Solution Idea, and Remaining Research Challenges. *In*: Bures, T., Weyns, D., Klein, M., and Haber, R.E. (eds.) 1st IEEE/ACM International Workshop on Software Engineering for Smart Cyber-Physical Systems, SEsCPS 2015, Florence, Italy, May 17, 2015. p. 48–51. *IEEE Computer Society*. https://doi.org/10.1109/SEsCPS.2015.17.

[10] Czarnecki, K., Helsen, S., Eisenecker, U. (2005). Formalizing cardinality-based feature models and their specialization. *Software Process Improvement and Practice*. 10, 7–29. https://doi.org/10.1002/spip.213.

[11] Daun, M., Brings, J., Obe, P. A., Weiß, S., Böhm, B., Unverdorben, S. (2019). Using View-Based Architecture Descriptions to Aid in Automated Runtime Planning for a Smart Factory. *In*: 2019 IEEE International Conference on Software Architecture Companion (ICSA-C). p. 202–209. https://doi.org/10.1109/ICSA-C.2019.00043.

[12] Borba, C., Silva, C. (2009). A Comparison of Goal-Oriented Approaches to Model Software Product Lines Variability. *In*: Heuser, C.A. and Pernul, G. (eds.) Advances in Conceptual Modeling - Challenging Perspectives. p. 244–253. *Springer Berlin Heidelberg*. https://doi.org/10.1007/978-3-642-04947-7_30.

[13] Silva, C., Alencar, F., Araújo, J., Moreira, A., Castro, J. (2008). Tailoring an Aspectual Goal-Oriented Approach to Model Features. SEKE 2008. 472–477.

[14] Semmak, F., Gnaho, C., Laleau, R. (2008). Extended Kaos to Support Variability for Goal Oriented Requirements Reuse. MoDISE-EUS.

[15] Semmak, F., Laleau, R., Gnaho, C. (2009). Supporting variability in goal-based requirements. *In*: 2009 Third International Conference on Research Challenges in Information Science. p. 237–246. https://doi.org/10.1109/RCIS.2009.5089287.

[16] Liu, Yanji., Su, Yukun., Yin, Xinshang., Mussbacher, G. (2014). Combined propagation-based reasoning with goal and feature models. *In*: 2014 IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE). p. 27–36. https://doi.org/10.1109/MoDRE.2014.6890823.

[17] Liu, Yanji., Su, Yukun., Yin, Xinshang., Mussbacher, G. (2014). Combined goal and feature model reasoning with the User Requirements Notation and jUCMNav. *In*: 2014 IEEE 22nd International Requirements Engineering Conference (RE). p. 321–322. https://doi.org/10.1109/RE.2014.6912277.

[18] Santos, E., Castro, J., Sanchez, J., Pastor, O. (2010). A Goal-Oriented Approach for Variability in BPMN. WER.

[19] Yu, Y., Leite, do Prado, J. C. S., Lapouchnian, A., Mylopoulos, J. (2008). Configuring Features with Stakeholder Goals. *In*:

Proceedings of the 2008 ACM Symposium on Applied Computing. p. 645–649. ACM, New York, NY, USA. https://doi.org/10.1145/1363686.1363840.

[20] Yu, Y., Lapouchnian, A., Liaskos, S., Mylopoulos, J., Leite, J. C. S. P. (2008). From Goals to High- Variability Software Design. *In*: An, A., Matwin, S., Ras, Z. W., and Slezak, D. (eds.) *Foundations of Intelligent Systems*. p. 1–16. *Springer Berlin Heidelberg*. https://doi.org/10.1007/978-3-540-68123-6_1.

[21] Asadi, M., Bagheri, E., Gaševi, D., Hatala, M., Mohabbati, B. (2011). Goal-driven Software Product Line Engineering. *In*: Proceedings of the 2011 ACM Symposium on Applied Computing. p. 691–698. ACM, New York, NY, USA (2011). https://doi.org/10.1145/1982185.1982336.

[22] Gonzales-Baixauli, B., Leite, J. C. S. P., Mylopoulos, J. (2004). Visual variability analysis for goal models. *In*: Proceedings. 12th IEEE International Requirements Engineering Conference, p. 198–207. https://doi.org/10.1109/ICRE.2004.1335677.

[23] António, S., Araújo, J., Silva, C. (2009). Adapting the i* Framework for Software Product Lines. In: Heuser, C.A. and Pernul, G. (eds.) Advances in Conceptual Modeling - Challenging Perspectives. p. 286–295. *Springer Berlin Heidelberg*. https://doi.org/10.1007/978-3-642-04947-7_34.

[24] Silva, C., Borba, C., Castro, J. (2011). A Goal Oriented Approach to Identify and Configure Feature Models for Software Product Lines. WER. (2011).

[25] Noorian, M., Bagheri, E., Du, W. (2014). From Intentions to Decisions: Understanding Stakeholders' Objectives in Software Product Line Configuration. *International Conference on Software Engineering & Knowledge Engineering*. 671–677.

[26] Noorian, M., Bagheri, E., Du, W. (2017). Toward automated quality-centric product line configuration using intentional variability. *Journal of Software Evolution and Process*. 29, e1870. https://doi.org/10.1002/smr.1870.

[27] Guedes, G., Silva, C., Soares, M. (2017). Comparing Configuration Approaches for Dynamic Software Product Lines. *In*: Proceedings of the 31st Brazilian Symposium on Software Engineering. p. 134–143. ACM, New York, NY, USA. https://doi.org/10.1145/3131151.3131162.

[28] Ali, R., Dalpiaz, F., Giorgini, P. (2013). Reasoning with contextual requirements: Detecting inconsistency and conflicts. *Information and Software Technology*. 55, 35–57. https://doi.org/10.1016/j.infsof.2012.06.013.

[29] Asadi, M., Gröner, G., Mohabbati, B., Gaševi, D. (2016). Goal-oriented modeling and verification of feature-oriented product lines. *Software and Systems Modeling*. 15, 257–279. https://doi.org/10.1007/s10270-014-0402-8.

[30] Lapouchnian, A., Mylopoulos, J. (2011). Capturing contextual variability in i* models. *In*: Proceedings of the 5th International i* Workshop. p. 96–101.

[31] Lapouchnian, A., Mylopoulos, J. (2009). Modeling Domain Variability in Requirements Engineering with Contexts. *In*: Laender, A. H. F., Castano, S., Dayal, U., Casati, F., and de Oliveira, J. P. M. (eds.) Conceptual Modeling - ER 2009. p. 115–130. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04840-1_11.