Reusable Models in Industrial Automation: Experiences in Defining Appropriate Levels of Granularity

CameliaR. Maga*, Nasser Jazdi**, Peter Göhner**

* Bosch Software Innovations GmbH, 71332 Waiblingen, Germany,
(Tel. +49-0711-811-58146; e-mail: Camelia.Maga@bosch-si.com)
**Institute of Industrial Automation and Software Engineering, University of Stuttgart, Germany
(e-mail:{Nasser.Jazdi; Peter.Goehner}@ias.uni-stuttgart.de)

Abstract: Domain engineering approaches are widely used in the software engineering. They focus on building reusable models in a domain. At the Institute of Industrial Automation and Software Engineering (IAS) of the University of Stuttgart, a new domain engineering approach has been developed. The approach considers entire industrial automation systems, including hardware and software. Models that capture structure and behavior of entire systems have to be carefully tailored, in order to be reusable. The level of granularity deserves particular attention, since too coarse or too fine grained reusable models discourage reuse. This paper addresses the issue of finding appropriate levels of granularity and proposes a set of recommendations for determining appropriate levels of granularity for reusable models created during domain engineering.

1. INTRODUCTION

In current industrial automation, engineering is rarely performed from scratch. In most cases, there is a legacy principle within the domain (Ebert et al., 2003). This enables the creation of product lines instead of single systems. Product lines promise to reduce development costs and shorten time to market. With all major efforts undertaken to create product lines systematically, there are still some aspects that highly depend on the experience of the developer. Such an aspect is the level of granularity of the reusable models, which is one of the important factors for successful reuse (Griffel, 1998), (Burd et al., 1996).

The intention in building product lines is to capture information about industrial automation systems only once, in form of reusable models. These are created during the so called domain engineering phase and then repeatedly used in different projects, during the application engineering phase. Reusable models are structural, behavioural or crosscutting models of industrial automation systems. Examples of reusable models are domain requirements, software, hardware description models, variability models, reference architectures, test cases, workflows, documentations or domain dictionaries. The difficulty is to decide during domain engineering, at which level of granularity the models shall be designed, in order to be reusable. For instance, is it better to prepare a class, an algorithm or the entire software architecture for reuse? The issue of finding appropriate levels of granularity is neither for software engineering nor for systems engineering answered. Nevertheless, the necessity to consider the level of granularity has been initially recognized by software engineering. Hence, the definition of granularity stem from software engineering.

The paper is structured as follows: Section 2 discusses the definition of the term granularity in industrial automation and gives an overview on related work. Section 3 offers a presentation of a case study, in which domain engineering has been applied for passenger elevators. Section 4 focuses on faced difficulties, causes and proposed solutions to determine the appropriate granularity level of reusable models. Section 5 concludes with a summary and a set of recommendations for determining appropriate levels of granularity.

2. RELATED WORK

In the literature, there are different definitions of granularity related to software components. In (Griffel, 1998), (Szyperski et al., 2002), (Szdzuy, 2002) granularity refers to the code size, the proportion of component’s software related to the entire application and the number of component’s interfaces. These aspects apply for software components, but unfortunately they are not matching the idea of granularity of reusable models in industrial automation. For instance, it is unclear how to assess the number of interfaces of a reusable test case or of a reusable system requirements specification.

In (Oates et al., 2002) the authors present two different definitions of granularity. The first one refers to the level of detail of a model. For example, a table with many details has a high level of granularity. In contrast, an aggregated or summarized table has a low level of granularity. The second definition relates granularity to the level of data depth. For instance, the date/time dimension could be at the year, month, quarter, etc. level of granularity. The discrepancies regarding granularity go even further. In (Griffel, 1998) the author identifies three levels of granularity: fine-grained, medium-grained, and coarse-grained models. Fine-grained models are small reusable units encompassing a reduced functionality. Coarse-grained models contain more complex functionalities and cover more percentages of the final application. In contrast, in (Oates et al., 2002) are considered only two possibilities, which are named “high level” and “low level” of
granularity. The more details are represented in the reusable model, the higher its level of granularity. In contrast, COMOS (COMOS, 2010), CATIA (CATIA, 2010), Delmia (Delmia, 2010) and Tecnomatix (Tecnomatix, 2010) consider a continuum in the levels of granularity. Reuse is supported “from a fine to a high level of granularity”. For these engineering tools, more details captured in a model are corresponding to a finer level of granularity. Confusingly, not only the number of the described levels of granularity (two, three or arbitrary), but also their understanding of a high level of granularity are different. Current engineering tools for industrial automation systems enable a different view upon granularity. COMOS, CATIA, Delmia, and Tecnomatix consider granularity as both level of detail and coverage proportion between the reusable model and the entire system. For example, a plant section is on a higher level of granularity as a working station. For industrial automation systems, we propose a compromise based on (Griffel, 1998), (Oates et al., 2002), (COMOS, 2010), (CATIA, 2010), (Delmia, 2010), and (Tecnomatix, 2010) by considering the following aspects regarding the granularity of a reusable model:

- Model size compared to the entire system. For example, an activity diagram describing the behaviour of a sensor is fine-grained. In contrast, an activity diagram for an entire elevator is coarse-grained. In this case, the granularity depends on the proportion of a model to the entire system.
- Detail level of the reusable model. The more details there are in the reusable model, the finer its granularity and vice versa. For instance, a reference architecture with generic hardware components is coarse-grained, while considering a circuit diagram of a hardware component is fine-grained.

Anyhow, there is no formula given to quantify the level of granularity (Griffel, 1998), (Burd et al., 1996). Moreover, there is a conflict described in the literature (Griffel, 1998), (Szyzerski et al., 2002), (Szdzuz, 2002). A maximum of flexibility and maintainability requires fine-grained models. In contrast, focusing on efficiency and robustness makes coarse-grained models necessary. This conflict is depicted in the Fig. 1.

![Conflict of interests regarding granularity of reusable models](Szdzuz, 2002)

Due to the mentioned conflict, we believe that it is impossible to propose an optimal level of granularity that matches all reusable models in a domain. Based on a case study, we propose a set of recommendations for supporting the determination of appropriate levels of granularity for reusable models.

3. REUSABLE MODELS IN THE DOMAIN “PASSENGER ELEVATORS”

The principle of systematic reuse for the development of industrial automation systems developed at the IAS and based on (Pohl et al., 2005) and (Czarnecki et al., 2000) has been concretely implemented in a case study for the domain of passenger elevators. The goal of the case study was twofold: first, to evaluate the approach and identify necessary refinements, changes or extensions. Second, it was necessary to analyse which granularity levels are appropriate for reusable models. Both domain engineering and application engineering processes have been performed within one year.

In order to accomplish the case study, we proceeded as follows. First, we accomplished the domain engineering process for different elevator types manufactured by the companies Kone, ThyssenKrupp Elevators, Otis and Schindler. These companies span a very broad field of products, from freight elevators over passenger elevators to escalators. For simplicity, we have considered passenger elevators.

In a next step, a set of domain requirements (meaning requirements that are valid for all passenger elevators within the domain of elevators) have been worked out. In addition, we identified typical stakeholders like customers or certification organizations. After analyzing different provided features for passenger elevators, a feature model has been developed. A detailed overview on the accomplished domain engineering activities is given in (Maga et al., 2009a).

The result of domain engineering applied for passenger elevators encompassed identified or newly created reusable models. They included a domain dictionary containing established terminology for passenger elevators, but also valid norms and directives, reusable functional trees, state machines for elevator components, activity diagrams, a reference architecture with physical elevator components, reusable documentation and cost calculation templates, CAD-drawings for standardized components and wiring diagrams. Regarding automation of passenger elevators, software components capturing the basic functionalities of an elevator were modeled in Matlab/Simulink and then implemented in C. To conclude, there are numerous different reusable models created during domain engineering, many of them using different tools. All these models have to face the difficulty of finding the appropriate level of granularity, in order to be used during application engineering. Reusable models created during domain engineering were grouped around a physically-based reference architecture. An overview on different types of reference architectures is given in (Maga et al., 2009b). The basic reference architecture for elevators has been modeled in SysML using the EmbeddedPlus SysML Toolkit v2.0 together with the IBM Rational Software Modeler v7.0. All in all, the modeled reference architecture contains more than 150 different blocks. Fig. 2 presents only the main four blocks. These were subsequently refined until required sensors and actuators were identified.
Already in the early stages of domain engineering it became evident that the issue of granularity should be treated in more details. For example, for the depicted reference architecture it was initially unclear, at which level of granularity it should be modelled. A medium- or fine-grained reference architecture for complex automation systems like passenger elevators implies disadvantages for understandability. In contrast, a coarse-grained reference architecture is easy to understand, but unfortunately not directly useful for the operational reuse. In order to mitigate this conflict, the reference architecture has been hierarchically modelled. A first look at the reference architecture is coarse-grained, containing four blocks and the basic relations between them. The next hierarchy level contains medium-grained blocks, in which physical components of the four blocks can be seen. For example, physical components for the cabin system are the cabin doors, the cabin console, the cabin lighting system and the emergency phone. The next hierarchical level contains fine-grained blocks representing components that cannot be divided into smaller components, like sensors and actuators. As mentioned before, other reusable models have been grouped around items of the reference architecture. For example, requirements, state machines, activity diagrams and test cases were connected to the blocks of the reference architecture. In order to be reused, behavioural models have to fulfil the same requirements concerning an appropriate level of granularity like structural models. The decision, at which level of granularity the behaviour shall be modelled, depends on the planned reuse. For example, an activity diagram capturing the behaviour of cabin doors has been used for derivation of test cases for all types of elevator doors. In addition, it has been reused for implementation of generic control software for the elevator doors. The level of granularity of the activity diagram was sufficient for this type of reuse. Nevertheless, the activity diagram was still too coarse-grained for automatic code generation.

After the accomplishment of the domain engineering process, the case study included the execution of the application engineering process. The reusable models created during domain engineering were exemplarily used. In order to achieve this, we have projected and virtually constructed two passenger elevators in an office building having seven floors. We aimed to execute all application engineering activities under usage of reusable models created previously. This included tender and project plan creation with help of reusable templates and calculation tables. Moreover, it contained the execution of a traffic analysis for the considered building. In addition, the configuration of the two passenger elevators was accomplished with a tool implemented in domain engineering. Further activities accomplished during application engineering included the mechanical and the electrical design. For this, the reusable reference architecture, the reusable CAD-drawings and the reusable wiring diagrams developed during domain engineering have been used. Finally, reusable state machines, reusable activity diagrams and reusable Matlab/Simulink models have been used to implement the control software for the two passenger elevators. The execution of application engineering activities confirmed that many project-specific models could be previously prepared. This makes an efficient engineering of new industrial automation systems possible. In addition, it revealed new issues regarding the level of granularity of reusable models.

4. PROBLEMS, CAUSES AND LESSONS LEARNT IN DEFINING APPROPRIATE LEVELS OF GRANULARITY

Based on the accomplished case study, we identified a number of topics that we believe to be relevant to many other domains other than passenger elevators. The topics are organised into observed problems, identified causes and lessons learnt, discussed in this section. For each problem of defining an appropriate level of granularity, first a description is presented. Subsequently, the causes are analyzed. Finally, we present the lessons learnt that help to define appropriate levels of granularity for reusable models under these conditions.

4.1 Conflicting Requirements

- Problem description

During domain engineering, it is difficult to create models that are equally detailed, since they cover more than requirements set on a single product. Many requirements are competing, so that they cannot be integrated in one model. For instance, an elevator cannot have glass doors and a firefighter-functionality at the same time. Attempting to cover an entire
product line with one model yields to coarse-grained models (e.g. model the doors, but not the material). Unfortunately, models that are valid for the entire product line are not directly useful for the operational reuse. In contrast, designing reusable models for a restricted set of functionalities leads to fine-grained models, which are difficult to search and reuse in specific projects.

- **Causes**

Numerous variants of automation systems within a domain yield to competing requirements. Models, which occur often in different automation systems, are generally fine-grained. It has been observed that these models cover a restricted set of domain requirements. Models that intend to cover a larger set of domain requirements are mostly coarse-grained. They can be fine-grained provided that the domain requirements are not competing. However, in most cases the domain requirements are competing, since they intend to fulfil a broad field of different functionalities. Domain engineering models are created before specific decisions regarding the physical realization can be taken. Hence, more than one realization possibility has to be considered. In case that the realization possibilities can be fully specified, the created models will be fine-grained. Otherwise, they are coarse-grained and contain generic information.

- **Lessons learnt**

We observed that models should be created as detailed as possible, in order to offer a concrete support during application engineering. In order to deal with competing requirements, we suggest building reusable models in a modular manner. With help of modular models, different functionalities should be enabled or disabled, depending on project-specific requirements. The modules should have a similar level of granularity, in order to be interchangeable. Furthermore, models that are reused without any changes in all projects should be as coarse-grained as possible. Advantages in this case are that large modules can be reused and, at the same time, few interfaces have to be managed.

### 4.2 Different Levels of Granularity necessary for Reuse in separate Disciplines

- **Problem description**

Reusable models for industrial automation systems are developed for different disciplines. All these disciplines have to handle the issue of finding appropriate levels of granularity (Stoics, 2010), (Palthepu et al., 1995). For example, a wiring diagram showing the electrical system of an elevator cabin is relevant for electrical engineers. The physical measurements of the same cabin are relevant for mechanical engineers. Both disciplines have their specific views upon the same model of the reference architecture – in our case the elevator’s cabin. It has been observed that different disciplines imply different levels of granularity, as well. Even within one discipline, we observed that different project phases require different levels of granularity. For example, coarse-grained models are very useful to determine the basic structure and the basic behaviour of an elevator in early phases of a project. Late project phases require detailed and fine-grained models.

- **Causes**

The problem mentioned in this section occurs very often. The causes therefore are twofold: First, the implicit characteristics of industrial automation systems require models with different levels of granularity for the disciplines involved. Second, the application engineering phase when reusable models are required is significant for their level of granularity.

- **Lessons learnt**

It is very difficult to propose a certain level of granularity for a reusable model, when this changes depending on the project phase in which it is instantiated. Therefore, it is reasonable to provide models that cover different levels of granularity. This can be achieved by designing reusable models in a hierarchical manner. We suggest therefore using nested models that are stepwise refined. The top-level of such a reusable model should be coarse-grained, with many configuration and parameterisation possibilities. They should be able to handle the issue of finding appropriate levels of granularity for the disciplines involved. Second, the application engineering phase when reusable models are required is significant for their level of granularity.
model or how a reusable model will be instantiated later. Some reusable models are designed to be fine-grained, as precaution for future tailoring activities. Second, domain engineering is an iterative process. As all iterative processes, it is sometimes unclear where an activity ends and where the following activity begins. This leads to vague stop-conditions for the modelling activity. In case that only a few iterations are executed, the models will be coarse-grained. After many iterations, the reusable models become fine-grained.

- Lessons Learnt

In order to mitigate this problem, we propose to analyze carefully the application engineering process. Depending on the application engineering process it should be decided, which models are required. After that it should be analyzed, which level of granularity is necessary for the reusable models created during domain engineering. In a following step it should be mentioned, if a reusable model created during domain engineering can provide the required level of granularity. A suggestion how to accomplish these analyses is to consider typical models used in already finished projects. Furthermore, in (Burd et al., 1996) is mentioned that a combination of lower level components and higher level template guides for the integration of these components is essential for successful reuse. Finally, we learnt that information concerning the level of granularity of reusable models should be included in the feedbacks between application engineering and domain engineering.

4.4 Mismatch between Reusable Models regarding their Levels of Granularity

- Problem description

Reusable models capture both structure and behaviour of industrial automation systems. In addition, they include crosscutting aspects like requirements or test cases. As mentioned at the beginning of this paper, reuse succeeds more efficiently, in case that the models are coarse-grained. In other words, structure, behaviour and crosscutting aspects should be bundled to large blocks, attached to the same model in the reference architecture and finally reused together. For example, extracting the model “elevator doors” from the domain repository would mean to extract CAD-drawings for the doors, the software components to control the doors, the wiring diagrams with the electrical devices mounted on the elevator’s doors and the concerned test cases. In case that these aspects are modelled at different levels of granularity, it is very difficult to group them to a bundle and to reuse them together.

- Causes

One cause for the observed problem has been already described in the previous section. It is the unclear stop-condition in designing the different models to be bundled. A second cause is the separate and detached creation of the different models. Creation of reusable models involves different persons from different disciplines with different technical backgrounds. They will create models with an optimal level of granularity for their discipline. In most cases, the resulting levels of granularity are so different, that the models cannot be grouped together in the reference architecture.

- Lessons Learnt

We observed that systematically executed domain engineering could solve this problem. In this case, the different disciplines are obligated to work together from the very beginning. This increases understanding for adjacent disciplines. More concretely, we learnt that a reusable reference architecture containing reusable models from different disciplines should be evaluated for all disciplines involved. Regarding the level of granularity of the contained models, we suggest to enable connections between reusable models having the same level of granularity. In our opinion, all involved disciplines should be able to create hierarchical, nested models. At the top-level, these models shall be coarse-grained and at the bottom-level fine-grained.

4.5 Lack of Documentation

- Problem description

In many cases, there are some reusable models with an appropriate level of granularity for a given problem. These models could be used in different projects. The experiences gained with the help of the case study confirmed once again an aspect that is well known in (Bosch, 1999). We observed that existence of reusable models with optimal level of granularity is not sufficient for their usage. Reusable models should be well documented, in order to be found in the domain repository, recognised as being appropriate for the specific project and finally for being reused.

- Causes

The obvious cause for the mentioned problem is a missing or incomplete documentation. Reusable models should contain both description of their functionality and description how they should be reused. Behaviour, structure, origin and quality of reusable models should be included in the documentation of each reusable model.

- Lessons Learnt

In case of coarse-grained models, we suggest to provide a concrete description of configuration and parameterisation possibilities. It should be clear, which variants are covered by the model and where changes are necessary, in order to obtain the variant required by a specific project. In case of fine-grained models, we suggest to provide a precise description of functionality and interfaces. Additionally, we recommend providing a search mechanism to find reusable models with an appropriate level of granularity for a concrete project phase or discipline. Furthermore, links to required, recommended or optional models should be offered.

5. CONCLUSIONS

Reusable models in industrial automation span a broad field of granularity levels. This contribution addressed the issue of defining appropriate levels of granularity for reusable models.
After a brief introduction in the used terminology, a case study of using domain engineering for passenger elevators was presented. The execution of the case study revealed a number of topics concerning the levels of granularity, which we hold for relevant for other domains, as well. In the following table, an overview on the identified problems, the causes and the lessons learnt is presented.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Causes</th>
<th>Lessons Learnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>conflicting requirements</td>
<td>numerous variants, different realization possibilities</td>
<td>create models as detailed as possible, create modular models</td>
</tr>
<tr>
<td>different levels of granularity necessary for reuse</td>
<td>involved disciplines or different project phases require different levels of granularity</td>
<td>create hierarchical, nested models</td>
</tr>
<tr>
<td>different levels of granularity in domain engineering and in application engineering</td>
<td>unclear usage of the reusable model, vague stop-condition</td>
<td>analyze carefully the application engineering process, feedbacks about granularity between application engineering and domain engineering</td>
</tr>
<tr>
<td>mismatch between reusable models regarding their levels of granularity</td>
<td>vague stop-condition, detached creation of the models</td>
<td>enable connections between reusable models having the same level of granularity</td>
</tr>
<tr>
<td>lack of documentation</td>
<td>missing or incomplete documentation</td>
<td>provide concrete descriptions of configuration and parameterisation possibilities, provide a precise description of the functionality and of the interfaces, provide a search mechanism</td>
</tr>
</tbody>
</table>

Depending on modelled aspects and on the purpose of the model, a fine, a medium or a coarse-grained model is more appropriate. The decision, which granularity is appropriate for a reusable model depends on both the considered domain and the intended reuse of the concerned model. As general recommendation, we suggest to use well-documented, hierarchical, nested models, which provide different levels of granularity depending on the required functionality. This apply for all reusable models - from domain requirements, over reference architectures, CAD-drawings, circuit diagrams, software components to test cases and documentations. The level of granularity of a model is crucial for its reusability. In case that it is appropriately chosen, it enables a transparent and highly efficient reuse with huge benefits for creation of product lines within any domain.

REFERENCES


Oates, J., Kelley, C., Barbudinski, L.: What does granularity mean in the context of a data warehouse and what are the various levels of granularity? Information Management Online (2002)

Palthepe, S., McCalla, G., Greer, J.: granularity in Reverse Engineering. ARIES Laboratory, Department of Computer Science, University of Saskatchewan, Canada (1995)


