Data-Dependent Controller Synthesis to Enable Reliable and Safe Interoperability of Medical Devices

Franziska Bathelt-Tok
Software Engineering for Embedded Systems
Technische Universität Berlin
bathelt-tok@soamed.de

Sabine Glesner
Software Engineering for Embedded Systems
Technische Universität Berlin
sabine.glesner@tu-berlin.de

Oliver Blankenstein
Endocrinology
Charité Berlin
oliver.blankenstein@charite.de

ABSTRACT

Due to the lack of formal data treatment within the controller synthesis process, a lot of manual effort is needed to enable a reliable and safe interoperability of medical devices. This manual effort makes the process time-consuming, expensive, error-prone and, thus, reduces the will to apply it to real-world questions. To tackle this problem we build up the basis for the generic modeling and correct composition of data related device behavior using algebraic Petri nets. Hence, we are able to synthesize controllers that ensure safety critical and data-dependent functional properties cost-efficiently and automatically so that devices can be combined properly and almost independently from their manufacturers.

Keywords

Controller Synthesis, Data-Dependent, Algebraic Petri Nets, Service Composition, Interoperability

1. INTRODUCTION

Enabling interoperability of arbitrary devices is as desirable as difficult. Especially, in safety critical domains like the medical area it is necessary to ensure the reliability and safety of the devices as well as their correct composition. One step on the way to its realization is to apply the paradigm of service-orientation to the medical domain. For this purpose, medical devices can be seen as services and, consequently, interoperability can be understood as the problem of service composition. To compose services which are generally developed by different providers, or in the device case produced by different manufacturers, a further component is necessary. This component, called controller, is responsible for the routing and modification of messages as well as for the compliance of behavioral requirements. However, approaches dealing with the synthesis of such controllers cannot cope with all kinds of behavioral requirements. Particularly, the lack of formal data treatment within the controller synthesis process leads to an abstraction from data and, thus, to a modeling of data-dependent choices as non-deterministic choices. This results in either invalid controllers or controllers that must be enriched with data manually to get one that ensures given data-dependent properties and, moreover, is correct w.r.t. safety-critical and functional requirements. That manual effort makes the synthesis process time-consuming, expensive, and error-prone which reduces the will to apply it to real-world questions.
To tackle this problem we propose an approach for the generic modeling of data related device-service behavior as well as for the correct composition of multiple device-services. As underlying formalism we adapt algebraic Petri nets to define service behavior from a data-centric point of view. Petri nets in general are used to model concurrent and asynchronous systems in a very eligible way [10]. Their graphical representation enables an easy understanding while their mathematical representation allows an well-founded analysis regarding functional properties. Algebraic Petri nets [11], as type of high level Petri nets, extend this formalism by including the symbolic modeling of data. Thus, we use algebraic Petri nets to model data related device-service behavior because they offer an abstract data modeling as well as well-founded analysis and verification methods. Here, we distinguish between two levels of abstraction. On the one hand, we define a formal representation of the symbolic data treatment, i.e., definition of data types and simple data modification. On the other hand, we introduce a formal representation for the data dependencies, e.g. causal relations and related complex modifications of data. This allows us

1. to model data related behavior of medical devices generically, i.e., independently of their real implementation,
2. to detect data dependent behavior that occurs in the composition, and
3. thus, to build up a basis for vendor independent and correct composition of device-services.

Hence, with our approach we build up the basis for enabling an automated, reliable, and safe interoperability of arbitrary medical devices independently of their real implementation.

The rest of the paper is organized as follows. In Section 2 we present our approach for the generic modeling and correct composition using a running example taken from the medical field. A brief comparison to related work is done in Section 3 We conclude and give a brief overview about our future work in Section 4.

2. GENERIC MODELING & CORRECT COMPOSITION

PervasiveHealth 2014, May 20-23, Oldenburg, Germany

DOI 10.4108/icst.pervasivehealth.2014.254949

To ensure a reliable and safe interoperability of medical devices, it is not sufficient to test the correctness of the composition of the devices regarding specific requirements. In fact, verification methods are necessary to cover each unlikely but possible misbehavior as well. As basis for this, it is essential to define the behavior of the devices and their composition formally. Based on a running example (Section 2.1), we distinguish between two levels of abstraction. Firstly, we focus on the formal representation of the services themselves by defining data and its modification as well as the behavioral processes generically. We demonstrate our modeling approach in Section 2.2 and explain how algebraic Petri nets can lead to vendor independence. Moreover, we regard the second level that deals with the composition of services and its correctness, i.e., compliance of requested properties, in Section 2.3.

2.1 Running Example: Artificial Pancreas

As reported in various medical journals, e.g. [4], the number of patients with type 1 diabetes increases continuously. Type 1 diabetes is an incurable disease that is mainly caused by gene defects, autoimmune diseases and viral infections. Affected people cannot produce insulin on their own. That is why they have to inject insulin each day of their lives. This entails that especially for children and handicapped persons the risk of an incorrect medication and, hence, the threat of health increases unpredictably. To reduce this risk, some companies, e.g. MiniMed®, have developed a new form of therapy that increases the quality of life by removing the need for patients to administer their own treatment. They developed a system, called artificial pancreas, which consists of a continuous glucose monitoring and an insulin pump.

For **continuous glucose monitoring**, different kinds of sensors can be used. In the Charité and, thus, in our running example, we focus on electro-chemical sensors, whose functionally [14] is sketched in the following. The enzymatic (chemical) component of the sensor reacts with the blood glucose. As a result, a current flow depending on the blood glucose concentration appears. This current flow is digitalized by an analogue-digital-converter. Afterwards, the digital signal is stored internally and sent via an interface to the environment. This signal can be seen on a display. The user has to decide if an insulin injection is necessary or not. A main drawback of such a sensor is that it is identified as a foreign substance. Thus, it must be exchanged every 4-7 days. This leads to the problem that, generally, a new system is constituted in this period of time, because a new component arises.

Besides the glucose sensor, there is a second component of the artificial pancreas, an insulin pump e.g. [2]. In general, an **insulin pump** consists of a catheter, a reservoir, a liquid level measuring instrument, and the pump itself. The reservoir serves 150-300 units of insulin. The amount of insulin which is still available in the reservoir is determined by the liquid level measuring instrument. Currently the insulin pump cannot react to signals from the glucose sensor. It only injects a fixed amount of insulin at regular intervals. This amount can be communicated through the interface. As the device receives the amount of insulin to be injected, it compares it with the amount available. If there is not enough insulin in the reservoir, the pump sends a warning to the environment. Otherwise, the pump injects the requested units of insulin via the catheter.

Because this kind of artificial pancreas has been developed as closed system, affected people must use a specific kind of glucose sensor. From an economical point of view, this creates a strong vendor dependence and leads to higher costs. This example shows pretty well that our long term target, to overcome vendor dependence, is highly desirable. By ensuring a safe and reliable interoperability of medical devices, the costs for a composed system can be reduced and the autonomy of patients can be increased. Here, two main questions must be answered during the composition of the involved devices: i) How can the device-services be represented formally? ii) How can we ensure the correctness of the composition with respect to data-dependent functional and safety-critical requirements?

2.2 Generic Modelling

To tackle the question, how to represent services formally, we focus on the formal definition of services and their behavior in this section (Figure 1). Here, there are two levels of abstraction we have to deal with: representation of the i) data flow and ii) control flow. The control flow as part of the service behavior can be easily represented by Petri nets and markings therein because they offer the possibility to formalize causal relationships. But for an application to real world problems this is by far not expressive enough. To increase the expressiveness, a possibility to represent the data flow simultaneously is needed. This can be done by using so called algebraic Petri nets. Informally, algebraic Petri nets are Petri nets that are inscribed by algebraic specifications.

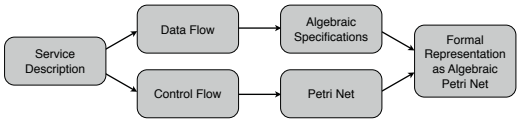


Figure 1: Modelling Approach

2.2.1 Data represented by Algebraic Specifications

For a generic modeling of real world's problems we need a formalism that is able to represent four kinds of data manifestation.

1. A **data type** describes the kind of a data. This kind can be primitive like Integer, Boolean, etc. or a complex combination of object sets.
2. A **data value** is a specific characteristic of the data type. E.g. a data of type Integer can have value 5.
3. A **data range** is a set of all possible data values.
4. A **data modification** is the change of the data value.

Algebraic specifications are an appropriate instrument for defining data types generically [7]. In the following, we briefly introduce algebraic specifications [3, 11] and explain their relationship to the generic definition of data types.

Definition 1. A **signature with variables** is a 3-tuple $\Sigma = (S, OP, X)$ with

- $S \neq \emptyset$ is a set of sorts
- $OP = (OP_{w,s})_{(w,s)\in S^* \times S}$ is a family of sets of operation symbols
  i.e. for $f \in OP_{w,s}$, $w \in S^*$ is called domain of f and $s \in S$ codomain of f
- $X = (X_s)_{s\in S}$ is a family of variables w.r.t. sorts S

With a signature defined in this way, a generic definition of data types and data modification is possible. Here, the set of sorts S represents different data types in an abstract and syntactical way. Moreover, the set of operations OP defines the mappings that are allowed during the modification.

Definition 2. Let $\Sigma = (S, OP, X)$ be a signature with variables. The family of sets $T_{\Sigma,s}(X)$ of **Σ-terms** according to sort $s \in S$ is the smallest subset $T_{\Sigma,s}(X) \subseteq S \cup OP \cup X \cup \{(;);,\}$ of words that fulfill:

1. $\forall x \in X_s . x \in T_{\Sigma,s}(X)$
2. $\forall c :\rightarrow s \in OP . c \in T_{\Sigma,s}(X)$
3. $\forall f : s_1 \dots s_n \rightarrow s \in OP . \forall t_i \in T_{\Sigma,s_i}(X), i \in \{1, \dots, n\}.$ $f(t_1, \dots, t_n) \in T_{\Sigma,s}(X)$

With help of this so defined term algebra as smallest subset that span the whole space of terms, a finite description of all possible terms is defined. If $X = \emptyset$ then $T_{\Sigma,s}$ is the family of **ground terms**.
First of all, these terms enable an inductive definition of the data range and, moreover, define the modification of data that are represented by variables generically.

Definition 3. Let $\Sigma = (S, OP, X)$ be a signature with variables, $s \in S$ an arbitrary sort ,and $t_l, t_r \in T_{\Sigma,s}$ terms according to sort s. Then $e \equiv t_l = t_r$ is a $\Sigma-$ **equation**.
The equations offer a possibility to abstractly define relationships between data and its modification occuring in the system to be formally represented.

Definition 4. An **algebraic specification** is a 2-tuple $SP = (\Sigma, E)$ where $\Sigma = (S, OP, X)$ is a signature with variables and E is a set of equations defined on Σ.
In summary, by using algebraic specifications a formal and generic definition of data and its modification is possible.

1. **data types** are **sorts** that are part of a signature
2. **data values** are assignments of **variables**
3. **data ranges** are **(ground) terms**
4. **data modifications** can be represented by **operations, equations,** and by **(ground) terms**

The **application to** the components of our **running example** leads to algebraic specifications shown in figure 2. For the glucose sensor (Figure 2(a)) we know from the description (seciton 2.1) that we have at least two kinds of data types with different ranges. The first one P is the current flow that is offered, the second one D is the digital signal that is stored and sent to the environment. To transform the measured current flow x to a digital signal y an operation $f : P \rightarrow D$ is needed. We extended the set of sorts and the set of operations by including T and the constant $\cdot :\rightarrow T$ respectively. With that we can define the end of one "measure send" cycle. Analogously the algebraic specification of the insulin pump which is shown in Figure 2(b) can be defined. Besides the representation of sorts, operations, and variables, there are two equations. They describe the relationship between the operations $Pair$ and $K1/K2$ and, thus, indirectly the modification of y/y'.

2.2.2 Petri nets extended by algebraic specifications

Algebraic Petri nets [11, 13] extend Petri nets by the assignment of algebraic specifications to their elements. This is exactly what is needed for a formal and data-related representation of device-service behavior, because algebraic specifications offer the possibility to define data generically (as shown in section 2.2.1) and Petri nets can be used to define causal relationships.

SPEC-GS = (S_{GS}, OP_{GS},X_{GS},EQ_{GS})

$S_{GS} = \{P, D, T\}$

$OP_{GS} = \{\cdot, f\}$
$\cdot :\rightarrow T$
$f : P \rightarrow D$

$X_{GS} = \{X_P, X_D\}$
$X_P = \{x\}$
$X_D = \{y\}$

$E_{GS} = \emptyset$

(a) glucose sensor

SPEC-IP = (S_{IP}, OP_{IP},X_{IP},EQ_{IP})

$S_{IP} = \{I, I-T, E\}$

$OP_{IP} = \{z, Pair, K1, K2\}$
$z :\rightarrow E$
$\cdot :\rightarrow T$
$Pair : I \times I \rightarrow I - T$
$K1, K2 : I - T \rightarrow I$

$X_{GS} = \{X_I, X_{I-T}\}$
$X_I = \{y, y'\}$
$X_{I-T} = \{x\}$

$E_{IP} = \{e_1, e_2\}$
$e_1 \triangleq (K1(Pair(y, y')) = y)$
$e_2 \triangleq (K2(Pair(y, y')) = y')$

(b) insulin pump

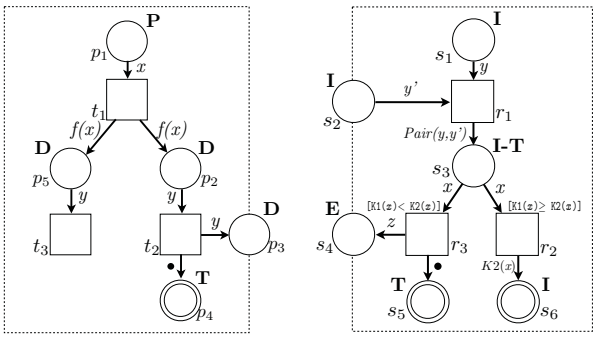
Figure 2: algebraic specification of running example

Definition 5. A **Petri net** is a triple $N = (P, T, F)$ where $P \neq \emptyset$ is a set of places that represent states, $T \neq \emptyset$ is a set of transitions that can be understand as actions, and $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation describing arcs between P and T, i.e., it describes which action can occur in which state and which action leads to which state.

Definition 6. Let $N = (P, T, F)$ a Petri net and $SP = (\Sigma, E)$ an algebraic specification. Then, the triple $AN = (N, (\varphi, M_0, \lambda), E)$ is an **algebraic Petri net**, where

- $\varphi : P \rightarrow S$ assigns sorts $s \in S$ to places $p \in P$
- $M_0 : P \rightarrow T_{OP}$ is the initial marking that assigns groundterms to places
- $\lambda : F \rightarrow T_{OP}(X)$ defines the arc inscription of N by assigning terms to arcs

The **application to** the components of our **running example** leads the Petri nets (Figure 3) that are inscribed by the algebraic specifications that are shown in Figure 2.
The Petri nets arise from the description that has been presented in section 2.1. Thus, the Petri net representing the glucose sensor can be understood as follows. Initially, a current flow occurs (in p_1) that can be measured (x). This current flow is transformed in a digital signal (applying f to x at t_1). Then, the digital signal "$y = f(x)$" on the one hand is stored internally in p_5 and can be deleted by t_3. On the other hand y is sent to the environment via the interface p_3. At the same time the final state p_4 is reached and the "measure-send" cycle is finished. Analogously, the Petri net representing the insulin pump (Figure 3(b)) can be interpreted.



(a) glucose sensor (b) insulin pump

Figure 3: Algebraic Petri nets of running example

2.3 Data Dependent Controller Synthesis

Besides the formal representation of the device-services, we have to tackle the question: How can we ensure the correctness of the composition with respect to data-dependent functional and safety-critical requirements? For this purpose, we presented an approach [5] that starts from the formal representation of the involved services given as algebraic Petri nets, the interface matching, and the data-dependent requirements (given in a specification language). As it can be seen in Figure 4, the approach consists of three steps. In the first step, the behavioral abstraction, the behavior of the services is reduced to the relevant behavior. This step leads to a set of properties, expressed by a specification language, which have to be fulfilled at least. The target of the second step, is to combine the resulting behavioral abstraction, the interface matching and the requirements the controller must fulfill. For this reason, we transform the properties into algebraic Petri nets using our generic and data-related transformation algorithm. Finally, the third step is the application of the well-defined composition and analysis methods of Petri nets for our composition algorithm. As a result, we obtain the requested controller if one exists. This controller in combination with the device-services creates a system that works correct with respect to the given data-dependent functional and safety critical requirements, because the controller is correct by construction.

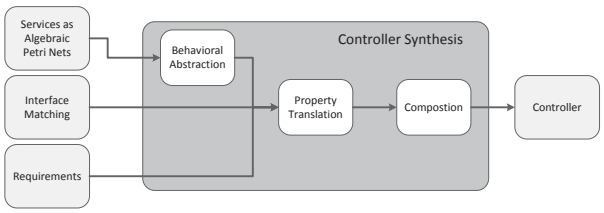


Figure 4: Basic Idea of the Controller Synthesis

3. RELATED WORK

Our controller synthesis approach requires a formal representation of data-dependent services. We considered several modeling languages: UML, π - calculus [9], finite-state automata (e.g. done in [8]) and (high level) Petri nets. None of them, except algebraic Petri nets, offers the opportunity to formally and symbolically define data-dependent behavior, to verify properties, and to represent data-dependencies in a user friendly way at the same time. Thus, we use algebraic Petri nets, as type of high level, because they are the most suitable formalisms regarding our requirements. There are several approaches dealing with service composition. Their main drawbacks are, that they either omit asynchronous communication [15] or abstract completely from data and data-dependent behavior [1, 6]. Another approach enables the time-related composition of services [12] but it is not able to represent data symbolically. To the best of our knowledge there is no approach dealing with formal definition of data and data-dependencies. Right now, there is no automated *data dependent* controller synthesis process available.

4. CONCLUSION & FUTURE WORK

In this work we have addressed the problem of the data-dependent controller synthesis to enable a reliable and safe interoperability of medical devices. We have presented a generic modeling approach that uses algebraic Petri nets as formal representation of data, its modification and behavior that occurs in the systems. For this purpose, we have focussed on an example taken from the medical field, the artificial pancreas. Based on that, we have shown how to receive an algebraic specification as well as an algebraic Petri net out of a device behavior description. Moreover, we have given the main idea of our correct composition algorithm to combine multiple device-services. With that we are able to synthesize correct controllers automatically and, thus, enable reliable interoperability of medical devices that is almost independently from their manufacturers.
In future work, we will automatize our modeling approach to achieve algebraic Petri nets out of device descriptions (patents, certifications, etc.) without manual effort.

5. REFERENCES

[1] A. Bracciali, A. Brogi, and C. Canal. A formal approach to component adaptation. *Journal of Systems and Software*, 74(1):45 – 54, 2005.

[2] S. B. Choi. Method for controlling insulin pump using bluetooth protocol, May 24 2004. US Patent App. 10/852,483.

[3] H. Ehrig, B. Mahr, F. Cornelius, M. Große-Rhode, and P. Zeitz. *Mathematisch-strukturelle Grundlagen der Informatik*. Springer DE, 2001.

[4] D. Elleri, D. Dunger, and R. Hovorka. Closed-loop insulin delivery for treatment of type 1 diabetes. *BMC medicine*, 9(1):120, 2011.

[5] F.Bathelt-Tok. Towards the automated synthesis of data dependent service controllers. In *PhD Symposium of the ICSOC 2013*. Springer, to appear.

[6] C. Gierds, A. J. Mooij, and K. Wolf. Reducing adapter synthesis to controller synthesis. *IEEE T. Services Computing*, 5(1):72–85, 2012.

[7] J. V. Guttag and J. J. Horning. The algebraic specification of abstract data types. *Acta informatica*, 10(1):27–52, 1978.

[8] C. in Automation (CiA). Cia 425 dsp v2.0.2 canopen application profile for medical diagnostic add-on modules – part 2: Injector, 03. 2014.

[9] R. Milner. *Communicating and mobile systems: the pi calculus*. Cambridge university press, 1999.

[10] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4), 1989.

[11] W. Reisig. Petri nets and algebraic specifications. *Theoretical Computer Science*, 80:1–34, 1991.

[12] D. Stöhr and S. Glesner. Planning in real-time domains with timed ctl goals via symbolic model checking. In *TASE 2013, IEEE*, 2013.

[13] J. Vautherin. Parallel systems specifications with coloured petri nets and algebraic specifications. In *Advances in Petri Nets 1987*, pages 293–308. Springer.

[14] J. Wang. Electrochemical glucose biosensors. *Chemical reviews*, 108(2):814–825, 2008.

[15] D. M. Yellin and R. E. Strom. Protocol specifications and component adaptors. *ACM Trans. Program. Lang. Syst.*, 19(2):292–333, Mar. 1997.