

# The Petri Net Baukasten: 2nd Installment<sup>1</sup>

J. Dehnert, M. Gajewsky, S. Lembke, and H. Weber<sup>2</sup>

*Institute for Communication and Software Technology  
Technical University Berlin  
Berlin, Germany*

---

## Abstract

Petri nets are a subject of interest to researchers since more than 30 years. They became popular since they allow the graphic representation of computational structures and have a formal underpinning that allows formal proofs of properties. Their use in engineering, however is limited to a rather small number of applications. Many of the graphic notations used in practice are very similar to the graphic representation of Petri nets but do not make use of the rich body of theory that exists for Petri nets. This is repeatedly attributed to the fact that Petri nets and the associated theory are hard to understand and real life problems modelled with Petri nets tend to require special brands of Petri nets which in turn led to a great variety of different net types. Finding the right Petri net types from the portfolio of existing types, finding corresponding techniques and tools, selecting them and applying them successfully to applications poses a problem and prevents their wide spread use. In order to improve this situation a Petri net Baukasten has been developed by the DFG-Forschergruppe PETRI NET TECHNOLOGY in Berlin. It aims at the development of a support environment for the use of Petri nets in new applications including theoretical and tool aspects. This paper presents the 2nd Installment of the Petri Net Baukasten, which extends the first installment of the Petri Net Baukasten, documented as Initial Realization in 1999, by providing interfaces, services, and data for specific user groups such as application developers, tool developers and Petri net experts. The user interface offers an intuitive presentation of Petri net concepts and notions. The Petri net Baukasten supports different user groups in their specific work with Petri nets by user specific assistance services. These services allow the users to find the right Petri net types in a variety of existing types and support their selection and use. Information is kept in an associated database. Furthermore the assistance services maintain this database in order to keep track of recent developments in theory, tools and application areas.

*Key words:* Application of Petri nets, Petri net assistance, Petri net classification, Petri Net Baukasten

---

## 1 Introduction

In the long history of Petri nets a universe of Petri nets has evolved consisting of an enormously rich theory, a wide variety of tools, and numerous successful applications and case studies in various application domains. This vast variety is difficult to handle for all kinds of people working with Petri nets, which results in the strong need of a structured access to Petri nets. The structured access has been the main aim of the “DFG-Forschergruppe PETRI NET TECHNOLOGY” which has developed the so-called Petri net Baukasten for this purpose. It is designed to support Petri net experts, application developers and tool developers alike in their specific work with Petri nets.

In [11, 15] it has been shown, that the Petri net Baukasten is an application-oriented Petri net technology, providing Petri net based techniques for system development. Figure 1 roughly sketches the conceptual structure of the Petri net Baukasten.

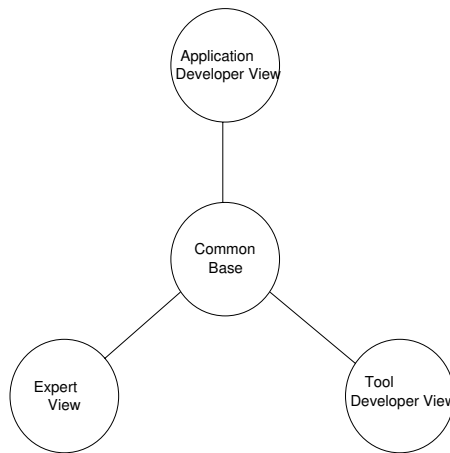


Fig. 1. The conceptual structure of the Petri net Baukasten

The Petri net Baukasten is divided into different views, namely the Application Developer View, the Expert View, and the Tool Developer View, which are linked via a so-called Common Base. The Common Base (for details also see [4, 17]) consists of a classification of Petri nets and corresponding notions independent of their use in applications, their formalizations, and tool support. These Petri net notions are represented in the specific views. They concern the use of the Petri net notions within a certain application domain, their formalization in the theory of Petri nets, and their implementation as Petri net tools.

<sup>1</sup> This work is part of the joint research project “DFG-Forschergruppe PETRINETZ-TECHNOLOGIE” between H. Weber (Coordinator), H. Ehrig (both from the Technical University Berlin) and W. Reisig (Humboldt-Universität zu Berlin), supported by the German Research Council (DFG).

<sup>2</sup> Email: dehnert, gajewsky, lembkes, hweber@cs.tu-berlin.de

It is important to note, that the Petri net Baukasten is given on a purely conceptual level. The contents of the specific views and Common Base as well as the way, how these relations are established, are not relevant on the conceptual level. Therefore, in order to apply the Petri net Baukasten in practice we have to realize or *install*<sup>3</sup> it, which can be accomplished in various ways. An installment of the Petri net Baukasten fixes the contents and the technical support of the Petri net Baukasten. There is already one installment, the so-called Initial Realization, which has been documented in [7]. This installment contains only medium amount of data on a low technical level and it does neither allow an interactive access to its contents nor a change of it. This is a major drawback. Therefore, the research group PETRI NET TECHNOLOGY proposes an advanced installment of the Petri net Baukasten in this paper, called “2nd Installment”, which is documented in this paper. In contrast to the Initial Realization the 2nd Installment provides an interactive access to the contents of the Common Base and the specific views including a dynamic change<sup>4</sup>. Clearly, the 2nd Installment is on a higher technical level, but a full implementation is beyond the capacities of the research group PETRI NET TECHNOLOGY. The 2nd Installment consists of a database, services operating on this database and an explicit access for each user group provided by specific user interfaces. The database may be searched or enriched by the services. Enrichment demands some kind of cooperation among the users in order to maintain the conceptual structure of the Petri net Baukasten. More precisely, in order to maintain a correspondence between the different views an inserted piece of data in one view has to be related to corresponding data of the other views. The different views include a Petri net notion, resp. a Petri net type, in the Common Base, its formalization in the Expert View, an application-oriented interpretation in the Application Developer View, and a tool in the Tool Developer View. As the knowledge of the corresponding data cannot be expected from a single user, the 2nd Installment has to support the cooperation among users as well. Intercooperation may comprise discussions of open problems as well as joint research within and also across the different user groups. It is therefore an important means for bridging the gap between different views on Petri nets.

A further difference to the Initial Realization is that the 2nd Installment focusses on a specific application domain, namely workflow management.

Workflow management has become critical in today's business life, where the efficient handling of business processes is decisively responsible for successful company development. Workflow management comprises modeling, analysis and execution support of business processes. The applicability of Petri nets in these areas has been examined and discussed extensively, see

---

<sup>3</sup> “Installment” has been called “realization” in previous papers about the Petri net Baukasten. The new term emphasizes the fact, that it is one part of a serial story.

<sup>4</sup> More precisely, we obtain a class of installments differing in their contents. For brevity, we consider this class as one single installment.

e.g. [1, 3, 6, 13]. It has been stated that Petri nets are highly suitable for workflow management systems due to their intuitive graphical representation, excellent formal foundation, early evaluation and verification, and their possibility to operate as a workflow engine. Although developed with special attention to the area of business processes, the conceptual structure of the Petri net Baukasten is not restricted to one particular application domain, but can be adopted to other domains as well.

The paper is organized as follows: In Section 2 we briefly review the aims and the concept of the Petri net Baukasten and give an overview of the 2nd Installment. In Section 3, 4 and 5 we explain its several constituents, namely User Interface, Services and Database. In Section 6 we propose a corresponding software architecture for the Petri net Baukasten. This architecture is based on the above constituents and comprises additional functionality which are essential for the software realization. The paper concludes with a user scenario in Section 7 outlining the benefit of the 2nd Installment.

## 2 The 2nd Installment of the Petri net Baukasten

In this section after reviewing the aims and the underlying concept of the Petri net Baukasten we will give a rough survey of the 2nd Installment.

### 2.1 *Aims of the Petri net Baukasten*

The main aim of the Petri net Baukasten is to provide support for the different users of Petri nets. This support is given with respect to specific application domains. As application domains we consider all areas where Petri nets have supported the development of a system in this domain, e.g. production automation, control systems, workflow management, etc.

The Petri net Baukasten enables an engineer who develops an application to use application-oriented interpretation of Petri net notions. The Petri net Baukasten supports the application developer to find and use adequate Petri net based methods, Petri net techniques or just Petri net types which fit the developed application. The Petri net Baukasten does not only provide the theoretical knowledge about the handling of Petri net notions, but also makes available practical examples and tool support.

The Petri net expert is supported in his work to elaborate and enhance existing formal techniques, to define new types and notions in a uniform way, to state properties of all variants of Petri nets in a formal and constructive manner, to transfer results between net types, and to make results and notions available for applications.

And last, but not least, the Petri net Baukasten enables tool developers to distribute their tool, to find appropriate tools (also for tool development), to fit tools together in a prototyping way, to add and change tools and tool references.

## 2.2 The Concept of the Petri net Baukasten

The conceptual structure of the Petri net Baukasten has been depicted already in Figure 1. The central notion of the Petri net Baukasten are Petri net techniques, illustrated in Figure 2. These are given as a consistent combination of a Petri net type, its formalization, a methodological procedure, and a corresponding tool.

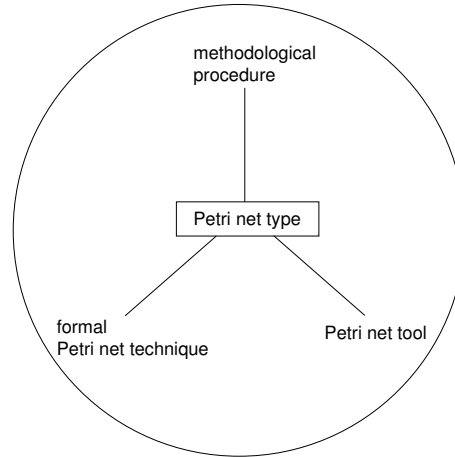


Fig. 2. Petri net technique

A Petri net type is given by a semiformal description of its characteristics. The description uses Petri net notions concerning the basic constituents of a Petri net variant like place, transition, marking, etc. and concerning the available Petri net operations for analysis, or structuring techniques, etc. The Petri net type is represented in the Petri net classification in the Common Base and holds links to the other views.

The formal description of a Petri net type is given in the Expert View. For each characteristic a mathematical formalization is given; basic constituents are captured in the definition of the Petri net class. Operational constituents correspond to definitions and results of Petri net operations.

In the Tool Developer View a tool is provided, which implements the Petri net type. That is, it works on Petri nets comprising the basic constituents and the operational characteristics are implemented.

A methodological procedure of how to work with the particular Petri net type in practical applications is given in the Application Developer View. It defines the order of development activities, including the available Petri net operations for verification, structuring, etc.

If only parts of a Petri net technique are available we speak of an incomplete Petri net technique. Addition of missing parts together with a corresponding link is called completion in the following sections. Although ideally each Petri net technique should be complete, it is important to consider also incomplete Petri net techniques, because the Petri net Baukasten or further research and development may be able to provide the missing parts. But more restrictive

strategies are also possible.

### 2.3 Survey of the 2nd Installment

The 2nd Installment of the Petri net Baukasten, depicted in Figure 3, roughly consists of three parts: The data which is to be provided, a user interface for communication between users and system, and thirdly an intermediate level which operates on the data. We call these parts user interface, database, and services, respectively. Each of these parts is structured internally in order to increase clarity and usability. This internal structure is derived from the conceptual structure of the Petri net Baukasten, as depicted in Figure 1. That is, each part is substructured into three views which corresponds to one of the views (application developer view, tool developer view, and expert view). This structure is motivated by the general aim of the Petri net Baukasten (see Section 2.1) to support each of these groups. We provide an individual access to information for each group, because we can then focus on the relevant information, which naturally differs for each group.

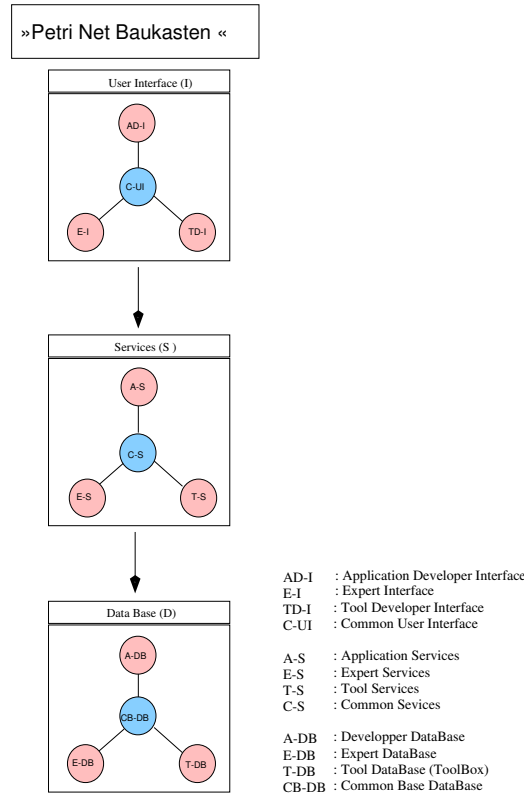


Fig. 3. The 2nd Installment

**Database** The contents of the 2nd Installment is stored in the database. In contrast to the Initial Realization it is not fixed, but may increase dynamically. According to the concept of the Petri net Baukasten it is substructured into specific data for the different user groups and the Common Base.

The Common Base contains the Petri net classification and Petri net notions and links to data of the specific views. In principle, we could start with an empty database, which is gradually enriched. For convenience we assume the data of the Initial Realization as its contents. More details about the database can be found in Section 5.

**Services** operate on the database. We distinguish between common-, application-, expert-, and tool-services, which are described in Section 4. Basically, there are search services which find data according to input parameters and maintenance services. Maintenance services modify the database and trigger an asynchronous cooperation between users (of different groups). This cooperation is essential for maintaining the basic concept of the Petri net Baukasten, described in Section 2.2. For search services the set of admissible input parameters is derived from the contents of the database, and may thus also be changed by maintenance services.

**User Interface** The upper part in Figure 3 shows the user interface. We provide a common user interface from where a specialized interface for any of the different user groups, application developers, experts, and tool developers, can be reached. Each specialized interface offers the corresponding services together with an informal description. Search services are accompanied by a set of admissible input parameters. Moreover, the user is informed, that a change of the database has been performed by another user. He may complete that piece of data by group-specific data in order to maintain the basic concepts of the Petri net Baukasten. The interfaces are described in Section 3.

### 3 User Interface

We provide a general user interface from where a specialized interface for any of the different user groups, application developers, experts, and tool developers, can be reached. Each specific user interface offers an individual set of services enabling the access to information about Petri nets, which are relevant to the respective user group. Such services are described more detailed in Section 4. As a result of this, a separation of concerns is achieved leading to a more need-oriented and easy use of the Petri net Baukasten.

The choice of one of the three different user groups correspond to the main objectives of the particular user. If their objectives change over the time they may chose a different access accordingly. For example, a user may be first interested in the formal development of a specific Petri net type and later in the implementation of an appropriate Petri net tool. This person should use the Petri net Baukasten as a Petri net expert first and as a tool developer afterwards. Accordingly, the person uses the Petri net expert user interface first and the tool developer interface afterwards. The specific sub-interfaces are reached through the common user interface, which realizes the user choice according to their role. A second task of the common user interface is the

information about changes in the database. In case new information has been added, all users are informed and asked to complete the data in order to maintain the basic concepts of the Petri net Baukasten.

The specific user group interfaces support the input of parameters required for the execution of selected services and the representation of results of a service execution. The interfaces are quite diverse regarding their offered services but their appearances are uniform. The available services are listed in menus and the input of parameters is enabled via forms. Moreover, each interface offers a help function explaining its specific functionality.

An explicit description of the application developer user interface can be found in [17]. The remaining user interfaces as expert- and tool developer user interface are implicitly described by the corresponding services in the following section.

## 4 Services

Services are invoked in the user interface and operate on the database. Basically, there are two different kinds of services: search and maintenance services.

Search services allow an interactive access to data. They fulfill the task of finding the requested data according to input parameters and present the search results in a manner which is adapted to the individual user preferences.

Maintenance services are essential for the dynamic aspect, as they perform modifications (addition or deletion) of the database. Moreover, they trigger a cooperation between users of different groups.

We distinguish between common-, application-, expert-, and tool-services. The group-specific services use common services in order to fulfill their tasks.

### 4.1 Application Services

The Petri net Baukasten supports application developers in the domain of Petri net based application management. Support can be given for the following tasks:

- (i) to find an adequate Petri net based method for developing the application,
- (ii) to use Petri net techniques that serve best in the development of the application,
- (iii) to find a Petri net type which fits the application,
- (iv) to find a suitable tool,
- (v) to provide the application developer with example solutions.

Accordingly, we define the following application services. The first five services stand in a one-to-one correspondence to the tasks listed above. The last service has been added in order to support the dynamic use of the 2nd Installment of the Petri net Baukasten.



**Petri net method selection** This service allows to find a suitable Petri net based development method, to search over objectives, intended application domains and development phases (analysis, design, etc.) of Petri net methods. After such a search the application developers can access information about each offered Petri net method, as e.g. intended objective, covered phases, underlying languages and techniques and supporting tools. Additionally, they may consult a method guide before selecting their suitable Petri net method.

**Petri net technique selection** This service allows to find a suitable Petri net technique. The application developers specify required characteristics of a Petri net technique. This can be done, among others, with the help of keywords which outline intended objective, application domain and underlying Petri net type. Petri net techniques fulfilling the specified characteristics are determined automatically and offered to the application developer.

Before application developers select their suitable Petri net technique from the offered set, they may use several meta informations about it, as e.g. application domain, objective, underlying Petri net types, or the essentials of its procedure. Moreover, they may use its informal description, the so-called technique guide, or its formal foundation. Last but not least, it is possible to search for a suitable Petri net tool supporting the preferred Petri net technique.

**Petri net type Selection** This service realizes the selection of a suitable Petri net type which fulfills the requirements of a particular application development process. There are several assistance methods for finding a suitable Petri net type as e.g. by an application aspect checklist or by navigation through the Petri net classification. Via the checklist it is possible to find a Petri net type which fits a combination of the most relevant aspects for their application purpose.

The service determines this Petri net type and offers it to the application developer. Together with the list of results more information about the different Petri net types is provided, as e.g. intended application domains, possible tools, informal and formal descriptions. These informations finally facilitate the Petri net type selection.

**Petri net tool Selection** This service supports application developers in selecting a suitable Petri net tool which enables the application of a Petri net technique, method, or type selected before. It provides the application developer as well with technical-functional criteria (as e.g. required operating system, purpose, usability) and socially-assessable criteria (as e.g. usage costs). These informations enable application developers to select the suitable tool or tool set for their development task. After the selection of a tool or tool set it can be downloaded respectively ordered from its supplier.

**Select a Sample Solution** This service is intended for finding proposed solutions suitable for fulfilling the requirements of a certain application develop-

ment. It enables application developers to find suitable sample solutions by their objectives, application domains, and their underlying Petri net types. Before application developers select a certain sample solution to integrate it within their own application development they can use information about it, for example about the solved problem or the underlying Petri net type and can consult a so-called solution guide for details.

**A.Maintenance** Via this service it is possible to add new information to the application developer database. This service can be used to enhance existing methods, to add new techniques or to make available examples for other users. It results in adding the given information in the application database and requires further information concerning the semiformal description on the common database. Thus, an incomplete Petri net technique is added, which invokes completion by other user groups via the common service **Petri net technique Completion** discussed below.

A more precise description of the application developer services, their realization and the terms introduced above can be found in [17].

#### 4.2 Expert Services

The Petri net Baukasten supports Petri net experts doing research in the area of Petri nets. Support can be given for the following tasks:

- (i) to elaborate and enhance existing formal techniques,
- (ii) to define new types and notions in a uniform way,
- (iii) to state properties of all variants of Petri nets in a formal and constructive manner,
- (iv) to transfer results between net types, and
- (v) to make results and notions available for applications.

Basic structure and notions of the expert view have been introduced in [10]. These include descriptions of formal Petri net techniques, transformations, abstract Petri net frames, and scheme. A formal Petri net technique is the formalization of a Petri net type in the Common Base, including formal definitions of its basic constituents and operations like e.g. transformations to other formal Petri net techniques. An abstract frame yields a generic description of a class of formal Petri net techniques. All descriptions are given in a structured way, the so-called schemes. The following services have been constructed accordingly, where **Add Formal Petri Net Technique** and **Add Transformation** services invoke the common service **Petri net technique Completion**.

**Search Formal Petri Net Technique** This service finds all formal Petri net techniques comprising a given Petri net notion. It uses the common service **Common Base Parameter Search**. For choosing suitable Petri net notions the expert may consult the glossary of Petri net notions.

**Search Transformation** This service is a special case of **Search Formal Petri Net**

**Technique**, because transformations belong to a formal Petri net technique. Input parameters are restricted to Petri net notions concerning transformations.

**Search Abstract Frame** This service comprises search for abstract Petri net frames, and instantiations, which are given by schemes in the Expert View. Optional input parameters specify the requested scheme. This service does not use common services as these schemes are specific to the Expert View.

**Search Precondition** This is a special case of searching for an abstract frame. It focusses on the preconditions of the formal parameters of that frame, which are presented as the result of the search.

**Search Case Study** The input of this service are Petri net notions or keywords concerning a certain problem. The service returns a set of case studies which are based on the Petri net notion or solve the problem.

**Add Case Study** The expert may add a description of a case study, provided that the used formal Petri net technique is already stored in the database. Moreover, keywords describing the solved problem are required as input. The service adds the case study and creates a link from the formal Petri net technique.

**E.Maintenance** The following services are maintenance services:

**Add Formal Petri Net Technique** This maintenance service adds a formal Petri net technique together with its semiformal description by a Petri net type to the database. The expert has to fill out the scheme for formal Petri net techniques. Moreover, he has to provide the semiformal description of the Petri net type using already existing Petri net notions and possibly defining new ones. Help is provided by the glossary of existing Petri net notions. The common services **Petri net type Insertion**, resp. **Link Construction** place the newly defined type in the Petri net classification and the service **Parameter maintenance** adds new Petri net notions.

**Add Transformation** Analogously to **Search Transformation**, this is a special case of **Add Formal Petri Net Technique**.

**Add Abstract Frame** The expert fills out the corresponding scheme for abstract Petri net frames, or instantiations, which are then added to the database.

**Add Precondition** Analogously to **Search Precondition**, this is a special case of **Add Abstract Frame**.

The above listed tasks are supported mainly by the following services: The service **Search Formal Petri Net Technique** supports the tasks (i) and task (ii) as the expert may use existing definitions for enhancement. The service **Search Abstract Frame** also supports tasks (i) and (ii) and additionally tasks (iii) and (iv). The service **Add Case Study** supports task v.

### 4.3 Tool Services

The task of the tool developer is supported by the Tool Developer View, see also [18]. The support comprises various aspects of tool development:

- (i) management of existing tools,
- (ii) facilities for tool development, and
- (iii) support for extension of existing tools.

Accordingly, the following tool-services are defined.

**Petri Net Tool Selection** This service supports tool developers in selecting a Petri net tool. It provides the tool developer with relevant information about the tool, such as functional range, implementation language, underlying programming concepts, operating system etc. These information enable tool developers to compare existing implementations and hence to find tools or specific modules which can be enhanced or reused for their own tool implementation.

**Search for Petri Net Input Formats** Most of existing Petri net tools use their own Petri net input format. This service provides a search function about existing formats and existing transformation scripts translating them into each other. It furthermore provides the tool developer with information about the current results concerning the development of a standard interchange format for Petri nets based on XML.

**Facilities** For building prototypes this service uses the Petri net kernel(PNK). The PNK can be considered as an infrastructure for building Petri net tools by offering standard functions and a graphical user interface. The PNK is not restricted to a particular Petri net type but covers all relevant Petri net types due to parameterization. It covers the following sub-services:

- The PNK supports the implementation of a new algorithm for analysis, simulation, or verification. Parser, graphical interface etc. are provided by the PNK, such that the tool developer may concentrate on the new algorithm.
- The PNK comes with an XML based file format which follows the development of the standard interchange format for Petri nets.
- Information about the net is easily accessible via a simple interface, which reflects the typical mathematical notions on Petri nets such as pre- and postsets. Knowledge of a particular software technique is not required, such that an unexperienced programmer is able to efficiently use the interface within short time.
- The PNK supports the integration of several algorithms which have been developed independently into a single tool. This enables tailoring to specific application domains from a collection of available algorithms.
- The implementation of a tool for newly defined Petri net types is facilitated—again without implementing additional parse operations or editor functions.

**T.Maintenance** This service realizes a tool administration component which stores relevant informations about tools in the tool data base. The administration is based on a form containing all relevant information about a tool, as e. g. a general description, the underlying Petri net type(s), the implemented Petri net operations, some technical instructions, information about licences and some evaluation notes. Similar to the service **A.Maintenance** this service triggers completion of the Petri net technique by other user groups via the common service **Petri net technique Completion**.

The above listed tasks are supported mainly by the following services: The services **Petri Net Tool Selection**, **Search for Petri Net Input Formats**, and **T.Maintenance** support the task (i). Tasks (ii) task (iii) are mainly supported by service **Facilities**.

#### 4.4 Common Services

Common services play a key role among the services. They are not invoked directly by a user but indirectly by a user specific service. The common services provide integrating functionality such as:

- (i) search over the common base, and
- (ii) providing a common platform for discussion, and
- (iii) care for a consistent and coherent update of the whole database.

The following common services have been defined:

**Common Base-Navigation** This service facilitates a search for a Petri net type through traversing the Petri net classification in the Common Base.

**Common Base-Parameter Search** This service searches for a Petri net type according to input parameters.

**Petri net technique Completion** This service cares for a consistent and, if available, complete descriptions of Petri net techniques. For the description of a Petri net technique data from all different views are combined. The Common Base contributes the Petri net type, the Application Developer View a Petri net methodological procedure, the Expert View the formalization and the Tool Developer View a corresponding tool or toolset. The service **Petri net technique Completion** is invoked, if one of the maintenance services of the user groups adds something new to their database or the Common Base. This information is published to all users of the Petri net Baukasten and is combined with the request to contribute to this piece of information in order to collect and establish links to the other views.

**Petri net type insertion** This services updates the Common Base-database. For a new Petri net type a new class with corresponding attributes and functions is generated and ranged into the classification hierarchy. Finally it is linked to the user specific views.

**Parameter maintenance** This service cares for a consistent update of all ad-

missible input parameters. The parameter list can be changed explicitly through the access of users or automatically through a search of the database contents.

**Link Construction** In case some new data has been added to the database this service is invoked to check for existing relations between this new piece of information and data belonging to other views. These relations are made explicit through the insertion of links.

The above listed tasks are supported mainly by the following services: The task (i) is supported by **Common Base-Navigation** and **Common Base-Parameter Search**. The service **Petri net technique Completion** realizes task (ii), and the remaining common services mainly support task (iii).

## 5 Database

The database contains all data which are made available through the Petri net Baukasten. According to the concept of the Petri net Baukasten it is substructured into specific data for the different user groups and the Common Base.

### 5.1 *Application Database*

The application-oriented view upon Petri nets presented in the application database mainly comprises Petri net (development) methods, Petri net (development) techniques, and sample solutions. These are described by their names, their objectives, their intended application domains, and application-oriented keywords. Their informal application-oriented descriptions are given by documents called method guides, technique guides, and solution guides respectively. The application database manages several tool recommendations supporting the application of Petri net development methods and techniques respectively. Moreover, application domains play a central role in the application data base. Application domains are characterized by a number of application-oriented aspects reflecting all relevant entities and questions. Moreover, several suitable means of expression to specify a certain application-oriented aspect using Petri nets are managed in the database. Thereby, the database also contains an application-oriented interpretation of Petri net notions. Furthermore, the database contains application-oriented glossaries describing all used application-specific notions.

The structural model of the application database corresponds to the structural model of the Application Developer View of the Petri net Baukasten which is described as a UML class diagram in [17].

### 5.2 *Expert Database*

The theory of Petri nets presented in the expert database is based on the notions of abstract Petri net frames, formal Petri net techniques, actualizations, and transformations. We use schemes for the structured representation of these notions in order to obtain a uniform structure. The expert database contains in its initial state all the data of the Initial Realization in [7]. Additionally, data for the service **Add** scheme and **SearchAbstract Frame** is provided. This comprises definitions and conditions for all operations, which are based on general techniques. E.g. the structuring technique of union is based on the categorical pushout, and correspondingly the definition is included in the expert database. Similarly, the definition of high-level replacement systems [8] is contained, which are the basis for the structuring technique of rule-based modification.

### 5.3 *Tool Database (Tool Box)*

The tool database comprises all data which are especially relevant for developing a Petri net tool. This incorporates first of all tools which are provided either physically in form of executable program code or virtually in form of a link to the source. Furthermore the tool database holds all available information about the use and the construction of a tool. Information about use comprises manuals, user guidelines, tool documentations and information about the required operating system, about its licence, licence fee or its usability. The support for construction includes source codes, source code documentations, programming guidelines and information about existing input formats for Petri nets and input format transformations.

### 5.4 *Common Database*

The common database contains in its initial state the current version of the classification diagram presented in the Initial Realization [7]. Every Petri net type and its properties is described within the class notation of UML. The relations between different Petri Net types are denoted through the hierarchy relationships, such as specialization- and generalization, within the class diagram. The data stored in the common data base plays a key role, because it links the data of the three different views.

## 6 **Architecture for the 2nd Installment**

In this section we describe an architecture for the Petri net Baukasten which follows the design described in the preceding sections. It is derived from the 2nd Installment from Figure 3 by taking into account implementational details. Each part illustrated in Figure 3 corresponds to an element in the architecture. Additional components like user session services, user database,

and API etc. are important for a software realization, but not specific for the Petri net Baukasten such that they have no counterpart in Figure 3.

### 6.1 The Architecture

The architecture follows the classical trisection into graphical user interface (GUI), functionality layer and data repository. The architecture of the Petri net Baukasten is illustrated in Figure 4. It is based on middleware technologies in order to support distribution of the different parts (GUI, Services and Repository) on different servers.

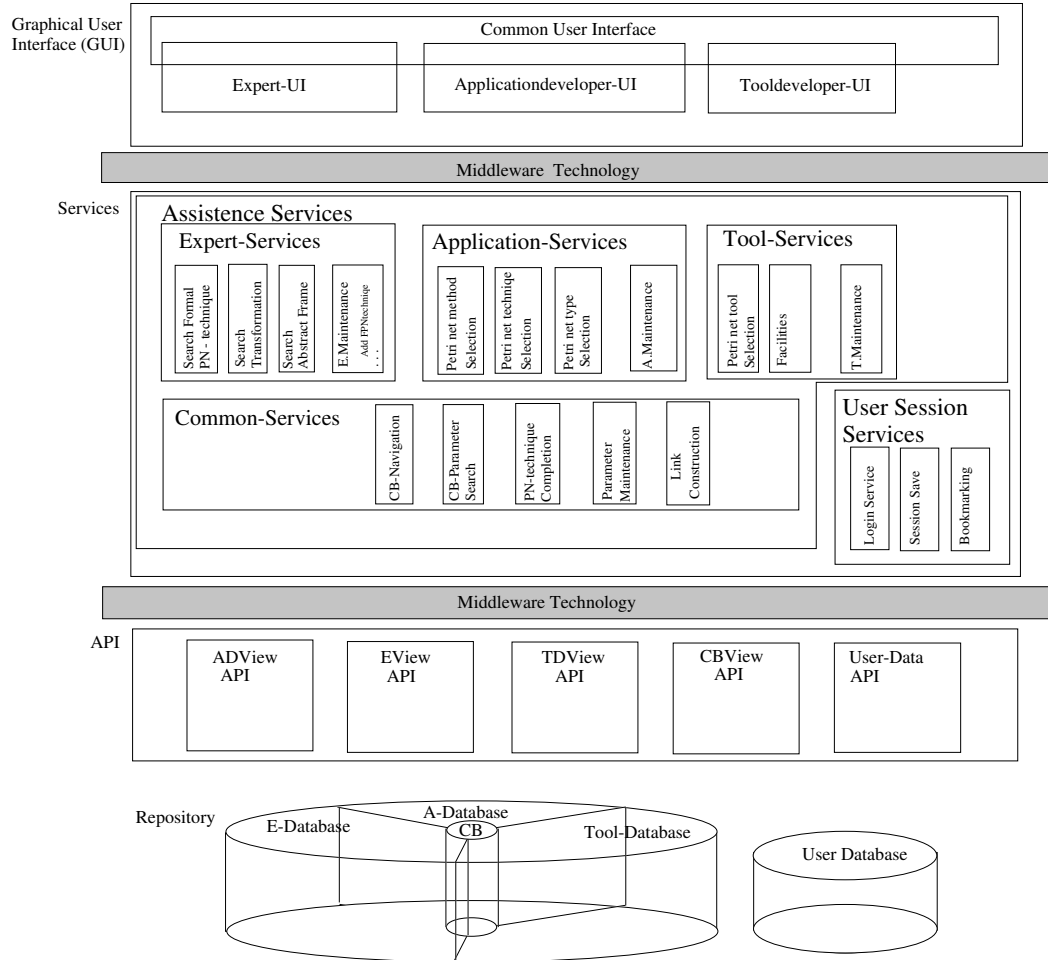


Fig. 4. An architecture for the Petri net Baukasten

The first part contains the user interface, which is subdivided into a common-, an expert-, an application developer-, and a tool developer-user interface. The specific sub-interfaces are reached through the common user interface, which realizes the choice of the user according to her role.

The user interface and the repository are related through services. The services are divided into *assistance Services* and *user session services*. Assistance services comprise *common services*, *application services*, *expert services* and



*tool services*. Each of these services is split up into search services and maintenance services. A particular user group service from *expert-, application- and tool services* is invoked through the corresponding user interface. The common services cannot be accessed directly by users but are called by the other assistance services. In Figure 4 we have only sketched some of the services of Section 4 for lack of space. The scope of the services and examples have been described in Section 4.

Besides the assistance services we consider the *user session services*. This set is accessed via the common user interface and takes over tasks like administration and supervision of user sessions. It comprises the save and load of login and session data as well as bookmarking.

Access to the repository is realized via an application programming interface (API). The API is a convenient interface, allowing abstraction from the repository realization.

The last part of the described architecture is the repository. Here all the relevant data are managed. There is one repository which holds the PN data discussed in Section 5. Additionally, we consider a user database which holds session data, bookmarks, notes etc. These data elements are generated during a session and are stored and provided for further use.

In [12, 17] an architecture of the Application Developer View is presented which refines the application developer-specific part of the architecture illustrated in Figure 4.

## 6.2 State of the Implementation

So far parts of the sketched system have already been implemented. The Petri net classification as main part of the common base was implemented using Rational Rose. The first upgrading level of this hierarchical order of Petri net types has been exported into an Oracle database. As part of the application developer assistance we implemented a service for the navigation through the Petri net classification to enable the selection of a suitable Petri net type. This service access the exported data of the Oracle database. Last, but not least, the Petri net kernel yielding an infrastructure for developing Petri net tools has been implemented already on an prototypical level, and is currently being upgraded.

## 7 Conclusion

The benefit of the 2nd Installment of the Petri net Baukasten is illustrated by a user scenario describing the use of the Petri net Baukasten in a distinguished application context. Scenarios have been described already in the Initial Realization [7]. They can now be described more precisely as a sequence of services explicitly offered by the 2nd Installment. As a conclusion of this paper we outline a specific scenario for an application developer. It describes the search

for a Petri net type suitable for the modeling of specific workflow aspects.

The *common user interface* is open and allows the user to login by calling the *user session service* **Login Service**. Here the system identifies the user, loads the status of the last interaction, and asks if he wants to continue his activity as application developer. In this case the *Application developer interface* is opened and the old session data is loaded. The application developer may now continue his work. Lets assume he wants to select a certain Petri net type that fits certain workflow aspects. He selects the service **Petri net type selection** which offers an application-oriented checklist or navigation through the Petri net classification. Opening a modeling related checklist the user may take his choice between a combination out of functional, data/document, roles, ressource and time aspects. Assume he wants to model functional aspects combined with ressource and time aspects. He selects these aspects, which are used by the system as input parameters. The system determines Petri net types, which comprise the demanded aspects. The resulting list, comprising e.g. Petri net types discussed in [2, 3, 5, 9], is presented in a new window. To facilitate the choice between the variety of Petri net types some more information (informal-, formal description of the used Petri net types, analysis methods, tool support, examples) about the results is provided. Assume our user wants to apply qualitative analysis methods and performance evaluation. This reduces the choice to the approaches based on stochastic Petri nets, whereby only the approach in [5] provides tool support. After the selection of one Petri net type, the system provides the user with relevant data and tool support.

The relevant assistance service in this scenario is **Petri net type selection**. The search is performed using a modeling related checklist of application oriented aspects, invoking a search service provided through the **common services: Common-Base Parameter Search**. Accessed data are the Petri net classification stored in the *Common Base-database*, (meta-)information from the *application-database*, tools, or links to tools from the *tool-box* and user data from the *user database*.

## Acknowledgement

This work is part of the joint research project “DFG-Forschergruppe PETRI-NETZ-TECHNOLOGIE” between H. Weber (Coordinator), H. Ehrig (both from the Technical University Berlin) and W. Reisig (Humboldt-Universität zu Berlin), supported by the German Research Council (DFG). The development of the Petri net Baukasten has been a major task, which has been achieved by an intensive cooperation among the members of the DFG-Forschergruppe. We would like to express our gratitude for very fruitful cooperation.

## References

- [1] Aalst, W.M.P. van der, *The application of Petri nets to workflow management*. The Journal of Circuits, Systems and Computers **8**(1):21–66, 1998.
- [2] Aalst, W.M.P. van der, K.M. van Hee, and G.J. Houben. *Modelling workflow management systems with high-level Petri nets*. In G. De Michelis, C. Ellis, and G. Memmi, editors, *Petri nets and related formalism*, second Workshop on Computer-Supported Cooperative Work, pages 31–50, 1994.
- [3] Adam, N.R., V. Atluri, and W-K. Huang. *Modeling and analysis of workflows using Petri nets*, Journal of Intelligent Information System, Special Issue on Workflow and Process Management **10**(2):131–158, 1998.
- [4] Dehnert, J., *The common base of the Petri net Baukasten*, In Weber et al. [16], pages 211–229.
- [5] Dehnert, J., A. Freiheit, and A. Zimmermann, *Workflow modeling and performance evaluation with colored stochastic Petri nets*, In *Bringing Knowledge to Business Processes, Workshop in the American Association for Artificial Intelligence (AAAI) Spring Symposium Series 2000*, 2000, accepted as poster.
- [6] Deiters, W., V. Gruhn, and R. Striemer. *Der FUNSOFT-Ansatz zum integrierten Geschäftsprozessmanagement*, Wirtschaftsinformatik **5**(37):459–466, 1995.
- [7] DFG-Forschergruppe PETRI NET TECHNOLOGY, Initial Realization of the “Petri Net Baukasten”, Informatik-Berichte **129**, Humboldt-Universität zu Berlin, October 1999.
- [8] Ehrig, H., M. Gajewsky, and F. Parisi-Presicce, *High-Level Replacement Systems with Applications to Algebraic Specifications and Petri Nets*, Volume 3: Concurrency, Parallelism, and Distribution, chapter 6, pages 341–400, World Scientific, Handbook of Graph Grammars and Computing by Graph Transformations edition, 1999.
- [9] Ferscha, A., *Qualitative and quantitative analysis of business workflows using generalized stochastic Petri net*, In G. Chroust and A. Benczu, editors, *CON’94: Workflow Management - Challenges, Paradigms and Products*, pages 222–234. Oldenbourg Verlag, 1994.
- [10] Gajewsky, M., *The expert view of the Petri net Baukasten*, In Weber et al. [16], pages 243–265.
- [11] Gajewsky, M. and H. Ehrig, *The “Petri Net Baukasten”: An Overview*, In H. Ehrig, G. Juhás, J. Padberg, and G. Rozenberg, editors, *Unifying Petri Nets*, Advances in Petri Nets, LNCS. 2001. To appear.
- [12] Lembke, Sabine, “Ein anwendungsorientierter Ansatz zur Verbesserung der praktischen Nutzbarkeit von Petrinetzen,” PhD thesis, Technical Uni-

versity of Berlin, 2001, in preparation.

- [13] Oberweis, A., “Modellierung und Ausführung von Workflows mit Petri-Netzen,” Teubener-Reihe Wirtschaftsinformatik. B.G. Teubener Verlagsgesellschaft, Stuttgart, Leipzig, 1996.
- [14] Object Management Group (OMG), <http://www.omg.org/>, “Unified Modeling Language Specification – Version 1.3,” March 2000.
- [15] Padberg, J., *The Petri net Baukasten: An application-oriented Petri net technology*, In Weber et al. [16], pages 191–209.
- [16] Weber, H., H. Ehrig, and W. Reisig, editors, “Int. Colloquium on Petri Net Technologies for Modelling Communication Based Systems, Part II: The ‘Petri Net Baukasten’,” Fraunhofer Gesellschaft ISST, October 1999.
- [17] Weber, H., S. Lembke, and A. Borusan, *Improving the usability of Petri nets with the “Petri Net Baukasten”*, In H. Ehrig, G. Juhás, J. Padberg, and G. Rozenberg, editors, *Unifying Petri Nets*, Advances in Petri Nets. Springer, 2001, To appear.
- [18] Weber, M., *The tool developer view of the Petri net Baukasten*, In Weber et al. [16], pages 267–277.