The Petri Net Baukasten of the DFG Forschergruppe PETRI NET TECHNOLOGY

H. Weber[1], H. Ehrig[1], W. Reisig[2], A. Borusan[1], S. Lembke[1], J. Dehnert[1], M. Weber[2], A. Martens[2], J. Padberg[1], C. Ermel[1], and A. Qemali[1]

[1] Technical University of Berlin
[2] Humboldt University of Berlin

Abstract. 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 not any more handable for anyone working with Petri nets, which results in the strong need of a structured access to Petri nets. This 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. This paper presents an overview of the concepts, initial and 2nd installment of the Petri Net Baukasten, which have been presented at the 1st and 2nd International Colloquium on Petri Net Technologies for Modelling Communication Based Systems in 1999 and 2001, respectively.

Keywords: Petri Nets, Petri Net Technology

1 Introduction

Petri Nets are a subject of interest to researchers since more then 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 notions used in practice are 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 has frequently been attributed to the fact that Petri nets and the associated theory are hard to understand. Moreover, people applying 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 exciting types, finding corresponding techniques and tools, selecting them and applying them successfully to the 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

H. Ehrig et al. (Eds.): Petri Net Technology ... , LNCS 2472, pp. 1–21, 2003.

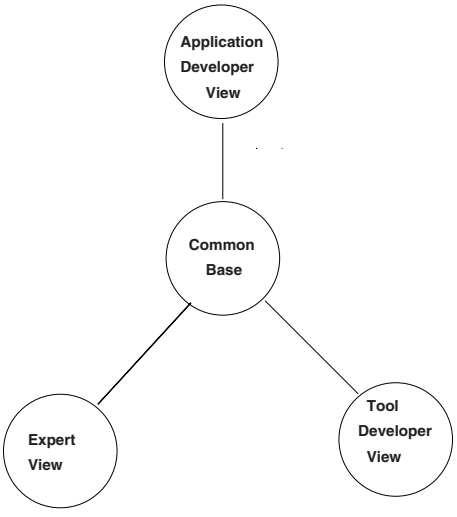

Fig. 1. The conceptual structure of the Petri net Baukasten

NET TECHNOLOGY in Berlin. It aims at the development of a support enviroment for the use of Petri nets in new applications including theoretical and tool's aspects. Figure 1 roughly sketches 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, all linked via a Common Base. The Common Base 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.

In this paper we present an overview of the Petri Net Baukasten, based on the initial installment, published as "Initial realization of the Petri Net Baukasten" in [18], and on the second installment in [19]. These installments have been presented at the 1st and 2nd International Colloquium on Petri Net Technologies for Modelling Communication Based Systems.

In Section 2 of this paper we review the concepts and the initial installment of the Petri Net Baukasten. The conceptual structure including the Common Base and three different views is shown in Figure 1. The initial installment comprises the contents of these views and the Common Base on a low technical level with direct access to the corresponding documents in a more or less static way. A more detailed presentation of the Petri Net Baukasten in the initial phase is given in [9].

In contrast to the initial installment, the second one presented in Section 3 of this paper provides an interactive access to the contents of the Common Base and the specific views including dynamic changes. A full implementation, however, is beyond the capacities of the research group PETRI NET TECHNOLOGY. The 2nd installment consists of a database, services operating on this database and explicit access for each user group (user interfaces). The database may be searched or updated by the services. Updating requires some mode 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 a Petri net notion, resp. 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 an inserted piece of data has to be related to corresponding data of the other views. 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.

In Section 4 we discuss the database aspect in more detail and in Section 5 an architecture for the 2nd Installment. Finally in Section 6, we summarise the main ideas and discuss how other contributions of the PETRI NET TECHNOLOGY researcher groups in this volume are related to the Petri Net Baukasten.

2 Concepts and Initial Installment of the Petri Net Baukasten

2.1 Aims of the Petri Net Baukasten

The main aim of the Petri Net Baukasten is to provide support for various 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 proven to successfully support the development of a system of that kind, e. g. automatic production, 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.

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 central notion of the Petri Net Baukasten are Petri net techniques, illustrated in Figure 2. A Petri net technique is given as a consistent combination of a Petri net type, its formalization, a methodological procedure, and a corresponding tool.

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 includes links to the other views.

The formal description of a Petri net type is given in the Expert View. For each aspect 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 tools are provided, customized to the Petri net type. That is, they work on Petri nets comprising the basic constituents and implement the operational characteristics of the type.

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

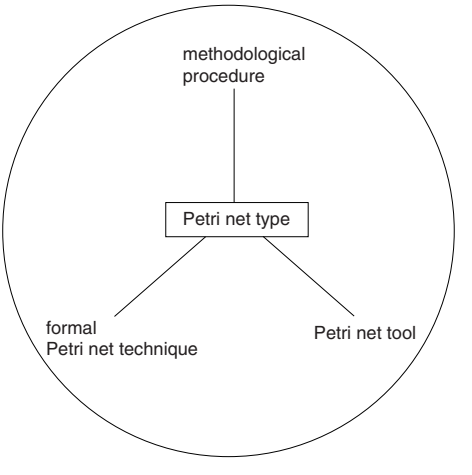

Fig. 2. The Petri net technique

2.3 The Common Base

The Common Base, see [4, 23], encompasses all Petri net types relevant for a specific application domain, it integrates all these types into a common scheme and defines the relationship between the Petri net types. It is intended that the Common Base is structured as simple as possible and understandable to application developers, tool developers and Petri net experts alike. The structuring concept that is considered to be simple and still powerful enough to capture all the properties of different Petri net types is a specialization/generalization relationship between the different Petri net types. The Petri net classification in the Common Base is a specialization hierarchy along distinguishing characteristics. This classification is represented in the Common Base using class diagrams of UML. The distinguishing characteristics are given in terms of attributes and attribute values. They describe Petri net notions on a conceptual level. The domain dependent interpretation of these concepts are part of the Application Developer View. The encoding into software belongs to the Tool Developer View and the mathematical formalization of these concepts is part of the Expert View.

2.4 The Application Developer View

The Application Developer View, see [15, 23], provides the prerequisites for an application-oriented assistance of Petri net based development projects. The planning of development activities in an application domain is supported in the following way: The Application Developer View provides information for the planning of development activities, i. e. methods of employing Petri net techniques. This information supports the application developer to choose the appropriate Petri net techniques, based on tools, methodological procedures and formal Petri net techniques for the development activities. Additionally, modelling activities are supported by sample and standard solutions, and methodological procedures for a chosen Petri net type. The Application Developer View includes information on Petri net notions and Petri net techniques in an application- and problem-oriented way. These so-called *assistance concepts* yield domain specific interpretations of Petri net notions of the Common Base.

2.5 The Expert View

The Expert View provides the formal foundation of Petri net types in the Petri Net Baukasten in terms of a mathematical presentation of the underlying notions and results. The Expert View is given in a structured way. It comprises formal Petri net techniques, abstract Petri net frames, actualizations, and transformations. Each of these notions includes a coherent and consistent piece of Petri net theory. Formal Petri net techniques, and transformations are directly related to the Common Base. Abstract Petri net frames, actualizations, and abstract transformations describe relation and dependencies of formal Petri net techniques and transformations on a more abstract level. This yields a uniform description of Petri nets as a foundation of the classification given in the Common Base.

2.6 The Tool Developer View

The main task of the Tool Developer View, [for details see [24, 12]], is to provide support for tool development. This comprises the management of existing tools, facilities for tool development as well as possibilities to extend tools. Petri net tools support Petri net based system development. They provide support for editing, simulating, structuring, and analyzing a Petri net variant. The management of existing tools is also important for the rest of the Petri Net Baukasten, since it offers tool support for the Petri net types of the Common Base. A Petri net tool corresponds to a Petri net type of the Common Base if it supports the notions comprised by that Petri net type. The Tool Developer View provides support for the development of tools in terms of object-oriented and parameterization concepts, a component-oriented approach for already existing tools and encoded algorithms. The Petri net type used in the encoded algorithm is given in the Common Base. The formal representation of the algorithm is given in the corresponding formal Petri net technique in the Expert View.

2.7 Relation of Common Base with Views

The Common Base is related to the views according to different representations of Petri net notions in the Common Base and the views.

The notions given in the Common Base in an informal way by attributes are represented in the Application Developer View by application-oriented notions within a methodological procedure, explaining the use of these notions in a specific application domain. Within the Tool Developer View these notions are represented either as algorithms or as tools. The Expert View provides consistent formal Petri net techniques for the Petri net type given in the Common Base.

As an example the notion *marking of a Petri net* may be considered

- in the Common Base to be a distribution of tokens over places indicating the state of the net,
- in the Application Developer View to be a representation of documents and business objects in different processes,
- in the Tool Developer View to be a record of places, and
- in the Expert View to be an element of the free commutative monoid over the set of places.

The specialization hierarchy of the Common Base can be used for the navigation with respect to all three views. It allows embedding of new tools or new theoretical results and making them available for practice.

2.8 Initial Installment of the Petri Net Baukasten

A detailed documentation of the initial installment is given in [3]. In the following we only summarise the main ideas.

Common Base
In the initial installment a class diagram representing the classification of Petri net types relevant for business processes has been established and is depicted in [7] using the UML tool Rational Rose. The classification comprises about 100 Petri net types and twelve different specialization paths, that can be partitioned into three categories: elementary, additional, and operational extensions.

Application Developer View
In the initial installment, an assistance system is provided on a conceptual level for the support of the application developer. There are two kinds of variants for the assistance that can be distinguished, namely assistance to find a suitable Petri net technique and assistance to find a suitable solution example. Both variants are supported by prescriptive, navigating, or descriptive assistance methods. The system architecture sketches the fundamental ideas about the assistance system in [7]. More details are given in [23].

Expert View
The structured representation of significant aspects of Petri net theory in the initial installment is based on schemes for abstract Petri net frames, formal Petri net techniques, actualizations, and transformations. Each of these schemes consists of a list of relevant keywords. The initial installment of the Expert View comprises instantiations of all these schemes, e. g. parameterized net classes as an actualization of abstract Petri net frames, or of algebraic high-level nets, coloured Petri nets, elementary nets, place/transition nets, and FunSoft nets as instantiations of formal Petri net techniques.

Petri Net Kernel in the Tool Developer View
The Petri Net Kernel in the initial installment is an object-oriented tool for the fast prototyping of simple Petri net tools. The basic idea of the Petri Net Kernel is the distinction between fixed and variable aspects of Petri nets. Variable aspects can be considered as parameters and allow the automatic generation of Petri net tools by actualization of these parameters. It implements the part of the Tool Developer View that affects tool development. The implementation of parameterized net classes with the Petri Net Kernel is discussed in [12] of this volume.

3 2nd Installment of the Petri Net Baukasten

3.1 Survey of the 2nd Installment

We are now going to give an overview over the 2nd installment of the Petri Net Baukasten. The 2nd installment roughly consists of three parts: The data which is to be provided, a user interface for communication between user and system, and thirdly an intermediate level which operates on the data. We call these parts *database*, *user interface* and *services*, respectively. Each of these parts is structured internally in order to increase clarity and handability. 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

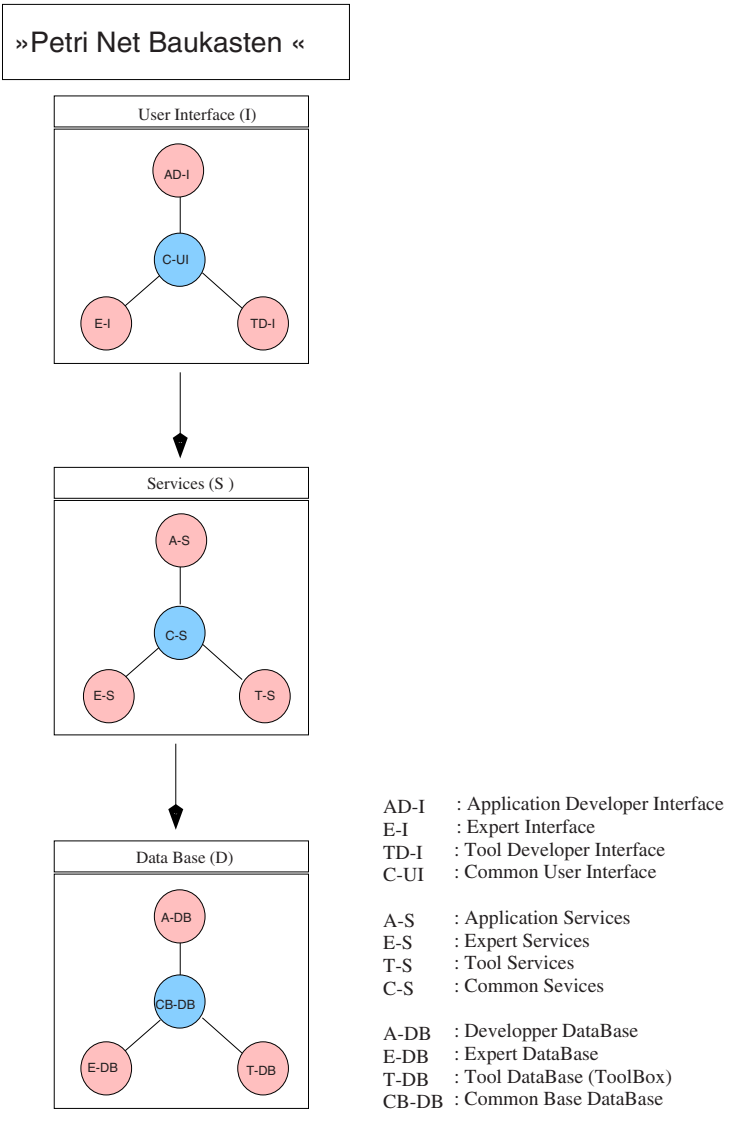

Fig. 3. 2nd Installment of the Petri Net Baukasten

correspond to one of the views (application developer view, tool developer view, and expert view). This structure is motivated by the general aim to support each of these groups. This view-oriented structure is shown in Figure 3. 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.

Database The contents of the 2nd installment of the Petri Net Baukasten is stored in the database. In contrast to the initial installment 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 various 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, to be gradually enriched. For convenience we assume the data of the initial installment as its contents. More details about the database can be found in Section 4.

Services operate on the database. We distinguish between common-, application-, expert-, and tool-services, which are described in Section 3. 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. 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 represents the user interface. We provide a general user interface from where a specialized interface, can be reached for any of the different user groups, application developers, experts, and tool developers. 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 2.

3.2 User Interface

We will describe the user interface of the Petri net Baukasten in more details here. We provide a general user interface as a basic for a specialized interface for any of the different user groups, application developers, experts, and tool developers. 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. As a result of this, a separation of concerns is achieved leading to a more requirement-oriented and easy use of the Petri Net Baukasten.

The belonging to a certain user group depends on the objectives of the particular users. If their objectives change over the time the belonging to a certain user group may also vary. For example, a user may be interested in the formal development of a specific Petri net type in one moment and in the implementation of an appropriate Petri net tool in the next moment. So, this person uses 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 general user interface, which realizes the user choice according to their role. A second task of the general 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; forms allow to update parameters. Moreover, each interface offers a help function explaining its specific functionality.

3.3 Services

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

Search services mainly realize the interactional aspect of the 2nd installment of the Petri Net Baukasten. They retrieve 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, which are discussed below. These group-specific services use common services in order to fulfill their tasks.

3.4 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:

1. to find an adequate Petri net based method for developing the application,
2. to use Petri net techniques that serve best in the development of the application,
3. to find a Petri net type which fits the application,
4. to find a suitable tool,
5. provide the application developer with example solutions.

Accordingly, we define the following application services, which are in a one-to-one correspondence to the above listed tasks.

Petri Net Method Selection This service assists finding a suitable Petri net based development method. The service allows to search over objectives, intended application domains and covered development phases (analysis, design, etc.) of Petri net methods. After the 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 the method guide before selecting their suitable Petri net method.

Petri Net Technique Selection This service covers the assistance in finding a suitable Petri net technique. The application developer specifies required characteristics of a Petri net technique. This can be done, among others, with the help of keywords which outline the 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.

Developers select their suitable Petri net technique from the offered set. They may use information 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 assists in finding a suitable Petri net type which fulfills the requirements of a specific application development process. There are several assistance methods for finding a suitable Petri net type , as for example, 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 then automatically determines Petri net types which support corresponding language concepts. The found Petri net types are offered to the application developer. Together with the list of resulting Petri net types, some more information about the different results is provided, as for exampl intended application domains, possible tools, informal and formal description. These information facilitates the final 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. the required operating system, purpose, usability) and external criteria (as e.g. usage costs). This enables 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 or ordered from its supplier.

Select a Sample Solution This service is intended for finding solutions suitable for fulfilling the requirements of a certain application development. 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 the so-called *solution guide* for details.

Application 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.

A more precise description of the application developer services, their realization and introduced terminology can be found in [22].

3.5 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:

1. to elaborate and enhance existing formal techniques,
2. to define new types and notions in a uniform way,
3. to state properties of all variants of Petri nets in a formal and constructive manner,
4. to transfer results between net types,
5. to make results and notions available for applications,
6. structured access to the theory of Petri nets.

Accordingly, we define the following expert services:

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.
This service supports the tasks (1) and task (2) as the expert may use existing definitions for enhancement.

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.

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.

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

Search Scheme 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. Again, tasks (1)and (2) are supported as well as tasks (2), (3), and (4).

Add Scheme The expert fills out the corresponding scheme for abstract Petri net frames, or instantiations, which is added to the database.

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. Consequently, there is no service "**Add Precondition**" as these are already added by adding an abstract frame.

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 which 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. This service supports task 5.

3.6 Tool Services

The task of the tool developer is supported by the Tool Developer View, also see [24]. The support comprises various aspects of tool development:

1. management of existing tools,
2. facilities for tool development, and
3. support for extension of existing tools.

Accordingly, we define the following tool-services.

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. This collection of information enables 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 the 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 mutually. 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 provides 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 new algorithms for analysis, simulation, or verification. Parsers, graphical interfaces 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 respects 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.

Tool Maintenance This service realizes a tool administration component which stores relevant information about tools in the tool database. The administration is based on a form containing all relevant information about a tool, as for example a general description, the underlying Petri net type(s), the implemented Petri net operations, some technical instructions, information about licences and some evaluation notes.

3.7 Common Services

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

1. search over the common base, and
2. care for a consistent and coherent update of the whole database, and provide a common platform for discussion.

The following common services have been defined:

Common Base-Navigation This service faciliates a search for a Petri net type by 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 contributes the formalization and the Tool Developer View contributes 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 service 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 admissible 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 new data have 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.

4 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.

4.1 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 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 of a particular application domain. 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.

4.2 Expert Database

The theory of Petri nets presented in the expert database is based on the notions *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. This database contains all the data of the initial realization, see [7]. Additionally, data for the service Instantiation are provided. This comprises definitions and conditions for all operations, which are based on general techniques. For example, the structuring technique of union is based on the categorical pushout, and correspondingly the definition is included in the expert database. Similarily, the definition of high-level replacement systems [8] is contained, because high-level replacement systems are the basis for the structuring technique of rule-based modification.

4.3 Tool Database

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 stores 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.

4.4 Common Database

The common database contains the current version of the classification diagram presented in the initial realization, see [7]. Every Petri net type and its properties are 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 play a key rule, because they link the data of the three different views.

5 Architecture for the 2nd Installment

In this section we describe an architecture for the Petri Net Baukasten which satisfies the above requirements. We outline its use with the help of a scenario.

5.1 The Architecture

The architecture follows the classical trisection into graphical user interface (GUI), functionality layer and data repository. The architecture of the Petri

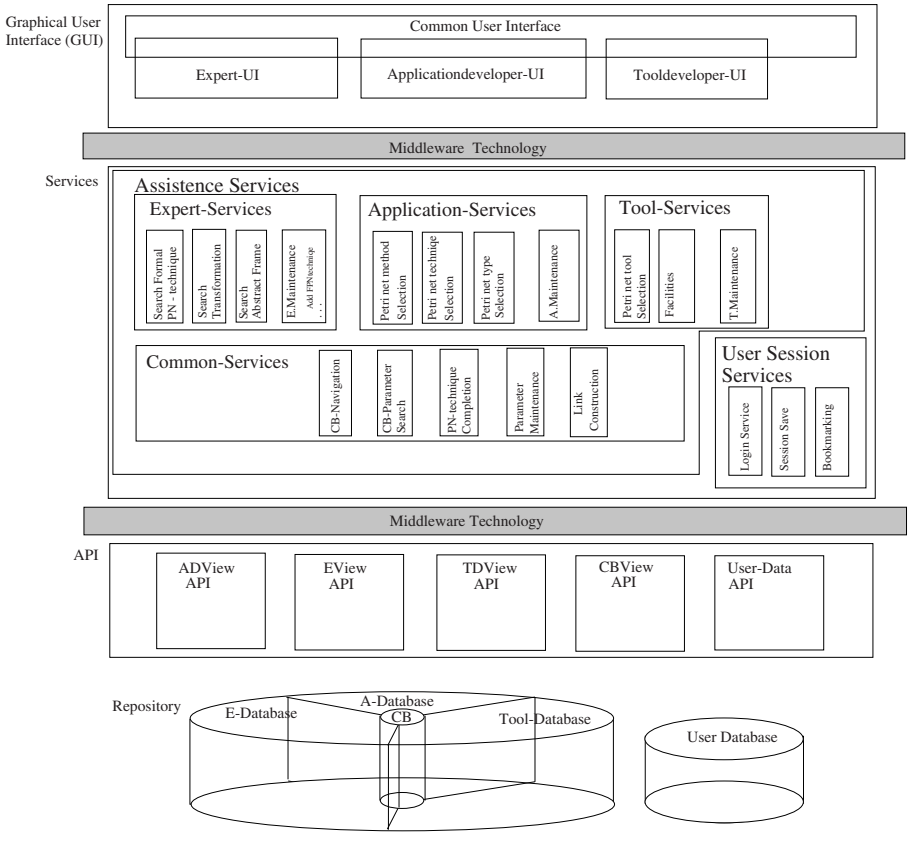

Fig. 4. Architecture of the Petri Net Baukasten

Net Baukasten is illustrated in Figure 4. It is based on middleware technologies in order to support distribution of the its parts (GUI, Services and Repository) on different servers.

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

The user interface and the repository are related through services. The services are divided into *common services*, *application services, expert services* and *tool services*. These assistance services are split up into *search services* and *maintenance services*. The specific user group services, as *expert-, application- and tool services* are invoked through the corresponding user interfaces. The common services take over an integrating part. Their search services over the common base are used by every user group; hence are accessed via the corresponding user inter-

faces. Their maintenance services are invoked through the general user interface, hence addressing all users indepently from a chosen role.

The scope of the services and examples have been described in Section 3.

Apart from specific user group and common services we consider a third group of services: the *user session services*. This group is accessed via the general user interface and takes over tasks like administration and supervision of user sessions. It comprises storing and loading 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 convienient interface, allowing abstraction from the repository realization.

The last part of the described architecture is the repository. Here all the relevant data are mangaged. There is one repository which holds all the databases discussed in Section 4. 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.

5.2 Scenario

For better illustration we will explain the architecture with the help of a user scenario. This scenario concerns the use of the Petri Net Baukasten by an application developer and describes the search for a Petri net type suitable for the modeling of specific workflow aspects.

The *general user interface* is open and allows the user to login. With the help of the user identity the system loads the status of the last interaction of this user and asks if he/she wants to continue his/her activity as application developer. In this case the *Application developer interface* is opened and the old session data are loaded. The application developer may now continue her work. Lets assume she wants to select a certain Petri net type that fits certain workflow aspects. The user may want to select the Petri net type with the help of a checklist. Opening a modeling related checklist the user may take his/her choice between a combination out of functional, data/document, roles, resource and time aspects. Assume the user wants to model functional aspects combined with ressource and time aspects. The system determines Petri net types, which comprise language concepts that depict the selected choice of workflow aspects (e.g. [2, 1, 13, 6]). The resulting list 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 the approach in [6] provides tool support. After the selection of one Petri net type, the system provides the user with relevant data (full papers) and tool support.

The relevant assistance service in this scenario was Petri net type selection. The search was performed using a modeling related checklist of application

oriented aspects, invoking a search service provided through the common services. Accessed data have been 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*.

6 Conclusion and Related Work

In the paper we have presented an overview of the concepts, initial and 2nd installment of the Petri Net Baukasten developed by DFG Forschergruppe PETRI NET TECHNOLOGY. For a more detailed presentation we refer to the proceedings of the 1st and 2nd International Colloquim on Petri Net Technologies for Modelling Communication of Based Systems[[25] [7]. One of the main aims of the Petri Net Baukasten is a unified presentation of the large variety of different Petri nets types and techniques. For this reason part I of the state of the art survey [9] on unifying Petri nets is mainly devoted to the Petri Net Baukasten, including an overview of the initial installment, assistance for the application development and an implementation of parameterized net classes with Petri Net Kernel. A main part of the expert view of the Petri Net Baukasten is based on parameterized net classes presented in part II of the same volume [10]. Other contributions closely related to the Expert View are the Petri net transformations [3], rule-based refinement [16], Petri net class transformations [17] and open Petri nets [14] presented in this volume. More details concerning the Application Developer View in general are given in [5], while specific case studies concerning logistics and train control systems are given in other contributions of this volume respectively. Last but not least the contributions [20, 21, 11] are most important for the Tool Developer View presenting the Petri Net Kernel, the Petri net markup language and tool support for animation of Petri nets respectively.

Finally we hope that the work on the Petri Net Baukasten will not only be continued by the Researcher group in Berlin, but also by other groups in different application domains. Moreover the general idea of the Petri Net Baukasten is not at all restricted to Petri nets, but can be extended to other kinds of specification and modelling techniques for communication based systems.

References

[1] W. M. P. van der Aalst, 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. 18

[2] Nabil R. Adam, Vijayalakshmi Atluri, and Wei-Kuang Huang. Modeling and analysis of workflows using Petri nets. *Journal of Intelligent Information Systems*, 10:131–158, 1998. 18

[3] B. Braatz, H. Ehrig, and M. Urbasek. Petri Net Transformation in the Petri Net Baukasten. In Ehrig et al. [10]. To Appear. 19

[4] J. Dehnert. The Common Base of the Petri Net Baukasten. In Weber et al. [18], pages 211–229. 5

[5] J. Dehnert. Four Steps Towards Sound Process Models. In Ehrig et al. [10]. To Appear. 19
[6] J. Dehnert, 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. 18
[7] DFG-Forschergruppe Petri Net Technology. Initial realization of the ≫Petri Net Baukasten≪. Informatik-Berichte 129, Humboldt-Universität zu Berlin, October 1999. 7, 16, 19
[8] H. Ehrig, M. Gajewsky, and F. Parisi-Presicce. *High-Level Replacement Systems with Applications to Algebraic Specifications and Petri Nets*, chapter 6, pages 341–400. Number 3: Concurrency, Parallelism, and Distribution in Handbook of Graph Grammars and Computing by Graph Transformations. World Scientific, 1999. 16
[9] H. Ehrig, G. Juhás, J. Padberg, and G. Rozenberg, editors. *Advances in Petri Nets: Unifying Petri Nets*, volume 2128 of *LNCS*. Springer, 2001. 2, 19, 21
[10] H. Ehrig, W. Reisig, G. Rozenberg, and H. Weber, editors. *Advances in Petri Nets: Petri Net Technologies for Modeling Communication Based Systems*. LNCS. Springer, 2002. To Appear. 19, 20
[11] C. Ermel, R. Bardohl, and H. Ehrig. Generation of Animation Views for Petri Nets in GENGED. In Ehrig et al. [10]. To Appear. 19
[12] C. Ermel and M. Weber. Implementation of Parametrized Net Classes with the Petri Net Kernel of the Petri Net Baukasten. In H. Ehrig, G. Juhás, J. Padberg, and G. Rozenberg, editors, *Advances in Petri Nets: Unifying Petri Nets*, LNCS. Springer, 2001. 6, 7
[13] A. Ferscha. 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. 18
[14] R. Heckel. Open Petri Nets as Semantic Model for Workflow Integration. In Ehrig et al. [10]. To Appear. 19
[15] S. Lembke. The Application Developer View of the Petri Net Baukasten. In Weber et al. [18], pages 231–241. 5
[16] J. Padberg and M. Urbasek. Rule-Based Refinement of Petri Nets: A Survey. In Ehrig et al. [10]. To Appear. 19
[17] F. Parisi-Presicce. A Formal Framework for Petri Net Class Transformations. In Ehrig et al. [10]. To Appear. 19
[18] H. Weber, 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. 2, 19, 20, 21
[19] H. Weber, H. Ehrig, and W. Reisig, editors. *2nd Int. Colloquium on Petri Net Technologies for Modelling Communication Based Systems*, Berlin, Germany, 2001. Research Group ≫Petri Net Technology≪, Fraunhofer Gesellschaft ISST. 2
[20] H. Weber and E. Kindler. The Petri Net Kernel. In Ehrig et al. [10]. To Appear. 19
[21] H. Weber and E. Kindler. The Petri Net Markup Language. In Ehrig et al. [10]. To Appear. 19
[22] H. Weber, S. Lembke, and A. Borusan. Improving the Usability of Petri Nets with the ≫Petri Net Baukasten≪. In G. Rozenberg, H. Ehrig, J. Padberg, and G. Juhás, editors, *Unifying Petri Nets*, Advances in Petri Nets. Springer, 2001. To appear. 12

[23] H. Weber, S. Lembke, and A. Borusan. Improving the Usability of Petri Nets with the Petri Net Baukasten. In Ehrig et al. [9], pages 54–78. 5, 7

[24] M. Weber. The Tool Developer View of the Petri Net Baukasten. In Weber et al. [18], pages 267–277. 6, 13

[25] M. Wermelinger. *Specification of Software Architecture Reconfiguration*. PhD thesis, Universidade Nova de Lisboa, 1999. 19