

INCOME2010 - a Toolset for Developing Process-Oriented Information Systems Based on Petri Nets

Stefan Klink, Yu Li, Andreas Oberweis

Institute of Applied Informatics and Formal Description Methods (AIFB)
University of Karlsruhe
76128 Karlsruhe, Germany

{stefan.klink | yu.li | oberweis}@aifb.uni-karlsruhe.de

ABSTRACT

Petri nets are widely accepted as a process modeling technique with formal semantics and mathematical foundation. They are well suitable for modeling (business) processes in executable form and for analyzing process models through simulative validation and formal verification. Nevertheless, there are currently only a small number of Petri net based tools available for developing process-oriented information systems, which have gained importance in recent years because of the awareness that business processes are one of the key success factors for organizations. In this paper we survey the project INCOME2010 aiming at developing an open source software toolset that facilitates the development of process-oriented information systems. It provides functionalities for modeling, analyzing, executing, and monitoring (business) processes using variants of high-level Petri nets and concepts of Service-Oriented Architecture.

Categories and Subject Descriptors

D.2.2 [Software Engineering]: Design Tools and Techniques – Petri nets.

General Terms

Design

Keywords

Petri nets, XML nets, software toolset, information systems, SOA

1. INTRODUCTION

In recent years process-oriented information systems (IS) have steadily gained importance in comparison to traditional function-oriented IS because business processes are one of the key success factors for organizations of all sizes. As a result, there has been a distinct increase in the number of tools that facilitate the development of process-oriented information systems based on various process modeling methodologies.

Petri nets are widely accepted as a well-founded process model-

ing technique with formal semantics, graphical nature, high expressiveness, analyzability and vendor-independence [15]. They are well suitable for modeling business processes with related process and data objects in executable form and for analyzing process models through simulative validation and formal verification. Since the introduction of high-level Petri nets in the 80's of the last century [5], Petri net theory has been greatly enriched through enormous research efforts resulting in a large number of theoretical publications. Nevertheless, there are still only a small number of Petri net based tools available for the development of process-oriented information systems.

With the globalization of the economy, the significance of inter-organizational business processes has been realized by more and more organizations and researchers. During execution of inter-organizational business processes, process objects (e.g., orders, invoices, protocols etc.) are exchanged among cooperating organizations. As a variant of high-level Petri nets, *XML nets* [10] have advantages especially in the description of process objects and inter-organizational exchange of standardized structured data (e.g., XML documents). However, there is currently still no software tool available that supports modeling, analysis, execution and monitoring of business processes using XML nets.

In this paper we survey the project INCOME2010¹ aiming at developing an open source software toolset² that facilitates the development of process-oriented information systems. It provides functionalities for modeling, analyzing, executing and monitoring (business) processes using variants of high-level Petri nets (e.g., XML nets) and concepts of Service-Oriented Architecture (SOA). The extensible plug-in architecture of INCOME2010 allows different Petri net variants to be implemented as plug-ins which can be easily added to the main platform and used for different types of process-oriented information system.

This paper is structured as follows: The next section examines related work. Section 3 gives a brief explanation of XML nets with a simple example. Section 4 discusses general require-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

International Workshop on Petri Nets Tools and Applications
PNTAP 2008, March 3, 2008, Marseille, France.

Copyright 2008 ACM ISBN 978-963-9799-20-2 ...\$5.00.

¹ INCOME stands for “Interactive Conceptual Modeling Environment”. An earlier version of INCOME2010 is described in [11]. INCOME2010 is being developed collaboratively by the Institute of Applied Informatics and Formal Description Methods (AIFB) at the University of Karlsruhe (TH), PROMATIS Software GmbH and the Research Center for Information Technologies Karlsruhe (FZI).

² The current version of the toolset can be freely downloaded from the project's website <http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/BIK/income2010/>.

ments for a software tool used to develop process-oriented information systems. In section 5 we describe the open plug-in architecture of INCOME2010. In section 6 some technical aspects in the implementation of INCOME2010 will be discussed. Section 7 presents the implemented functional features of the toolset. Section 8 concludes with a summary and an outlook on future work.

2. RELATED WORK

In recent years, a great number of Petri net tools have been developed³. Most of them are designed for high-level Petri nets for running on different operating systems. More sophisticated tools provide a graphical editor for modeling Petri nets and a token game animation which makes the system more user-friendly. But only a few provide an interchange file format, e.g. PNML, for further use or execution in other (commercial) tools.

Representative work related to INCOME2010 includes for example VIPtool, CPN-Tool, Woflan, PetriNetKernel and YAWL-Tool. None of them is supporting XML nets.

VIPtool is a tool for editing, simulating and validating systems modeled with P/T Petri nets. It implements the research done in a project named VIP (Verification of Information systems by evaluating partially-ordered Petri net runs) to establish a simulation and validation concept for Petri nets based on the generation and evaluation of partially ordered execution runs (processes) instead of totally ordered sequences of transition occurrences [2].

CPN-Tool is a high level Petri net tool which supports the basic Petri nets plus Colored Petri nets and timed Petri nets. It includes a simulator and a state space analysis tool [4].

The tool Woflan enables users to model business processes by workflow nets, and to check whether the designed model fulfills desired properties, such as soundness etc. [14].

PetriNetKernel is an infrastructure for developing Petri net tools. It relieves the programmer from implementing standard functions on Petri nets such as loading and saving nets, accessing and modifying the net structure, and building graphical user interfaces [8].

Based on YAWL (Yet Another Workflow Language), a Petri net dialect extended with additional features and constructs to facilitate the modeling of complex workflows [13], the YAWL-Tool provides a graphical editor and a SOA-based engine supporting both the control flow and data perspectives. It uses

XML-based standards like XML Schema, XPath and XQuery to define, transform, evaluate and exchange workflow data. It also supports workflow analysis (e.g., soundness test) at build time.

Modeling systems supporting graphical design of models are also related to INCOME2010 in the sense that Petri nets are also a graph-based modeling formalism. A representative of such modeling systems is Draw-Net [3], a Java-based open framework supporting the multi-formalism modeling of systems. It includes a formalism generator for defining and editing (multi-)formalism and a graphical Draw-Net tool for the creation of models using the defined formalism. Nevertheless, using the formalism definition languages provided by Draw-Net, it would be difficult to map all constructs of XML nets without semantic loss.

3. FOUNDATIONS

XML nets represent a variant of high-level Petri nets, in which each place is inscribed and typified by an *XML Schema*. Places are therefore interpreted as containers of XML documents that must conform to the place types defined in the respective schemas. Transitions can also be inscribed by predicate logical expressions, whose free variables must be contained in the inscriptions of adjacent edges, the so-called *Filter Schemas* used to read or manipulate XML data objects.

A transition in an XML net is enabled (i.e. it may occur) for a given marking and for a given instantiation of the variables in the transition's environment if the following three conditions hold: Each place in the pre-set of the transition contains at least one valid XML document that conforms to the Filter Schema inscribing the edge from the place to the transition. Additionally, places in the post-set of the transition have to be checked: If the transition modifies existing XML documents in post-set places, the existence and validity of these documents must be ensured. If the transition creates new XML documents in post-set places, then the places must not contain the documents to be created. Finally, the transition inscription must be evaluated to be true for the given instantiation of the variables, in order to enable the transition. If an enabled transition occurs, XML documents of pre-set places are read or (partially) deleted for the given instantiation of variables, and in the marking of post-set places, new XML documents are inserted or existing documents are modified.

Figure 1 shows a simplified XML net “pay bill” with XML Schema and Filter Schema diagrams assigned to places and edges, respectively. The XML element “isPaid” is of type Boo-

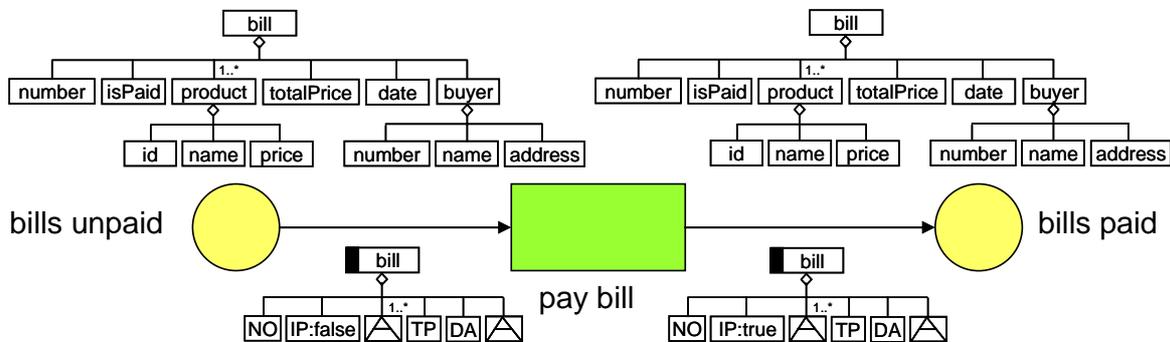


Figure 1: XML net "pay bill"

³ A comprehensive overview of Petri net tools is given at <http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/>.

lean and indicates whether the bill has been paid. If the transition “pay bill” occurs, XML documents containing a bill number, a total price, an issue date and an element “isPaid” with the value “false” are deleted from the place “bills unpaid” according to the Filter Schema assigned to the ingoing arc of the transition. According to the Filter Schema inscribing the outgoing arc of the transition and the XML Schema of the place “bills paid”, XML documents are created where the value of element “isPaid” is set to “true”. The black bars in the Filter Schema diagrams represent *manipulation filters* used to create or delete XML documents in this context. The rectangles with an inscribed “A” stand for *element placeholders* of the XML data type *AnyType* and can be instantiated by elements of any type.

4. REQUIREMENTS

Requirements for a software tool supporting the development of process-oriented information systems can be classified in *functional* and *non-functional* requirements. The functional requirements relate to behavior and internal functionalities of the software and may generally include following aspects:

- *Graphical modeling of process flows (control flows)*: A graphical editor should be provided to build and edit process flow models in executable form. Auxiliary graphical elements (shapes, images etc.), attributes (e.g., color, font, alignment) and related functions are also desired to increase the visuality of the models.
- *Modeling of data flows*: The tool should support the modeling of data flows and the (structural) description of relevant process objects and data objects.
- *Definition and evaluation of (business) rules*: Functionalities should be supplied for the definition and evaluation of logical rules ranging from global strategic business rules to process rules bound to specific usage points in process models.
- *Modeling of organizational structure*: It should be possible to model organizations involved in a business process and their relationships and to assign roles and related resources to organizational entities.
- *Definition and management of process metrics*: Process metrics like time, cost, and resource consumption are performance indicators used to monitor and optimize process-oriented information systems. The tool should allow defining, assigning, storing, removing, and editing process metrics.
- *Hierarchy modeling*: It should be allowed to build process and organization models hierarchically by refining process nodes or organization entities to provide views at different level of detail.
- *Reusable process fragments*: To reduce the time and cost for building new process models, the tool should allow users to reuse process fragments (e.g., sub-processes) defined previously or to select process prototypes from a process library.
- *Analysis of process models*: In order to improve process models and to reduce costs in handling execution faults, process models should be analyzed through simulative validation and, if possible, through formal verification prior to the execution of models.

- *Execution of process models*: A workflow engine should be provided to execute process models. Tasks of the workflow engine include for example instantiation of workflows, interpretation of business rules, creation of work lists, assignment of resources and invocation of (remote) functional units (e.g., applications, services).
- *Monitoring of process execution*: A monitor should be provided to record and report the execution of process models using process metrics. Mechanisms for exception handling (e.g., error signaling, interruption, rollback, compensation) should also be provided.
- *Dynamic process management*: Functionalities should be supplied to support ad-hoc process deviations at run-time by modifying process models and related objects, rules and metrics during the process execution without violating data consistency and other constraints.

More specifically, the incorporation of SOA concepts can also be regarded as a functional requirement for the system. By treating process activities or fragments as reusable and loosely coupled *Services*, the flexibility and interoperability of process models built with the tool would be enhanced.

Besides usability, availability, performance, and other non-functional requirements for measuring and judging run-time quality of the tool, some requirements for development-time quality like extensibility, scalability, composability, and reusability should also be taken into account. An open architecture is therefore desirable, especially for an open source software tool with features developed in parallel and cooperatively by different teams.

5. ARCHITECTURE

INCOME2010 is a rich client Java application based on Eclipse Rich Client Platform (RCP). Using the plug-in technology of Eclipse, it arranges its components in an open plug-in architecture in order to increase feature extensibility as stated in Section 4. Figure 2 depicts this architecture where the component dependencies are indicated by arrows.

The *Core Plug-in* serves as the foundation of other plug-ins and provides definition and implementation of basic GUI elements (e.g., global menus, toolbars) and related (retargetable) actions. It also defines system-wide configurations and the main structure of the integrated help system which can be extended and refined by other plug-ins.

The *Petri Net Plug-in* supplies necessary facilities for the graphical creation, editing and animated simulation of elementary Petri net models by implementing Petri net specific GUI elements (perspective, views, menus, toolbars, wizards, etc.) and related actions. As one of the fundamental components of the system, it provides basic functionalities, extension points and interfaces that can be extended and commonly used by plug-ins implementing higher Petri net variants. In order to support business processes, it also enables the definition, editing and assignment of roles, resources, documents, and process metrics like time and cost.

Based on the Core and the Petri Net Plug-in, a variety of functional modules can be added as plug-ins. For instance, the *Structural Analyzer Plug-in* is utilized to analyze the structural properties of Petri net models (e.g., free choice property). The *Monitoring Plug-in* allows keeping track of the execution of process models based on predefined metrics or indicators. The *Workflow Engine Plug-in* supplies a workflow engine used to

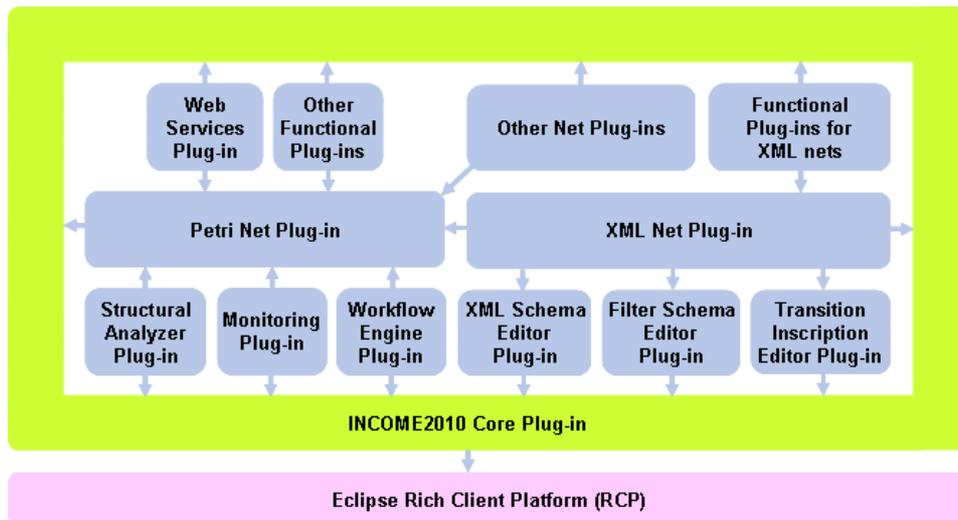


Figure 2: Architecture of INCOME2010

execute process models. The *Web Services Plug-in* uses SOA concepts and provides a BPEL⁴ generator to convert Petri net models (partially) in executable BPEL codes and a UDDI⁵ interface to search and access the available public Web Services that can be integrated as process activities. In addition, by taking advantage of this extensible architecture, other functional plug-ins for Petri nets can also be added to the system.

Functionalities for modeling and simulating XML nets are provided by the *XML Net Plug-in*, which depends on the Core Plug-in, the Petri Net Plug-in, and three Editor Plug-ins. The

latter three are used to create and edit XML Schemas, Filter Schemas and transition inscriptions, respectively. All functional plug-ins for Petri nets can also be extended to meet specific requirements of XML nets.

6. IMPLEMENTATION

INCOME2010 is a platform independent software toolset developed in Java. Its earlier version [1] was a standalone application based on SWT/JFace⁶ toolkits and the graph library JGraph⁷. Since version 0.2.0, INCOME2010 has migrated to

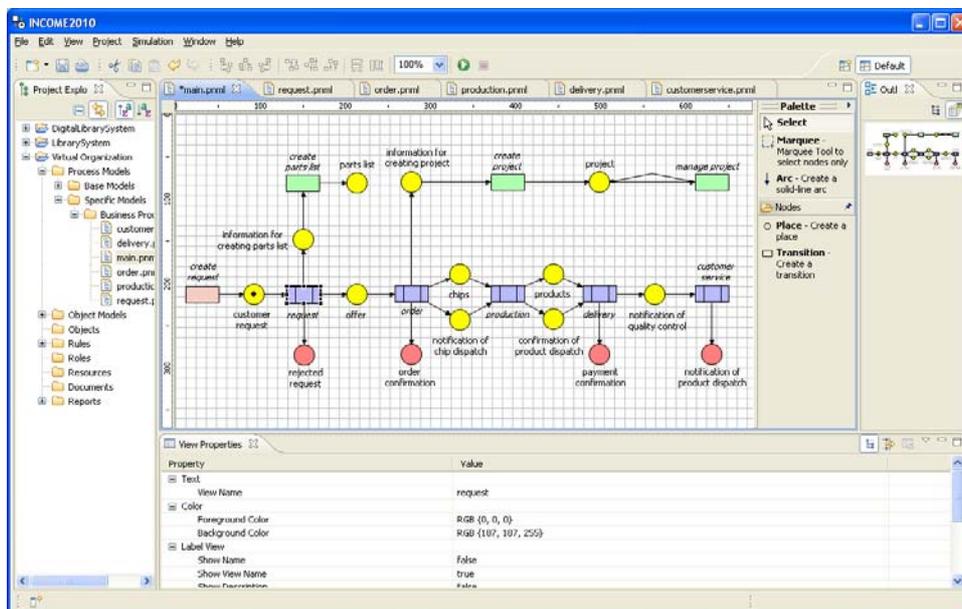


Figure 3: User interface of INCOME2010

⁴ BPEL (Business Process Execution Language) is an executable business process modeling language for the orchestration of Web Services. The specifications of BPEL can be found at <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>.

⁵ UDDI (Universal Description, Discovery and Integration) is a platform-independent, XML-based registry for publishing and discovering Web Services. For details see <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm>.

⁶ SWT (Standard Widgets Toolkit) and JFace are Java widget toolkits for the creation of graphical user interfaces (GUI) by accessing native GUI libraries of the underlying operating system. <http://www.eclipse.org/swt>.

⁷ JGraph is an open source Java Graph visualization and layout library. <http://www.jgraph.com/>.

Eclipse RCP in order to utilize basic functionalities provided by Eclipse and to increase extensibility of the system. It also uses Eclipse Graphical Editing Framework (GEF) for the construction of rich graphical editors for process modeling and analysis. Figure 3 shows the user interface of INCOME2010⁸.

Similar to JGraph and other graph libraries, GEF is also built on the architectural pattern *Model-View-Controller* (MVC) [12]. The model in GEF is any data that can be stored persistently. Changes made in the model are notified to the controller using *PropertyChangeListener* defined in JavaBeans. The controller is named *EditPart*. It contains helpers called *EditPolicies* which are responsible for handling editing tasks identified in user

requests or property change events. The view in GEF is a *Figure* or a *TreeItem* which can only be changed by the controller.

Within the implementation of the Petri Net Plug-in, generic interfaces for Petri net element models are defined so that facilities provided by this plug-in can easily be used and extended by plug-ins implementing other Petri net variants. Figure 4 shows the UML class diagram for these interfaces with generic operations that characterize them. The root interface *IPetriNetElement* represents the most generic interface from which all other model interfaces are derived. It ensures that all Petri net elements can be identified by IDs. The interface *IPetriNetSubpart* extends *IPetriNetElement* with bounds information (e.g.,

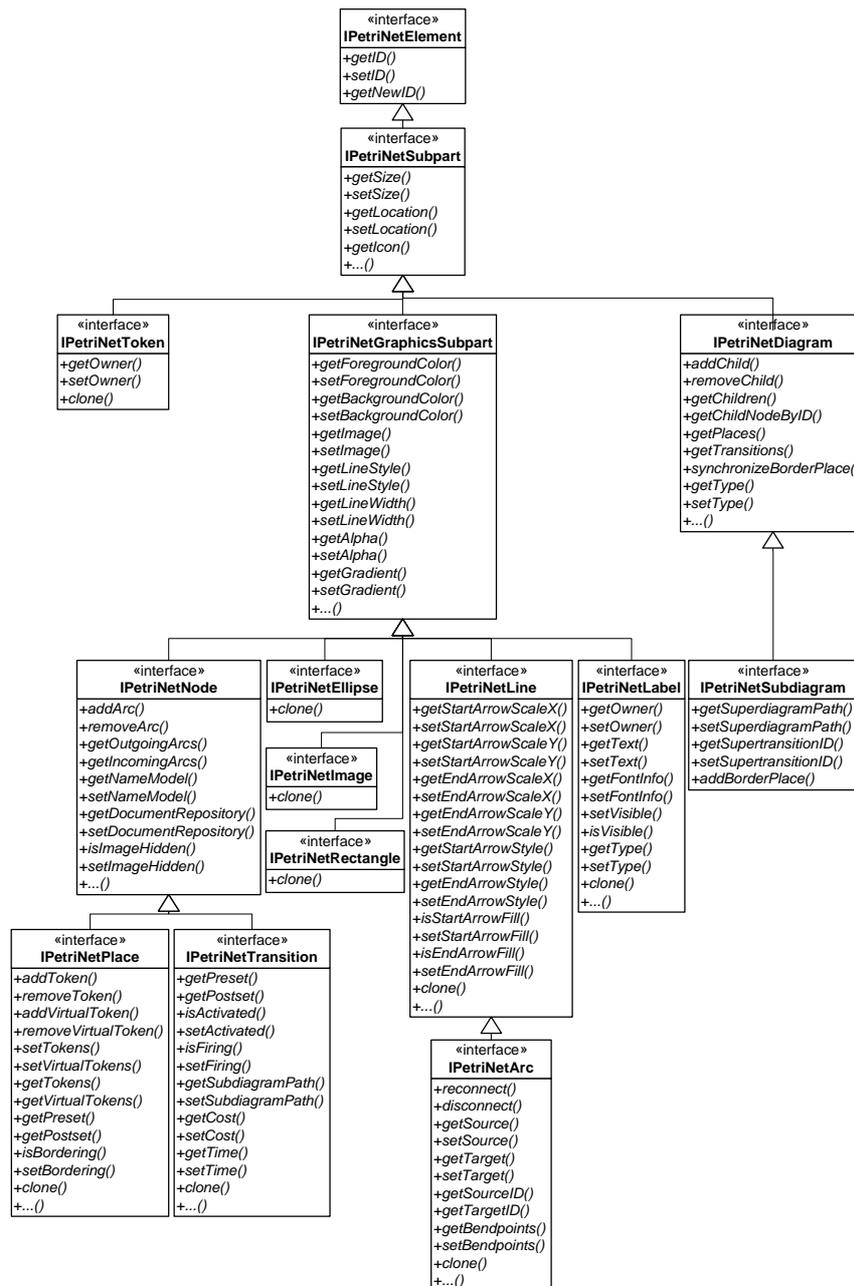


Figure 4: Interfaces for Petri net element models

⁸ For more screenshots see <http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/BIK/income2010/screenshots.htm>.

size, location) for resizable and/or movable net elements. Within the interface *IPetriNetGraphicsSubpart* which specializes *IPetriNetSubpart*, common graphical properties (e.g., color, line style, transparency, gradient) of Petri net node, label, image and geometric shape (e.g., ellipse, rectangle, line) are aggregated. The interface *IPetriNetSubdiagram* extends *IPetriNet-Diagram* with super-diagram path, super-transition ID and the operation for adding border places to the sub-diagram while ensuring the consistency between border places and the corresponding places in the super-diagram. Common properties of Petri net place and transition like references to arcs, labels and other nodes are incorporated in the operations of *IPetriNet-Node*. All element model interfaces together with the related *EditParts*, *EditPolicies* and *Figures* have been implemented in the Petri Net Plug-in and extended by the XML Net Plug-in.

INCOME2010 is implementing element libraries used to manage reusable net elements and the references generated in the creation of sub-diagrams. The current version provides implicit element libraries for each project to add, remove, update, and search references between border places and the corresponding referenced places. The element libraries stored as XML files are net specific, which means that several libraries can coexist in a project if it contains process models based on different Petri net variants. The Petri Net Plug-in offers an interface *IElement-Library* that can be implemented by other net plug-ins to build their own specific element libraries.

By refining transitions with subnets, process models can be built hierarchically. As a result, simulation of process models should be allowed to run at any level of the hierarchy. Once started on a process diagram, the simulator must take all related sub-diagrams into consideration. The simulation process of each diagram must be coordinated by a central functional unit. To achieve this for the token game simulation provided by the Petri Net Plug-in, a token game simulator is divided into a *Token Game Coordinator* and several *Token Game Controllers*. The Coordinator is responsible for registering Controllers, starting and stopping token game, controlling virtual token flow, creating simulation report, and delegating checking tasks and user requests to the registered Controllers. A Controller is instantiated and assigned to each diagram involved in the simulation to perform net checking tasks and to restore token and transition status after the token game is stopped. Because one sub-diagram can be used in several model hierarchies, it is important that no more than one Coordinator can be instantiated at the same time and thus no more than one token game instance can run simultaneously to avoid deadlocks and errors in the simulation. This simulation mechanism also provides interfaces that can be extended by other net plug-ins.

Within the implementation of the XML Net Plug-in, XQuery is used to parse process rules expressed by Filter Schemas and transition inscriptions and to manipulate XML data objects. XQuery statements are executed using the XSLT and XQuery processor Saxon⁹. To store and exchange XML net models, an XML-based file format *XNML* has been defined, which extends the standard Petri net interchange format PNML [6] with XML Schema elements for XML net specific information like transition inscription and Schema path.

To speed up the development of INCOME2010, it is necessary to integrate existing Petri net tools (e.g., Woflan) in the system. The difficulty of the integration may vary depending on functionalities, interfacing, data exchange format, system require-

ments and other natures of the tool. For instance, it would be easier to integrate a tool supporting PNML standard. Otherwise an interface or an adapter for data transformation has to be provided. EAI (Enterprise Application Integration) technologies (e.g., bus/hub, connector, middleware) and Web Services may also be used to integrate more heterogeneous tools.

On the other hand, to ensure the reusability of functional plug-ins of the system, in addition to the adherence to PNML, interfaces should be provided to enable direct procedure call (DPC) with simple parameter passing.

7. FUNCTIONAL FEATURES

INCOME2010 is in early stage of development. The current version implements the Core Plug-in, the Petri Net Plug-in, and the XML Net Plug-in and includes the following main functional features:

- **Graphical modeling:** The creation and editing of Petri net and XML net models are supported with net specific graphical editors enabling simultaneous work on several net diagrams. Net elements can be drag-and-dropped to the diagram easily.
- **Hierarchy modeling:** Net models can be built hierarchically by creating reusable sub-processes that refine transitions and can be referenced and embedded in more than one net diagram. Places in the pre- and post-set of a refined transition are referenced by border places in the sub-processes. In order to guarantee the semantic consistency between places in super- and sub-processes, some properties (e.g., name, initial marking) of border places can only be modified by making changes on the top level of the hierarchy. The system then propagates the changes downwards to all related levels. To reduce complexity, other forms of hierarchy modeling (e.g., place refinement) are currently not supported.
- **Animated token game simulation with report:** Net models can be simulated by playing interactive token games in which the movement of tokens is animated. After stopping a token game, a simulation report recording transition activation and token movements is created. A token game can be started at any level of a hierarchy.
- **PNML-compliant file format:** Net models are stored on the local file system in tool-independent file formats compliant to PNML, which enables exchanging net models with other Petri net tools. For XML net models a PNML-based file format *XNML* with XML net specific extensions is used.
- **Structured standard information of net nodes:** Net nodes (places and transitions) are described and specified by structured standard information or metrics like name, view name, description, notes, cost, time, etc. This information is used to identify its owner or to analyze the net behavior in simulation.
- **Role and resource:** Roles and resources are fundamental elements for resource planning and management which can be part of business process management. INCOME2010 allows the definition of roles and resources that can be allocated and assigned to transitions.
- **Graphical editing support:** Grid, ruler, guide and zoom functions are provided to facilitate the positioning of diagram elements, which can also be resized and aligned ac-

⁹ <http://saxon.sourceforge.net/>.

ording to the size and position of other elements. In addition, net nodes can be displayed using images. It is also possible to draw geometric shapes and to add image and text at an arbitrary location.

- **Graphical export of diagram:** Model diagrams can be easily exported as images in BMP, JPEG, GIF and PNG formats.
- **Categorized project structure:** INCOME2010 is project-oriented, which means all resources created by the user are managed in projects with predefined categorized structure. INCOME2010 provides a project explorer that gives a hierarchical view of the resources and categories of projects in the workspace.
- **Import and export of project:** INCOME2010 provides wizards and functionalities to import projects from an existing workspace or an archive file and to export projects to an archive file.
- **Project handbook:** Project handbooks used to document projects and related resources can be generated with configuration possibilities, i.e. users can predefine the style and content of a project handbook by activating some standard reporting options. Project handbooks are stored in XML format and can be exported as HTML, PDF or Microsoft Word documents using XSLT stylesheet.
- **Integrated help system:** Documents guiding users and developers of INCOME2010 are arranged in an integrated Eclipse help system based on an XML table of contents referencing HTML files. The contents of the help documents are searchable. Help topics related to a user interface widget selected by the user can be listed dynamically by searching all available topics on the query formed from current context.

8. SUMMARY AND OUTLOOK

In this paper we surveyed INCOME2010, an open source software toolset facilitating the development of process-oriented information systems using variants of high-level Petri nets (e.g., XML nets) and SOA concepts. Although in early stage of development, INCOME2010 can already be used for modeling and simulating (business) processes in an effective and user friendly way. It offers an integrated help system and comprehensive functionalities supporting graphical modeling, hierarchy modeling, animated token game simulation, PNML-conform exchange format, project-oriented resource management, and the definition and assignment of process metrics, documents and other process-relevant information.

Next steps in the development of INCOME2010 include the implementation of functional plug-ins for Petri nets and XML nets, especially plug-ins for editing XML Schemas, Filter Schemas, and transition inscriptions. In addition, the Petri Net Plug-in will also be expanded, e.g., with the following features and functionalities:

- **Automated simulation with detailed report:** A module for automated simulation will be added to the current system. Time, cost, resource consuming, and other process metrics will be considered in the simulation. The simulation result will be reported in detail and in different text formats.
- **Explicit element library:** The current internal element library will be expanded to enable users to add, remove or edit reusable net elements or fragments conveniently.

- **Organization model:** A graphical editor will be implemented for creating and editing organization models that describe organization entities and their relationships. The editor also enables users to assign roles and resources to organization units. Similar to process models, an organization model can also be built hierarchically by refining organization entities with sub-organigrams.

Taking advantage of the extensible open architecture, we also plan to enrich INCOME2010 with other Petri net variants (e.g., Front-end nets [7]) and tools (e.g., tool for semantic annotation of Petri nets [9]) that are currently under development.

In cooperation with PROMATIS Software GmbH and the Research Center for Information Technologies Karlsruhe (FZI), INCOME2010 will be applied and evaluated in different (industrial) projects.

Acknowledgement:

We wish to thank the anonymous referees for many valuable comments on an earlier version of this paper.

9. REFERENCES

- [1] Betz, S., Karle, T., Klink, S., Koschmider, A., Li, Y., Mevius, M., Oberweis, A., Ried, D., Trunko, R., and Zaich, M. 2006. Ein Framework zur Modellierung und Analyse von XML-Netzen. In Moldt, D., Tagungsband 13. Workshop Algorithmen und Werkzeuge für Petri-Netze (AWPN'06), 18-24. Universität Hamburg (in German).
- [2] Desel, J., Juhas, G., Lorenz, R., and Neumair, C. 2003. Modelling and Validation with VipTool. LNCS 2678, 380-389.
- [3] Gribaudo, M., Mazzocca, N., Moscato, F., and Vittorini, V. 2005. Multisolution of Complex Performability Models in the OsMoSys/DrawNet Framework. In Proc. 2nd Int. Conf. on the Quantitative Evaluation of Systems, Torino, Italy.
- [4] Jensen, K., Kristensen, L.M., and Wells, L. 2007. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. In International Journal on Software Tools for Technology Transfer (STTT).
- [5] Jensen, K. and Rozenberg, G. (eds.) 1991. High-Level Petri Nets. Theory and Application. Springer-Verlag.
- [6] Jünger, M., Kindler, E., and Weber, M. 2000. Towards a Generic Interchange Format for Petri Nets. In: Bastide, R., Billington, J., Kindler, E., Kordon, F. and Mortensen, K. H. (eds.): Meeting on XML/SGML based Interchange Formats for Petri Nets. 1-5, Århus, Denmark, 21st ICATPN.
- [7] Karle, T. and Oberweis, A. 2006. Unterstützung von Kollaboration im Rahmen der Softwareentwicklung auf Basis service-orientierter Architekturen. In: EMISA 2006 - Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen, Volume P-95 of Lecture Notes in Informatics, 77-90. GI (in German).
- [8] Kindler, E. and Oschmann, F. 1998. The Petri Net Kernel: An INA-Pilot. In Desel, J., Kemper, P., Kindler, E., Oberweis, A.: Workshop Algorithmen und Werkzeuge für Petri-Netze.
- [9] Koschmider, A. and Oberweis, A. 2005. Ontology based Business Process Description. In Castro, J., Teniente, E.

Proceedings of the CAiSE'05 Workshops, No. 2, 321-333. Porto/Portugal.

- [10] Lenz, K. and Oberweis, A. 2003. Inter-Organizational Business Process Management with XML Nets. In: Ehrig, H., Reisig, W., Rozenberg, G., Weber, H. (eds.): Petri Net Technology for Communication-Based Systems, Advances in Petri Nets volume 2472 of LNCS, 243-263. Springer-Verlag.
- [11] Oberweis, A., Scherrer, G., and Stucky, W. 1994. IN-COME/ STAR: Methodology and Tools for the Development of Distributed Information Systems. In: Information Systems, Vol. 19, No. 8, 643-660.
- [12] Reenskag, T. M. H. 2003. The Model-View-Controller (MVC), Its Past and Present. JavaZone 2003, Oslo, Norway.
- [13] Van der Aalst, W.M.P. and ter Hofstede, A.H.M. 2005. YAWL: Yet Another Workflow Language. Information Systems, 30(4):245-275.
- [14] Van der Aalst, W.M.P. and van Hee, K. 2002. Workflow Management, Models, Methods and Systems. The MIT Press, Cambridge, Massachusetts.
- [15] Van der Aalst, W.M.P. 1998. The Application of Petri Nets to Workflow Management. In: The Journal of Circuits, Systems and Computers, 8(1), 21-66.