An overview to S-BPM oriented Tool Suites

Albert Fleischmann
Interaktiv Unternehmensberatung
Burgriedenstrasse 16, 85276 Pfaffenhofen a.d. Ilm, Germany
albert.fleischmann@interaktiv.expert

Stephan Borgert
Urban Software Institute GmbH
Zwickauer Str. 223a, D-09116 Chemnitz, Germany
stephan.borgert@the-urban-institute.de

Matthes Elstermann
Karlsruhe Institute of Technology (KIT), Institute for Information Management in Engineering
Bldg. 20.20, KIT Campus South, 76131 Karlsruhe, Germany
matthes.elstermann@kit.edu

Florian Krenn
Johannes Kepler Universität
Altenberger Straße 69, 4040 Linz, Austria
florian.krenn@jku.at

Robert Singer
FH Joaneum- University of Applied Sience
Alte Post Straße 147, 8020 Graz, Austria
robert.singer@fh-joanneum.at

ABSTRACT

This paper gives an overview about available S-BPM tools. These seven tools cover either modeling or execution aspects.

Keywords

Business Process Management Tool Suite; S-BPM; Process Modeling; Execution of processes

1. INTRODUCTION

Several tool sets have been developed for defining and executing business processes described in a subject oriented way. In this paper these tools are outlined together with their development status and their strengths and weaknesses. A Business Process Management suite should cover following major aspects [3], [4]:

- Modelling aspects
  - Describing business processes and creating process models (modeling behaviour)
  - Definition of the business objects used in a process (modeling data)
  - Design of related user interfaces (designing user interfaces)
- Execution aspects
  - Embedding process models in their organizational and technical environment (embedding environment)
  - Executing process models in their organizational and technical environment (execution environment)

The modeling aspects are managed with a repository. Whereas the execution environment is handled by administration and monitoring tools. Figure 1 shows an architecture adapted to the specialties of S-BPM.

Some subject oriented business process management suites do not cover all of these aspects in an integrated monolithic way and these various aspects are not always sharp separated in corresponding tool blocks. In the following sections we want to outline which aspect is supported by which tools. Some of the suites do not support all aspects defined above. The expressivness of all these tool suites is based on the subject oriented kernel described in [1]. An overview can be found in [2]. Some tool suites have extended this basic concepts for various reasons. These specialties are outlined in the coresponding sections.

S-BPM One March 30-31, 2017, Darmstadt, Germany

2. OUTLINE OF S-BPM SUITES

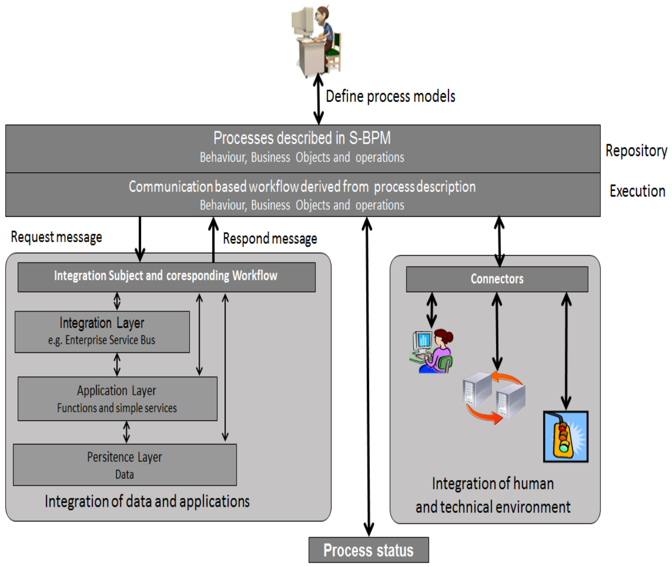
Define process models
Processes described in S-BPM
Behaviour, Business Objects and operations
Repository
Communication based workflow derived from process description
Behaviour, Business Objects and operations
Execution
Request message
Respond message
Integration Subject and coresponding Workflow
Integration Layer
e.g. Enterprise Service Bus
Application Layer
Functions and simple services
Persitence Layer
Data
Integration of data and applications
Connectors
Integration of human
and technical environment
Process status

Figure 1: Workflow Architecture

2.1 KIT S-BPM Modeler

The KIT-S-BPM suite is a pure modelling tool based on Microsoft Visio and supports modeling of subject interaction and behavior. It comes in form of a mobile set of stencils for Microsoft Visio and therefore is highly distributable. It is somewhat compatible with the Visio-based modelling tool of actnconnect.

- User Interface for modeling
  In general, the user interface for modeling business processes is based on Visio which in turn is part of the Microsoft Office product family. So the tool makes full use of standard and proven interface and editing concepts of MS Office including concepts like grouping, group formatting, spell checking, or shared editing (marginal).

  Editing is done in general by dragging and dropping shapes from a stencil set onto a modelling page or by using build in auto-complete functions. Interaction and Behavior models are programmatically linked (e.g. renaming messages and actors will have an impact on underlying behavior diagrams)

  It is possible to link several Visio files via external subjects to allow modelling of sub- or neighboring systems in different files. The modeling tool supports following language dialects/elements:

  - Standard PASS (subjects, messages, send+ receive+function states and transitions)
  - Standard PASS specialties (Multi-subjects, Interface subjects)
  - E-PASS (Checkboxes, Guard-Receive (simple), Time-Trigger/Time-out, User-Cancel)
  - Abstract Layered PASS (ALPs) (multiple Layers, Subject and State inheritance and implementations)

- Storage format
  The models are stored in standard Visio 2013-2016 (.vsdx) format

- Exporters/generators exist for the ActnConnect execution environment (business-actor .xml files) and for the proposed RDF/OWL exchange format.

- Modelling Data

  - No native support for data modelling
  - The KIT S-BPM modeler is compatible with the ActNConnect Visio tool and theoretical can use the additional data modelling stencil from there. The ActNConnect Shapes allow to model complex hierarchical data structures and their mapping to subjects/actors and messages.

- Designing User Interfaces
  The design of user interfaces is not supported.

- Process Model Repository
  No specialized repository technology is implemented yet. Process descriptions are stored either as standard Visio-Files or an ActnConnect XML File. Future plans based on the RDF/OWL Data model concept envision exchange of process descriptions/models identifiable via unique URI-identifies and semantic web technologies.

- Embedding Tools and Execution Environment
  The KIT S-BPM suite does not offer an execution environment itself yet. It can export XML files compatible with ActnConnect Suite for execution purposes.

- Experiences
  This tool suite has been used for describing several complex process systems in various settings (see [8])

- Availability, status and further plans
  The shapes are freely available for non-commercial purposes and can be downloaded via www.i2pm.net or directly from
  http://subjective-me.jimdo.com/downloads/

- Future development plans include an importer for RDF/OWL exchange format and a simulation environment within Virtual Realties that will be incorporated into the opensource

  - VR authoring software Poly-VR also under development at the KIT. The simulation-environment will be able to executed multi-layered process models based on the arbitrator-pattern concept presented by Elstermann, Seese, and Fleischmann in [7].
  - Next to the simple stencil files, a fully-integrated plug-in for Visio is in development. The plugin will enable full editing functionality for E-PASS and ALPS concepts, including the extension and inheritance of multi-layered subject-oriented process descriptions. [6], [5].
  - On a larger horizon, development plans also envision the incorporation of full object-oriented datamodelling (including inheritance, polymorphism,

abstract and passive-object-oriented process modelling). Also the before discussed repository technology within the semantic-web technology framework false in this planning category.

2.2 InFlow S-BPM Suite

InFlow is a Multi-enterprise Business Process Platform based on S-BPM. It is built on functionalities of the Microsoft .Net Framework and has been discussion in [9],[10],[11], [12].

- Modeling Behavior
  Modeling is done via a browser window; that means modeling can be done on all operating systems and mobile devices (based on Bootstrap). Model design is based on creating a Subject Interaction Diagram (SID) and a Subject Behavior Diagram (SBD) for each subject. InFlow supports the following modeling artifacts: subject, message, receive-, function-, send-states, timeouts, and refinements. States can be defined as start and end states. Furthermore, subjects can be defined as multi- and ex-ternal subject. Behind the scenes is the modeling functionality based on Javascript and several open source libraries.

- Storage Format
  Process descriptions are stored as XAML files, a declarative XML-based language developed by Microsoft that is used for initialising structured values and objects. XAML elements map directly to Common Language Runtime object instances. Modeling Data and Business Objects Data models can be created based on primitives and business objects (complex data). Business objects are based on the JSON format, which is also used internally: that means, the message exchange is implemented as an exchange of JSON data structures.

- Designing User Interfaces
  User interaction is based on forms, which are automatically generated; nevertheless, business objects based on JSON data can also be supported by a hand crafted HTML in the integrated business object designer, including default data for initialization. All „states "are derived from standard .NET classes; that means any enhanced or specific behavior can be implemented via Microsoft Visual Studio. A process can even be embedded into an application, as has been demonstrated in one the keynotes of the 2015 S-BPM ONE by Stefan Raßand Peter Resnik (Case „Riegler and Partner").

- Process Model Repository
  A process repository is integrated and based on an SQL database. It supports, for example, model versioning and rights management (who is allowed to change the model).

- Embedding Tools and Execution Environment
  The architecture for the suite (modeling and execution) is based on Microsoft .NET Frame-work 4.5, IIS, Service Bus, Workflow Manager, and SQL Server (see references).

- Experiences
  The book „S-BPM Illustrated "(based on Metasonic Suite 4.x) has been rewritten for InFlow and is available via
  https://robert62.gitbooks.io/structured-communication /content/.
  It demonstrates that the examples from the original book can be easily executed with InFlow without restrictions. Integration with an ERP-system is even much more convenient to implement, as demonstrated in the book.

- Availability, status and further plans
  The source code is available via
  https://github.com/InFlowBPM/InFlow-BPMS. The code is commented, but there is no installation manual yet. The suite is designed to be executed on Microsoft Azure, but can also be installed on a stand-alone server; in that case, a couple of server licenses are needed. Furthermore, commercial and supported use can be ordered from IMA Solutions GmbH (http://www.strict-solutions.com). The architecture uses heavily functionality of the Microsoft Workflow Manager 1.0. In the moment of writing, there is no solid information about further plans for this product from Microsoft and the code can not be compiled with the latest version of Microsoft Visual Studio anymore. Until a statement from Microsoft about further plans about Workflow Manager, no new features will be implemented.

2.3 UeberFlow Suite

The UeberFlow Suite combines a process modeling editor with a workflow execution environment. Ueber Flow is developed using the Actor Model by Hewitt as the foundational computation model.

- Modelling Behaviour

  The specification of processes and workflows in UeberFlow can either be done text-based or using a web-based editor (currently in development, alpha state). For the purpose of process specification a meta model has been developed. Workflow specifications in UeberFlow Lang incorporate these workflow elements using three basic elements. UeberFlow Lang Workflow specifications, as illustrated in Figure 2, comprise of WorkflowUnits, WorkflowSteps and WorkflowFunctions. Each component defined in the meta model can be understood as a parallel executable unit allowing for parallel instantiation execution of tasks, roles, and models. An entire executable model is represented by a WorkflowSpecification. It acts as container for the WorkflowUnits and stores the relevant meta-data of the workflow model (e.g., version or creation date).

  WorkflowUnits group tasks according to the responsible process role. For each role participating in the specified workflow a WorkflowUnit is created which contains and supervises all WorkflowSteps the corresponding role is responsible for. Additionally, WorkflowUnit functions as a data space for the underlying WorkflowSteps and WorkflowFunctions. In the course of execution all data accessible by the associated role are made available to the WorkflowSteps via the WorkflowUnit. WorkflowSteps. WorkflowSteps represent the activities

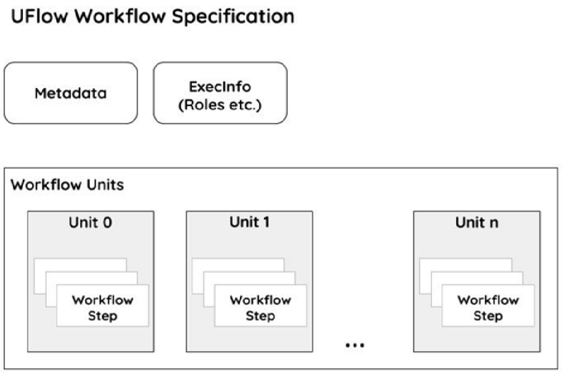

Figure 2: Workflow-Metamodel

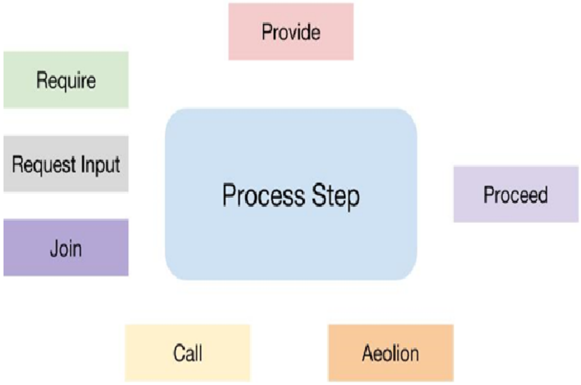

Figure 3: WorkflowStep and Workflow-Function overview.

a workflow comprises. The actual execution logic of a WorkflowStep, its prerequisites and results, are solely defined by its WorkflowFunctions. Each WorkflowStep contains a sequence of WorkflowFunctions which are executed sequentially, when the corresponding step is triggered. WorkflowFunctions can be assigned to a WorkflowStep without any limitations concerning order or quantity. The execution of a step is complete, once all its WorkflowFunctions have been executed successfully. WorkflowFunctions are the most fine-grained units of execution in the meta-model. WorkflowFunctions are the elements in the specification were the actual task is encoded. Each WorkflowFunction represents an atomic action of workflow execution. In order to define the workflow execution logic on a very fine-grained level, for each WorkflowFunction an optional condition can be specified which limits the set of situations the WorkflowFunction is triggered based on current instance data. Since WorkflowFunctions encapsulates all actions of a workflow specification, different types of WorkflowFunctions are needed for runtime purposes. In the herein presented basic version of UeberFlow Lang six Workflow-Function-types are defined (see 3.

- RequireFunction
  The RequireFunction allows specifying a set of values required for the execution of subsequent functions defined in the WorkflowStep. These values can either represent an event triggered during the workflow execution or a set of data. The execution of the process step stops until the required values are available for the process unit. The RequireFunction has an optional convert expression, which allows modifying the data before it is made available in the context of the WorkflowStep. Since the RequireFunction completely abstracts from the source of the data it is agnostic to whether the incoming (or already available) data was provided via a message or by the previous step.
- ProvideFunction
  Upon execution, the ProvideFunction sends a set of values to any WorkflowUnit defined in the workflow. Analogous to the RequireFunction, these provided values can either be a set of data (e.g., completed order form) or an event. Thus, the ProvideFunction can be used in combination with the RequireFunction. It implements asynchronous messaging/data exchange between WorkflowUnits. In this way, data become available for subsequent steps in the same unit.
- ProceedFunction
  The ProceedFunction triggers the execution of another WorkflowStep. The execution of the current step has not to be complete, in order to trigger the next step, i.e., other functions can be executed after the ProceedFunction has been executed. AND-, OR- and, XOR-Gateways can be implemented by specifying multiple sequential ProceedFunctions and corresponding conditions within one WorkflowStep. Besides simply triggering the execution of a subsequent step, the ProceedFunction offers an alternative to the ProvideFunction. It is also possible to directly pass data to the triggered step. For example, the result of a calculation performed by step A can be passed on the subsequent step B without adding it to the data context of the WorkflowUnit.
- JoinFunction
  JoinFunctions enable synchronizing two or more parallel execution flows by halting execution of the containing ProcessStep until all of the defined previous steps have been executed. It is also possible to define a subset of steps required in order to realize a partial join according to the workflow patterns as described by [13]. The JoinFunction does not distinguish between the synchronization of paths within a single WorkflowUnit or synchronizing parallel paths of different WorkflowUnits.
- RequestInputFunction
  The RequestInputFunction is used to define required user input. Based on a specified input form, the current user (or users) associated with the role of the WorkflowUnit is requested to provide input. The execution of the WorkflowStep is halted until the required input is provided.
- Call Function
  The CallFunction allows extending the workflow capabilities by using external services. Such an extension can be achieved by defining a code snippet, which is interpreted at run-time, once the

WorkflowFunction is executed.

- Storage format
  Processes and workflow modelled in Ueber Flow are stored in a document-oriented database. Models can be exported and imported using a JSON-like representation.

- Modelling Business Objects
  The UeberFlow Business Object supports complex types which can be arbitrarily nested. A graphical editor for creating business objects is not yet available and has to be done text-based. The data flow in Ueber Flow is modelled using the Provide/Require-Functions

- Designing User Interfaces
  User interfaces are generated based on the defined form in the RequestInput-Function. The form definition allows to select the required input type (e.g., text, checkbox, radio button etc.).

- Model Repository
  Models are stored in a document-oriented database allowing importing and exporting models. Additionally, a versioning of the process model is implemented. This versioning support does not only allow to track changes of the process model but can furthermore be used to allow instance-based deviations from the process model at run-time.

- Embedding Tools
  The UeberFlow Suite has been integrated with the Aeolion Platform currently in development at the Department of Business Information Systems - Communication Engineering at JKU. Aeolion is a flexible middleware allowing the combination of tools provided as „Aeolion plugins "and other (external) web services on a fine-grained level. Currently, using this architecture Ueber Flow has been integrated with the learning platform Ueber Learn in order to enrich the platform with executable learning processes.

- Execution Environment
  The execution environment of UeberFlow supports to execute the models in the process model repository. A validation mode, were a single user can „step through "the model in order to validate the sequence and communication flow, as well as a „production mode "for executing a workflow with different users is available.

- Availability, status and further plans
  The UEBER Flow Suite is under active development, open-source and available at http://www.i2pm.net/open-tools.

2.4 TUD-S-BPM Groupware

The basic idea of the S-BPM Groupware was to merge important tools of a usual groupware in a way that makes it easy to model or generate S-BPM processes. The considered tools are the eMail-app, the task management app and the calendar app of a groupware. eMail-Apps support „Send "and „Receive "actions, task management apps „Internal actions "and calendar apps „Timeouts "and these features are not well connected. Often we receive tasks in a mail or define time blocks in the calendar for doing certain task. Thus we have to look at all three tools to get an overview of all tasks. Furthermore we can not connect „Send ", „Receive "and „Internal actions "with each other to describe a S-BPM process model which can be executed later on. The S-BPM Groupware was a first try to overcome these drawbacks. Although no sending of mails is supported, the solution supports modeling end execution of S-BPM process models. The execution of cross company business process models is also supported.

- Modeling of Behaviour
  - User Interface
    The user interface is a web-based app providing an automatic lay-outing of both types of S-BPM diagrams, the subject interaction and the internal behavior view. The lay-outing is semi automatic. New nodes are placed automatically when they are added but the model designer can replace them manually and persistent. The new nodes will not be removed when other nodes are replaced. Next to the view for designing processes a view for the execution of the models has been developed. The current tasks of all running process instances are displayed as far as they become active.
  - Storage format
    The process models are stored in a non SQL data base on the backend. Alternatively they can be exported and imported to a self defined JSON format.
  - Modelling Business Objects
    The number of business objects is restricted to one so far and the only supported data type is String.

- Modelling Data
  Data can not be modeled

- Designing User Interfaces is not supported.

- Process Model Repositories
  The S-BPM Groupware provides functions for generating processes interface descriptions and storing them on an internal or external model repository. Other users can find and import them into their process models in order to finish a cross company process.

- Embedding Tools
  No support.

- Execution Environment
  On the backend a process engine is running. It is implemented in Scala and Akka and supports distributed execution of S-BPM processes.

- Availability, status and further plans
  A link to a git repository and further information of the S-BPM groupware can be found on https://sbpm-groupware.atlassian.net. The development stopped in November 2015 and at present no further developments are planed.

2.5 SAReI

The S-BPM ASM Reference Implementation (= SAReI) is based on the CoreASM engine. Abstract State Machines are an appropriate means to specify complex systems in a formal manner and a first ASM specification of an S-BPM interpreter was developed and published in [1]. CoreASM is a tool to execute ASM specifications. Thus these specifications can be validated and every execution is absolutely compatible to this specification what is an ideal prerequisite for a reference implementation of official or internal standards. SAReI is a combination of a modeling and an execution tool.

- Modelling Behaviour
  - User Interface
    The yED editor [yworks.com/yed] is an all purpose graph editor. Its strength are the integrated lay-outing algorithms. We use yED to describe the S-BPM process models. As shapes BPMN shapes are used.
  - Storage format
    The process models are stored as .graphml files what is an XML file.
  - Modelling Business Objects
    Every message type can contain an abbreviator number of parameters which are the business objects
- Modelling Data
  While it would be possible to model any kind of data models in the editor, the execution unit is only supporting strings.
- Designing User Interfaces
  Not possible and not planned.
- Process Model Repository
  Not available
- Embedding Tools
  CoreASM has a plug-in interface so other tools can be embedded if they offer an API
- Execution Environment
  As already stated the execution environment is the CoreASM engine.
- Availability, status and further plans
  SAReI is under construction and not yet available. It will be provided as open source after all work on the current ASM specification is finished. On top of the execution unit a verification unit will be developed in order to be able to verify process models on interaction soundness automatically.

2.6 Actorsphere Suite

Actorsphere is an execution environment for subject oriented business processes. The subjects in the actorsphere called actors are not coupled tightly. The communication partners are identified by the behavioral interface during the deployment phase (see [15]).

- Modelling Behaviour
  The ActorSphere modelling environment is based on the „KIT S-BPM modeler "(see subsection „KIT S-BPM modeler "). This special Actorsphere modeler allows to define Subject Interaction Diagrams (SID) and Subject Behavior Diagrams (SBD). If complex process systems have to be defined a model can be distributed over several Visio Files. Because the Actorsphere execution environment does not support all modeling aspects of the KIT S-BPM Modeler the expressivness of the ActorSphere modeler has the coresponding restrictions.
- Modelling Business Objects
  The ActorSphere Modeler supports the definition of complex business objects. Hierarchical data structures can be defined graphically. The business objects are assigned to subjects in which they are used. These data elements assigned to a subject can be embedded into the SBD states. In each internal state it can be specified which data elements are used in the coresponding internal function. For each message it can be specified which data items are transported by a it.
- Storage format
  The process models defined with the ActorSphere Modeler are stored in the standard Visio Format. But there is an exporter which allows to create a XML-file which can be understood by the Actorsphere execution environment. There is a separate file for each subject. These XML files for each subject can be stored and managed in standard repositories like GIT.
- User Interface
  The XML file format used by the execution environment has a special section called „interface ". In this section the user interface for a subject can be defined. This allows to create masks for the input or output of values of business objects. For defining the layout of these masks special XML data structures are defined. If there is no interface section in the XML file a standard layout is applied.
- Process Model Repository
  As already mentioned above there is no special repository available. The XML files for each subject can be stored and managed in standard repositories like GIT or CVS.
- Embedding Tools
  In the actorsphere two categories of subjects have been introduced. Business actors cover the interaction with the users. The details of the user interfaces are described in the interface section (see above). Service actors do not have human interfaces. Service actors wrap all automatic activities based on existing applications. Functions of existing applications are triggered by messages sent to the service subject wrapping the existing application.
  The embedding of subjects into an organisation is covered by the execution environment. After uploading a subject into the actorsphere (now a subject becomes an actor or agent) a user or a group of users is assigned to the actor.
- Execution Environment
  The execution environment called actorsphere is totally cloud oriented. The execution environment runs in a

cloud and each subject can be uploaded separatly. The execution environment uses the concept of behavioral interface (see [1]) in order to find the right partner actors. This means during the upload phase the execution environment checks which actors uses the same messages with the same parameters and the send and receive sequences fit to each other. This concept allows to replace actors individually and supports business agility. Actors are connected with each other during upload time. This allows also that actors can be created by different parties and can be combined for more complex services.

- Experiences
  The Actorsphere has been used in designing and implementing a business process for the order and invoice management for insolvency administrators (see [14]).

- Availability, status and further plans
  The architecture of the ActorSphere execution environment itself is also based on actors. This means all services of the actorsphere like uploading actors, user management, assigning users to actors e.g. are also implemented with actors. Adding a new service to the Actorsphere like deleting actors is just uploading the corresponding actor. This concept allows to extend the management and monitoring functionality in a very straight way. For teaching and research purposes free accounts are available via the Institute of Innovative Process Management (http://www.i2pm.net).

2.7 Metasonic Suite

The Metasonic Suite was the first subject-oriented tool suite. The development started around the year 2000. The Metasonic suite concists of four components: Metasonic Build, Metasonic Proof, Metasonic Flow and Metasonic Base.

- Modelling Behavior
  Part of the Metasonic suite is a dedicated modelling tool called Metasonic Build. It allows the definition of Subject Interaction Diagrams and Subject Behavior Diagrams. The modeling tool runs on Windows PCs. The behavior of a process can be tested with Metasonic Proof.

- Modelling Business Objects and User Interface
  The modeling tool allows also the definition of business objects including a form editor for screen layouts.

- Embedding Tools
  Existing applications can be integrated as refinements of internal functions. This is supported by a template for configuring refinements. Embedding existing applications as refinements is a functionality of Metasonic Build. Whereas the embedding in an organisation is part of the Metasonic Base functionality. This component allows the assignment of subjects to users and user groups.

- Storage Format and Process Model Repository
  Process models (SID, SBD, Business objects, Layout, Refinements and assignment of subjects to users) are stored in a proprietary database.

- Execution Environment
  Process models are uploaded to the execution environment. The upload function is part of Metasonic Base. The models are executed by Metasonic Flow.

- Availability, status, further plans
  The Metasonic Suite is now owned by Allgeier IT SOlutions GmbH. It is not clear whether this suite is still maintained. The last release 5.2 was anounced May 2015. Other information is not available at https://www.metasonic.de/ (last access January 2017).

3. SUMMARY

Following tables summarize the features of the tools outlined in the sections before. Table 1 summarizes the modeling aspects whereas table 2 gives an overview of execution aspects.

Table 1: Tool overview modeling aspects

Tool	Behavior	Data	User Interface	Repository
KIT S-BPM modeler	Visio shapes	Visio shapes	Visio shapes for actorsphere	Files in OWL format
InFlow S-BPM suite	Modeling environment available via browser	no special modeler a availbale	no special modeler a availbale	XAML file
UeberFlow suite	Text based modeler, graphical modeler in development	text based modeling	generated automatically depending on the request input function	Document-oriented database which allows export and import of Json files
TUD-S-BPM Groupware	Web based modeler	not supported	not supported	SQL data base
SAReI	Shapes of the yED editor	Shapes in yED editor	not supported	.graphml files what is an XML file
actnconnect actorsphere	special version of the KIT S-BPM modeler	Extension of the KIT S-BPM modeler	XML Files	XML-based modeling
Metasonic S-BPM suite	Integrated modeler	Integrated modeler	Integrated modeler	Database

Table 2: Tool overview execution

Tool	execution	availability
KIT S-BPM modeler	no execution environment	Shapes are available for free (see i2pm.net)
InFlow S-BPM suite	Execution in the cloud, Microsoft based	available as open source
UeberFlow suite	Execution in the cloud	available as open source
TUD-S-BPM Groupware	Execution in the cloud	not supported anymore
SAReI	Execution in the cloud	available as open source
actnconnect actorsphere	Execution in the cloud, open actorsphere platform	Product and services offered by actnconnect
Metasonic S-BPM suite	Execution on dedicated server	Product and services offered by Allgeier IT Solutions GmbH

4. CONCLUSIONS

This article shows that during the last year several subject oriented tool suites have been developed. Each of these suites focus on different aspects. The KIT modelling tool focuses on the modeling aspect and does not consider execution aspects. This modeling tool allows to use new modeling constructs which are not contained in [1]. Its main purpose is to describe process models in a simple way using a standard tool (Visio).
The Inflow tool suite uses a totally different platform than the other tool suites. The objetive of the development of this tool suite was to find out how Microsoft tools can be used to create a tool suite with less programming effort as possible just using existing Microsoft components.
The Ueberflow suite has the objective to create a workflow framework which can be used for implementing workflow systems following various philosophies (control flow orientation, subject orientation etc.).
The TUD-S-BPM Groupware focuses on the integration aspect. How can processes be combined with calenders, todo lists, mails etc.. Additionally the architecture of TUD-S-BPM is subject based and implemented in Scala/Akka, which allows to execute business processes across clouds.
SAReI covers the complete functionality as described in [1]. The objective of this implementation has been to create a reference system in order to compare it with other implementations to find out the their gap to the complete functionality.
The Actorsphere is a commercally available product/service. It desined for clouds and for creating business choreographies, which means combining sactors for creating new complex services.
The Metasonic suite is the oldest subject oriented tool suite. All the other implementations learned from this tool suite.

Currently there is an initative to bring the various tool suite closer to each other. There is a discussion to define a standard storage format which should allow to create models with different tool suite and exchange them between the various tool suite. Additionally the execution of a process should be possible on any execution environment independent from the used modeling tool.

5. REFERENCES

[1] Fleischmann A., et. al. , Subject oriented Business Process Management, Springer Verlag, Heidelberg 2012

[2] Fleischmann A., Schmidt w., Stary C., Subject-Oriented Business Process Management, Handbook on Business Process Management Volume II, pp 601-621, Springer Verlag, Berlin Heidelberg 2015

[3] Hollingsworth D., The Workflow Reference Model, TC00-1003 Issue 1.1, Workflow Management Coalition, 24 November 1994

[4] Hollingsworth D., The Workflow Reference Model: 10 Years on the Workflow Handbook 2004, (Workflow Management Coalition, Cohasset 2004) pp. 295-312

[5] Elstermann, M., Ovtcharova, J. , Abstract Layers in PASS - A Concept Draft. S-BPM One 2014 (S. 125-136). Heidelberg: Springer.

[6] Elstermann, M., Ovtcharova, J. , An Editing Concept for PASS Layers. S-BPM One 2014 (S. 137-146). Heidelberg: Springer.

[7] Elstermann, M., Seese, D., Fleischmann, A. . Using the Arbitrator Pattern for Dynamic Process Instance Extension in a Work-Flow Management System. Abstract State Machines, Alloy, B, VDM, and Z, (S. 323-327). Pisa (2012), Italy.

[8] Fleischmann C., Riha K, Stangl G., Logistics Processes modelled in S-BPM and implemented in SAP to reduce Production Lead Time, S-BPM-ONE 2016, ACM Digital Library 2016.

[9] Singer, R., Raß, S., Structured Communication - Approaching S-BPM with Microsoft Technologies, in Albert Fleischmann et al. (eds.), S-BPM in the Wild, Springer, 2015.

[10] Singer, R., Kotremba, J., Raß, S., Modeling and Execution of Multienterprise Business Processes, 16th IEEE Conference on Business Informatics, Workshop on Cross-Organizational and Cross-Company BPM (XOC-BPM), Genf, Switzerland, 14-17 July 2014.

[11] Raß, S., Kotremba, J., Singer, R., The S-BPM Architecture: A Framework for Mul-ti-Agent Systems, In Proceedings of the 2013 IEEE/WIC/ACM International Confer-ence on Web Intelligence and Intelligent Agent Technology (WI-IAT), Atlanta, USA, 17-20 November 2013.

[12] Kotremba J., Raß, S., Singer, R., Distributed Business Processes - A Framework for Modeling and Execution, September 2013, arXiv:1309.3126.

[13] Workflow Pattern Initiative, Workflow Pattern Home Page (2001), http://www.workflowpatterns.com, last access January 2017

[14] Strecker F. Gniza R., Hollosy T., Schmatzer F., Order and Invoice Management for Insolvency Administrators: A Software-Solution based on

Business Actors, Proceedings of the S-BPM-ONE 2016.

[15] Strecker F. Gniza R., Hollosy T., Schmatzer F.,Business Actors as Base Components of complex and distributed Software Applications, S-BPM-ONE 2016, ACM Digital Library 2016.