DPIL Navigator 2.0: Multi-Perspective Declarative Process Execution

Stefan Schönig, Lars Ackermann, Stefan Jablonski

University of Bayreuth,
Universitätsstraße 30, 95447 Bayreuth, Germany
{firstname.surname}@uni-bayreuth.de

Abstract. The Declarative Process Intermediate Language (DPIL) is a declarative process modelling language that allows for specifying multi-perspective and multi-modal flexible, processes. The DPIL Framework provides a tool set for supporting flexible and multi-perspective business processes based on the language DPIL. The DPIL Navigator 2.0 depicts a declarative execution engine for enacting models. It comprises a web-based responsive user interface which allows process participants to choose and perform tasks and offers enterprise content management (ECM) functionality. In this demo paper, we present a declarative execution system that covers all process perspectives, in particular data and resource handling. The presented process execution software enhances and extends previous versions of the tool by several new features.

Keywords: Process Execution, Declarative Modelling, Multi-Perspective

1 Background and Significance to BPM

In flexible business processes the exact flow of activities cannot be fully determined at design time. These processes require highly flexible IT support. Flexible processes are common in healthcare where, e.g., patient diagnosis and treatment processes require flexibility to cope with unanticipated circumstances. Declarative models define execution constraints that the process has to satisfy [1] and focus on crosscutting relations instead of the flow of activities. Research has shown that declarative approaches are well-suited for modelling flexible processes [1, 2]. Declarative modelling provides means for increasing the number of alternative paths without explicitly modelling them. Given the vast amount of alternatives, a differentiation between mandatory and only recommended actions is reasonable [3]. Independent from a specific modelling paradigm different perspectives on a process exist. The data perspective deals with process data and documents. Business documents and other objects which are used within activities as well as local variables of the process, may reflect pre- and post-conditions of activity executions. Typically, process data is passed into and out of applications through interfaces, allowing manipulation of the data. The resource perspective manages the involvement of human resources in processes, e.g., it provides an anchor to the process in the form of human roles responsible for executing activities.

In this demo paper, we present *a declarative process execution system that covers all process perspectives*, in particular *data integration and resource handling*. The *Declarative Process Intermediate Language (DPIL)* [4] is a multi-perspective and multi-modal process modelling language on a textual basis. Unlike other declarative languages it allows for representing several perspectives, namely, behaviour, data and resources and their crosscutting relations. It is multi modal, meaning that both mandatory and recommended actions can be specified. The DPIL framework offers a rich toolset for modelling and executing process models: *(i)* DPIL models can be defined using a textual editor the *DPIL Modeller*. DPIL models can be *(ii)* executed by a declarative execution engine, the *DPIL Navigator 2.0* that is directly accessible at `http://navigator.kppq.de`. The expressiveness of DPIL and its suitability for business process modelling have been evaluated with respect to the well-known Workflow Patterns.

2 DPIL Navigator 2.0: Overview and Demo Guidelines

In this section, we will briefly explain the functionality of the new DPIL Navigator 2.0 by means of a concrete example process.

2.1 DPIL Clinical Example Process and Case Study

The following model is a sketch of a flexible *clinical evidence application process* and contains many of the keywords and structures of the DPIL grammar:

```
use group Doc
use relationtype memberOf

use repository FileStore {
    url "http://..."  user "UserID1"  password "pw" id "-default-"
}

sequence(t1, t2) iff start(of t2 at :t) implies complete(of t1 at < t)
produces(t, d) iff complete(of t at :t) implies write(of d at < t)

process ClinicEvidence "Impose Clinical Findings" {

    task NF "Impose neurological finding"
    task CD "Get confinement of discretion"
    task PF "Impose psychopathological finding"

    document _NF "Neurological finding" at FileStore
    variable _CD "Confinement of discretion"
    variable _PF "Psychopathological finding"

    ensure produces(NF, _NF)
    ensure produces(PF, _PF)
    advise "Confinement of discretion should exist": sequence(CD, NF)
    advise "Confinement of discretion should exist": sequence(CD, PF)
    advise "Confinement of discretion should be obtained by a doctor":
    start(of CD by :i) implies relation(subject i predicate memberOf object Doc)

    milestone "findings imposed ": complete(of PF) and complete(of NF)
}
```

First, the group *Doc* and the relation type *memberOf* of the organizational management system is referenced to make them usable in the model. Afterwards the connection to a CMIS[1] compatible ECM system *FileStore* is defined which enables the process to access documents of this repository. Furthermore the macro definitions *sequence* and *produces* conclude the head section of the model. The macro *sequence* depicts a simple temporal dependency between two tasks while *produces* requires a value for a certain data object before a certain activity can be *completed* and, thus, describes an outgoing data dependency. The model describes the process of a clinical evidence application and separates it into the activities *NF* (impose neurological finding), *CD* (Get confinement of discrection) and *PF* (Impose psychopathological finding). Three data objects are part of the process. These are the document *_NF* (Neurological finding), and the in-process variables *_CD* (Confinement) and *_PF* (Psychopathological finding). Below the described entities of the model two mandatory (*ensure*) rules require that the data objects *_NF and _PF* have to be written before the corresponding tasks *NF and PF* can be completed. The subsequent soft rules (*advise*) recommend that the confinement of discretion is available before performing the examinations. The third advise recommends that the confinement is obtained by a doctor. A *milestone* is reached and the case is completed after imposing both findings.

2.2 The DPIL Navigator 2.0

The demo will show the functionality of the *DPIL Navigator 2.0*. The tool is based on the execution principle as visualized in Fig. 1. Tasks of a DPIL model undergo a life cycle composed of events that is managed by the engine. A task, e.g., can be started and completed. The current state of a process is then the

[1] http://www.oasis-open.org/committees/cmis

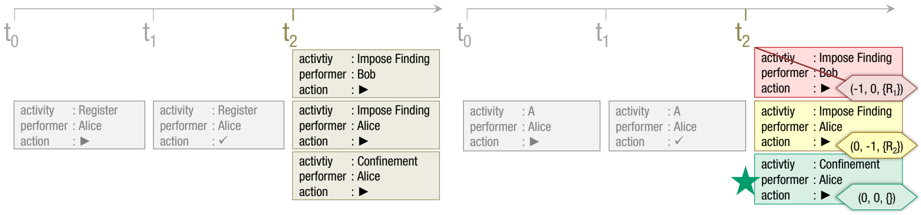
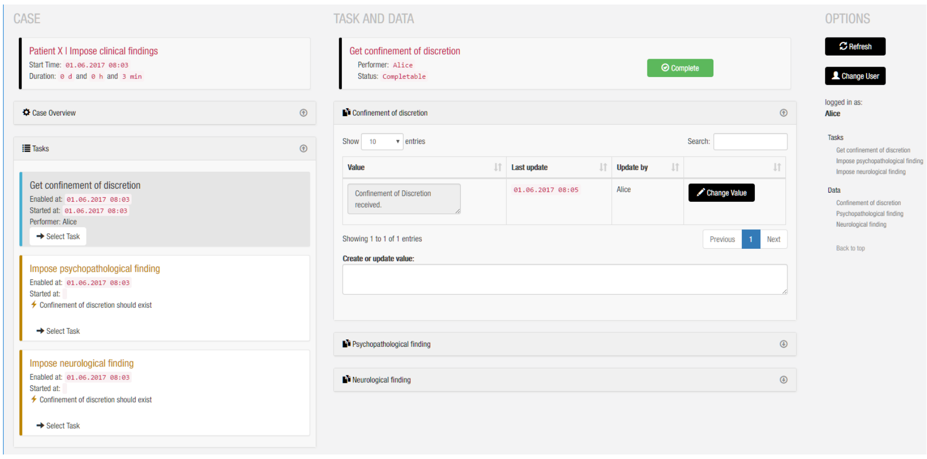


Fig. 1: Principle of declarative process execution of the DPIL Navigator 2.0



CASE

Patient X | Impose clinical findings
Start Time: 01.06.2017 08:03
Duration: 0 d and 0 h and 3 min

Get confinement of discretion
Enabled at: 01.06.2017 08:03
Started at: 01.06.2017 08:03
Performer: Alice

Impose psychopathological finding
Enabled at: 01.06.2017 08:03
Started at:
Confinement of discretion should exist

Impose neurological finding
Enabled at: 01.06.2017 08:03
Started at:
Confinement of discretion should exist

TASK AND DATA

Get confinement of discretion
Performer: Alice
Status: Completable

Confinement of discretion

Show 10 entries

Value	Last update	Update by	
Confinement of Discretion received.	01.06.2017 08:05	Alice	Change Value

Showing 1 to 1 of 1 entries

Create or update value:

OPTIONS

logged in as:
Alice

Case Overview
Tasks
Select Task
Complete
Search:
Previous
1
Next
Psychopathological finding
Neurological finding
Refresh
Change User
Tasks
Get confinement of discretion
Impose psychopathological finding
Impose neurological finding
Data
Confinement of discretion
Psychopathological finding
Neurological finding
Back to top

Fig. 2: Web-based user interface of the DPIL Navigator 2.0

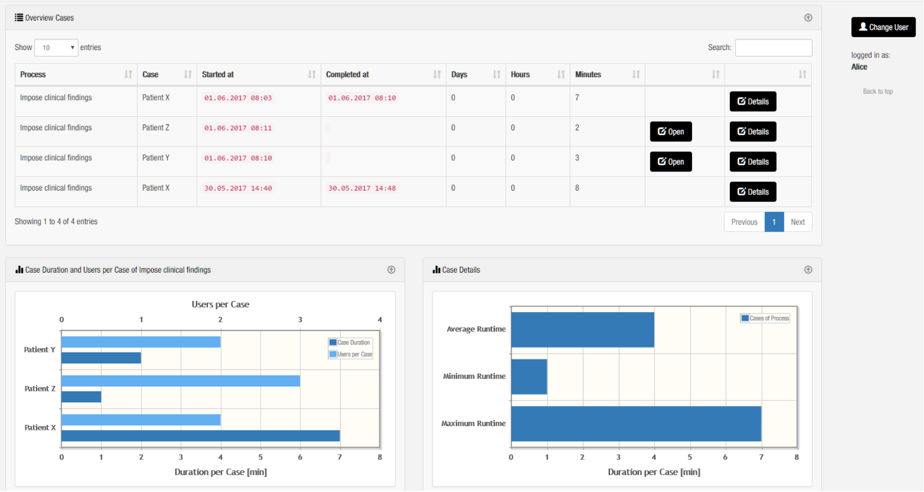
series of past events. Besides elements like tasks a process may specify rules constraining that series of events. When the model is executed, the engine simulates one event ahead for every element and evaluates the resulting series of events. Each event that does not violate any *ensure* rule is related to an action that the engine interprets immediately. A simulated start of a task by a certain participant is interpreted as the assignment of this task to the user. If an event violates an *advise* rule, the action is marked as not recommended.

The tool offers *enterprise content management (ECM)* functionality in form of the integration of the Alfresco CMIS system[2] where process data is stored and managed. The *web-based responsive interface* of the DPIL Navigator is visualized in Fig. 2. Currently available human tasks and reached milestones are shown on the left hand side of the screen. Different modalities are encoded by means of different colours, i.e., actions that violate certain soft rules are highlighted by orange colour. The right hand side shows activity details, data variables and documents from the connected document store that the current user is allowed to interact with. The DPIL Navigator with some example processes deployed is accessible at `http://navigator.kppq.de`.

2.3 New Features and Advances

The presented software enhances and extends previous versions of the tool [5] with several features: *(i)* the user interface has been redesigned responsively based on user feedback to be also applicable on mobile devices; *(ii)* the interface has been extended to offer typical process management functionality (e.g., task lists, monitoring dashboards with graphs and a document workbench); *(iii)* the system now integrates directly with the Alfresco ECM system for document man-

[2] https://www.alfresco.com/products/enterprise-content-management/community



Process	Case	Started at	Completed at	Days	Hours	Minutes		
Impose clinical findings	Patient X	01.06.2017 08:03	01.06.2017 08:10	0	0	7		Details
Impose clinical findings	Patient Z	01.06.2017 08:11		0	0	2	Open	Details
Impose clinical findings	Patient Y	01.06.2017 08:10		0	0	3	Open	Details
Impose clinical findings	Patient X	30.05.2017 14:40	30.05.2017 14:48	0	0	8		Details

Showing 1 to 4 of 4 entries

Fig. 3: Monitoring dashboard of the DPIL Navigator 2.0

agement; *(iv)* process execution is now logged enabling process mining and *(v)* resource management has been integrated to manage and change user accounts.

3 Maturity, Prototype, Screencast and Future Work

The DPIL Navigator 2.0 is a well-evaluated prototype that is used for demonstration in academia as well as in industry. The DPIL Navigator with the example process deployed is accessible at `http://navigator.kppq.de`. A screencast is available at `https://youtu.be/X0nr0Hr4Bzk`. *The contribution of this tool demo is to show that also declarative process execution systems are ready to use for practical applications.* For future extensions, we are currently developing a graphical DPIL editor to facilitate modelling.

References

1. D. Fahland, D. Lübke, J. Mendling, H. Reijers, B. Weber, M. Weidlich, and S. Zugal, "Declarative versus imperative process modeling languages: The issue of understandability," in *BPMDS*, pp. 353–366, 2009.
2. R. Vaculín, R. Hull, T. Heath, C. Cochran, A. Nigam, and P. Sukaviriya, "Declarative business artifact centric modeling of decision and knowledge intensive business processes," in *EDOC*, pp. 151–160, 2011.
3. G. Regev and A. Wegmann, "A regulation-based view on business process and supporting system flexibility," in *CAiSE*, vol. 5, pp. 91–98, 2005.
4. M. Zeising, S. Schönig, and S. Jablonski, "Towards a Common Platform for the Support of Routine and Agile Business Processes," in *CollaborateCom*, 2014.
5. S. Schönig and M. Zeising, "The DPIL Framework: Tool Support for Agile and Resource-Aware Business Processes," in *BPM Demos*, 2015.