Mixing Paradigms for More Comprehensible Models

Michael Westergaard

Eindhoven University of Technology

National Research University Higher School of Economics

Tijs Slaats

IT University of Copenhagen

Exformatics A/S

Overview

TS2

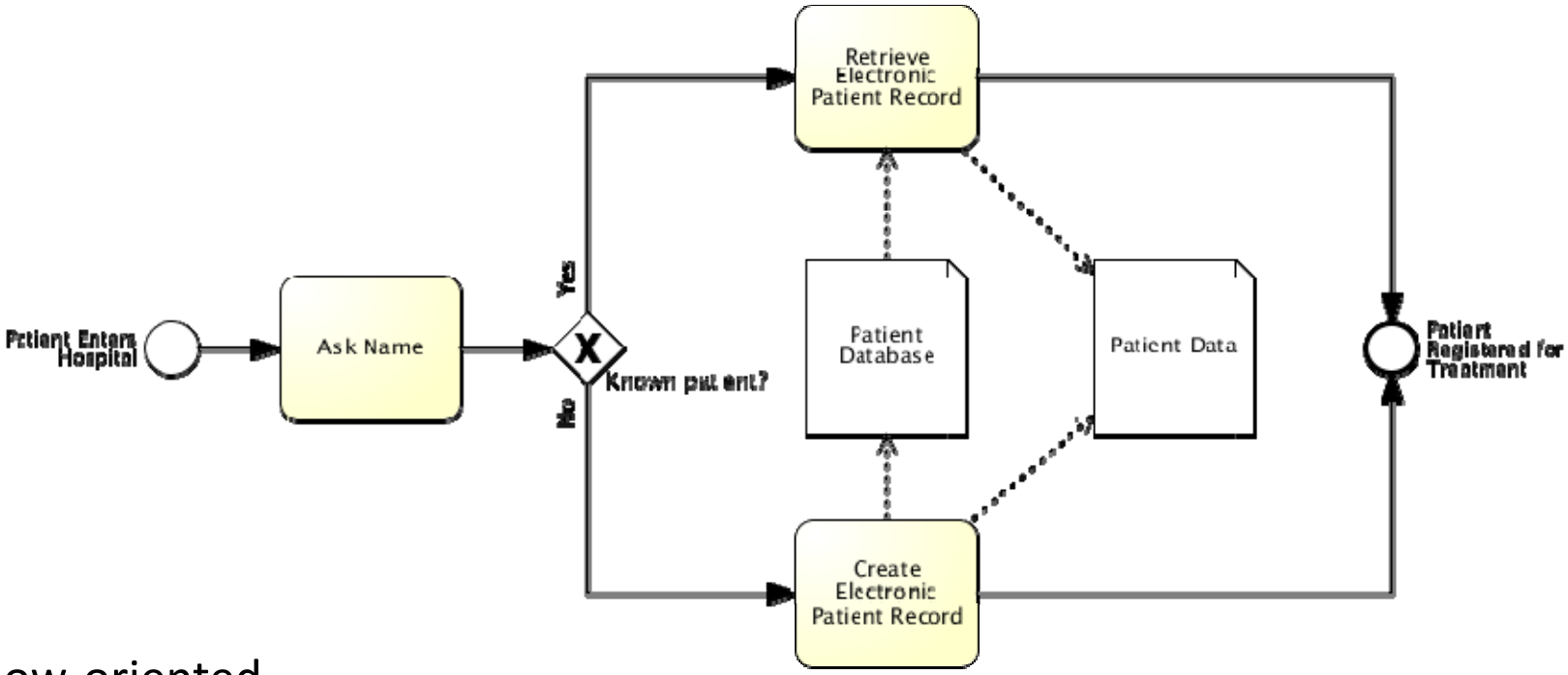
- Motivation
- Implementation
- Related work
- Future Work
- Conclusion

TS2

Industrial PhD

Tijs Slaats, 2013/1/23

Motivation
Imperative Workflow Languages

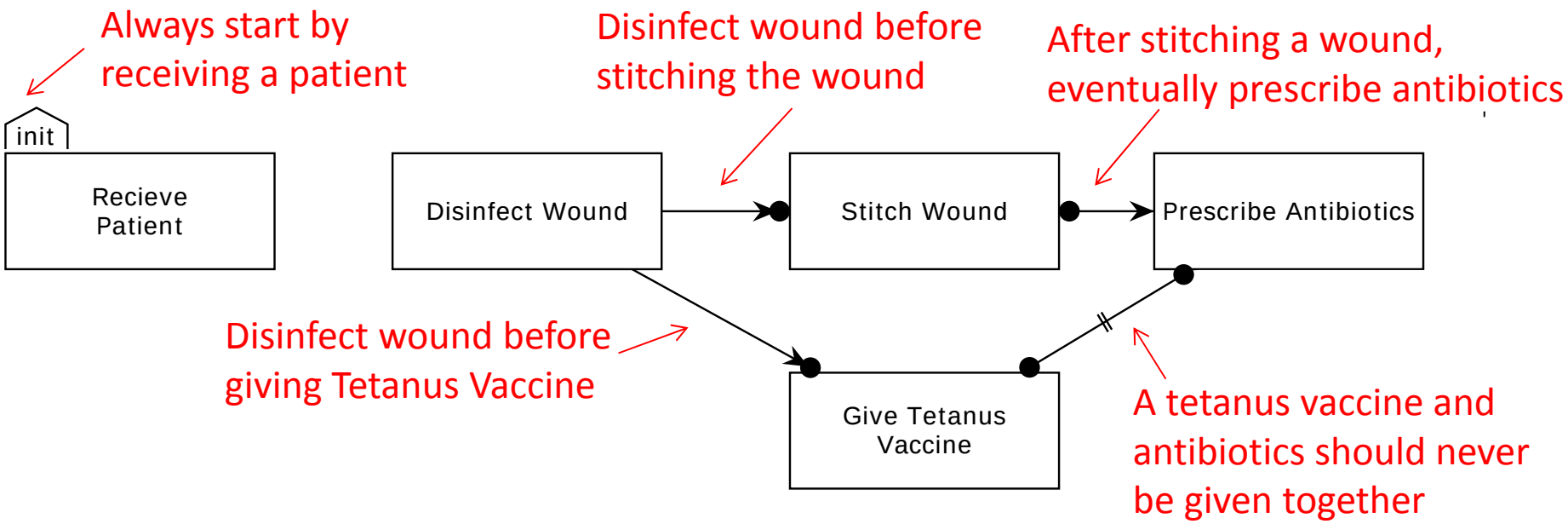


Flow-oriented
Well-suited to rigid processes
In a model with no flow nothing can happen
Adding flow allows for more possible behaviors
Common in academia and industry

eXformatics

IT University of Copenhagen

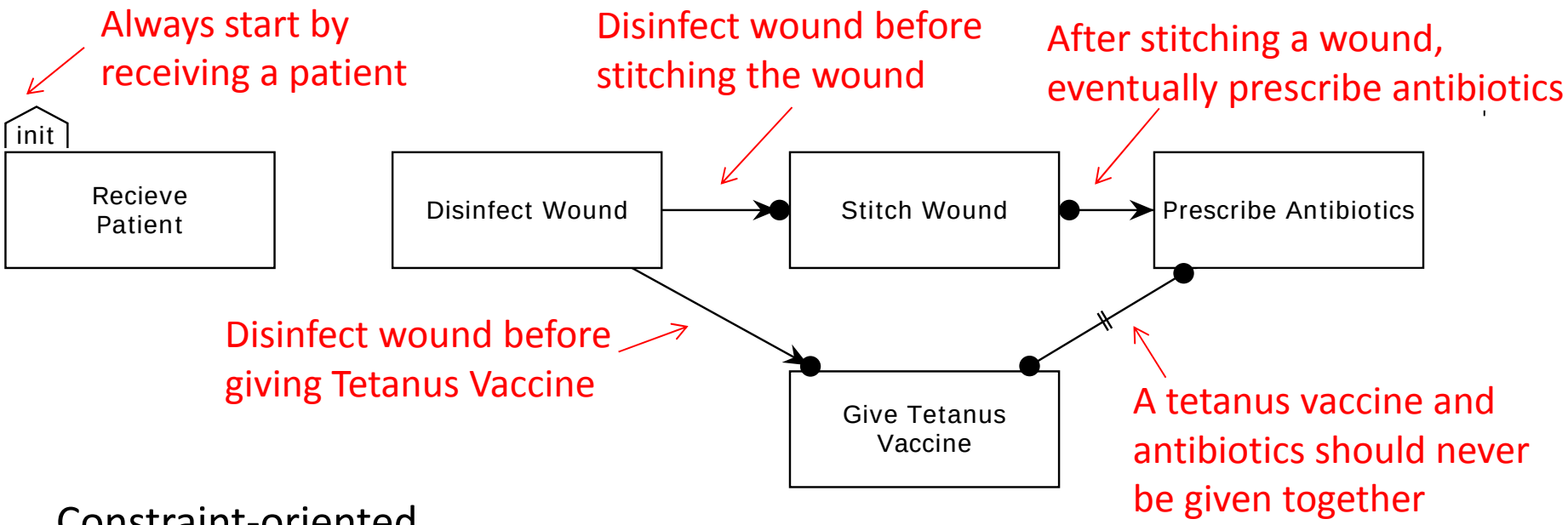
Motivation
Declarative Workflow Languages



eXformatics

IT University of Copenhagen

Motivation
Declarative Workflow Languages

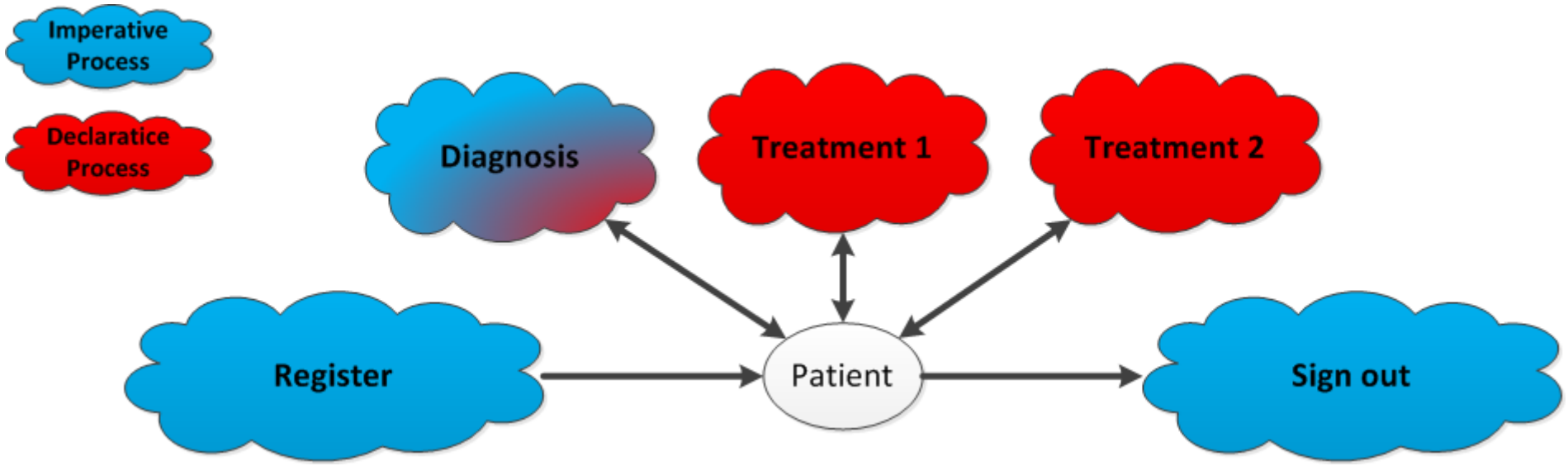


Constraint-oriented
Well-suited to flexible processes
In an unconstrained model anything can happen
Adding constraints limits behavior
Still a novelty in industry

eXformatics

IT University of Copenhagen

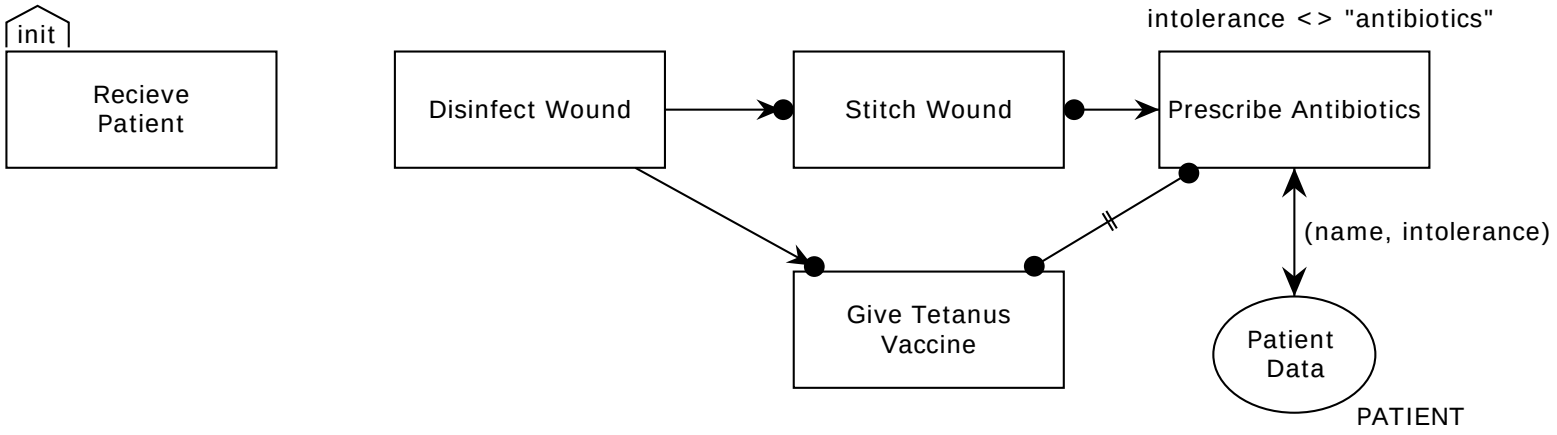
Motivation
Combined Workflow Languages



- Different sub processes of the same system may be more imperative or declarative in nature
- Modeling an declarative process imperatively, or the other way around often leads to incomprehensible models.

Motivation
Combined Workflow Languages

- Allows the use of advanced features of imperative languages in declarative models
- For example, data:

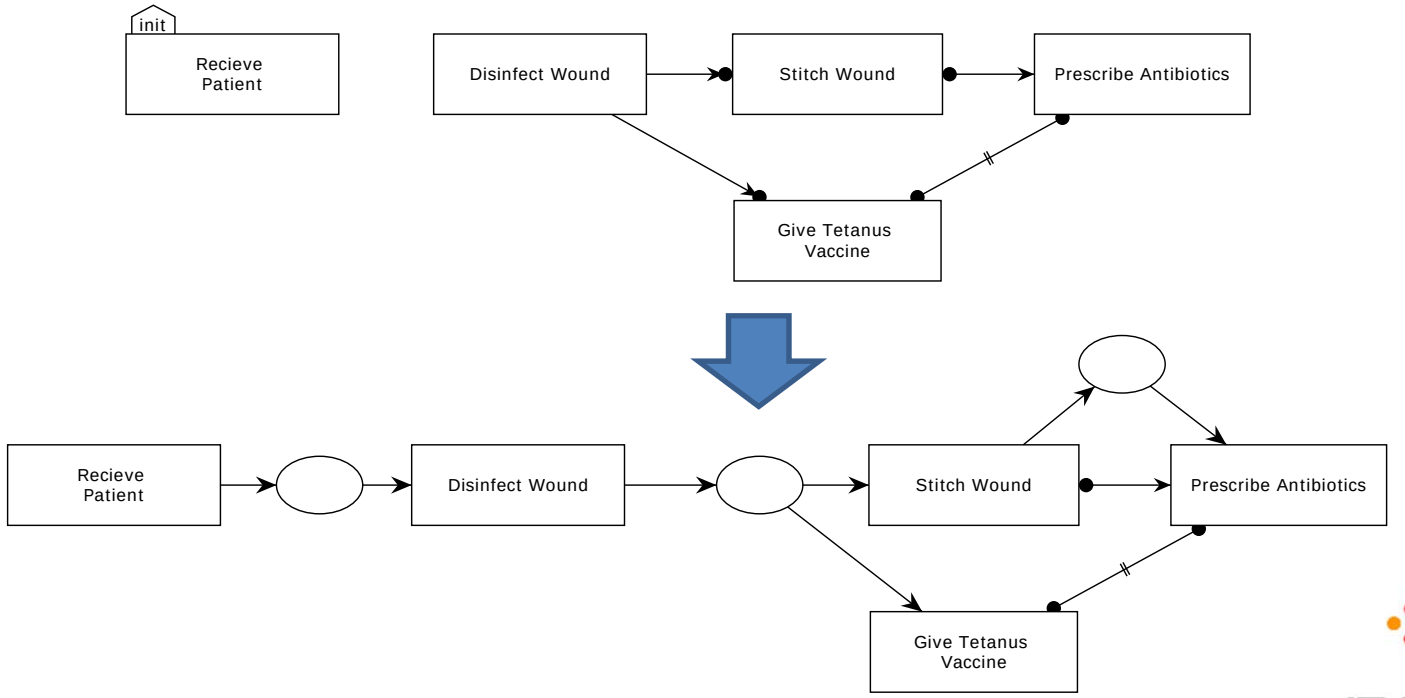


eXformatics

IT University of Copenhagen

Motivation
Combined Workflow Languages

- Provides a tool for a refinement methodology where one refines an abstract declarative model into a (more strict) imperative model.



eXformatics

IT University of Copenhagen

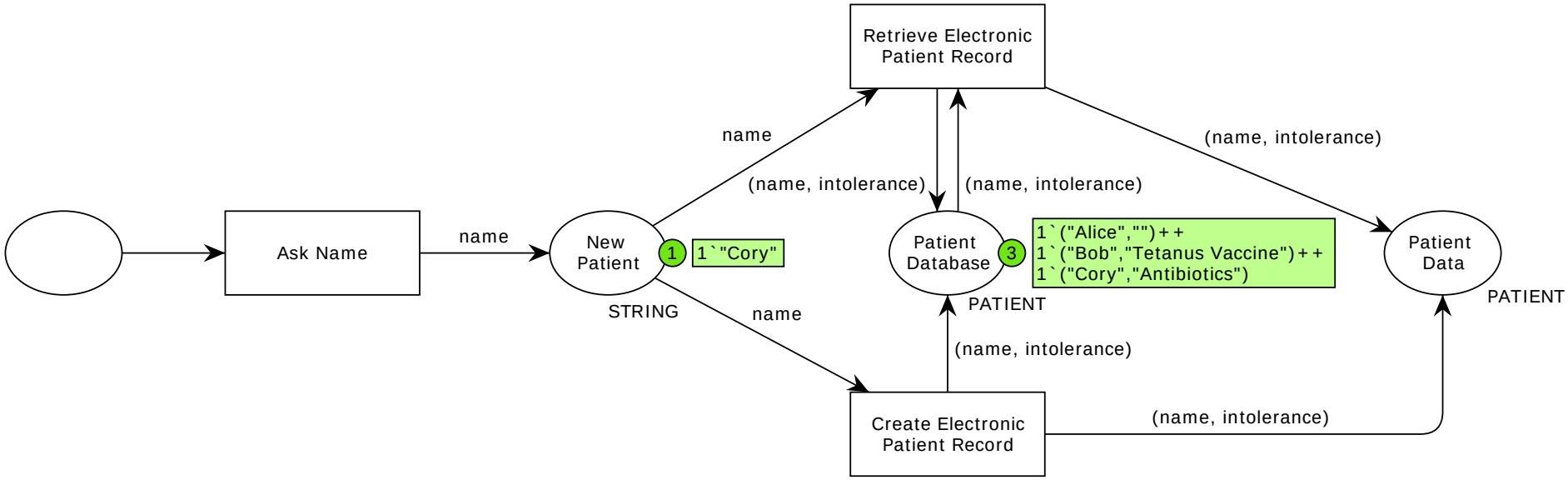
Motivation
Combined Workflow Languages

- Recent study done with BPM practitioners at a Dutch BPM provider showed interest in a mixed imperative and declarative approach to modeling.

eXformatics

IT University of Copenhagen

Implementation
Coloured Petrinets

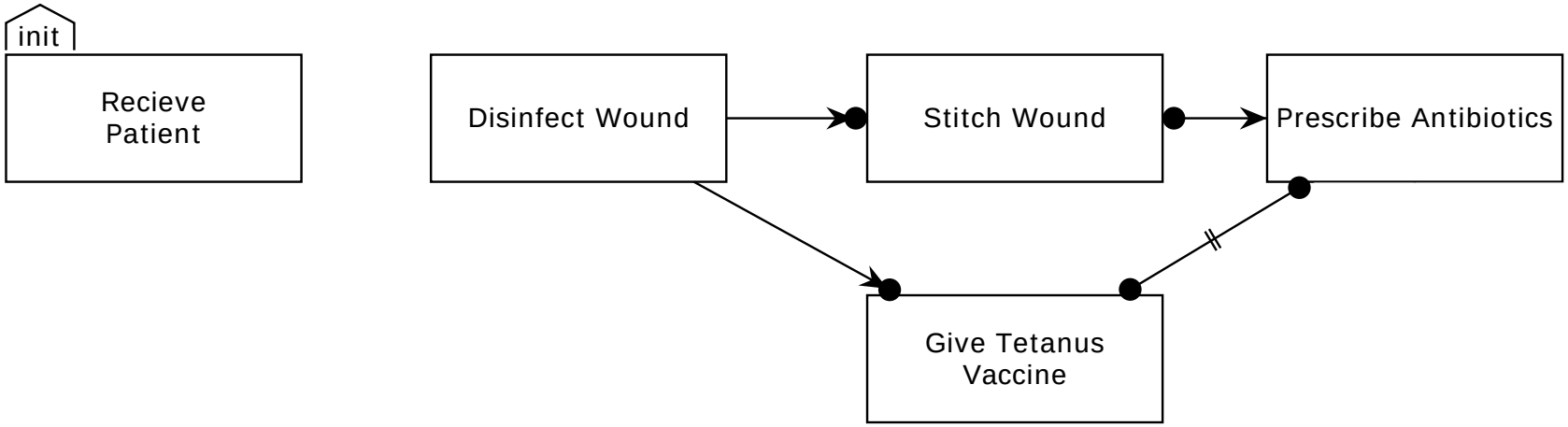


Commonly used for describing (business) processes
Strong following in academia
Popular for formalizing the BPMN semantics
Matches well with the declarative approach
Allows for modeling data in processes
Has seen use in industry

eXformatics

IT University of Copenhagen

Implementation
Declare

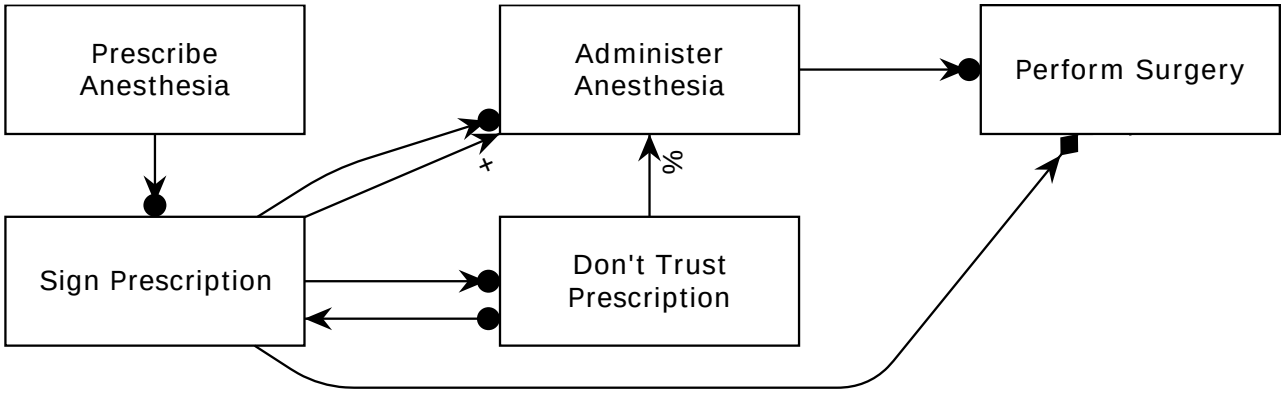


First and commonly used declarative language for workflows
Offers a large set of constraints aimed in particular at BPM

eXformatics

IT University of Copenhagen

Implementation
DCR Graphs



Formal declarative model aimed at process modeling in general.
Offers strong formal expressiveness with only four basic constraints.

eXformatics

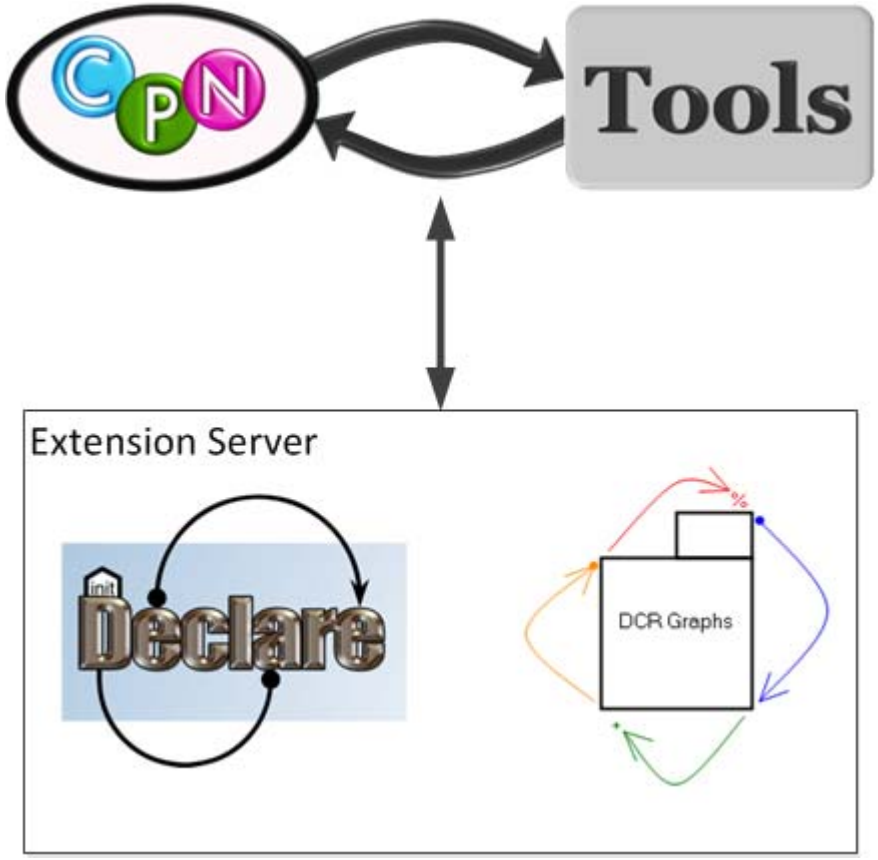
IT University of Copenhagen

Implementation
CPN Tools

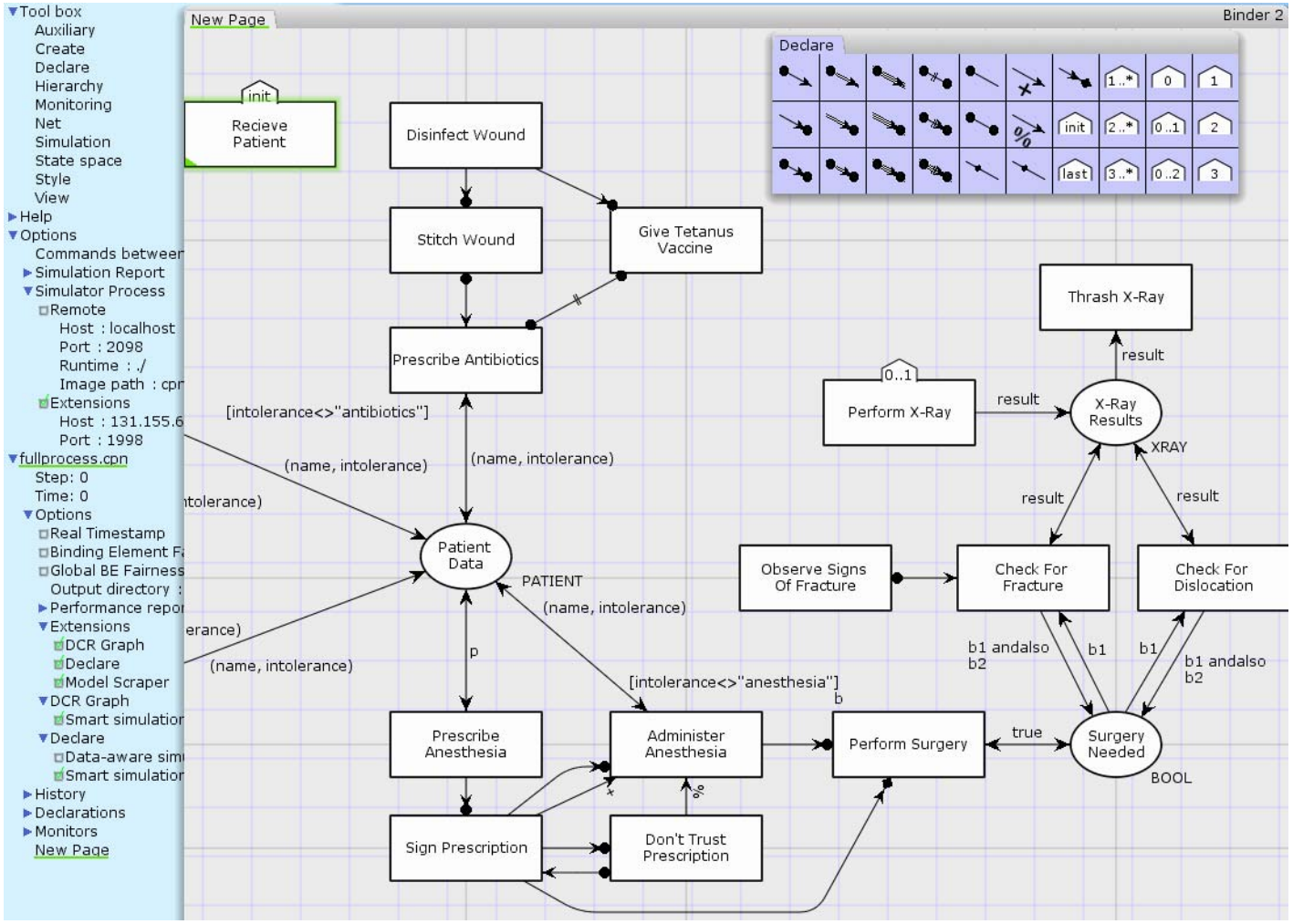
- Tool for modeling Coloured Petrinets with support for simulation and statespace analysis
- Originally developed at Aarhus University
- Now supported by Michael Westergaard at the AIS group, Eindhoven University of Technology
- Available at: http://cpntools.org/

Implementation
CPN Tools

- Latest version of CPN Tools supports *extensions.*
- Extensions available for:
  - Declare
  - DCR Graphs



Implementation
CPN Tools



eXformatics

IT University of Copenhagen

Implementation
Simulation of Combined Models

- We consider three modes of simulation for the combined language:
  - Simple simulation: we only allow transitions that do not violated the declarative constraints
  - Smart Simulation: we only allow transitions that allow us to reach an accepting state, but only considering the declarative constraints
  - Data-aware Simulation: we only allow transitions that allow us to reach an accepting state, considering the combination of the CP-Net and declarative constraints. This is generally not decidable (the transition system of the CP-Net may be infinite).

- First two simulation modes are implemented in the tool.

Related Work

- Pockets of flexibility in workflow services [Sadiq et al.] proposes an approach where workflows are specified with “declarative” pockets of flexibility, but at runtime one still has to pick a specific imperative instantiation of the workflow that fits the definition.

- Flexibility as a Service (FAAS) [Aalst et al.] proposes combining Declare, YAWL and Worklets, where subprocesses can defined in any of these languages, but within a single process fragment the same language is used, therefor the result becomes more of an interleaving of Declarative and Imperative parts.

- The Guard-Stage-Milestone model [Hull et. al.] provides both imperative and declarative constructs and can be seen as a hybrid model.

Future Work

- Development of a formal refinement methodology based on using combined workflow languages
- Finding the right combination of imperative and declarative languages to use in the combined approach
- Finding the right methodology for combining these languages for user satisfaction

Conclusion

- First attempt at combining (as opposed to interleaving) existing Imperative and Declarative languages
- Positive interest from industry
- Tool support though extensions for CPN Tools
- Further research encouraged!
- Thank you for your time – questions?