

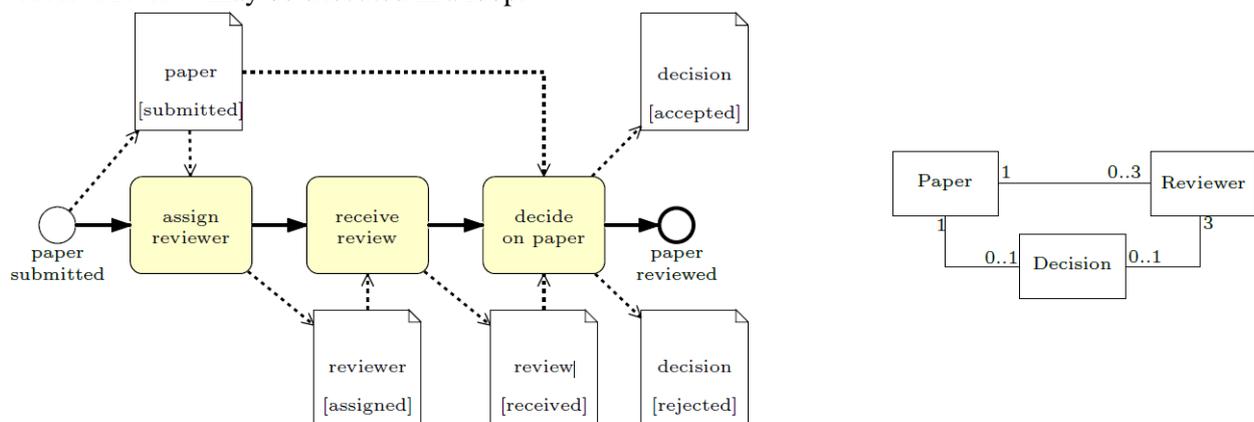
## Design-Time Support for Fragment-Based Case Management

Traditionally, business process modeling focuses on highly structured processes that have few variants. However, many real world processes are more flexible. Knowledge-intensive processes (KiPs) are primarily driven by informed decisions of experts, so called knowledge-workers. KiPs are multi-variant, rapidly changing, and data-driven. Classical business process models are incapable of modeling such behavior concisely.

Fragment-based Case Management (fCM) is an approach for modeling multi-variant processes by a data model, object behaviors, and a set of process fragments. fCM combines ideas from both activity-centric and data-centric process modeling. However, KiPs are inherently complex as are their models. Process behavior and data are connected. Therefore, the fragments pose requirements on the data model and vice versa. Correct integration is crucial.

Consider the process model and data model in the figure below. The process describes that for every paper that is received, a reviewer is assigned, and a review is received before the paper is accepted or rejected. The data model describes that each paper is linked to up to three reviews and up to one decision. Each decision is linked to exactly three reviews and one paper, and each review is linked to up to one decision.

The models have inconsistencies. First, the reviewer object created in the process is not part of the data model. Secondly, the review is created without the paper being read – no link can be established. Finally, only one review is read to create a decision instead of three. To resolve this issue, a class ‘Reviewer’ may be introduced, the paper object may be read by every activity, and ‘assign reviewer’ and ‘receive review’ may be executed in a loop.



Software tools can be an effective means to cope with complexity. IDEs, for example, offer syntax highlighting, linting, auto-completion, auto-correction, and interactive simulation. The goal of this master project is exploring the possibilities to support modeling and comprehending fCM models.

- How are domain model, object behavior, and case behavior connected?
- What kind of errors can occur?
- How can errors be resolved (semi-)automatically?
- How can the parts be synchronized (auto-completion)?
- How can the joint semantics/behavior be explained to the modeler?

The feasibility of the approaches should furthermore be evaluated by implementing a prototype. Ideally, the prototype is a web-based solution. We encourage the participants to build upon the state-of-the-art open source process modeling tools, such as [bpmn.io](http://bpmn.io).

**Contact.** For more information or questions contact Stephan Haarmann and Dorina Bano:  
[stephan.haarmann@hpi.de](mailto:stephan.haarmann@hpi.de); [dorina.bano@hpi.de](mailto:dorina.bano@hpi.de)