Using the $i^*$ framework:

**Which tool?**

- $i^*$ is useful in:
  - Requirements engineering
  - Organizational analysis
  - Business process reengineering

- **Strong points:**
  - Visual utility
  - Representation of intentional concepts
  - Easy to understand and use

- **For using $i^*$ we need to:**
  - Adopt the most suitable variant of $i^*$
  - Establish guidelines for defining the models
  - Use tool support

**Which $i^*$?**

**J-PRiM: Tool support for the PRiM methodology**

**Requirements:**
- Programming language: JAVA
- Development environment: Eclipse
- Database: MySQL

**PRiM: Process Reengineering $i^*$ Methodology**

**Phase 1:** Analysis of the Current Process

- Step 1.1. Analysis of the Current Process
- Step 1.2. Documenting the Current Process

Many projects consist of reengineering current activities.

**Phase 2:** Construction of the $i^*$ Model

- Step 2.1. Actor Identification and Modelling
- Step 2.2. Building the Operational $i^*$ Model
- Step 2.3. Building the Intentional $i^*$ Model
- Step 2.4. Checking the Resulting $i^*$ Model

$i^*$ models describe the current process and the intentionality behind the process.

Alternatives may include new actors and activities.

**Phase 3:** Generation of Alternatives

- Step 3.1. Reengineering the Current System
- Step 3.2. Adding/Removing System Actors
- Step 3.3. Reallocating Responsibilities
- Step 3.4. Checking the Resulting $i^*$ Model

Automatic generation of alternative $i^*$ models

- Automatic generation is achieved by distributing activity responsibility among the chosen actors.
- $i^*$ Models and their alternatives can also be built from scratch

Future work:

- Transform $i^*$ models into its UML specification
- Add a graphical representation of the models
- Support for other $i^*$ frameworks and methods

**Phase 4:** Evaluation of Alternatives

- Step 4.1. Choosing Suitable Properties
- Step 4.2. Defining Property Metrics
- Step 4.3. Evaluating Alternative Models
- Step 4.4. Evaluation Trade-off Analysis

The reengineering process is iterative.

**Phase 5:** Specification of the New System

**References:**