Modeling Software Engineering Education with i*
Michael Koch – Dieter Landes
Coburg University of Applied Sciences and Arts
Introduction – The Research Project EVELIN

• Experimental improVEment of Learning software engINeering

• Creating and Maintaining a course profile including:
  • competencies
  • teaching goals / intended learning outcomes
  • tasks and resources to reach these goals

→ Helps to make the evolution of a course more:
  • transparent
  • comparable
  • traceable

iterative process
Motivation

We are looking for a way to...

• make the evolution of (SE) courses more transparent, comparable and traceable

• impose structure on the complex environment of educational goals

• highlight interdependencies between goals and their environment

• supply goal profiles for CORE (Competency Repository), a recommender system for learning arrangements under development
Why i* ?

• Text-based goal profiles may be very complex and hard to read

• Interdependencies between relevant issues remain hidden in the text

• Graphical notations allow focusing on particular elements while hiding others

• Graphical notations supplement rather than replace text-based descriptions

• Competencies and teaching goals are educational goals
  \( \Rightarrow \) Goal-Modeling notations from Requirements Engineering are obvious candidates for a suitable visual notation

• i* is one promising candidate – among others
Educational Goals

• Competencies may be seen as (abstract or soft) goals of education and qualification

• A competence cannot be taught directly, it has to be acquired individually by each student

• We use competencies as a basis for deducing teaching goals

• We distinguish between...
  • learning goals (goals set individually by each student)
  • teaching goals (intended learning goals defined by the lecturer)
Actors

- Student
- Faculty
- Government
- Industry
- Ethics Committee
- Lecturer
- Legal Department
- University Board

- Out of scope / passive
- Background / active
- Focus / active

...
Scenario: Informatics undergraduate program at Coburg University

- **Lecturer**
- **Student**
  - **Software Engineering**
    - Know / remember appropriate methods and notations in order to specify requirements
    - Introduction in software engineering / basics
  - **Software Modelling**
    - Be able to apply appropriate methods and notations in order to specify requirements
    - Detailed view on requirements engineering and process modelling
  - **Software Engineering Project**
    - Synthesis / capstone

Focus: t
present a more tangible impression of the term "requirements"
create an understanding on the importance of requirements and the ability to act accordingly
be able to apply popular approaches to complexity and cost estimation for software systems
be able to estimate the functional size
be able to apply popular approaches to complexity and cost estimation for software systems
be able to explain algorithm-based estimation techniques
solve exercise 1.b on exercise sheet 3
solve exercise 1.a on exercise sheet 3
be able to understand COCOMO II
be able to estimate the costs
be able to apply the Function Points method
exercise sheet 3
Help
Some+
Help
Help
...
Experiences

• The SR diagram is complex

• Yet, we reached our goal to model teaching goals and their relationships

• The modeling process was helpful to reflect the course design (e.g. isolated topics)

• Using a taxonomy with implicit relationships reduces the complexity significantly

  • Bloom

  • Anderson & Krathwohl

  • EVELIN-Taxonomy for Software Engineering
    remember – understand – explain – use – apply – develop
Experiences

• There is no standard way to achieve an educational goal

→ It is hard to break the process down to hard goals and means-end links (In some cases it may be possible on the task-decomposition level)

→ We can only describe aspects that help to achieve (or inhibit) an educational goal
Missing Aspects

We are looking for a way to integrate lecturers’...

• motivations
• considerations
• hypotheses
• rationales which give rise to a teaching goal

→ It might be a possible use case for the Belief element
Main Benefits of using i*

- The elements and link types defined by i* fit to describe the characteristics of educational goals
- We were able to describe many relevant aspects of an educational goal profile
- Relationships between goals and their environment as well as semantic gaps can be highlighted
Future Work

• We will do more in-depth analysis of goal-modeling approaches
  
  • istar
  
  • other notations, e.g. KAOS

• Design and implementation of an editor for modeling educational goals

  → We see potential in i* that can be revealed by offering a good software tailored to our needs

• Integration in our recommender system CORE (Competency Repository)
Thank you for your attention!

Michael Koch
michael.koch@hs-coburg.de

The research project EVELIN is funded by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung) under grant no. 01PL12022A.