

CS research and everyday life

Margus Freudenthal
Cybernetica

EXCS kick-off
18.09.2008

What does Cybernetica do?

- ▶ Among other things, we develop software
 - Distributed
 - With complex business logic
 - Complex communication protocols
 - Sometimes involving cryptography, low-level development, soft real-time

Examples

- ▶▶ X-Road – system for connecting state agencies
- ▶▶ VHF-VTS – controlling and listening to VHF radios from distance
- ▶▶ E-voting software
- ▶▶ eLabor – IS for medical laboratory
- ▶▶ Customs Engine – software for processing customs documents

Customs Engine

- ▶ Consists of several applications/modules, each module processes one kind of customs document
 - Customs declaration
 - TIR-carnet
 - Manifest
 - Warehousing notice
 - etc.
- ▶ Modules communicate with each other and their counterparts in other EU countries

Good opportunity for reuse

- ▶ All modules share some common functionality
 - Editing documents
 - Checking correctness of documents
 - Performing risk analysis and control
 - Responding to various events throughout document's life cycle
 - Communication with the EU
- ▶ Modules use common components/mechanisms
- ▶ Modules have similar architecture

Research topic: reuse

- ▶ Modules are constructed by putting together various mechanisms
 - Mechanism is often not component but a way of doing things (or a design idea)
- ▶ To maximise reuse, existing mechanisms should be used as much as possible
- ▶ This affects not just implementation, but also analysis, requirements and even business modelling
- ▶ Question: how to model these mechanisms and constraints, how to model the system based on these constraints

Research topic: describing things

- ▶▶ Customs Engine contains lot of business logic
 - Rules for verifying documents
 - Document's life-cycle
 - Configuration of standard components
- ▶▶ Ideally they are described directly by systems analyst
 - No need for programmers to play broken telephone
 - The language must be high-level, nontechnical and directly executable
- ▶▶ Question: what should these languages look like (graphical? textual?) and how to implement them?

Thank you

▶▶ Questions, comments?