Case WeCoTin –
A practical tool for
mass-customising
configurable products

Andreas Anderson

Outline

- Requirements for a practical configurator supporting e-commerce
- WeCoTin: our configurator implementation
  - Architecture
  - Modeling
  - Theoretical foundation
  - Core features
- Conclusions
Practical requirements 1

- Check a configuration for *completeness* and *consistency*
  - inconsistent or incomplete configuration cannot be ordered
- Support the user by deducing fully the consequences of previous selections
- Support for e-commerce → Web-based
- Efficient deployment of configuration models to sales people and customers
  - without risk of using out-dated configuration models
- Deployment with minor company specific effort

Practical requirements 2

- Support for long-term management
  - Configuration models must reflect changes in product offering
  - Product experts should be able to model the products
  - Declarative (instead of procedural), object-oriented modelling
  - Automatic or semi-automatic user interface generation
  - Programming should not be needed to create a configurator
- Multiple simultaneous configuring users
- Export of configurations to e-commerce, ERP, or PDM
- High-quality multi-lingual user interface for end user
- Management of pricing and delivery time
WeCoTin Architecture

WeCoTin Configuration Tool

- Web Browser
- ConfigServlet
- Configuration etc. (RMI)

WeCoTin Modelling Tool

- Modelling tool
- User Interface
- Smodels interface
- PCML core
- Ipars

Legend
- Component
- Information flow (protocol)

HELSINKI UNIVERSITY OF TECHNOLOGY

WeCoTin Modelling Tool
- Visually create and edit configuration models
- PCML Conceptual model
  - Feature types: compositional structure as subfeature definitions
    + properties + constraints, object-oriented (types, inheritance)
  - Compositional structure similar to generic product structures

constraint c2 Engine individual of Diesel22 implies GearBox individual of Auto_transm
constraint c3 GearBox individual of Auto_transm implies Styling="Elegance"
Logic-based Configurator Implementation

- PCML is mapped to weight constraint rules
  - a form of logic programs
  - Smodels [Simons, Niemelä, Soininen. Artificial Intelligence, 2002]: an efficient inference engine to calculate configurations and to deduce consequences
  - Mapping provides formal semantics for PCML
- Provides modeling with high-level domain-specific concepts with an underlying implementation that has clear formal semantics

User interface generation for end user

Requirements
- It should be easy to create UI for end user, preferably without programming
- When products change, changes to the UI should be minimal
- Visual look of a company must be taken into account
- The configurator should be available with desired languages

Features
- No end user UI programming: automatic generation of configuration model dependent part
  - display order, form-elements, contacting inference engine
  - modeller can override
- Visual appearance and product-dependent part separated
  - combine with templates
- Visual appearance can be utilized in new product generations with little effort
Other features and mechanisms

- Price and delivery time calculation mechanisms
- Defaults and default value packages
- Hard and soft constraints
- Graphical constraint editor
- Table-based constraints
- Can be integrated with e-commerce systems

WeCoTin Configuration Tool

- Configure over www with a standard browser
- Semi-automatically generated user interface
- Free and guided configuration order, graying, …
Flexible use & Long Term Management
- Flexible use
  - Configuration decisions in guided order or free navigation through configuration tree
  - Undo + redo
  - Configuration overview in configuration tree
  - Messages for violated constraints shown – reduces burden on user memory
- Long Term Management
  - Conceptual model of configuration as basis
  - Built-in support for multiple languages
  - Centralized maintenance due to architecture
  - Easier maintenance of user interfaces (separate slide)
  - Automatic component updates from PDM (not yet)
  - Versioning and often effectivity (not yet)

Conclusions
- A novel configurator prototype
  - Web-based architecture, suitable for e-commerce
  - High-level object- and product structure-oriented modelling
  - PCML configuration model is translated to a logic program
    - clear formal semantics, provides sound and complete inference on basis of configuration model and requirements
  - Visual modelling tool
  - Semi-automatically created web-based user interface
  - Empirical results: inference engine is efficient enough
  - Our experiences: the prototype is suitable for e-commerce
  - Modeled successfully roughly 10 products
    - PCML adequate & expressive enough for case products
  - Need separate modelling concepts for sales features or functions
  - Resource and connection oriented concepts needed in some domains