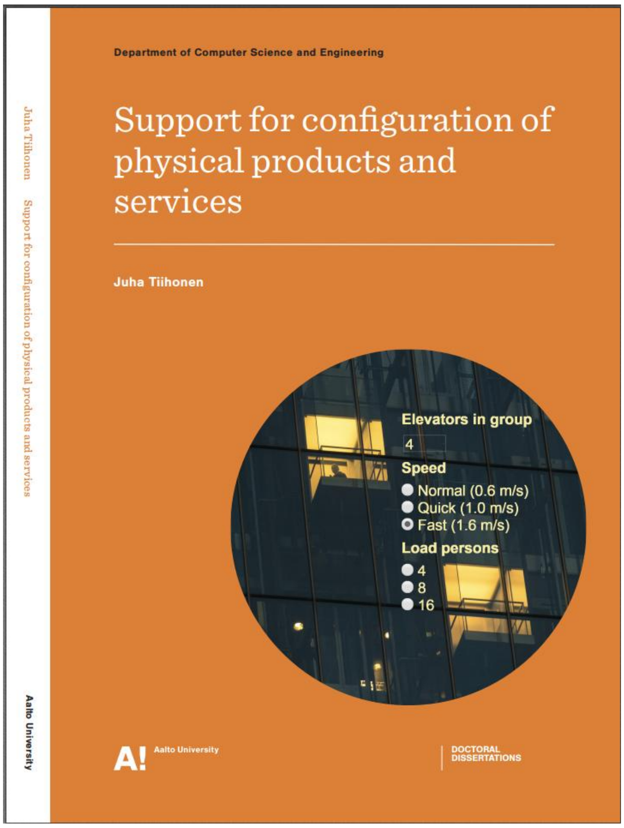
Support for configuration of physical products and services

AMICT'2015
Petrozavodsk, May 13, 2015
Juha Tiihonen

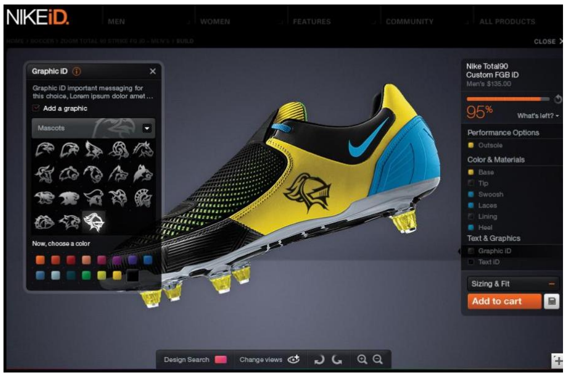


Doctoral dissertation

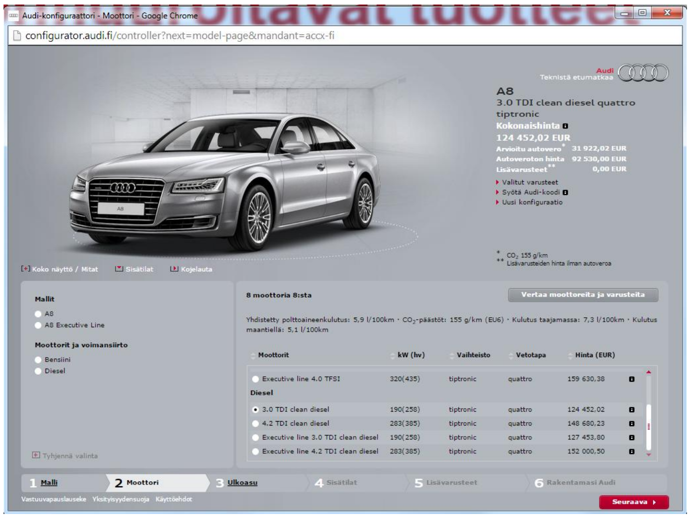
- 10/2014
- Supervisor
  - Prof. Casper Lassenius, Aalto University, Department of Computer Science and Engineering, Finland
- Advisors
  - Prof.Tomi Männistö, University of Helsinki, Department of Computer Science, Finland;
  - Prof. Emeritus Sulonen, Reijo, Aalto University, Finland
- Opponent
  - Prof. Gerhard Friedrich, Alpen-Adria Universität Klagenfurt, Austria

http://urn.fi/URN:ISBN:978-952-60-5895-5
= http://bit.ly/1zdpixN

Mass customization (Pine, 1993)

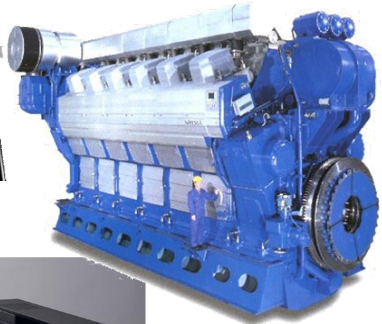


- Ideal: products meet **individual requirements** at the **price of a mass product**
  → competitive advantage
- Key capabilities (Salvador, de Holan, & Piller, 2009)
  - Understand customer needs and their variability
  - → solution space development
  - → product family: *configurable product*
  - A robust process to fulfill differentiated customer needs
  - Choice navigation capability to support customers in identifying their own solutions



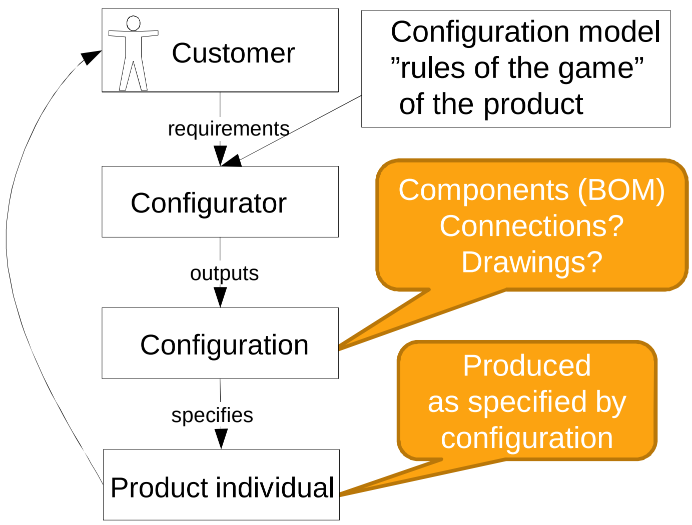
Configurable products

- Pre-designed to meet a given range of different customer requirements (*product family*)
- Each delivered product individual is tailored to the individual needs of an individual customer
- Each product instance is specified as a combination of pre-designed components or modules.
- The product has a pre-designed general structure or architecture or a set of these
- There is no need to do creative design as a part of the sales-delivery process



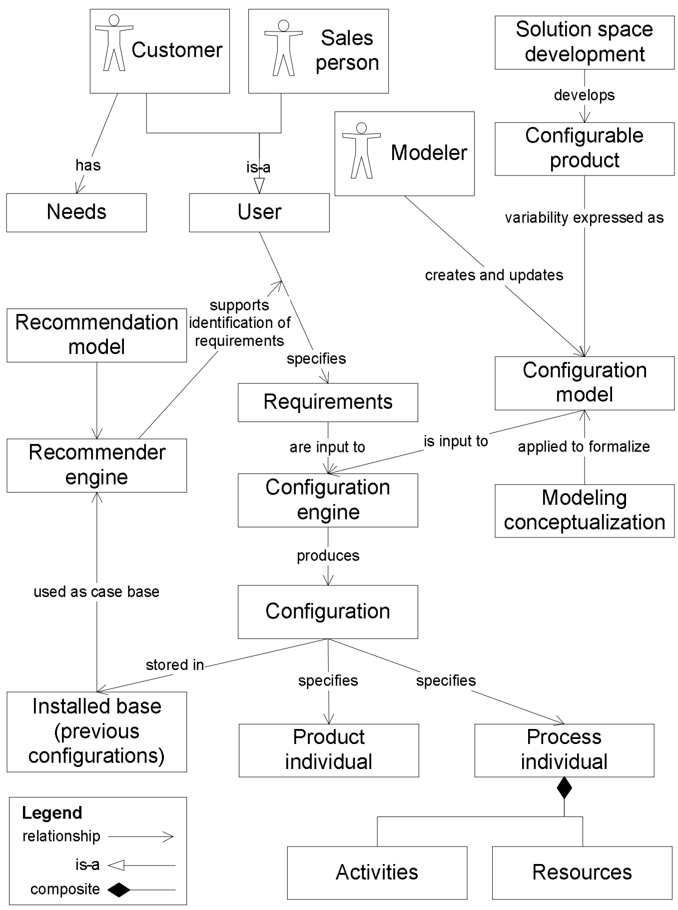
(Tiihonen & Soininen, 1997)

Configurator



- Software system that supports efficient and errorless specification of individualized products or services
- *Configurators* enable representing the variability of configurable products by creation and management of *configuration models*
- A tool for configuration modeling
- Another tool for the customer or sales person
- Customer receives a product individual according to the specification

Core concepts and their major relations

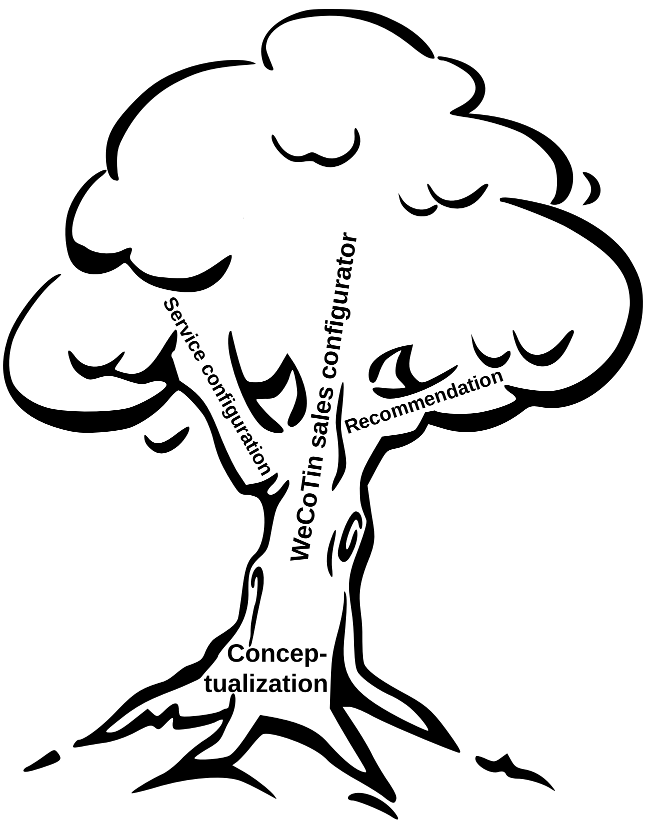


Research questions

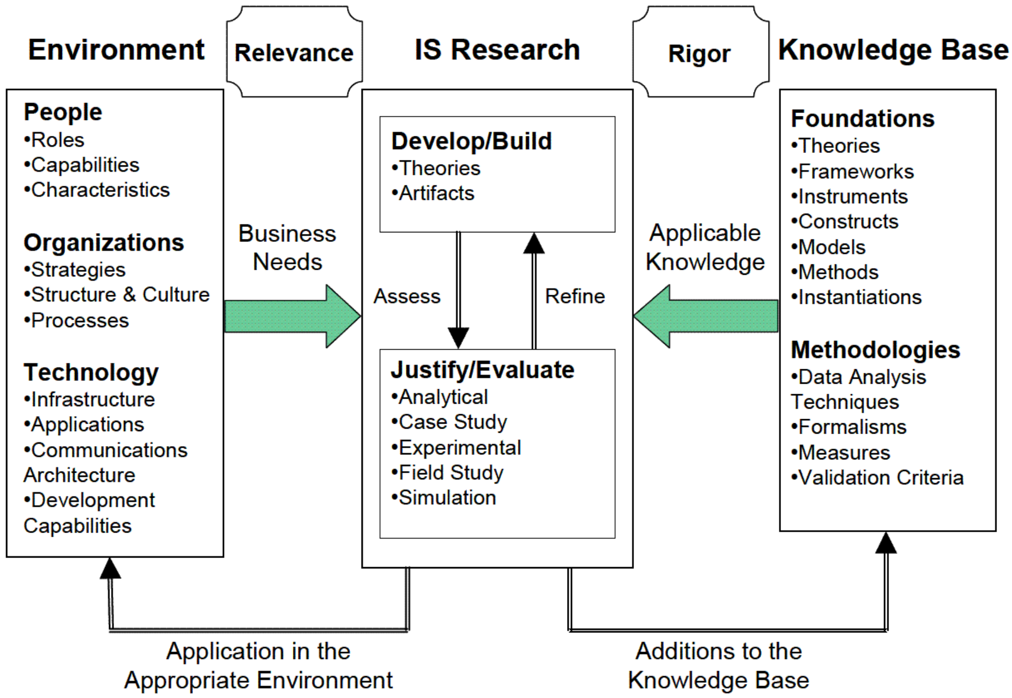
- RQ1: What are the concepts central to configuration knowledge?
- RQ2: How to construct a practical and computationally well-founded sales configurator?
- RQ3: Can users be effectively supported in finding suitable products and services with personalized recommendations?
- RQ4: How does service configuration differ from the configuration of physical products?

Overview

- IT tools to support business based on mass customization
- Conceptual model for representing configuration models
- Novel configurator WeCoTin
- How does service configuration differ from the configuration of physical products?
- How to apply recommendation technologies that support selection making for/in context of configurable products?
- Sales configurator information systems design theory



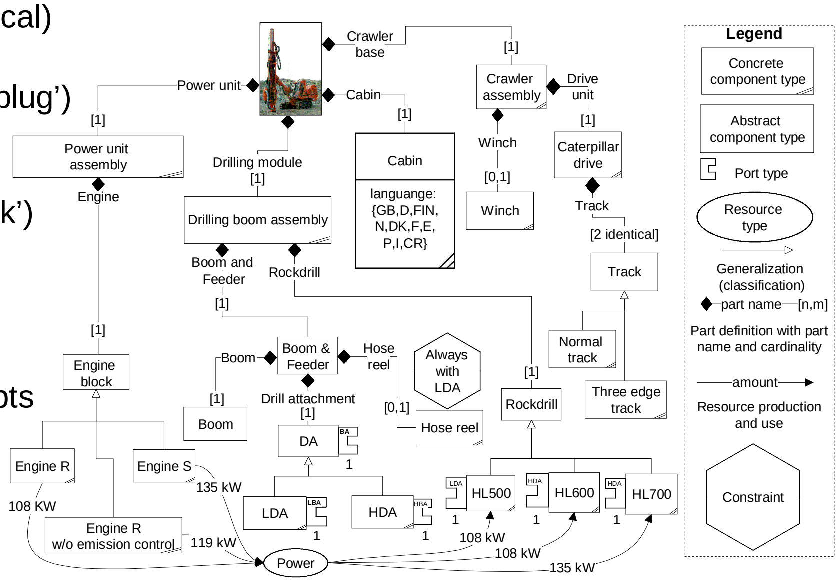
Method: Design Science



(Hevner, March, Park, & Ram, 2004, redrawn)

RQ1: What are the concepts central to configuration knowledge?

- Modeling conceptualization describes ‘the rules of the game of product’
- **Components**: physical (or logical) building blocks
- **Ports**: connections (‘socket & plug’)
- **Resources**: balance production and consumption (‘big enough fuse’, ‘enough disk’)
- **Functions/feature**s
- **Constraints**
- **Attributes** for main concepts
- **Classification** for main concepts
- **Compositional structure** (components, functions / features)

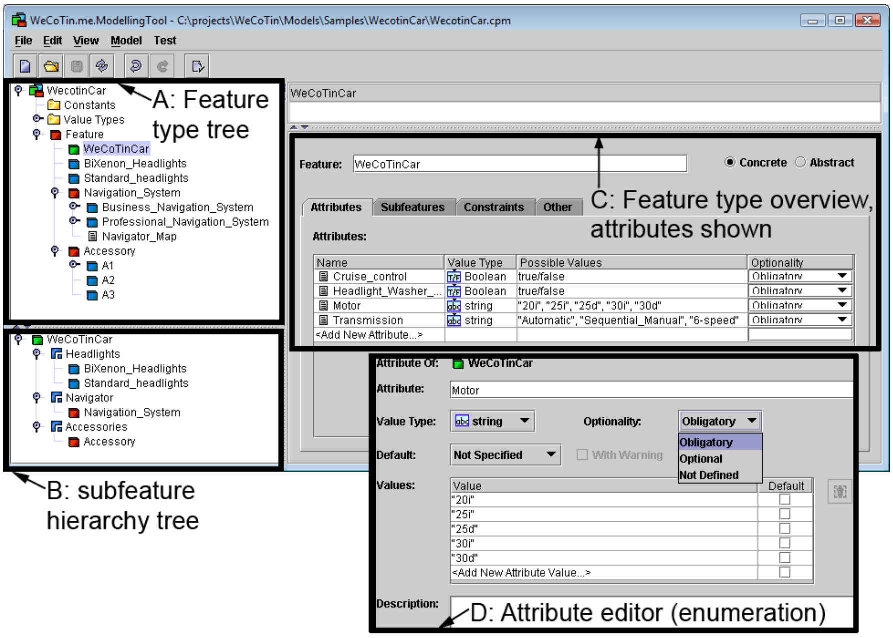


(I, Tiihonen, Lehtonen, Soininen, Puikkinen, Sulonen & Riitahuhta, 1999)

RQ2: How to construct a practical and computationally well-founded sales configurator?

WeCoTin Modeling tool

- Modeling conceptualization enables to describe 'the rules of the game of product' graphically
- In addition, textual language Product Configuration Modeling Language (PCML)
- Automatic translation to a logic program
  - Answer set programming
- Tried with over 20 products



(II)

Generated Smodels program (WCRL, ASP)

```
in(C2) :- pa(C1,T,C2,Pn), ppa(T,C1,C2,Pn).
:- 2{pa(C1,T,C2,Pn):ppa(T,C1,C2,Pn)}, compT_Feature(C2).

isa(X,Z):- isa(X,Y), isa (Y,Z),  compTDom(X), compTDom(Y), compTDom(Z).
isa(X,X):- compTDom(X).

isa(compT_Navigation_System,compT_Feature).
compTDom(compT_WeCoTinCar).
compT_Feature(C) :- compT_WeCoTinCar(C).
isa(compT_WeCoTinCar,compT_Feature).
compTDom(compT_Standard_headlights).
compT_Feature(C) :- compT_Standard_headlights(C).
isa(compT_Standard_headlights,compT_Feature).
compTDom(compT_BiXenon_Headlights).
compT_Feature(C) :- compT_BiXenon_Headlights(C).

1{in(C):compT_WeCoTinCar(C)}1.

pan(part_Navigator).
0{pa(C1,compT_WeCoTinCar,C2,part_Navigator):ppa(compT_WeCoTinCar,C1,C2,part_N
avigator)}1 :- in(C1),compT_WeCoTinCar(C1).
:-
compT_WeCoTinCar(C1),pa(C1,compT_WeCoTinCar,C2,part_Navigator),ppa(T,C1,C2,pa
rt_Navigator), not ppa(compT_WeCoTinCar,C1,C2,part_Navigator).
ppa(compT_WeCoTinCar,C1,C2,part_Navigator) :-
compT_WeCoTinCar(C1),compT_Navigation_System(C2),
for(compT_WeCoTinCar,C1,C2,part_Navigator).
emptyPart(ind_compT_WeCoTinCar_1,part_Navigator) :-
in(ind_compT_WeCoTinCar_1), not
pa(ind_compT_WeCoTinCar_1,compT_WeCoTinCar,ind_compT_Professional_Navigation_
System_1,part_Navigator), not
pa(ind_compT_WeCoTinCar_1,compT_WeCoTinCar,ind_compT_Business_Navigation_Syst
em_1,part_Navigator).
```

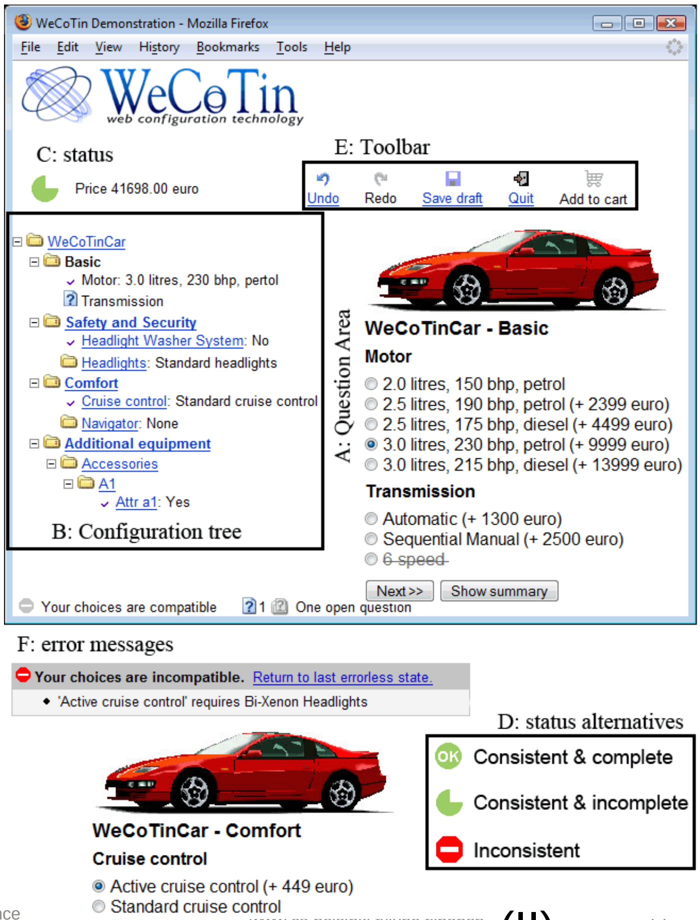
```
1{prop_WeCoTinCar_Motor(X,compT_WeCoTinCar,Y):prSpec_2(Y)}1 :-
in(X),compT_WeCoTinCar(X).
prSpec_2("20i").
prSpec_2("25i").
prSpec_2("25d").
prSpec_2("30i").
prSpec_2("30d").

% constraint WeCoTinCar:Manual_6_speed_Incompatible_with_3Litre
:- not sat(compT_WeCoTinCar, C1, constr_compT_WeCoTinCar_2),
compT_WeCoTinCar(C1), in(C1).
sat(compT_WeCoTinCar, C1, constr_compT_WeCoTinCar_2) :- not
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_6),compT_WeCoTinCar(C1).
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_6) :-
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_7),sat(compT_WeCoTinCar,C1,s
ubexpr_compT_WeCoTinCar_8),compT_WeCoTinCar(C1).
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_7) :-
prop_WeCoTinCar_Transmission(C1,T_C1,V_C1_prop_WeCoTinCar_Transmission),isa(T
_C1,compT_WeCoTinCar),prSpec_3(V_C1_prop_WeCoTinCar_Transmission),V_C1_prop_W
eCoTinCar_Transmission=="6-speed",compT_WeCoTinCar(C1).
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_8) :-
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_9),compT_WeCoTinCar(C1).
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_8) :-
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_10),compT_WeCoTinCar(C1).
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_9) :-
prop_WeCoTinCar_Motor(C1,T_C1,V_C1_prop_WeCoTinCar_Motor),isa(T_C1,compT_WeCo
TinCar),prSpec_2(V_C1_prop_WeCoTinCar_Motor),V_C1_prop_WeCoTinCar_Motor=="30i
",compT_WeCoTinCar(C1).
sat(compT_WeCoTinCar,C1,subexpr_compT_WeCoTinCar_10) :-
prop_WeCoTinCar_Motor(C1,T_C1,V_C1_prop_WeCoTinCar_Motor),isa(T_C1,compT_WeCo
TinCar),prSpec_2(V_C1_prop_WeCoTinCar_Motor),V_C1_prop_WeCoTinCar_Motor=="30d
",compT_WeCoTinCar(C1).
```

RQ2: How to construct a practical and computationally well-founded sales configurator?

WeCoTin Configuration tool

- Customer or sales person selects suitable alternatives
- Makes sure that all selections will be done and that it is possible to produce the specified product individual
- It is not possible to order incompatible features/alternatives
- Deduction of the consequences of selections
- Implementation utilizes logic-based AI techniques



(II)

A Taxonomy of Theory Types in Information Systems Research

Theory Type	Distinguishing Attributes
I. Analysis	**Says what is.** The theory does not extend beyond analysis and description. No causal relationships among phenomena are specified and no predictions are made.
II. Explanation	**Says what is, how, why, when, and where.** The theory provides explanations but does not aim to predict with any precision. There are no testable propositions.
III Prediction	**Says what is and what will be.** The theory provides predictions and has testable propositions but does not have well-developed justificatory causal explanations.
IV. Explanation and prediction	**Says what is, how, why, when, where, and what will be.** Provides predictions and has both testable propositions and causal explanations.
V Design and action	**Says how to do something.** The theory gives explicit prescriptions (e.g., methods, techniques, principles of form and function) for constructing an artifact.

S. Gregor, "The nature of theory in information systems," *MIS Quarterly,* vol. 30 (3), pp. 611-642, 2006

Components of Information Systems Design Theory

Component	ISDT component Description (Gregor & Jones, 2007).
Core components	
1) Purpose and scope	"What the system is for," the set of meta-requirements or goals that specifies the type of artifact to which the theory applies and in conjunction also defines the scope, or boundaries, of the theory.
2) Constructs	Representations of the entities of interest in the theory.
3) Principle of form and function	The abstract "blueprint" or architecture that describes an IS artifact, either product or method / intervention.
4) Artifact mutability	The changes in state of the artifact anticipated in the theory, that is, what degree of artifact change is encompassed by the theory.
5) Testable propositions	Truth statements about the design theory.
6)Justificatory knowledge	The underlying knowledge or theory from the natural or social or design sciences that gives a basis and explanation for the design (kernel theories).
Additional components	
7) Principles of implementation	A description of processes for implementing the theory (either product or method) in specific contexts.
8) Expository instantiation	A physical implementation of the artifact that can assist in representing the theory both as an expository device and for purposes of testing.

RQ2: How to construct a practical and computationally well-founded sales configurator?

Sales configurator information systems design theory (SCISDT)

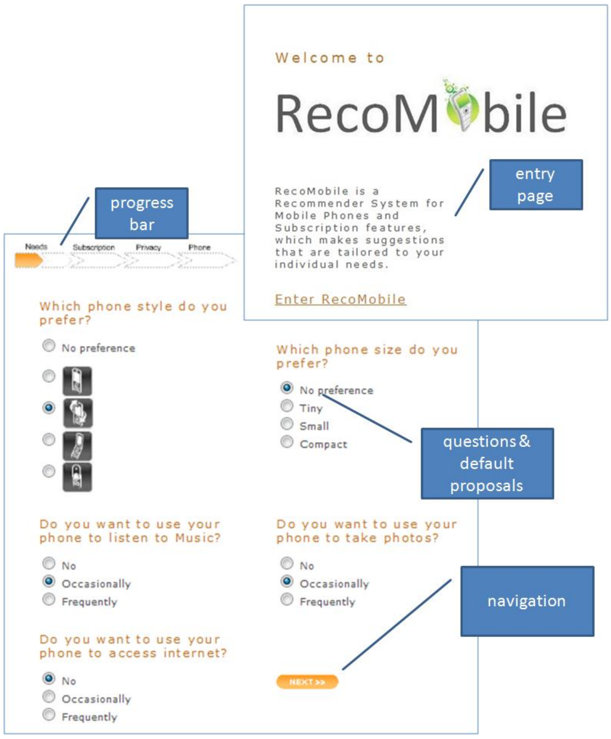
- An information systems design theory is a recipe-like description that makes it possible to construct an artefact described by the theory (Gregor & Jones, 2007)
- Ideas and experiences about WeCoTin configurator were distilled into a design theory
- This design theory is not a complete recipe

Component	SCISDT component description
1) Purpose and scope	A web-based sales configurator that fulfills a set of major requirements
2) Constructs	Concepts of configuration knowledge, product configuration modeling language PCML, weight constraint rule language.
3) Principle of form and function	A high-level architecture and main functions of components was presented along with main working principles
4) Artifact mutability	WeCoTin has several internal interfaces that enable replacement of major components. It has also been designed to be flexible in numerous aspects, such as different ways to determine prices, and support for several languages.
5) Testable propositions	The main propositions were capability to model and configure real products. Another proposition is adequate performance. These aspects were tested with highly satisfactory results.
6)Justificatory knowledge	The modeling constructs of PCML were given clear formal semantics by mapping them to the weight constraint rule language. This mapping also enables sound and complete inference by the Smodels system.

RQ3: Can users be effectively supported in finding suitable products and services with personalized recommendations?

Recommendation

- What recommendation techniques are applicable for configurable products?
- How to recognize alternatives that fit the user best?
- Extensions to recommendation techniques
- Utility of recommendation was examined empirically; N=546 users
  - were more satisfied with the recommendation-supported configuration process,
  - perceived that recommendation-supported configuration was clearly better in finding the best options
  - perceived that expectations regarding the solution were better fulfilled

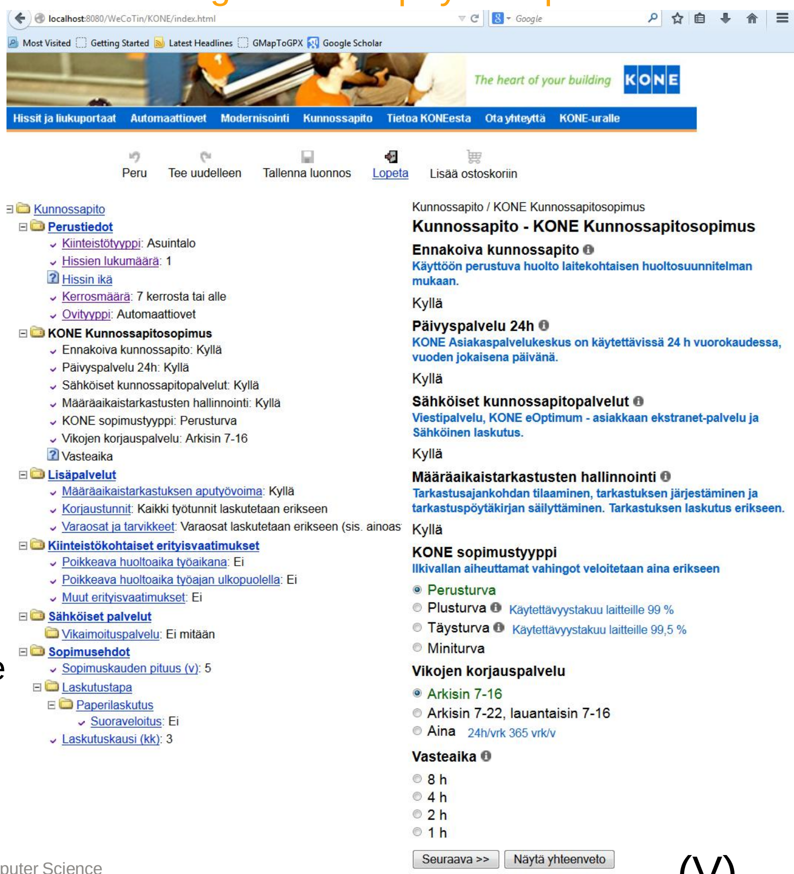


(III, IV; Felfernig, Mandl, Tiihonen, & Schubert, 2010)

Service configuration

RQ4: How does service configuration differ from the configuration of physical products?

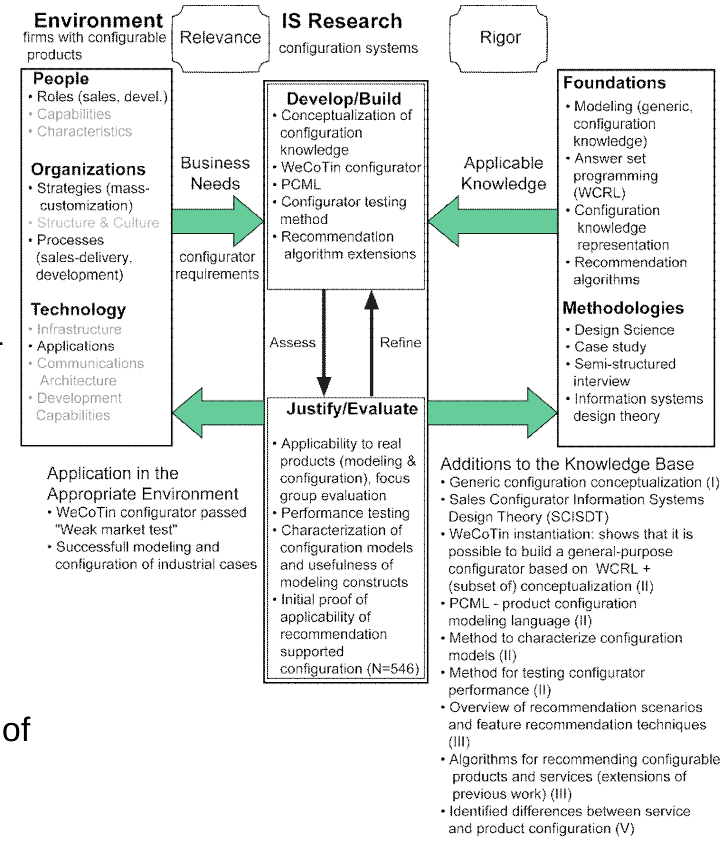
- Variability of services in three industries; contracts:
  - Maintenance contracts
  - Mobile phone & broadband subscriptions
  - Insurance
  - what is varied?
- Are configurators designed for physical products suitable for configuring services?
  - Yes, but with some concerns (e.g. conceptual match, service process modeling)



(V)

Contributions & Conclusions

- Purpose: advance practical support of configuration
- Sales configurator information systems design theory SCISDT
- Principles of WeCoTin implementation
- Modeling conceptualization
- Extensions of recommendation technologies
- Differences of configuration of services and physical products
- Methods to characterize configuration models and testing of configurator performance



Articles of the dissertation

I. Soininen, T., Tiihonen, J., Männistö, T. & Sulonen, R. (1998). Towards a general ontology of configuration. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* (*AI EDAM*), *12*(4), 357–72.

II. Tiihonen J., Heiskala M., Anderson A., and Soininen T. (2013). WeCoTin—a practical logic-based sales configurator. *AI Communications, 26*(1), 99–131. doi 10.3233/AIC-2012-0547.

III. Tiihonen J., Felfernig A. (2010). Towards recommending configurable offerings. *International Journal of Mass Customisation, 3*(4), 389–406.

IV. Felfernig, A. Mandl, M., Tiihonen, J. Schubert, M., and Leitner G. (2010). Personalized user interfaces for product configuration. 15th International Conference on Intelligent User Interfaces (IUI ’10), Hong Kong, China, 317–20. doi 10.1145/1719970.1720020.

V. Tiihonen J., Heiskala M., Paloheimo K.-S., Anderson A. (2006). Configuration of contract based services. 17th European Conference on Artificial Intelligence (ECAI 2006), configuration workshop, Riva del Garda, Italy, 25–30.

References

- Felfernig, A., Mandl, M., Tiihonen, J., & Schubert, M. (2010). Personalized Product Configuration. Paper presented at the *Multikonferenz Wirtschaftsinformatik 2010, 24. PuK-Workshop: Planung/Scheduling und Konfigurieren/Entwerfen,* Göttingen, Germany. 2251-2263.
- Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information Systems, 8*(5), 312-335.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.
- Pine, B. J. (1993). *Mass customization: the new frontier in business competition* (1st ed.). Boston, MA, USA: Harvard Business School Press.
- Salvador, F., de Holan, P. M., & Piller, F. T. (2009). Cracking the code of mass customization. *MIT Sloan Management Review*, 50(3), 71-78.
- Tiihonen, J., & Soininen, T. (1997). *Product configurators - information system support for configurable products*. (No. TKO-B 137). Espoo: Helsinki University of Technology.
- Tiihonen, J., Lehtonen, T., Soininen, T., Puikkinen, A., Sulonen, R., & Riitahuhta, A. (1999). Modeling configurable product families. *12th International Conference on Engineering Design (ICED'99),* Munich, Germany,. 1139-1142.

Thank you for your attention!

Questions?