Sales Configurator Information Systems Design Theory

Juha Tiihonen[1] & **Tomi Männistö**[2] & **Alexander Felfernig**[3]

[1]Department of Computer Science and Engineering, Aalto University, Espoo, Finland. juha.tiihonen@aalto.fi

[2]Department of Computer Science, Helsinki University, Helsinki, Finland. tomi.mannisto@cs.helsinki.fi

[3]Institute for Software Technology, Graz University of Technology, Graz, Austria. alexander.felfernig@ist.tugraz.at

Outline

- Background & motivation
- Research goal
- Design Science approach & ISDT
- Sales Configurator Information Systems Design Theory
- Future work
- Conclusions

Background & Motivation

- Significant scientific knowledge on different approaches and means for building configurators, e.g.
  - problem solving methods
  - modeling conceptualizations
  - procedures for implementing configurators (Hvam, Mortensen,&Riis 2008)
  - sound principles and requirements on user interaction e.g. (Abbasi, Hubaux, Acher, Boucher, & Heymans, 2013; Trentin, Perin, & Forza, 2013)
- Configurators (instantiations & frameworks) are numerous

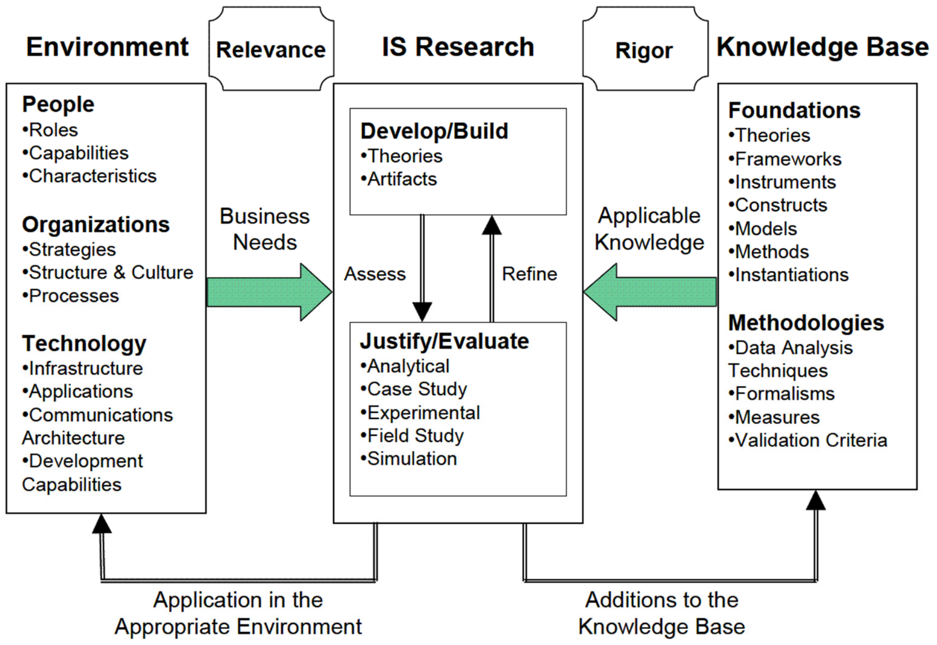
$\rightarrow$ scientific contributions require deeper principles / novelty

- Theories from the design perspective of generic configurator systems are still non-existent

Research goal: sales configurator ISDT

- Often configurator construction could have been conducted within a *Design Science* (Hevner, March, Park, & Ram, 2004) framework but have not been presented as such
- Possible: advance configuration systems via rigorous and methodological research with the aim of *theory creation: Information Systems Design Theory* (*ISDT*) (Gregor & Jones, 2007)
- The underlying idea is that an ISDT can be applied as a prescription when constructing similar artefacts
  - Not to be followed blindly, rather must be applied and interpreted
- Construction of the WeCoTin sales configurator as a basis for the theory and as an example for illustrating the theory

Information Systems Research Framework



Aalto University
School of Science

A. R. Hevner, S. T. March, J. Park and S. Ram, "Design science in information systems research," *MIS Quarterly,* vol. 28 (1), pp. 75-105, 2004.

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.

Aalto University
School of Science

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.

Aalto University
School of Science

Sales Configurator Information Systems Design Theory (SCISDT)

Component	SCISDT component description (as explicated by WeCoTin)
1) Purpose and scope	A web-based sales configurator that fulfills a set of major requirements
2) Constructs	Concepts of configuration knowledge [30], 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 [2, 65, 66]
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.

Aalto University
School of Science

WeCoTin sales configurator
(Tiihonen et al., 2003, 2013; Tiihonen, 2014)

- *WeCoTin Modeling Tool* for creating and editing configuration models (graphically), semi-automatic generation of user interfaces
- *Product Configuration Modeling Language (PCML)*.
  - object-oriented and declarative
  - conceptually based on a function-oriented subset of a configuration knowledge conceptualization (Soininen et al., 1998)
- *WeCoTin Configuration Tool* : configure products over the web using a standard browser
  - dynamically generated user interface for end users
  - computationally well founded: translation of configuration knowledge into weight constraint rules (ASP) (Soininen et al., 2001)

Purpose and scope

- Provide choice navigation capability for companies with a mass customization strategy
- Generic configurators, aka configuration toolkits, enable the creation of configurator instantiations for individual companies or product lines.
- Some requirements
  - easy set-up without programming (excluding integrations),
  - fluent modeling of products by product experts based on a well-founded high-level modeling conceptualization,
  - easy maintenance of configuration knowledge.
- Wanted to experiment with ASP for problem solving + high-level configuration modeling + consistent and complete inference

Constructs

- Somewhat challenging line between the constructs and principles of form and function
- Constructs include at least
  - conceptualization of configuration knowledge
  - object-oriented product configuration modeling language (PCML)
    - Compositional structure & configurable attributes are the main mechanisms for capturing variability
    - Taxonomy with (multiple)inheritance
  - A sales configurator (WeCoTin) as a whole and its major parts (Modeling Tool, Configuration Tool)
  - Underlying subsystems
    - inference engine Smodels (Simons, Niemelä, & Soininen, 2002)
    - its modeling language weight constraint rule language (WCRL),
    - and the method of translating configuration knowledge to WCRL [53].

Principle of form and function

- Layered architecture with a clear separation of
  - formal inference (here: ASP)
  - high-level modeling constructs
    - match how the product experts think of configuration,
    - can be provided with formal semantics,
    - can automatically mapped to a form suitable for inference
  - the end-user interface
    - creation of which does not require programming
  - A hierarchy of modeling languages to match the layered architecture
- The main functions include checking for the consistency and completeness of a configuration
  - Price is an integral element that must be managed

Artifact mutability

- Internal interfaces enable replacement of major components
  - E.g. Smodels could be relatively easily replaced
  - E.g. new user interfaces enabled by interfaces for configuration model manipulation and manipulation of configurations.
- Flexibility, e.g.
  - different ways to determine prices & taxes
  - support for several end-user languages
- Product changes without programming of end user interface
- Architectural mutability and suitability for generic tasks (e.g. dimensioning and connections) could be higher

Testable propositions (1)

- Capability to model and configure real products and adequate performance in this context
- 26 sales configuration models; characterized in terms of size and modeling constructs that were applied
  - 14 real-world products was modeled in their entirety; 8 partially
  - From 10 organizations representing machine industry, healthcare, telecommunications services, insurance services, maintenance services, software configuration, and construction
  - The created models were small, but representative of the Finnish industry
  - 'Broadband': 66 feature types, 453 effective attributes, 43 type level "generic" constraints
  - Linux: 626 feature types, 4369 'effective attributes', 2380 constr.

Tiihonen, J. (2010). Characterization of configuration knowledge bases. *ECAI-2010, Workshop on Intelligent Engineering Techniques for Knowledge Bases (IKBET)*

Testable propositions (2)

- WeCoTin had demonstrably adequate performance with the four models that were systematically tested.
- Configured all the characterized products using the WeCoTin user interface (Linux only partially)
  - feeling of instant response, except the “Broadband” model’s response time was slightly more than 3 seconds before an attribute with 436 possible values was specified, after which the response time decreased to less than a second.
  - Linux was too slow to be usable.
  - compilation time from PCML to WCRL and then to BCRL was very satisfactory: a script that compiled all the characterized configuration models, except Linux, and a few additional test and sample models ran in 32 seconds

Tiihonen, J., Soininen, T., Niemelä, I., & Sulonen, R. (2002). Empirical testing of a weight constraint rule based configurator. *ECAI-2002, Configuration Workshop,* 17-22.

Justificatory knowledge

- The underlying configuration knowledge conceptualization is a synthesis of previous work & additional experiences
- PCML allows the variability of products to be expressed on a high level that product experts can understand.
- Constructs of PCML have clear formal semantics via mapping to WCRL (except defaults)
- Sound and complete inference by the Smodels system

→ a working sales configurator can be built on the well-founded principles, and choice navigation support can be provided

Future

- More & improved Design Science theories in configuration!
- Future sales configurator ISDTs should address user interaction more thoroughly
  - E.g., capabilities to avoid the product variety paradox (Trentin et al., 2013): focused navigation, flexible navigation, easy comparison, benefit-cost communication, and user-friendly product-space description capabilities.
  - Support users in choice navigation with recommendations
- There is lots of practical work to do: many sales configurators even struggle on aspects like consistency checking! (Abbasi, Hubaux, Acher, Boucher, & Heymans, 2013).

Conclusion

- Constructed an ISDT for sales configurators (SCISDT)
  - based on the design of WeCoTin (Tiihonen et al. 2003, 2013)
- Using WCRL and Smodels to provide inference seems to be a feasible proposition for building a sales configurator
- The Design Science approach can potentially be applied in other configuration related contexts
  - can help to ensure the rigor and relevance of configuration research.
  - contributions can be additions to the knowledge base as suggested by Hevner et al. (2004), or (ISDT) theories (Gregor & Jones, 2007).