Modelling Ontology-Based Decision Rules for Computer-Aided Design

Ioana Ciuciu[1,2] and Yan Tang Demey[2]

[1] Laboratoire Informatique de Grenoble - University Joseph Fourier, France
[2] STARLab - Vrije Universiteit Brussel, Belgium
{iciuciu,yan.tang}@vub.ac.be

Abstract. A challenge in computer-aided design is how to parameterize shape functions in order to model a desired shape. Such design phases often include both systematic design and user-oriented design. In the paper, we want to focus on the latter case by bringing ontology-based decision rules to a computer-aided design system. Domain specific constraints and operative rules will be modelled in an artefact called semantic decision table.

Keywords: ontology, semantic decision table, decision table, computer-aided design.

1 Introduction

Computer-aided Design (CAD) is one of the fields where decision-making is intensively used in processes involving computer-human interactions. In this context, a challenge is how to create intuitive modeling tools in order to support and guide the modeler during the design phase, to ensure that the best design decisions are taken starting early in the design process. Therefore, a major aspect to be taken into account is how to integrate decision-making with computer-aided design tools.

This paper discusses the extension of an existing intuitive Computer-Aided Design system by integrating semantic decision tables with the purpose of enhancing the human-computer interaction. In our previous work [1], we have introduced a representation of the user-specific knowledge by means of ontologies and its major contribution to the computer-human interaction. Ontologies are used to store the semantics of the CAD model at five architectural levels, from business-level (non-expert level) to technical geometric level (expert level). We use Semantic Decision Table (SDT), which is a decision table enhanced with ontology technologies, to assist the user in specifying domain-specific design rules in a user-friendly approach. The proposed concept is illustrated on a case study related to the insertion of dies in a car parcel shelf while respecting the volumetric constraint, among others.

2 Background

A Semantic Decision Table (SDT [2]) is a decision table containing well specified meta-data and meta-rules. Its decisional environment and settings are studied in the

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 566–572, 2013.

specifications. It allows decision makers, rule modelers, knowledge engineers or evaluators to analyze a decision table using domain ontologies. It is also possible to assist a group of experts or stakeholders to draw a group decision.

An SDT consists of a set of formal agreements called commitments, grounded on domain ontology, and specified by a community of business stakeholders (domain experts). A commitment specifies how to use a binary fact types defined in the ontology. It can be 1) instantiation of a concept or a binary fact type, 2) a constraint, 3) selecting/grouping binary fact types from one or several contexts, 4) instantiation of a value for a concept if its value range is defined in a constraint, 5) articulation, which is a mapping between a concept and the glosses defined in a glossary, dictionary and thesaurus, 6) interpretation and implementation of role pairs, and 7) alignment of concepts within/across contexts.

3 Use Case Scenario

Our decision-based approach is demonstrated on a use case scenario from the automotive industry around the computer-aided design of a car parcel shelf. In this scenario, the user virtually interacts with the surface of the car parcel shelf in order to arrive to the desired shape, by applying dies onto the surface at various abstraction levels, according to his expertize [1] (see Fig. 1).

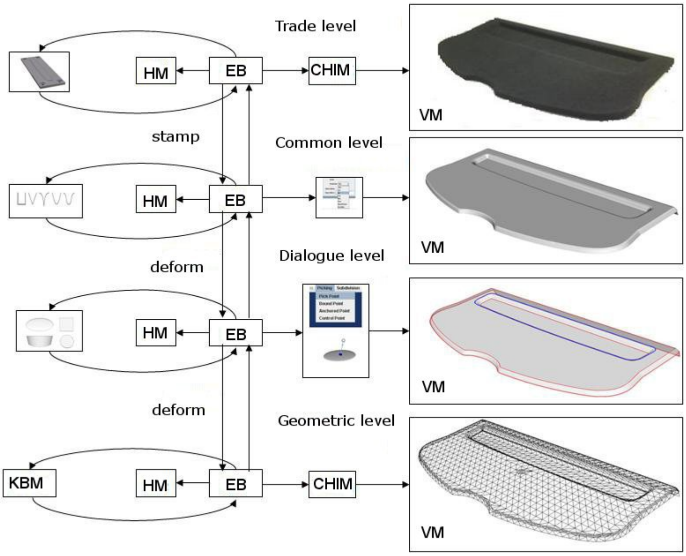

Fig. 1. Scenario: Die insertion on a car parcel shelf

This operation is done according to a specific parameterization, which integrates application-specific constraints related to spatial limitations and material resistance (e.g., the minimum distance between two dies is 2 cm when the dies diameter is less than 10 cm). These operations require intensive decision-making and reasoning, since for every user-system interaction multiple constraints must be verified and enforced, according to the user design intent.

For the concrete example in this study, the user interactively applies dies on the surface of the parcel shelf, using the virtual tools available in the system and an intuitive visualization technique. However, the dies must not exceed a certain number and their dimensions must respect the parameters specified at the beginning of the deformation, according to well-established business rules. The specific parameters are illustrated in Table 1.

Table 1. Parameters for the insertion of dies on the surface of a car parcel shelf

Parameter	
maximum number of dies allowed	$N_{\max}$
distance between 2 dies	$d_{\max}$
maximum volume of a die	$V_{\max}$
maximum height (depth) of a die	$h_{\max}$

In case the user does not respect the above-mentioned constraints (parameters) while inserting a die on the surface, the system will propagate the deformation either upwards, or on the die's neighboring surface (if possible).

Let us describe the case when the user makes a deformation which does not respect the height constraint. Suppose, in the first case, that the user selects a height h for the die which exceeds the maximum allowed height, $h_{\max}$ (i.e., $h > h_{\max}$). In this case, the system must make several pre-computations in order to decide what action to take. Therefore, the system first computes the volume (i.e., v) of the die corresponding to the height indicated by the user. If the volume is inferior to the maximum volume allowed for a die (i.e., $v \le v_{\max}$), then this volume is imposed (fixed) by the system and the deformation iteratively propagates on the non-deformable neighboring zone on the surface until the height converges towards the maximum allowed value (i.e., $h \rightarrow h_{\max}$).

If on the contrary, for the second case, the volume computed for the selected height exceeds the maximum volume allowed (i.e., $v > v_{\max}$), then the system imposes (fixes) the maximal height $h_{\max}$ and the deformation propagates on the neighboring zone on the surface until the volume converges towards the maximal volume allowed (i.e., $v \le v_{\max}$).

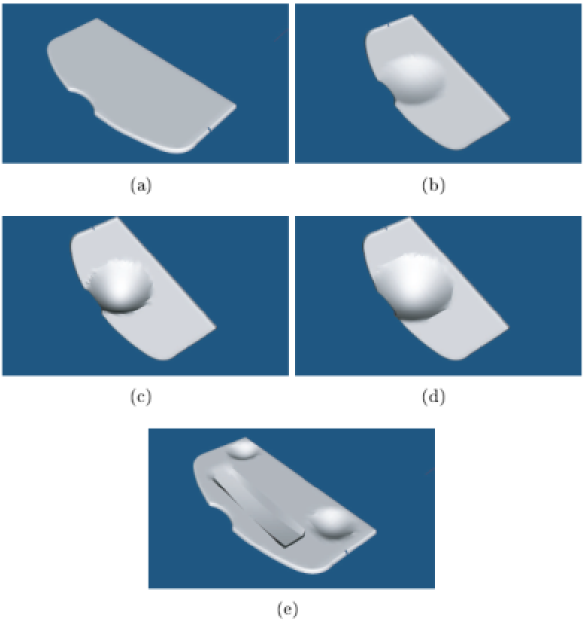

Fig. 2. Design of a car parcel shelf obtained by applying dies on the initial surface; (a) initial parcel shelf (courtesy of FAURECIA Acoustics and Soft Product Line Mouzon R&D Center, France); (b) die style when $h \leq h_{\max}$; (c) die when $h = h_{\max}$; (d) die style "bombed" when $h > h_{\max}$; (e) example of parcel shelf obtained by insertion of three dies with different contours while respecting the specified constraints

The two cases are illustrated in Fig. 2. In the following section, we will discuss how semantic decision tables can be used in this scenario.

4 An SDT Approach for CAD

An SDT contains a decision table, a set of lexons and a set of commitments [3,4]. The decision tables of the corresponding SDTs that are designed for the deformation function are illustrated in the following tables, namely SDTable 1, SDTable 2 and SDTable 3. Table 5 contains an extract of the binary fact types (lexons) that are used to understand the decision tables (see [1,3,4] for more details on the ontology engineering formalism). More precisely, it provides the formal representation of the deformation tools at different architectural levels of the CAD system.

Based on the above fact types, we present in Table 6 the rules that compose the Commitment layer. These rules represent application axiomatizations, where the application is the insertion of a parcel shelf using a "Thermoforming" deformation tool.

Table 2. ID: SDTable 1

Condition	1	2	3	4
h	$> h_{max}$		$\leq h_{max}$	
v	$> v_{max}$	$\leq v_{max}$	$> v_{max}$	$\leq v_{max}$
Action	*	*		
Compute v		*		
Impose v		*		
CALL SDTable 2				
Impose h_{max}	*			
CALL SDTable 3	*			

Table 3. ID: SDTable 2

Condition	1	2
$h > h_{max}$	Yes	No
Action		
Propagate deformation	*	
Call SDTable 2	*	

Table 4. ID: Table 3

Condition	1	2
$v > v_{max}$	Yes	No
Action		
Propagate deformation	*	
Call SDTable 3	*	

Table 5. Lexons representing the 3D surface deformation tools [1]

Context (archit. level)	Head Term	role	co-role	Tail Term
Trade	Thermoforming	deforms	is deformed by	Surface
Trade	Thermoforming	has	characterizes	Height
Trade	Thermoforming	produces	is produced by	Loading Zone
Common	Die Stamp	deforms	is deformed by	Surface
Common	Die Stamp	has	characterizes	Shape
Common	Die Stamp	has	characterizes	Height
Common	Die Stamp	has	characterizes	Volume
Dialogue	Deformation	affects	affected by	Surface
Dialogue	Deformation	has	characterizes	Contour
Dialogue	Deformation	has	characterizes	Initial Point
Dialogue	Deformation	has	characterizes	Target Point
Dialogue	Deformation	has	characterizes	Height
Dialogue	Deformation	has	characterizes	Volume

Table 6. Commitments based on the binary facts (in *Italic*) from Table 1

Abstraction Level	Rule
Trade level	Every *thermoforming* has exactly one maximal *height*.
Common level	Every *die stamp* has exactly one maximal *height*.
Common level	Every *die stamp* has exactly one maximal *volume*.
Dialogue level	The *contour* of the *deformation* lies on the *surface*.
Dialogue level	The *initial point* lies on the *surface*.
Dialogue level	The *initial point* lies inside the *contour*.

5 Related Work

One advantage of using SDTs is that we can easily validate and verify (V&V) decision rules using domain semantics. There are a few existing V&V approaches for decision tables. Shwayder [5] proposes combining decision columns in a decision table in order to reduce redundancies. Pooch [6] illustrates a survey on decomposition and conversion algorithms of translating decision tables in order to check for its redundancy, contradiction and completeness.

Vanthienen et al. [7] illustrate using PROLOGA5 (a decision table tool) to discover the intra-tabular anomaly, which is caused by a cyclic dependence between a condition and an action, and inter-tabular anomaly, which is caused by redundancy, ambivalence and deficiency. Qian et al. [8] use the approach of approximation reduction to managing incomplete and inconsistent decision tables. Incomplete and inconsistent decision tables are reduced into complete and consistent sub tables.

Compared to their work, our approach is focused on using ontological axioms as the meta-rules for validating a decision table. As the ontology is shareable and community based, the SDT validation process thus supports group activities in a natural way. Decision modelers and rule auditors share their common view through this process. By doing so, misunderstanding is minimized and the cost is consequently reduced.

6 Conclusion

In this paper, we demonstrate how an SDT can be used to unambiguously model and parameterize decision rules for a computer-aided design system. The advantages of using an SDT-based approach are multiple: 1) it is community-grounded and therefore enables knowledge (decision rules) sharing in highly collaborative contexts (e.g., for CAD); 2) it closes the gap between domain knowledge (ontologies) and end users (rule modelers); and 3) it ensures consistency and completeness of the decision rules.

Future work includes the integration of an existing SDT-based visualization client with the CAD system for setting the user's design preferences (parameters). Usability tests are also envisaged in order to evaluate the usability of SDTs. This implies extending the real-world industrial scenario presented here to be tested with groups of end users.

References

1. Ciuciu, I., Meersman, R., Perrin, E., Danesi, F.: Semantic Support for Computer-Human Interaction: Intuitive 3DVirtual Tools for Surface Deformation in CAD. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 645–654. Springer, Heidelberg (2010)
2. Tang, Y., Meersman, R.: Towards directly applied ontological constraints in a semantic decision table. In: Olken, F., Palmirani, M., Sottara, D. (eds.) RuleML - America 2011. LNCS, vol. 7018, pp. 193–207. Springer, Heidelberg (2011)
3. Meersman, R.: Ontologies and databases: More than a Fleeting Ressemblance. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS, vol. 1609, pp. 30–45. Springer, Heidelberg (1999)
4. Spyns, P., Tang, Y., Meersman, R.: An ontology engineering methodology for DOGMA. Journal of Applied Ontology 3(1-2), 13–39 (2008)
5. Shwayder, K.: Combining decision rules in a decision table. Commun. ACM 18(8), 476–480 (1975)
6. Pooch, U.W.: Translation of decision tables. ACM Comput. Surv. 6(2), 125–151 (1974)
7. Vanthienen, J., Mues, C., Aerts, C.: An illustration of verification and validation in the modelling phase of kbs development. Data Knowl. Eng. 27(3), 337–352 (1998)
8. Qian, Y., Liang, J., Li, D., Wang, F., Ma, N.: Approximation reduction in inconsistent incomplete decision tables. Knowl. -Based Syst. 23(5), 427–433 (2010)