Vino4TOSCA: A Visual Notation for Application Topologies Based on TOSCA

Uwe Breitenbücher, Tobias Binz, Oliver Kopp,
Frank Leymann, and David Schumm

Institute of Architecture of Application Systems, University of Stuttgart, Germany
Universitätsstraße 38, 70569 Stuttgart, Germany
lastname@iaas.uni-stuttgart.de

Abstract. A major difficulty in enterprise computing is the modeling of complex application topologies consisting of numerous individual components and their relationships. Especially in the context of cloud computing, the Topology and Orchestration Specification for Cloud Applications (TOSCA) has been proposed recently for standardization to tackle this issue. However, TOSCA currently lacks a well-defined visual notation enabling effective and efficient communication in order to transport the semantics of the encoded information to human beings. In this paper, we propose a visual notation for TOSCA based on established usability research which provides additional concepts for visual modularization and abstraction of large application topologies.

Keywords: TOSCA, Modeling, Visual Notation, Application Topologies.

1 Introduction

Cloud computing enables significant benefits in terms of cost, flexibility, and scale compared to traditional IT. An important issue is the automation needed to achieve these advantages. The Topology and Orchestration Specification for Cloud Applications (TOSCA [9]) provides a well-defined way to model composite applications and to provide plans for automating their management [4].

Visual notations enable an effective communication because of the powerful and highly parallel human visual system [7]. A well-designed visual notation eases the comprehension of the content structure and enables an easier navigation. In addition, visual notations are generally easier to learn and can be remembered faster than textual syntax. Thus, they are appropriate to complement languages which only provide textual notations such as XML. This leads to a separation of concerns: Visual notations are used for fast and effective communication, the original notation for the actual purpose and more detailed information processing. However, TOSCA does not specify a visual notation to map the language constructs to visual elements. The lack of a visual notation in other specifications resulted in a number of different graphical renderings of the same model. One

R. Meersman et al. (Eds.): OTM 2012, Part I, LNCS 7565, pp. 416–424, 2012.

example is the Business Process Execution Language (BPEL) [8] for which different approaches to visualize elements regarding their shapes, icons, layout, etc. exist. This becomes a problem when diagrams must be communicated between people using different representations as it might lead to misunderstandings and wrong interpretations. Thus, we advocate using only one common visual notation in addition to the actual notation provided by the original language.

A major problem of existing visual notations is that they currently correspond to what Alexander [1] calls an unselfconscious design culture: The design rationales are not based on explicit design principles. They are based on instinct, imitation, and tradition. As a consequence, many visual notations, like UML, focus on semantics and lack an explicit design process for the visual syntax which results in problems decreasing the usability of the notation. Therefore, we present a Visual Notation for TOSCA (Vino4TOSCA) which is based on an explicit requirements analysis regarding human cognition, usability, ergonomic influences, and evidence-based principles. The remainder of the paper is structured as follows: In Sect. 2 we describe the fundamentals and related work whereon the requirements for the notation in Sect. 3 are based. Section 4 describes the notation and Sect. 5 concludes the paper and gives an outlook on future work.

2 Fundamentals and Related Work

TOSCA [9] is a language to formally describe cloud applications and their management. The structure of an application is captured by a so-called Topology Template, a graph with Node Templates and Relationship Templates, serialized in XML. Node Templates represent the components of an application, for example, an "application server". Relationship Templates define how a particular node relates to another node, for example, the "application server" node is "hosted on" an "operating system" node. Templates are typed with Node Types and Relationship Types respectively. Types define the meaning of the nodes and relationships by specifying their properties and states of their lifecycle. TOSCA additionally defines policies on nodes, management operations provided by the node, and deployment artifacts implementing the functionality of the node. TOSCA does not define concrete Node Types and Relationship Types as it only provides a way to model them and to compose templates of several individual types into a topology. Therefore, the modeler of an application is able to define new Node Types and new Relationship Types. Node Types can be provided by software vendors as building blocks to simplify the integration of their products into cloud applications. The operational aspects, key for each automated cloud environment, are captured in so-called plans which are workflows capturing the management tasks of an application. The management operations defined by nodes are orchestrated by plans into higher level management functionalities, like deploying or scaling up the application. This enables software developers to model their management knowledge and experience explicitly into these plans which enables operating an application without having all the deep technical knowledge required before.

Moody [7] contributed a design theory, called "The Physics of Notations", focusing on the physical perceptual properties of notations regarding human capabilities. The principles defined in this design theory are synthesized from theory and empirical evidence. They are based on a theory of how visual notations communicate and provide the basis for the development of Vino4TOSCA.

Existing enterprise architecture modeling languages and notations such as Acme [5] do not provide a visual notation which can be used for TOSCA, because they do not fulfill all requirements we introduce in Sect. 3 and consider as absolutely necessary. In addition, the concept of managing applications by plans differentiates TOSCA fundamentally from other application modeling languages and thus needs special consideration.

3 Requirements Analysis

In this section we present requirements and design principles on the visual notation which are necessary for an effective usage. They have been identified, discussed, and validated by TOSCA users and members of the OASIS TOSCA Technical Committee. In the following sections we use the symbol Rx, with x being the number of the reference, to reference a certain requirement.

The prescriptive component in [7] defines nine principles for designing cognitively effective visual notations to increase speed, ease, and accuracy with which information can be understood by humans: R1 Semiotic Clarity, R2 Perceptual Discriminability, R3 Semantic Transparency, R4 Complexity Management, R5 Cognitive Integration, R6 Visual Expressiveness, R7 Dual Coding, R8 Graphic Economy, and R9 Cognitive Fit.

TOSCA-related requirements address the semantic constructs to obtain a tailored notation: R10 Completeness (all information contained in Topology Templates must be representable), R11 Semantic Correctness (a valid Vino4TOSCA diagram has to be a representation of a valid TOSCA Topology Template), R12 Extensibility (be extensible to show additional information), R13 Compact Representation (support compact visual representation to tackle space problems).

The following requirements shall improve usability and user experience to achieve a broad acceptance and user satisfaction. They are inspired by usability standards (e. g., EN ISO 9241) and [10]. R14 Suitability for the Task (optimization for modeling TOSCA Topology Templates), R15 Self-descriptiveness (diagram and graphical symbols describe their meaning themselves), R16 Simplicity (graphical elements must be easy and fast to draw), R17 User Satisfaction (account for human preferences and enable visually appealing designs).

4 The Notation

Vino4TOSCA covers the modeling of TOSCA Topology Templates by Topology Template Diagrams which mainly consist of Node Templates, Relationship Templates, and Groups (R11, R14). Modeling of plans is not part of the notation as there are already existing languages and notations available (e. g., BPMN).

The notation allows defining profiles which are domain-specific visual languages devised for specific needs, knowledge, and capabilities of users in a certain application domain [6] (R9, R17). The main advantage of profiles is that tailoring enables a strong cohesion to the domain properties and being effective and intuitive for the task to be performed, e. g., a "Whiteboard Profile" defines how to draw a Topology Template Diagram effectively by hand while an "Electronic-Design Profile" can be used for creating a modeling software supporting more details (R14). Profiles are allowed to constrain, but not to structurally modify the notation or change its basic shapes. The notation provides visual variability for profiles as described in Sect. 4.1 (R8), e. g., profiles may forbid the usage of groups or define line colors. Some shapes offer Additional Information Areas which can be used by profiles to add any information which is not natively reflected by the notation (R10, R12, R15). These areas may contain any text or graphic. Nevertheless, a profile has to regard the requirements defined in Sect. 3, too.

To reduce the complexity of large diagrams, the notation must provide mechanisms to group multiple elements visually into one single element. TOSCA Group Templates are not sufficient as they are applied at the TOSCA model level. Thus, grouping would influence the effective application topology. Therefore, the notation provides two additional concepts for visual grouping which are not part of TOSCA itself: Visual Group and Visual Relationship Group. Both may reduce complexity as they enable visual modularization and abstraction as well as reducing the number of symbols by collapsing (R4, R13). The Visual Group may also be used for integrating external diagrams homogenously (R5).

4.1 Visual Design Rationales

As the basic shapes of the notation must not be changed by profiles (R1) and there is a need for a hand-drawable "Whiteboard Profile", the notation has to tackle human drawing issues (R16). We decided to use rounded shapes wherever possible because of human drawing skills and preferences [2].

The notation uses the eight elementary visual variables of the Design Space defined by Bertin [3] to visually encode information: Horizontal and vertical position, shape, size, color, brightness, orientation, and texture (R6). They are classified into three categories: (i) Fixed variables defined by the basic notation, (ii) constrained variables defined by profiles, and (iii) free variables. While the first two ones are defined strictly and must be followed when applying the notation (limitations and constraints), free variables can be used for individual modeling of concrete diagrams (points of variability). The fixed visual variables are the retinal variables shape, orientation, and texture of lines: The basic shapes, their surrounding lines, and orientation are defined strictly and must not be changed by profiles or instantiation. All other variables are free if they are not constrained by a profile. Thus, if a profile does not constrain the visual variable color it is also a free variable and different colors can be used wherever the basic notation allows it. A profile is allowed to constrain the retinal variables color, value, texture, and size for enabling a high value of cognitive effectiveness. Constraining variables by profiles enables adding new semantics for different tasks

and/or audiences by creating visual dialects. Generally free visual variables are the horizontal and vertical position of an element.

The basic notation employs icons for describing elements as they have a higher information density and need less space for information presentation than text (R3, R6, R7). Icons are recognized, processed, remembered, and learned more easily and faster than textual information and preferred to abstract shapes by humans [2]. They enable tailoring domain-specific visual languages by using different icons for each domain. All icons are placed on the top most left position of the visual element because humans spend most of their attention to this place.

Text fonts are not defined by the basic notation. This may be done by profiles as especially hand drawn profiles need this variability. To enable fast recognition of textual information, text used to identify semantic elements differs in visual appearance: A name is not decorated, an id is underlined, and the name or id of the corresponding element type is enclosed by two parentheses (R3, R7, R15).

The notation does not define shapes for all elements of TOSCA, e. g., there are no shapes defined for types (e. g., Node Types). This can be modeled as additional information contained in the Additional Information Areas of the template shapes.

4.2 Shapes

This section defines the visual syntax of the notation, i. e., the visual elements and shapes. For each visual element we describe its shape in terms of form, contained information, semantics, variability points, and visual design freedom.

The Node Template Shape shown in Fig. 1 represents the Node Template as a rectangle with rounded corners surrounded by a solid line. There are five possibilities to describe the corresponding Node Template: (i) An icon contained in the Icon Area may represent the Node Template or the corresponding Node Type, (ii) using name or (iii) id of the Node Template to identify it, (iv) using name or (v) id of the corresponding Node Type to identify its type. A valid Node Template Shape contains at least one of these five information items. If multiple textual variants are combined, the visual order is given by the vertical ordering in Fig. 1 (R7, R13). The Additional Information Area is a rounded rectangle surrounded by a solid line which may be optionally attached below the main shape to provide any additional visual information (variability point). It is positioned behind the main shape so that both upper corners are hidden, as depicted in Fig. 1. The Icon Area is allowed to contain any graphic or symbol. The main shape is allowed to contain any graphic as background image, i. e., behind the Icon Area and the text blocks. Thus, various designs are possible as well as monochrome-colored Node Template Shapes. The surrounding lines are allowed to be colored, but the solid style must not be changed (R1).

The Relationship Template Shape shown in Fig. 2 represents the Relationship Template as a single line with a small shape at each end (shown as question marks in Fig. 2), e. g., an arrow. It visually connects any two *Relational Elements*, which are Node Template Shape, Collapsed Group Template Shape, and Collapsed Visual Group Shape. The semantics of a Relationship Template Shape sourcing

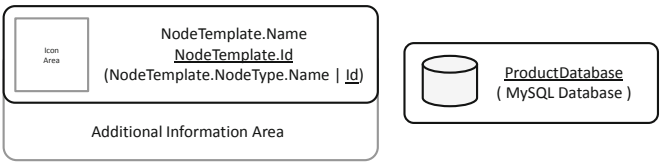

Fig. 1. Node Template Shape and example

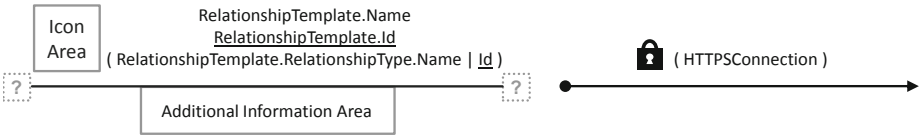

Fig. 2. Relationship Template Shape and HTTPSConnection example

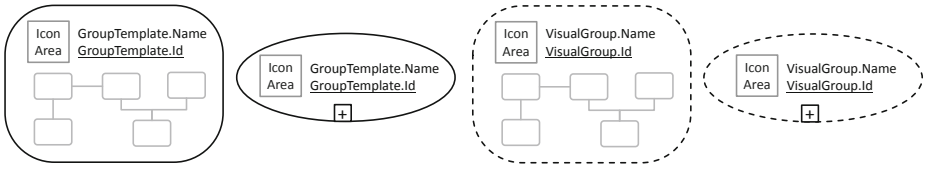

Fig. 3. Expanded / Collapsed Group Template Shapes and Visual Group Shapes

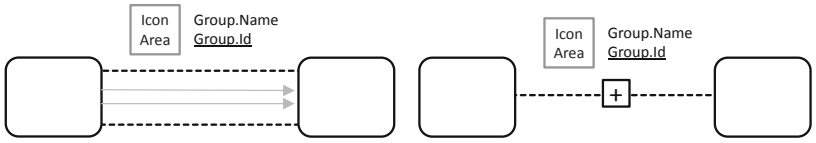

Fig. 4. Expanded and Collapsed Visual Relationship Group Shapes

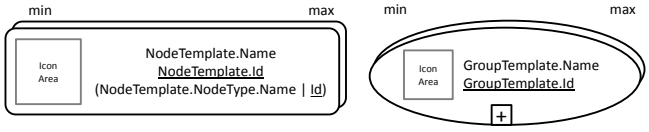

Fig. 5. Multiple instances of Node Template and Group Template

or targeting a Collapsed Visual Group Shape or Collapsed Group Template Shape is that it points to a hidden Relational Element inside the group. The basic notation neither defines a special line style nor shapes which may be used at the line endings. It is only prohibited to use the same dashed line style as the Visual Group Shapes (R2). The two shapes at the line endings may use any free visual variable. There are five possibilities to describe the corresponding

Relationship Template: (i) An icon contained in the Icon Area representing the Relationship Template or the corresponding Relationship Type, (ii) using name or (iii) id of the Relationship Template to identify it, (iv) using name or (v) id of the corresponding Relationship Type to identify its type (R7). The Icon Area is placed on the left of the textual information which is positioned above the line if it is horizontal or at any side if the line is vertical or diagonal. A valid Relationship Template Shape contains at least one of these five information items. The visual order of the elements and an HTTPSConnection example are shown in Fig. 2. The Additional Information Area is a rectangle surrounded by a solid line which may be attached below touching the line if it is horizontal or sideways otherwise in order to provide additional information about the template.

The Group Template is represented by two different shapes shown in Fig. 3 on the left: The Expanded Group Template Shape is a solid line surrounding elements which are grouped. There is no special shape defined. At the top left position there is the possibility to describe the Group Template Shape by a left-aligned icon or textually by using a name or id. All combinations of them are allowed but at least one has to be used. The collapsed Group Template Shape is an oval surrounded by a solid line. A small square with a plus sign positioned at the bottom center of the shape indicates the collapsed state hiding the contained elements. Inside the oval there must be at least one of the following: An icon, the name, or the id of the Group Template (R7). The free usage of the Icon Area and color of the surrounding line are the only free variables. The background of these two shapes must not be filled with any color or image.

The Visual Group Shapes shown in Fig. 3 on the right are equal to Group Template Shapes with the difference that the surrounding lines are dashed. The semantics is to group elements visually only. The shapes may also be used to represent the integration of other diagrams (R5): Its name or id is used to identify the integrated diagram, especially in collapsed state. The visual variability is equal to Group Template Shapes.

The two Visual Relationship Group Shapes shown in Fig. 4 are used to visually group and hide Relationship Template Shapes connecting Relational Elements. The expanded variant consists of two dashed lines between any two Relational Elements and at least contains two Relationship Template Shapes. The element can be described by a left-aligned icon or textually by using a name or id of the group above the lines. All combinations of them are allowed but at least one has to be used. The Collapsed Visual Relationship Group Shape (right shape in Fig. 4) is a dashed line with a small square containing a plus sign inside positioned at the center of the line connecting any two Relational Elements. The shape's semantics is that it visually groups and hides the Relationship Templates between two Relational Elements. Above the line there must be at least one of the following: An icon, the name, or id of the group (R7). The free usage of the Icon Area and the color of the lines are the only free variables which can be used to design these two shapes. The background must not be filled with any color or image.

TOSCA Node Templates and Group Templates have two attributes representing the number of allowed instances, e. g., multiple instances of a service component: min and max. Multiple instances are represented by drawing a second solid line partly covered by the original shape and writing the min value at the left and the max value at the right above the main shape as shown in Fig. 5.

5 Conclusion and Outlook

We presented a Visual Notation for TOSCA based on a scientific development approach taking the "Physics of Notation" theory and a well-defined requirements analysis into account. The presentation includes a visual model which explicitly defines the elements, relations, and representations. For the time being, an evaluation of the notation was not possible for the following reasons: First, TOSCA has been published quite recently and the users of this language are collecting experience with the language itself just now. Second, to judge the expressiveness of the visual model, one needs to really work with them. Therefore, we implemented the proposed visual notation in an open source web-based TOSCA modeling environment prototype[1] in the CloudCycle[2] project, which is one early adopter of TOSCA. After a broader usage, we will evaluate the notation using a questionnaire. In addition, we plan to evaluate several new profiles as well as to develop a new diagram type to integrate the proposed notation with process modeling notations such as BPMN. On the official Vino4TOSCA Web page[3] we present application examples, profiles, and the meta-model of the notation.

Acknowledgments. This work was partially funded by the BMWi project CloudCycle (01MD11023).

References

1. Alexander, C.: Notes on the Synthesis of Form. Harvard University Press (1964)
2. Bar, M., Neta, M.: Humans prefer curved visual objects. Psychological Science 17(8), 645–648 (2006)
3. Bertin, J.: Semiology of graphics. University of Wisconsin Press (1983)
4. Binz, T., Breiter, G., Leymann, F., Spatzier, T.: Portable Cloud Services Using TOSCA. IEEE Internet Computing 16(03), 80–85 (2012)
5. Garlan, D., Monroe, R.T., Wile, D.: Acme: Architectural Description of Component-Based Systems. In: Foundations of Component-Based Systems, pp. 47–68. Cambridge University Press (2000)
6. de Lara, J., Vangheluwe, H.: Defining visual notations and their manipulation through meta-modelling and graph transformation. J. Vis. Lang. Comput. 15(3-4), 309–330 (2004)
7. Moody, D.L.: The "physics" of notations: a scientific approach to designing visual notations in software engineering. In: ICSE, pp. 485–486 (2010)

[1] `http://www.cloudcycle.org/en/valesca/`
[2] `http://www.cloudcycle.org/en/`
[3] `http://www.vino4tosca.org`

8. OASIS: Web Services Business Process Execution Language Version 2.0 – OASIS Standard (2007)
9. OASIS: Topology and Orchestration Specification for Cloud Applications Version 1.0 Working Draft 07 (June 2012), http://www.tosca-open.org
10. Petre, M., de Quincey, E.: A gentle overview of software visualisation. Psychology of Programming Interest Group (PPIG) (September 2006)