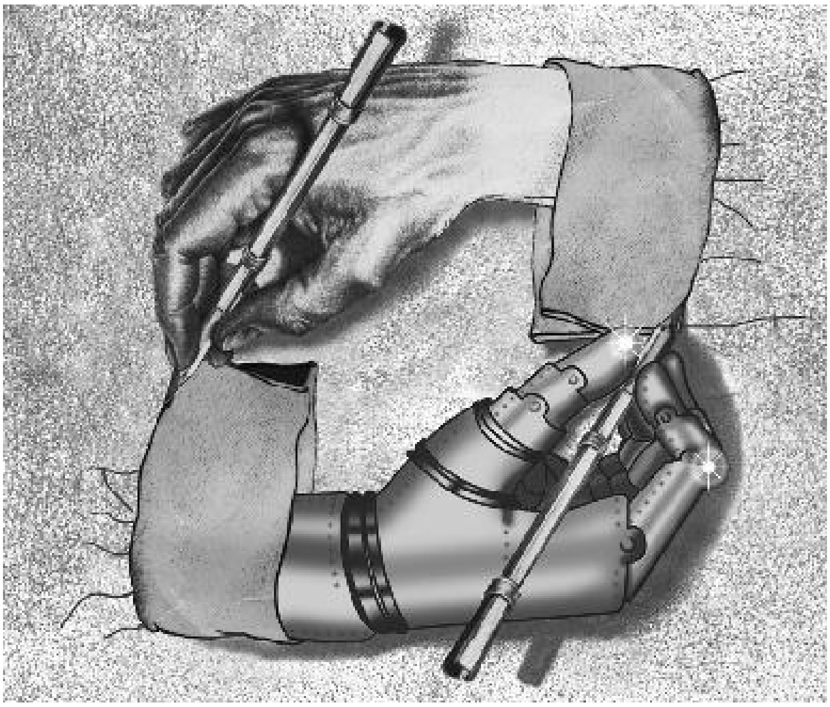
Francesca Alessandra Lisi, Stefano Borgo (Eds.)



AIDE 2015

Proceedings of the 1st Workshop on Artificial Intelligence and Design

Ferrara, Italy, September 22, 2015

Editors' address:
Università degli Studi di Bari "Aldo Moro"
Dipartimento di Informatica
Campus Universitario "E. Quagliariello"
Via E. Orabona, 4
70126 Bari, Italy
francesca.lisi@uniba.it

CNR-ISTC Trento
Via alla Cascata, 56 C
38123 Povo (Trento), Italy
stefano.borgo@cnr.it

Preface

This volume contains the papers presented at the 1st Workshop on Artificial Intelligence & Design (AIDE 2015, `http://www.di.uniba.it/~lisi/AIDE2015/`) held on September 22, 2015 in Ferrara, Italy, and organized by Stefano Borgo (ISTC-CNR Trento, Italy), Domenico Camarda (Politecnico di Bari, Italy) and Francesca A. Lisi (Università degli Studi di Bari "Aldo Moro", Italy).

Design is a fundamental activity in all human disciplines that require to decide in advance what to do. This encompass most of, if not all, the applied sciences: from engineering to architecture, from biology to computer science. Design is based on the assumption that we can positively improve our personal, social and cultural life by choosing how to change the environment in which we live. This approach has been proven successful enough to be applied to larger and larger parts of our world. We indeed live in a designed rather than a natural world, and spend most of our time in or surrounded by designed, that is, artificial environments (house, office, city, transportation system). Design research aims to develop an understanding of design both as a theory and as an activity, and to produce models that can be used to aid design.

The AIDE workshop provides a forum for researchers and practitioners interested in the interplay between AI and design via the presentation and discussion of state-of-the-art and cutting edge research, and developments of material that integrates these two areas.

The programme of AIDE 2015 included a keynote presentation, three technical sessions and a thematic panel.

The keynote talk, entitled "AI tools in the design process of industrial products", was given by Carlo Poloni, Professor of Mechanical Engineering at the University of Trieste, Italy, and co-founder and current President of ESTECO.

The technical sessions covered topics spanning from the ontological foundations of Design to applications of AI to Design in architecture and engineering. The workshop received 10 submissions, 8 of which as regular papers. After the review process, 7 papers were accepted for presentation.

The panel on the theme "Spatial Planning and Spatial Organizations: The challenges to AI" was chaired by Dino Borri, Professor of Urban Planning at the Politecnico of Bari, Italy, and involved the following experts:
- Amedeo Cesta (ISTC-CNR Roma, Italy)
- Grazia Concilio (Politecnico di Milano, Italy)
- Raffaele Giordano (IRSA-CNR Bari, Italy)
- Nicola Guarino (ISTC-CNR Trento, Italy)
- Giovanni Rabino (Politecnico di Milano, Italy)

AIDE 2015 was an event supported by the Italian Association for Artificial Intelligence (AI*IA) and co-located with the 14th Conference of the AI*IA (`http://aixia2015.unife.it/`).

October 1, 2015 | Francesca A. Lisi
Bari | Stefano Borgo

Programme Chairs

Borgo, Stefano	ISTC-CNR Trento, Italy
Lisi, Francesca Alessandra	Università degli Studi di Bari “Aldo Moro”, Italy

Programme Committee

Besold, Tarek Richard	University of Osnabrück, Germany
Bhatt, Mehul	University of Bremen, Germany
Camarda, Domenico	Politecnico di Bari, Italy
Colabella, Enrica	Politecnico di Milano, Italy
De Liddo, Anna	The Open University, United Kingdom of Great Britain and Northern Ireland
Dentsoras, Argyris	University of Patras, Greece
Esposito, Floriana	Università degli Studi di Bari “Aldo Moro”, Italy
Guarino, Nicola	ISTC-CNR Trento, Italy
Kutz, Oliver	Free University of Bolzano/Bozen, Italy
Mizoguchi, Riichiro	Japan Advanced Institute of Science and Technology, Japan
Pauwels, Pieter	University of Ghent, Belgium
Rabino, Giovanni	Politecnico di Milano, Italy
Renso, Chiara	ISTI-CNR Pisa, Italy
Schultz, Carl	University of Muenster, Germany
Stufano Melone, Maria Rosaria	Politecnico di Bari, Italy
Terkaj, Walter	ITIA-CNR Milano, Italy

Additional Reviewers

Sanfilippo, Emilio	ISTC-CNR Trento, Italy
Sojic, Aleksandra	ITIA-CNR Milano, Italy

Contents

Abstract of Invited Talk

Research Contributions

AI tools in the design process of industrial products

Carlo Poloni

University of Trieste, Via A.Valerio 10, 34142 Trieste, Italy
ESTECO S.p.A., AREA Science Park, Padriciano 99, 34012 Trieste, Italy
poloni@units.it

Abstract. The definition of Artificial Intelligence that can be found on the pages of Wikipedia is the intelligence exhibited by machines or software which clearly has a rather broad and vague meaning that in many circumstances has been misunderstood.

I would therefore focus more on Artificial Intelligence Tools, i.e. the spectrum of mathematical procedure that can be used to gain, explore and exploit knowledge during a design process.

To gain knowledge means to probe design opportunities in a systematic way in order to collect sufficient data to be able to understand and predict product behaviour. To explore knowledge means to be able to drive automatically through the design options using optimization techniques. To exploit knowledge means to be able to take rational decisions about the configuration of a product to be produced.

All these actions can be performed by means of software components based on AI-related tools: Neural networks, Evolutionary Computing, Classifier Systems just to name a few.

In the development of decision support software for design optimization there is not one technique that would prevail but a blending of tools, including more traditional mathematical algorithms, that contribute to the finding of the best design configuration.

In this presentation a selection of industrial application form transportation industry to consumer goods will be used to showcase the use of AI-tools in daily design activity while possible future needs will be identified by looking at the opportunity offered by collaborative environments.

Industrial Collaborative Robot Design

A guideline for future design activity

Daniele Baratta[1]

[1] PhD student, University of Bologna, Architecture department, Bologna, Italy
daniele.baratta6@unibo.it

Abstract. This study explores the rapidly expanding Collaborative Robot market and the recent literature in *social robotics*. An attempt is made to interpret both resources to define macro-parameter trends that could be useful guidelines for robot designing. The market shows a trend towards anthropomorphism in collaborative robot design, though some studies suggest that functionality of robots (not sociality) should always be explicit and evident in their design. An unknown area of convergence between these two trends is hypothesised; here hides the most anthropomorphic design accepted by human collaborators.

Keywords: Design, Collaborative Robot, Robot Design, Industrial goods, HRI

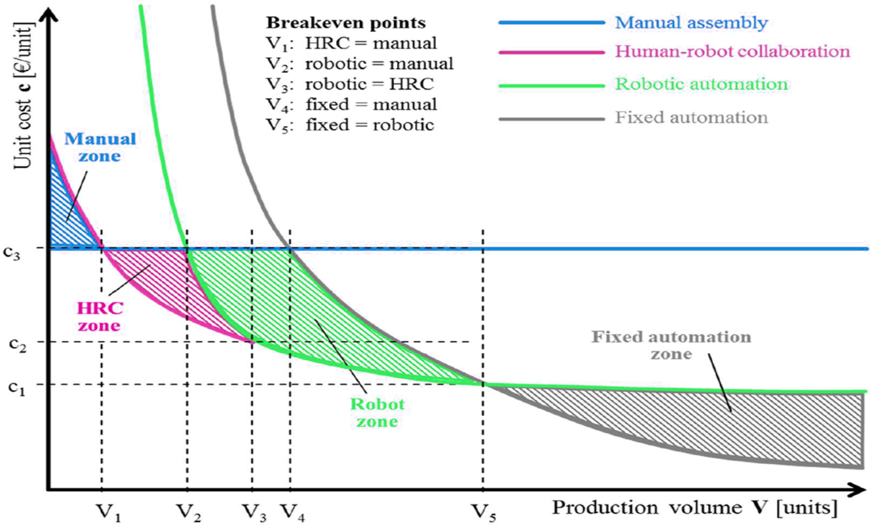
1 Introduction

Scientific sub-fields within *Robotics* are increasing and exploring its specificity in depth. Increasing results in the last decade underlines how this discipline is becoming central in shaping future society. The Human Robot Interaction (HRI) conference is an example of the latter: in its 10th annual congress it has collected many of remarkable works confirming its landmark status in robotic studies [15,16]. Being strongly interdisciplinary the HRI board suggests reflecting on the current and future relation we have with robots, underlining the pervasive presence of this "entity" constantly less recognized as simple product or industrial good. There are an increasing number of *Social Robotics* studies that analyse and test humanoid robot's effects designed for social or service purposes, where interaction dynamics are the main focus of interest. Between 2006 and 2015, 8 out of 10 most cited articles focus on interaction between people and humanoid robots, some of these are [17,18,19,20,21].

2 Collaborative Robots

The technological expertise that the world of robotics boasts has been largely exploited in industries, especially where the flexibility of the productive system is a strategic competitive factor. The rapid growing benefits brought manufacturing companies to an increase use of robots alongside human operators, since they are becoming safer, smarter and cheaper. Collaborative Robots (CR) or Cobots formerly introduced by Colgate, Wannasuphoprasit and Peshkin [8], are now in expansion; working cooperatively with humans, they reach high efficiency in operations that previously were handled only by them, for instance complex product assembly lines. In Fig.1 the pink coloured area shows the production volume gap in which HRC is convenient.

Fig. 1. Adjustment from Matthias [14]



The CR is a safe robot that interacts and works alongside operators with no cages required, opening a large number of new possibilities. Human robot collaboration (HRC) will be a competitive factor for a number of undiscovered or untested tasks and functions. Main robot manufacturing firms, committed to fabricating robots with multiple purposes, are now offering at least one CR product on the market. The main products and producers are listed below.

- Apas (Bosch)
- Baxter and Sawyer (Rethink Robotics)
- Biorb (Bionic Robotics)
- DexterBot (Yaskawa Motoman)

- iiwa (Kuka)
- M1 (Meka Robotics)
- Nextage (Kawada Industries)
- PF 400 and PP100 (Precise Automation)
- PRob (F&P Personal Robotics)
- Roberta (Gomec, ABB group)
- Speedy10 (Mabi)
- UR3, UR5 and UR10 (Universal Robots)
- YuMi (ABB)

The growing scientific community focused on CRs is largely committed to studies related to functional interaction between robots and humans: the literature supplies designers with tools to increase cooperation efficiency in some specific tasks as [9,10,11,12]. Thus, if our goal is to test the robot's formal design and the effect that this produces, we need to set specific focus parameters to our study. The past ten years of social robotics study results could be invaluable for CR optimization. Along this we think that Design, with its multidisciplinary and user centred predisposition, could be a strategic mediator between these two worlds.
Donald A. Norman in his essays *Emotional Design* and *The Design of future Things* had already reflected on the implications of robot introduction in our private and professional environment [6,7]. His thought, developed during his groundbreaking studies on cognitive sciences, analyses interactions on three levels: visceral, behavioural and reflective elaboration. Similarly [1] proposes a recent model that categorizes interaction levels based on influential factors: visceral, social mechanics and social structure. If we acknowledge this representation as a shared model of analysis, it could be useful not only in social robotics but also in mapping results with implications in CR. As previously mentioned, the aim is to focus on design activity for formal and aesthetic purposes, not functional ones.

3 Visceral factor of interaction with CR

Kaplan in his famous study [13] gives an interesting and effective panoramic to understand the cultural phenomenon of robots in both Japan and occidental countries. By analysing the differences of the two contexts, the obstacle factors for humanoid robots diffusion in USA and Europe are well explained; hence in occidental countries mechanomorphic robots are more widespread than humanoid ones. In other studies like [2], the user's aesthetic preferences are analysed and vary depending on the engaging activity. Human-oriented robots are preferred for social tasks and product-oriented robots for functional tasks. General industrial robots belong to the second group, while CR, sharing proximity and working in symbiosis with human operators, require some social skills and mannered movement that may improve work environment.

Collaborative robots Baxter and Sawyer are an example of how Rethink Robotics, leading US robot producer, interprets this opportunity: the robot eyes, always visible on the monitor, have the functional aim of revealing to the human operator the next task area and as a second effect they create a personal link with him increasing the perception of active agency [1]. People collaborating with it for a long time start to consider it as a human entity rather than a machine, as Allison Sauppé confirm studying some Baxters at Steelcase [8]. Another experience that guides us to the same conclusion comes from the research project that Kuka lead developing iiwa [3]: the design shift from the second to the third version of the product results in wire integration and more organic shape styling. The result of augmented anthropomorphism is one of the preferred upgrades their customers recognize. In a similar way YuMi (fig.2) from ABB has human-like shape properties too that can stimulate our unconscious in a visceral way, producing a natural empathy for him. In [5] is analysed how the redundancy of the 7 axis arm mimics the movements of a human arm and how this feature promotes better human-robot coexistence: co-operators are less stressed with this kind of kinematics. The market shows us a general trend for anthropomorphic shapes though sometimes there is no functional need for them. DexterBot from Yaskawa is provided with two 7 axis arms like Dexter and YuMi and, for promotional needs, is exhibited with a non functional head (fig.3).

Fig. 2. Co-worker stress test with YuMi [5]

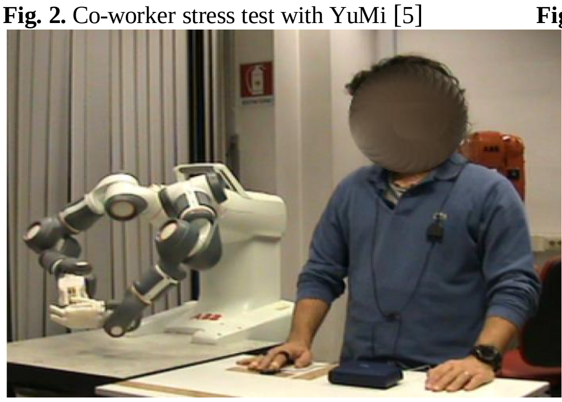
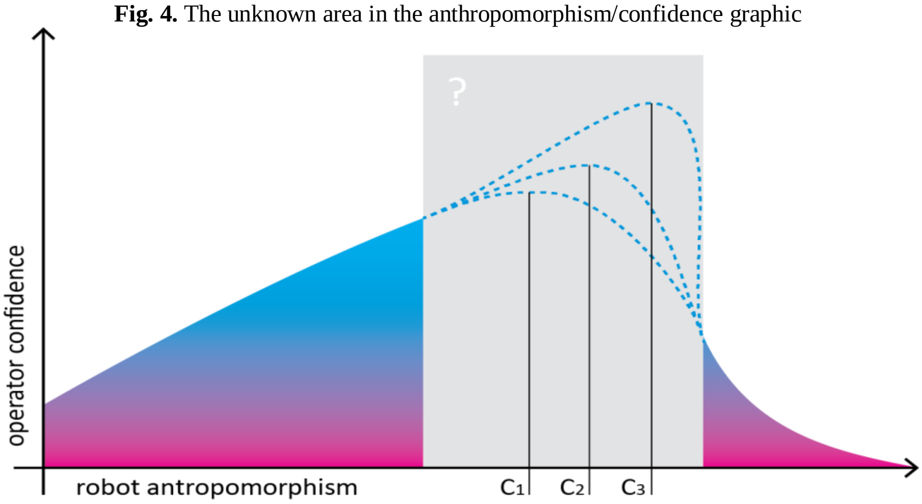


Fig. 3. Non-functional head - DexterBot

4 Future implications.

An in depth reading of the uncanny valley theory [4] suggests a guideline for future robot design activity: coherency between the anthropomorphism level of the robot's diverse features. The presence of human-like and machine-like characteristics could provoke a natural repulsion for this entity. For instance, a human-like robot performing a 360° degree shoulder rotation is unnatural and unsettling for human coworkers. Since it is not possible to design collaborative robots without key functional kinematics, beyond human limits, a strong "functional design" is required. In conclusion, the market analysis suggests that insurmountable limit for anthropomorphism is yet to be found; at the same time literature warns of its existence (C_1, C_2, C_3 in fig.4). An unknown area of convergence between these two trends is hypothesised in the graphic below. Designers should design future collaborative robots following this balance. Yumi and Dexter-bot robots are CR that explore this unknown area; due to their anthropomorphic double arm configuration they are more likely to evoke active agency, and humans tend to behave socially with them. Baxter goes farther and promotes interaction through its monitor eyes. The guideline suggested in this paper is to develop CR with these kind of anthropomorphic features, designed with a clear mechanical aesthetic: the result shouldn't mimic humans just inspire their social way of interaction.

Fig. 4. The unknown area in the anthropomorphism/confidence graphic



5 Conclusions & Aknowledgment.

This study reflects upon CR market trends and the future role of designers. The hypothesis suggests the possibility of mediation among robot functional, technical, social and interactional characteristics due to designer's multidisciplinary study background. After, a view on contemporary anthropomorphic CR design trends is suggested to supply designers with a useful guideline for their projects.

This study about CR, is part of a broader work entitled "Design driven innovation for Industrial Goods", doctoral research that the author is conducting at the Department of Architecture, University of Bologna.

References

1. Young J., Sung J.Y., Voida A., Sharlin E., Igarashi T., Christensen H., Grinter R. (2011). Evaluating human-robot interaction. Int J Soc Robot 3(1),53–67.

2. Choi J.J., Kwak S.S. (2015) The effect of robot appearance types and task types on Service evaluation of a robot. In Extended Abstract (HRI’15). ACM, Portland, OR, USA, 121-122.

3. Bischoff R., Kurth J., Schreiber G., Koeppe R., Albu-Schaeffer A., Beyer A., Eiberger O., Haddadin S., Stemmer A., Grunwald G., Hirzinger G. (2010). The KUKA-DLR Lightweight Robot arm: a new reference platform for robotics research and manufacturing. Robotics (ISR), 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK) , 741-748.

4. Chin-Chang Ho, MacDorman K.F., Dwi Pramono Z.A.D.. 2008. Human emotion and the uncanny valley: a GLM, MDS, and Isomap analysis of robot video ratings. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction* (HRI '08). ACM, New York, NY, USA, 169-176.

5. Zanchettin A.M., Bascetta L., Rocco P. (2013). Acceptability of robotic manipulators in shared working environments through human-like redundancy resolution. Applied Ergonomics, volume 44, issue 6, pp. 982–989.

6. Norman D. A., (2004). Emotional design. New York: Basic Books.

7. Norman D. A., (2007). The design of future things. New York: Basic Books.

8. Smith J., Interview found in http://www.news.wisc.edu/23419.

9. Dragan A.D., Bauman S., Forlizzi J., Srinivasa S.S. (2015). Effects of Robot Motion on Human-Robot Collaboration. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI '15). ACM, New York, NY, USA, 51-58.

10. Allen J., MacLean K.E. (2015). Personal Space Invaders: Exploring Robot-initated Touch-based Gestures for Collaborative Robotics. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts (HRI'15 Extended Abstracts) 185-186.

11. Dragan D.,Bauman S., Forlizzi J.,Srinivasa S.S. (2015). Effects of Robot Motion on Human-Robot Collaboration. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI '15). ACM, New York, NY, USA, 51-58.

12. Fang R., Doering M., Chai J.Y. (2015). Embodied Collaborative Referring Expression Generation in Situated Human-Robot Interaction. In Proceedings of the Tenth Annual

ACM/IEEE International Conference on Human-Robot Interaction (HRI '15). ACM, New York, NY, USA, 271-278.

13. Kaplan F. (2004). Who is afraid of the humanoid? Investigating cultural differences in the acceptance of robots. Int J Humanoid Robot 1(3):1–16.

14. Mathias, B. (2014). Industrial safety requirements for collaborative robots and applications. Slide presentation for ERF2014 workshop, Rovereto, Italy.

15. Hoffman G., Zuckerman O., Hirschberger G., Luria M., Sherman T.S. (2015). Design and Evaluation of a Peripheral Robotic Conversation Companion. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction* (HRI '15). ACM, New York, NY, USA, 3-10.

16. Kato Y., Kanda T., Ishiguro H. (2015). May I help you? Design of Human-like Polite Approaching Behaviour. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction* (HRI '15). ACM, New York, NY, USA, 35-42.

17. Powers A., Kiesler S., Fussell S., Torrey C. (2007). Comparing a computer agent with a humanoid robot. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction* (HRI '07). ACM, New York, NY, USA, 145-152.

18. Calinon S., Billard A. (2007). Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction* (HRI '07). ACM, New York, NY, USA, 255-262.

19. Steinfeld A., Fong T., Kaber D., Lewis M, Scholtz J., Schultz A., Goodrich M. (2006). Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction* (HRI '06).

20. Mutlu B., Shiwa T., Kanda T, Ishiguro H, Hagita N. (2009). Footing in human-robot conversations: how robots might shape participant roles using gaze cues. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction* (HRI '09). ACM, New York, NY, USA, 61-68.

21. Gockley R., Forlizzi J, Simmons R. (2007). Natural person-following behavior for social robots. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction* (HRI '07). ACM, New York, NY, USA, 17-24.

A formalization of Ashok Goel's SBF concept of function

Stefano Borgo[1], Massimiliano Carrara[2], Pawel Garbacz[3], and Pieter E. Vermaas[4]

[1] Laboratory of Applied Ontology, ISTC CNR, Trento, Italy
stefano.borgo@cnr.it
[2] FISPPA Department – Section of Philosophy, University of Padua, Italy
massimiliano.carrara@unipd.it
[3] Philosophy Department, John Paul II Catholic University of Lublin, Poland
garbacz@kul.lublin.pl
[4] Philosophy Department, Delft University of Technology, The Netherlands
p.e.vermaas@tudelft.nl

Abstract. We formalize within the DOLCE foundational ontology the Structure-Behavior-Function model (SBF) proposed by Ashok K. Goel and colleagues. Our work focuses in particular on the notion of function. This work on SBF is part of a larger project that includes the formalization of the concepts of function by Chandrasekaran and Josephson and by Stone and Wood. The overall goal is to make engineering functional descriptions of technical artifacts based on different concepts of function, exchangeable by separately formalizing these different concepts in a single ontological framework. The formalization is a necessary step towards the development of an integrated information system for engineering design.

Keywords: function, formal ontology, SBF model, Ashok K. Goel, DOLCE.

Introduction

The aim of this contribution is to formalize the concept of function of technical artifacts as advanced by Ashok K. Goel and his colleagues [10] as part of the *Structure-Behavior-Function* (SBF) model. The SBF concept of function is developed in [10, pp. 25-26], [11] from the so-called *Functional Representation* (FR) approach towards modeling functions, proposed by Chandrasekaran and Josephson [8]. The SBF model extends this original modeling, for instance, by describing the structure of technical artifacts in terms of components and substances, and by adding the assumption that there exists a limited set of primitive functions.

Given the relationship between the SBF model and the FR approach we arrive at a formalization of the SBF concept of function using as a starting point our earlier formalization of the FR approach. The formalization of SBF functions includes also formal characterizations of the SBF concepts of structure and behavior: we take a behavior in SBF to be a discrete sequence of states, and an

SBF function to be an SBF behavior with a fixed input state and a fixed output state, both specified by a set of values for state parameters (pre-conditions and post-conditions). An SBF function is then formalized as constraints on these parameters of states.

A central starting point in our larger project is to formalize all engineering concepts of function within the same ontology, seen as a unifying structure for the analysis and the formalization of these concepts, namely the *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE) [12].

The paper opens in section 1 with a brief description of our larger project. Section 2 outlines the central concepts of DOLCE. Then, in Section 3, we describe the SBF model in some detail and relate it with the FR approach. In section 4 we focus on the formalization of the SBF concept of function.

1 The larger project

This work on SBF functions is part of a larger project aimed at making engineering functional descriptions of technical artifacts (based on different concepts of function) interoperable. This is obtained by separately formalizing the main different concepts within a single ontological framework. The approach is described and argued for in [5]:

> [It] does not aim directly at a single concept of function, but tries to reconstruct the main meanings that engineers attach to this term by means of a series of formalizations within one single formal framework. In this strategy one focuses still on well-defined and specific concepts of function, which are taken as classical concepts, but now different [meanings] of such concepts are formalized. [It] is also in conformance with engineering practice by describing and formalizing the concepts of function used. [...] Yet this [...] strategy disambiguates functional descriptions only in a weak sense. Each meaning that is formalized on this strategy is analyzed in detail, assessed for consistency, and if needed at points corrected. And if such corrections are not feasible, particular meanings may even be discarded as untenable ones [...]. Yet, after formalization it still amounts to different concepts of function that co-exist in one formal system. By their co-existence in one formal system, these functional concepts may be compared and related, just as any other set of concepts can be compared and related. [...] [5, p. 152]

In our larger project we thus accept the co-existence of different meanings of function as a feature of engineering [15], and proceed by formalizing those different meanings. In [1] we formalized the concept of function by Chandrasekaran and Josephson [8], which represents the FR approach towards modeling functions. In [2] we formalized the Stone and Wood [14] concept of functions, representing the FB modeling approach. And in [9] we provided a formal comparison between these two formalizations and showed how automatic exchange of functional descriptions originating in these approaches may look like. With this contribution we proceed in our project by including a formalization of Goel's SBF concept of function.

2 A very brief introduction to DOLCE

2.1 The general structure of DOLCE

DOLCE is a foundational ontology of particulars with a clear cognitive bias since its categories are obtained by analyzing the surface structure of language and cognition. Consequences of this approach are that DOLCE's categories are at the so-called *mesoscopic level*, the level of the middle-sized objects we, as humans, perceive. The DOLCE's taxonomic structure is pictured in Figure 1. Each node in the graph is a category of the ontology. A category that is a direct subcategory of another is depicted by drawing the latter higher in the graph and linking them with an edge. PARTICULAR is the top category. The set of direct subcategories of a given category forms a partition unless dots are inserted.

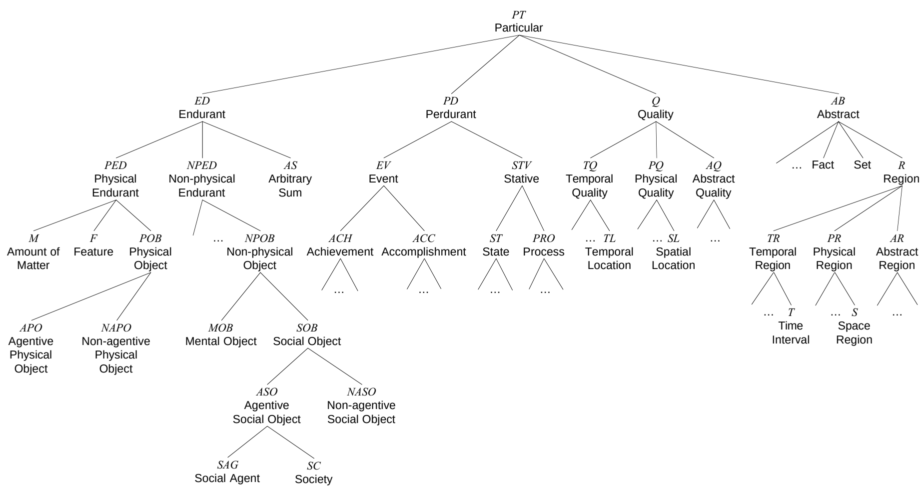


Fig. 1. The DOLCE taxonomy (from [12]).

The DOLCE ontology category ENDURANT comprises objects, e.g., a *hammer*, and amounts of matter, e.g., *the amount of water in this glass, the amount of gold in my wedding ring*, while the category PERDURANT comprises events like *making a hole* or *a soccer game*, that is, things that happen in time. The term 'object' is used in the ontology to capture a notion of unity as suggested by the partition of the class PHYSICAL ENDURANT into classes AMOUNT OF MATTER, FEATURE, and PHYSICAL OBJECTS (see Figure 1). Among those we need to explain in more detail the DOLCE notion of FEATURE. In DOLCE, features are dependent entities which are wholes, thus distinguished from individual qualities:

> Typical examples of features are "parasitic entities" such as holes, boundaries, surfaces, or stains, which are generically constantly dependent on physical objects (their hosts). All features are essential wholes, but, as in the case of objects, no common unity criterion may exist for all of them. However, typical features have a topological unity, as they are singular entities. Some features may be *relevant parts* of their host, like a bump or an edge, or *places* like a hole in a piece of cheese, the underneath of a table, the front of a house, which are not parts of their host. [12, p. 16]

2.2 DOLCE categories and relations we focus on

In this section we present the categories of DOLCE in Figure 1 that are relevant to our work. Note that the terminology adopted departs sometimes from that in engineering design, knowledge representation, and conceptual modeling since affected in part by the philosophical literature.

ED(x) stands for "x is an endurant". An endurant is an entity that is wholly present at any time it is present. It is physical if located in space and time: a *hammer_#321*, a *mover machine_#111*, an *amount of plastic*, and *the cavity in which a piston moves.*

PED(x), a subcategory of ED, stands for "x is a physical endurant. A *hammer*, a *mover machine*, an *amount of plastic*, and *the cavity in which a piston moves*, are all examples of physical endurants. We will use two subcategories of physical endurants: physical objects POB and features F.

NPED(x) stands for "x is a non-physical endurant." NPED is a subcategory of ED that includes mental objects, e.g., beliefs, intentions, etc., and social objects (SOB), e.g., norms, shares, peace treaties.

PD(x) stands for "x is a perdurant", i.e., an entity that is only partially present at any time that is present. For instance, consider the perdurant *producing an item of type #234* that consists of riveting two metal pieces and painting the resulting piece. While the *painting* goes on, the (temporal) part corresponding to *riveting* is no longer present and when this is present, the *painting* still has to come. We will use also the basic distinction between events (EV) and states (ST) among perdurants. A perdurant is stative or eventive according to whether it holds of the mereological sum of two of its instances, i.e., if it is cumulative or not. A sitting is a state since the sum of two sittings is still a sitting, while a sitting down is an event since the sum of two sitting downs is not a sitting down.

Among the ontological relations in DOLCE we will make use of the *parthood* relation: "x is part of y", written P(x,y). The formal theory based on parthood is called *mereology* [13]. In DOLCE the *parthood* relation applies to pairs of endurants and to pairs of perdurants. For instance, if a = 'writing article A' and b = 'writing the introduction to article A', then P(b,a) holds. For endurants, the relation of parthood is temporalized since an endurant may loose and gain parts throughout its existence: $\mathsf{P}(\mathtt{e}, \mathtt{e}', \mathtt{t})$ says that the endurant $\mathtt{e}$ is part of the endurant $\mathtt{e}'$ at the instant or interval $\mathtt{t}$. In the setting of SBF holds the simplifying assumption that the time interval is fixed: consequently, the temporal relativization of mereological parthood between endurants is here neglected.

A number of auxiliary definitions, like proper part, overlap and sum, can be introduced from P. (Symbol $\triangleq$ indicates a definition.)

$$\mathsf{PP}(\mathtt{x}, \mathtt{y}) \triangleq \mathsf{P}(\mathtt{x}, \mathtt{y}) \wedge \neg \mathsf{P}(\mathtt{y}, \mathtt{x}) \tag{1}$$

A perdurant is a *proper part* (PP) of another if it is part of the second and not *vice versa*. *Example*: Reading this section is a proper part of reading the paper.

$$\mathsf{O}(\mathtt{x}, \mathtt{y}) \triangleq \exists \mathtt{z}(\mathsf{P}(\mathtt{z}, \mathtt{x}) \wedge \mathsf{P}(\mathtt{z}, \mathtt{y})) \tag{2}$$

Two perdurants *overlap* (O) if a perdurant exists which is simultaneously part of both. *Example*: 'My drinking on the couch' and 'my watching TV on the couch' have 'my sitting on the couch' as part of both. Regarding mereological sum (+), a perdurant $\mathtt{z}$ is the sum of $\mathtt{x}$ and $\mathtt{y}$ provided that $\mathtt{x}$, $\mathtt{y}$ are parts of $\mathtt{z}$, and that whatever overlaps $\mathtt{z}$ also overlaps $\mathtt{x}$ or $\mathtt{y}$. Formally,

$$\mathtt{x} + \mathtt{y} \triangleq \iota \mathtt{z} \, \forall \mathtt{w}(\mathsf{O}(\mathtt{w}, \mathtt{z}) \leftrightarrow (\mathsf{O}(\mathtt{w}, \mathtt{x}) \vee \mathsf{O}(\mathtt{w}, \mathtt{y}))) \tag{3}$$

This definition can be easily extended for ternary, quaternary, etc., operations.

3 Functions in the SBF model and in the FR approach

Here we report the terminology from [10] and connect the concepts used in the SBF model and the concepts advanced in the FR approach. In Section 4, when we formalize SBF concepts, we add more details to the description of these concepts.

An SBF model of an artifact includes submodels of the artifact's structure, behavior and function. These submodels are characterized as follows:

The structural submodel of an artifact consists of a description of the *elements* of the artifact and the *connections* between these elements. In these structural models a distinction is made between elements that are *components* and elements that are *substances*. The connections between components are called *connecting points*.

The behavioral submodel captures the behavior of an artifact in terms of *transitions* between *states* of the artifact, where these states refer to properties of the connecting points of the artifact, that is, of the artifact's structure. The behavioral submodel moreover gives *causal explanations* of these transitions.

Finally, SBF functions describe the role an element in an artifact plays in the operation of the artifacts; an SBF function gives a *purpose* of the element and refers to a behavior by which the element realises the purpose. Some *primitive functions* are listed, e.g., 'create', 'destroy', 'expel', 'allow', 'pump' and 'move'.

Let us now bring in the FR approach as described in [8]. In this approach the term behavior is undestood to have five engineering meanings and the term function to have two. The meanings of behavior are characterized with the help of the primitive notion of *state variable* (the examples are from [8]):

1. the value of some state variable of the artifact or a relation between such values at a particular instant.
2. the value of a property of the artifact or a relation between such values.
3. the value of some state variable of the artifact over an interval of time.
4. the value of some output state variable of the artifact at a particular instant or over an interval.
5. the values of all the described state variables of the artifact at a particular instant or over an interval.

Note that for all meanings, a behavior of a technical artifact is in part *objective* and in part *subjective*. Objective because it eventually depends on the properties or features of the artifact. Still, the very same behavior depends on the designer(s) and, indirectly, on engineering practice for the choice of the variables.

The two meanings of function in the FR approach are called *device-centric* and *environment-centric* meanings. A *device-centric function* of an artifact is a behavior of the artifact that is selected and intended by some agent. The function is described in terms of the properties and behaviors of the artifact only; an example is "making sound" in the case of an electric buzzer. An *environment-centric function* is in turn an effect or impact of this behavior of the artifact on its environment provided this effect or impact is selected and intended by some agent. This kind of function is conceptually separated from the artifact that performs or is expected to perform this function; "enabling a visitor to a house to inform the person inside the house that someone is at the door" is an environment-centric function of the buzzer.

When comparing the concepts advanced in the SBF model and the FR approach, it can be noted that the notions of *behavior* are fairly similar. Moreover, *functions* are derived notions in both: functions give the agent's viewpoint on behaviors although agents are only implicit in the SBF framework.

In a nutshell: in SBF and in FR *functions* provide the purpose of an entity in a given situation while the entity's *behavior* is the way the purpose is accomplished. The distinction *device-centric* and *environment-centric* functions is not part of SBF. Here, we will consider the SBF concept of function as typically an FR device-centric function, since – as we will see – SBF functions refer to SBF behaviors and purposes of components that are typically described in terms of properties of the artifact itself, a specification given in FR to device-centric functions.

As concerns *behavior*: in FR the behavior of a technical artifact is the specific way in which the artifact occurs in an event, it is specified by the meanings (1-5) given above, and characterized using the primitive notion of *state variable*; in SBF behavior is also conceived as a specific way in which a technical artifact occurs in an event. Differently from FR, in SBF there is an emphasis on the state-transition construction of behaviors.

Finally, the notion of *structure* is in SBF somewhat more complex than in FR since there is in SBF, and not in FR, a basic distinction between the elements of a device and the connections between the elements.

4 Formalizing SBF Functions

We now develop the formalization of the SBF model starting from our previous work on the FR approach [1], and then extend it to cover the SBF system including the notion of function. The notion of technical artifact (or device) is introduced in SBF without a specific characterization as it happens in FR and the notion of behavior is developed from similar assumptions. Note, however, that the different setting of SBF will later lead us to make some alternative formalization choices.

We identified the following main categories of SBF

- (technical) device and its physical components
- substances
- connections and connection points
- devices' states and behaviors
- functions

Following the methodology described in [5] we first align these categories to the DOLCE taxonomy.

4.1 Ontological categorization

SBF uses a notion of device which is richer than that exploited by FR. SBF can describe to some extent the structure of the device itself. In particular, a basic distinction is set between elements (parts) of the device and connections among them. Elements are clearly divided in: a) physical components, i.e., the physical parts of a device, and b) substances, like fluids and forces. Ontologically these entities are DOLCE's endurants ($\oplus$ stands for the exclusive disjunction):

$$\texttt{Elem}(x) \to \texttt{PhComp}(x) \oplus \texttt{Subst}(x) \tag{4}$$

$$\texttt{Elem}(x) \to \mathsf{ED}(x) \tag{5}$$

More specifically, a physical component is a rigid or semi-rigid material object of a subclass (called `RigidPOB`) of physical objects (POB). We do not attempt to constrain this class here since the distinction is not clarified by the authors and does not play a role in the system. A substance can be characterized as an amount of matter (M) or a non-physical endurant (NPED), although not a NPOB, i.e., it is neither a mental nor a social object.

$$\texttt{PhComp}(x) \to \texttt{RigidPOB}(x) \tag{6}$$

$$\texttt{RigidPOB}(x) \to \mathsf{POB}(x) \tag{7}$$

$$\texttt{Subst}(x) \to \mathsf{M}(x) \vee [\mathsf{NPED}(x) \wedge \neg\mathsf{NPOB}(x)] \tag{8}$$

Components, and not substances, may have connection points (`ConnPt`) with which to be connected to other components. In DOLCE these connection points are classified as features (F):

$$\texttt{ConnPt}(x) \to \mathsf{F}(x) \tag{9}$$

Two connection points in two components can be connected. There is a fixed number of possible connection types depending on how force can be transferred across the connection points: parallel, series, touching, adjoining, bolted, fused, hinged, jointed, tied, telescoped, threaded, frictionally embedded, sewn, nailed, clipped, ball&socket installed and glued. Since connections are relationships needed to discuss force transfer or lack of it, in DOLCE we look at their temporal behavior and classify them in the category of states (ST). Thus by stating that there is a connection of type X between two points we mean that their two components are in a state to exchange force in as much as allowed by the type X of the connection. Classifying connections as states we implicitly add a temporal parameter to the connections. However, as anticipated, we do not exploit temporal information in this formalization.

To capture this, we introduce a ternary relation $\mathtt{Connect}(x, y, z)$ whose intended reading is "connection x holds between connection points y and z (in this order)."

$$\mathtt{Connect}(x, y, z) \to \mathsf{ST}(x) \wedge \mathtt{ConnPt}(y) \wedge \mathtt{ConnPt}(z) \tag{10}$$

It goes without saying that connections relate different connection points:

$$\mathtt{Connect}(x, y, z) \to y \neq z \tag{11}$$

As said, behaviors in FR and SBF are similar but the state-transition construction in SBF leads to a somewhat different formalization of behavior, in particular to include causes or explanations for the transitions, an important aspect of SBF. Starting from the notion of behaviour in FR, in the formalization of SBF we add a notion of *system behavior* ($\mathtt{SysBeh}$), namely a perdurant which is a non-empty sequence of states describing at least a connection and at least one transition. We classify transitions as achievements or accomplishments, i.e., in the eventive category EV, see Figure 1. (An interesting alternative would be to model transition types as simplified descriptions of events, this choice would amount to introduce transitions as black box entities.) We use relations $\mathtt{BehStart}$ and $\mathtt{BehEnd}$ to indicate the initial and final states of a transition, respectively, i.e., "$\mathtt{BehStart}$(x,y)" ("$\mathtt{BehEnd}$(x,y)") means that x is the initial (final) state of y.

$$\mathtt{SysBeh}(x) \to \mathtt{Transition}(x) \tag{12}$$

$$\mathtt{Transition}(x) \to \mathsf{EV}(x) \tag{13}$$

$$\mathtt{BehStart}(x, y) \vee \mathtt{BehEnd}(x, y) \to \mathsf{ST}(x) \wedge \mathtt{Transition}(y) \tag{14}$$

We are now ready to discuss functions in SBF. Functions are embedded in the SBF language via a precise list of primitives inspired by the work of Bylander [4]: *create*, *destroy*, *expel*, *allow*, *pump* and *move*. While functions are taken as intended input-output relationships, resembling once again the FR approach, there is an explicit commitment to interpret the behaviors from these elements.

To capture the specific role of these primitives, we add the following axioms (where $\mathtt{Func}(x)$ stays for "x is an SBF function"):

$$[Create(x) \vee Destroy(x) \vee Expel(x) \vee Allow(x) \vee Pump(x) \vee Move(x)] \to \mathtt{Func}(x) \tag{15}$$

However, these functions are not taken as exhaustive in the SBF language, not even in the sense that any other function should or could be seen as a specialization or a combination of these. Indeed, SBF allows the user to add new functions without restrictions. A basic separation in functional types is given by the mandatory classification of function in achievement (*Achieve*), maintenance (*Maintain*), prevention (*Prevent*) and negation (*Negate*).

$$\texttt{Func}(x) \rightarrow [Achieve(x) \oplus Maintain(x) \oplus Prevent(x) \oplus Negate(x)] \quad (16)$$

From the SBF's examples, these special cases and functions can be classified as social objects in the terminology of DOLCE:

$$\texttt{Func}(x) \rightarrow \mathsf{SOB}(x) \quad (17)$$

However, differently from FR the SBF system makes no direct reference to agents.

4.2 Ontological description

In this section we provide a more detailed ontological characterization of SBF in terms of the four relationships that relate:

1. physical components with physical components: `PhCompOf`
2. physical components with connection points: `HasConnPt`
3. physical components with functions: `HasFunc`
4. functions with behaviors: `FBCorr`

We add relation $\texttt{PhCompOf}(x, y)$, stating that x is a component of (device or component) y, to make explicit the components' structure. We also enforce the existence of a maximal component, namely, the device itself (axiom (20) makes explicit that the SBF models are contextualized to the chosen device). Then, we enforce each component to refer to only one larger component so that the component hierarchy is a tree as requested by SBF:

$$\texttt{PhCompOf}(x, y) \rightarrow \mathsf{PP}(x, y) \quad (18)$$

$$\texttt{PhCompOf}(x, y) \rightarrow \texttt{PhComp}(x) \wedge \texttt{PhComp}(y) \quad (19)$$

$$\exists x \forall y \neg \texttt{PhCompOf}(x, y) \quad (20)$$

$$\texttt{PhCompOf}(x, y_1) \wedge \texttt{PhCompOf}(x, y_2) \rightarrow y_1 = y_2 \quad (21)$$

There is no real difference between components and devices in SBF, thus we do not introduce a specific predicate for devices. The distinction is a matter of focus: components are seen as (functional) parts of larger devices. A component is itself a device from the perspective of any of its subcomponents. Since SBF always concentrates on a single device, any other element in the modeling is a component and components can be nested.

Since the notion of connection point (`ConnPt`) involves the relation of having a connection point, and $\texttt{HasConnPt}(x, y)$ means that x has y as a connection point, we can define the former in terms of the latter:

$$\texttt{ConnPt}(x) \triangleq \exists y \; \texttt{HasConnPt}(y, x) \quad (22)$$

In turn, it seems that $\mathtt{HasConnPt}(x, y)$ is ontologically subsumed by the relation of parthood:

$$\mathtt{HasConnPt}(x, y) \rightarrow \mathsf{PP}(y, x) \tag{23}$$

Note that definition 22 and axioms (6), (7), (9) imply, in DOLCE system, that $\mathtt{HasConnPt}(x, y) \rightarrow \neg\mathtt{PhComp}(y)$. Since components, and not substances, may have connection points, we need:

$$\mathtt{HasConnPt}(x, y) \rightarrow \mathtt{PhComp}(x) \tag{24}$$

Recall that connection points are features, axiom (9), and that substances are material (M) or non-physical endurants (NPED), axiom (8), thus it follows from DOLCE that connection points and substances are distinct.

To relate physical components with their functions we introduce $\mathtt{HasFunc}(x, y)$ to mean that component x has function y.

$$\mathtt{HasFunc}(x, y) \rightarrow \mathtt{PhComp}(x) \wedge \mathtt{Func}(y) \tag{25}$$

$$\mathtt{Func}(x) \rightarrow \exists y\ \mathtt{HasFunc}(y, x) \tag{26}$$

$$\mathtt{PhComp}(x) \rightarrow \exists y\ \mathtt{HasFunc}(x, y) \tag{27}$$

As said above, in both SBF and FR functions are derived notions: they select a "reading" of behaviors, and so (perhaps implicitly) provide the agent's viewpoint. The reading is given by selecting the purpose of an entity in a given situation and by considering the entity's behavior as the way that purpose is accomplished. We already stated in (25) that each component in SBF has a function. We can now state a correspondence ($\mathtt{FBCorr}$) between functions and behaviors:

$$\mathtt{FBCorr}(x, y) \rightarrow \mathtt{Func}(x) \wedge \mathtt{SysBeh}(y) \tag{28}$$

$$\mathtt{Func}(x) \rightarrow \exists y\ \mathtt{FBCorr}(x, y) \tag{29}$$

$$\mathtt{FBCorr}(x, y_1) \wedge \mathtt{FBCorr}(x, y_2) \rightarrow y_1 = y_2 \tag{30}$$

Finally, we provide a further characterization of the SBF notion of behavior. Axiom (31) states that a system behavior is the event sum of the states of 'behavior start' and 'behavior end' *plus* the transition event between them (the sum is ordered since they have a temporal dimension). Axiom (32) states that these system behaviors are uniquely identified by their input and output states.

$$\mathtt{SysBeh}(x) \rightarrow \\ \exists y, v, z\ [\mathtt{BehStart}(y, x) \wedge \mathtt{BehEnd}(v, x) \wedge \mathtt{Transition}(z) \wedge x = y + z + v] \tag{31}$$

$$(\mathtt{BehStart}(x, z_1) \wedge \mathtt{BehEnd}(y, z_1)) \wedge (\mathtt{BehStart}(x, z_2) \wedge \mathtt{BehEnd}(y, z_2)) \rightarrow \\ z_1 = z_2 \tag{32}$$

Note that the transition (an event in DOLCE) is naturally directed from the initial state to the ending state and provides the information on how the state change happens, that is, it also includes the causal explanation(s) requested by SBF.

Further formal characteristics. Our ontological characterization of SBF has modeled the explicit ontological aspects of SBF. Below we characterize some key SBF notions in more detail, but this is rather an extension than an explication.

Since PhCompOf is subsumed by the relation of parthood, following [6] we assume that it is a (strict) partial order:

$$\neg\texttt{PhCompOf}(x, x) \tag{33}$$

$$\texttt{PhCompOf}(x, y) \wedge \texttt{PhCompOf}(y, z) \to \texttt{PhCompOf}(x, z) \tag{34}$$

Furthermore, while the system seems to be extensional, it is unclear whether the mereological reconstruction of these notions requires more specific principles like, e.g., strong supplementation [13, p. 29].

We know that PhComp and HasConnPt are related via axiom (24), but there seem to be an implicit relationship among them stating that each physical component has at least one connection point:

$$\texttt{PhComp}(x) \to \exists y\ \texttt{HasConnPt}(x, y) \tag{35}$$

Together, they amount to the following equivalence which is easily justified within the engineering perspective:

$$\texttt{PhComp}(x) \leftrightarrow \exists y\ \texttt{HasConnPt}(x, y) \tag{36}$$

Another implicit assumption seem to bind connection points to unique bearers:

$$\texttt{HasConnPt}(x_1, y) \wedge \texttt{HasConnPt}(x_2, y) \to x_1 = x_2 \tag{37}$$

5 Conclusions

We have studied the concepts underlying Goel's SBF model and proposed a formalization of the system within the DOLCE foundational ontology. The formal characterization of the SBF concepts aimed to cover three key elements: structure, behavior and function. The analysis and the subsequent formalisation show that notions like *component*, *substance* and *connection point*, are only partially characterized and that further information should be collected from other sources, for instance by directly analyzing SBF software packages on component and functional information. We have not investigated this type of material here.

While there are strong connections between the SBF and FR models of function, our analysis shows some important differences which have not been highlight in the literature. The notion of function in SBF does not admit a direct dependence on agents as in FR and, while remaining compatible with the latter, seems to carefully introduce a framework where agents have no explicit role. Furthermore, SBF introduces a short list of functions, showing that function classification is relevant for the framework, but does not include the general distinction between device-centric and environment-centric functions which is at the core of the FR model. Finally, SBF provides the tool for a mereological description of the structure of devices by introducing components and connection ports, while FR focuses mainly on the relations between the devices and their environment.

With this analysis and formalization we are now in the position to formally compare the SBF concept of function with other engineering concepts of function and to extend the means for interoperability across engineering functional descriptions of technical artifacts based on different concepts of function. This will be a subject of future research.

References

1. Borgo, S., Carrara, M., Garbacz, P., Vermaas, P. E., (2009), "A Formal Ontological Perspective on the Behaviors and Functions of Technical Artifacts", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 23 (1), 3-21.
2. Borgo, S., Carrara, M., Garbacz, P., Vermaas, P. E., (2011), "A Formalization of Functions as Operations on Flows", Journal of Computing and Information Science in Engineering, 11, 031007-031020.
3. Borgo, S., and Leitão, P., (2007), Foundations for a Core Ontology of Manufacturing, in Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems, Integrated Series in Information Systems Vol. 14, ed. by Kishore R., Ramesh R., Sharman R., Springer, New York, pp. 751-776.
4. Bylander, T., (1991), "A Theory of Consolidation for Reasoning about Devices", Man-Machine Studies, 35, 467-489.
5. Carrara, M., Garbacz, P., Vermaas, P. E., (2011), "If Engineering Function is a Family Resemblance Concept: Assessing Three Formalization Strategies", Applied Ontology, 6 (2), 141-163.
6. Casati, R., Varzi, A. C., (2003), Parts and Places: The Structures of Spatial Representation, MIT Press, Cambridge, MA.
7. Chandrasekaran, B., (2005), "Representing Function: Relating Functional Representation and Functional Modeling Research Streams", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 19 (2), 65-74.
8. Chandrasekaran, B., Josephson, J. R., (2000), "Function in Device Representation", Engineering with Computers, 16 (3/4), 162-177.
9. Garbacz, Borgo, S., Carrara, M., Vermaas, P. E., (2011), "Two Ontology-Driven Formalisations of Functions and Their Comparison", Journal of Engineering Design, 22, 733-764.
10. Goel, A. K., Rugaber, S., Vattam, S., (2009), "Structure, Behavior, and Function of Complex Systems: The Structure, Behavior, and Function Modeling Language", Art. Intelligence for Engineering Design, Analysis and Manufacturing, 23 (1), 23-35.
11. Goel, A. K., (2013), "One Thirty Year Long Case Study; Fifteen Principles: Implications of an AI Methodology for Functional Modeling", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 27 (3), 203-215.
12. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L., (2002), WonderWeb Deliverable D18. Ontology Library (final), WonderWeb European Project, 2003. http://wonderweb.man.ac.uk/deliverables/documents/D18.pdf
13. Simons, P., (1987), Parts: A Study in Ontology, Oxford University Press, Oxford.
14. Stone, R., Wood, K., (2000), "Development of a Functional Basis for Design", Journal of Mechanical Design, 122 (4), 359-370.
15. Vermaas, P. E., Eckert, C., (2013), "My Functional Description is Better!", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 27, 187-190.

The Isolation System Design in Hydraulic Networks

Marco Gavanelli, Maddalena Nonato, and Andrea Peano

Engineering Department - University of Ferrara,
Via Saragat 1 - 44121 Ferrara - Italy,
{marco.gavanelli,maddalena.nonato,andrea.peano}@unife.it

Abstract. The positioning of isolation valves on water distribution networks is a hard design issue in hydroinformatics. Hydraulic engineers usually solve it by way of genetic algorithms, which do not exploit the constrained structure of the problem. Several solving approaches, based on constrained optimisation, have been developed in Artificial Intelligence, and prove that this discipline can surely have a prominent role in hydraulic networks design.

Keywords: hydroinformatics, isolation system design

1 Introduction

Water Distribution Systems (WDSs) are complex systems whose mission is to supply water to the communities living in their service area. A WDS is made of several components, the main ones being: a set of reservoirs feeding the WDS, a set of pipes delivering water to the system users, a set of junctions connecting two or more pipes to each other; each pipe has then a user demand to satisfy, and it can be quantified by the average water consumption (litres per second $[l/s]$). We illustrate these components on the toy network depicted in Figure 1. This hydraulic network has a single reservoir T, 8 junctions and 10 pipes with positive demand, plus a 0-demand pipe which connects the reservoir to the rest of the network.

Many design issues in hydraulic engineering come up as constrained optimisation problems, e.g., the design of pipes' diameters [4], the positioning of various hydraulic devices such as quality sensors [21], valves [11], pumps [18]. Artificial Intelligence provides suitable declarative paradigms and languages to define these problems, e.g., Logic Programming [16], Answer Set Programming [14, 10], and Constraint Programming [22] among all, many dedicated algorithms, and very efficient off-the-shelf solvers [9, 23, 8, 15]. Several of these technologies have been exploited to optimise the positioning of valves, which is introduced just below; this success case shows that Artificial Intelligence also designes solutions in hydraulic engineering and extends hydroinformatics with powerful tools.

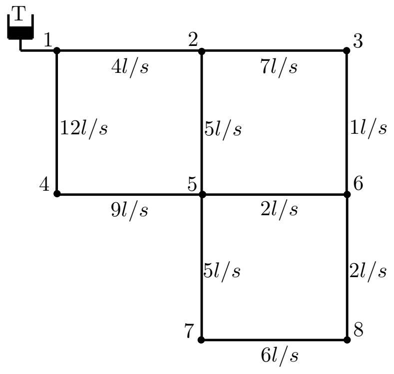


Fig. 1. A toy network

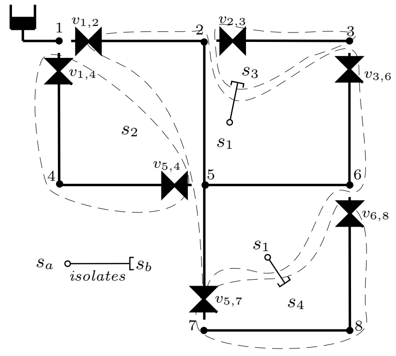


Fig. 2. A feasible sectorization

The isolation system. Failure of ageing pipes frequently occurs. In such a case, the leaking pipe is isolated on purpose, to be de-watered and fixed. Isolation is achieved by closing some of the isolation valves purposely located on the network, in such a way that the failed pipe gets disconnected from the reservoirs. In the ideal situation, each pipe would have one such valve positioned at each of its two extremes, so that only that pipe could be disconnected in case of maintenance by closing just its two valves, and it would require twice as many valves as the network pipes. However, the number of valves is limited due to cost, and their intelligent location poses a challenge, as described hereafter.

First, valves must be properly located at pipe extremes, right in adjacency to the junctions; in fact, manholes are typically available there, so junctions are accessible for maintenance purposes. Also, every pipe can get broken, thus any pipe must be isolable by closing some valves. Consequently, when all valves are closed the network should be subdivided into a set of subnets (or *connected components* in graph theory). We call *sectors* the subnets that are induced by closing all valves. The valves delimiting a sector s are the *boundary valves* of s, and have to be closed to isolate s whenever any pipe gets broken in it.

Figure 2 reports a feasible isolation system made of 7 valves, where $v_{a,b}$ tells that the valve lies close to junction a of the generic pipe (a, b); similarly $v_{b,a}$ tells that the valve is on the extreme b of pipe (a, b). So, the installed valves are: $v_{1,2}$, $v_{1,4}$, $v_{2,3}$, $v_{3,6}$, $v_{5,4}$, $v_{6,8}$, and $v_{7,5}$. This positioning yields 4 sectors, named s_1, s_2, s_3, and s_4 in Figure 1; s_2 has the greatest internal demand (ID), i.e., $ID(s_2) = 21l/s$, whereas $ID(s_1) = 17l/s$, $ID(s_3) = 7l/s$, and $ID(s_4) = 8l/s$.

When a sector is isolated, all its users experience supply disruption that is measured by the amount of their unsatisfied demand (UD). The WDS engineers who design the network aim to reduce and equally distribute the service disruption among users in case of maintenance operations. Graph partitioning problems recall several aspects of this problem structure and they can be exploited to compute a feasible sectorization of the network; however, they are able to represent only the internal demand of the sectors, which is often lower than the whole unsatisfied demand due to its isolation.

Unintended Isolations. A sector for which all connections to the reservoirs go through other isolated sectors will be isolated as well. Having Figure 2 at hand, pipes $(2,3)$, $(6,8)$ and $(7,8)$ are isolated whenever a pipe in s_1, e.g., $(5,6)$, gets broken. Closing s_1's boundary valves also determines the isolation of s_3 and s_4, so $UD(s_1) = ID(s_1) + ID(s_3) + ID(s_4) = 17 + 8 + 7 = 32l/s$, that makes s_1 the worst isolation case in terms of unsatisfied demand; s_3 and s_4 are called *unintended isolations* of s_1.

In general we have that $ID(s) \leq UD(s)$, which means that the quality measure of the sectorization does not depend only on the internal demand of the sectors. To include the missing quantity and achieve the entire unsatisfied demand of a sector isolation, the unintended isolations should be modelled.

Next section defines the isolation system design as a constrained optimisation problem, then recalls related works and describes existing solution approaches in Artificial Intelligence. Section 3 shows results and Section 4 draws conclusions and future works.

2 Optimising the Valves Positioning

The design of the isolation system of WDSs can be formulated as a constrained optimisation problems that consists of computing the optimal placement of a limited number of valves; the positioning should draw a sectorization of the network, so that any pipe can be isolated. What an optimal placement is may depend on several criteria that give rise to different objective functions; in particular, in the hydraulic engineering literature a bi-objective optimisation minimizes i) the maximum undelivered demand and ii) the number of valves [11]. Accordingly, having fixed a number of valves N_v, the objective function can be stated in a general fashion as $\min : \max_s\{UD(s)\}$, and the Pareto front can be computed by a sequence of single-objective problems.

2.1 Related Works

In the literature of hydraulic engineering, a multi-objective genetic algorithm for the near-optimal design of the isolation system is described in [11]; the isolation system's cost is also optimised by a genetic algorithm in [3]. Both cannot ensure that the found solutions are indeed the Pareto optimal.

The first mathematical model for this constrained optimisation problem was a Mixed Integer Linear Programming (MILP [17]), it integrates Graph Partitioning and Maximum Flows modules [20] and it has been further generalized in [19]. A stochastic formulation of this model has been proposed in [1].

2.2 Solution Approaches in Logic Programming

Two main exact approaches have been proposed in Artificial Intelligence, and in particular they are based on Logic Programming, as follows.

Constraint Logic Programming. The first exact approach for the design of the isolation system was implemented in Constraint Logic Programming on Finite Domains (CLP(FD)) [13]. It models the problem as a two-player game, and solves it with a minimax approach [2]. The moves of this game are: i) the first player places N_v valves in the network, ii) the second player selects one pipe to be damaged, iii) the first closes a set of valves isolating the damaged pipe. The cost for the first player (and reward for the second) is the undelivered demand: the total demand of all users that remain without service when the broken pipe is isolated. Sectors are built up on the fly and not explicitly defined by this approach, so no symmetry on sectors' names is induced.

Answer Set Programming. In Artificial Intelligence, Answer Set Programming (ASP) [14, 10] is another logic paradigm that allows for solving constrained optimisation problems. Several ASP programs have been developed for the design of the isolation system [6, 5]. Some programs measure the worst isolation case by computing the reachability of each pipe from the sources, so enumerating the paths from the sources to the demand points. Other programs group the isolated pipes into sectors and compute the sectors reachability from the sources; in this way, the exponential explosion of paths is reduced at the cost of a huge symmetry on sectors' names, however symmetry breaking constraints can be imposed and effectively help the search [5]. All these programs count a few logic rules, about 25.

The mathematical program described in Section 2.1 can be solved by branch and bound and, like the CLP(FD) and the ASP programs, provides optimal solutions. We show a computational comparison of these three methodologies in the next section.

3 Results

The CLP(FD) and ASP programs in Section 2.2 were solved with ECLiPSe [23] and Clasp [8], respectively, whereas the MILP program in [20, 19] was solved with Gurobi [12]. These algorithms are complete, so they are able to find the optimal solutions and prove optimality.

The chart in Figure 3 shows the optimal Pareto front [2] for a real hydraulic network, consisting of 33 pipes, and it improves the approximated front in [11] of about the 10% for some points; notice that all exact approaches are able to compute the very same optimal front, though computing times may be quite different. In particular, the computational comparison in [19] shows that with a timeout of $10'000$ seconds the MILP program is solved up to 10 valves, the ASP one up to 11, and the CLP(FD) up to 14, as shown in Figure 4. The MILP model suffers of a huge number of symmetries, but symmetry breaking through hard constraints has no effect [19], whereas it is very helpful in ASP and in constraint propagation systems. The ASP programs can be improved further, as discussed in [19]. It is worth noting that both solution approaches in Artificial Intelligence overcome the MILP program in terms of computing time.

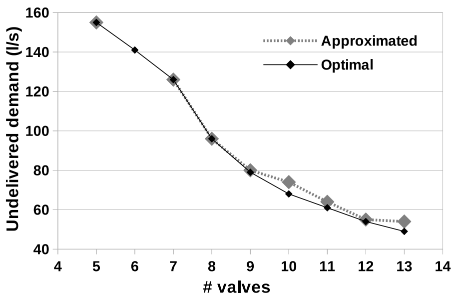


Fig. 3. Pareto fronts in [2]

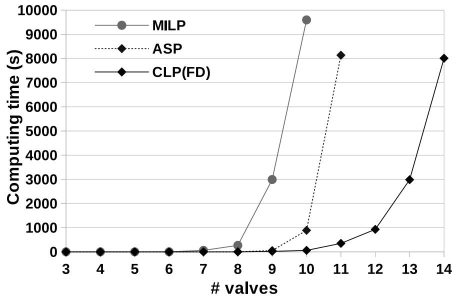


Fig. 4. Computational comparison in [19]

4 Conclusion

We summarized two existing approaches that have been developed in Artificial Intelligence to address the isolation system design, i.e., a constrained optimisation problem in hydraulic engineering. These approaches improved the state of the art in the hydraulic engineering literature in terms of solution quality and in the Operations Research literature in terms of computing time. This proved that Artificial Intelligence provides suitable technologies to address design issues arising in hydraulic engineering, and we believe it will be integrated more and more into the hydroinformatics in the next future. As the results show, exact approaches do not scale up on larger instances, so future work aims to develop hybrid methodologies and heuristics. MILP and ASP technologies could be coupled together to solve decompositions of the MILP model. Also genetic algorithms could be coupled with ASP, in analogy to the work in [7]; in this way the search capability of genetic algorithms on combinatorial spaces is enriched with an ASP optimisation layer, whose role would be to tighten the search space.

References

1. M. Bruni, P. Beraldi, and D. Conforti. A stochastic programming approach for the strategic valve locations problem in a water distribution system. *Procedia - Social and Behavioral Sciences*, 108(0):129 – 138, 2014. Operational Research for Development, Sustainability and Local Economies.
2. M. Cattafi, M. Gavanelli, M. Nonato, S. Alvisi, and M. Franchini. Optimal placement of valves in a water distribution network with CLP(FD). *Theory and Practice of Logic Programming*, 11(4-5):731–747, 2011.
3. E. Creaco, M. Franchini, and S. Alvisi. Optimal placement of isolation valves in water distribution systems based on valve cost and weighted average demand shortfall. *Water Resources Management*, 24(15):4317–4338, 2010.
4. M. Franchini. Un excursus sugli algoritmi per la progettazione e la riabilitazione delle reti di distribuzione idrica. *L'Acqua*, 2:15 – 22, 2010.
5. M. Gavanelli, M. Nonato, and A. Peano. An ASP approach for the valves positioning optimization in a water distribution system. *Journal of Logic and Computation*, 2013. Accepted for publication.

6. M. Gavanelli, M. Nonato, A. Peano, S. Alvisi, and M. Franchini. An ASP approach for the valves positioning optimization in a water distribution system. In F. Lisi, editor, *9th Italian Convention on Computational Logic (CILC 2012), Rome, Italy*, volume 857 of *CEUR workshop proceedings*, pages 134–148, 2012.
7. M. Gavanelli, M. Nonato, A. Peano, S. Alvisi, and M. Franchini. Scheduling countermeasures to contamination events by genetic algorithms. *AI Communications*, 28(2):259–282, 2015.
8. M. Gebser, B. Kaufmann, and T. Schaub. Conflict-driven answer set solving: From theory to practice. *Artificial Intelligence*, 187-188:52–89, Aug. 2012.
9. Gecode Team. Gecode: Generic constraint development environment, 2006. Available from `http://www.gecode.org`.
10. M. Gelfond. Answer sets. In F. van Harmelen, V. Lifschitz, and B. Porter, editors, *Handbook of Knowledge Representation*, pages 285–316. Elsevier Science, 2008.
11. O. Giustolisi and D. A. Savić. Identification of segments and optimal isolation valve system design in water distribution networks. *Urban Water Journal*, 7(1):1–15, 2010.
12. Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2014. http://www.gurobi.com.
13. J. Jaffar and M. J. Maher. Constraint logic programming: a survey. *The Journal of Logic Programming*, 1920, Supplement 1(0):503 – 581, 1994. Special Issue: Ten Years of Logic Programming.
14. N. Leone. Logic programming and nonmonotonic reasoning: From theory to systems and applications. In C. Baral, G. Brewka, and J. Schlipf, editors, *Logic Programming and Nonmonotonic Reasoning*, volume 4483 of *Lecture Notes in Computer Science*, pages 1–1. Springer Berlin Heidelberg, 2007.
15. N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic (TOCL)*, 7(3):499–562, July 2006.
16. J. W. Lloyd. *Foundations of Logic Programming; (2nd ext. ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1987.
17. R. Martin. *Large Scale Linear and Integer Optimization: A Unified Approach*. Springer US, 1999.
18. I. Narayanan, V. Sarangan, A. Vasan, A. Srinivasan, A. Sivasubramaniam, B. Murt, and S. Narasimhan. Efficient booster pump placement in water networks using graph theoretic principles. In *Green Computing Conference (IGCC), 2012 International*, pages 1–6, June 2012.
19. A. Peano. *Solving Real-Life Hydroinformatics Problems with Operations Research and Artificial Intelligence*. PhD thesis, University of Ferrara, 2015.
20. A. Peano, M. Nonato, M. Gavanelli, S. Alvisi, and M. Franchini. A bilevel mixed integer linear programming model for valves location in water distribution systems. In S. Ravizza and P. Holborn, editors, *3rd Student Conference on Operational Research*, volume 22 of *OpenAccess Series in Informatics (OASIcs)*, pages 103–112, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
21. S. Rathi and R. Gupta. Sensor placement methods for contamination detection in water distribution networks: A review. *Procedia Engineering*, 89(0):181 – 188, 2014. 16th Water Distribution System Analysis Conference, {WDSA2014} Urban Water Hydroinformatics and Strategic Planning.
22. F. Rossi, P. van Beek, and T. Walsh. *Handbook of Constraint Programming*. Foundations of Artificial Intelligence. Elsevier Science, 2006.
23. J. Schimpf and K. Shen. ECL^{i}PS^e - from LP to CLP. *TPLP*, 12(1-2):127–156, 2012.

On the ontological status of design objects

Nicola Guarino*, M.Rosaria Stufano Melone°

*ISTC-CNR , Via alla Cascata, 56/C , 38123 Trento , Italy
nicola.guarino@cnr.it

° Politecnico di Bari, via E.Orabona, 4, Bari, Italy
rossellaok@gmail.com

Abstract. In the design process, architects tend to choose and arrange together primitive geometrical elements according to their own cognitive environment (as reflected by culture and education), taking into account as well the broader social environment from which planning requests emerge.
Therefore, each of such elements plays a very specific role in the global system, resulting from a strong intentional choice. Design creativity emerges exactly from such strong intentional choices.
During their activities, architects tend to talk of these design elements using terms that reflect the role they have in the system, talking of them as "real entities" even if they don't exist (yet) in the real world, and are just on paper or even only in the architect's mind.
In this paper we shall discuss the ontological nature of design elements and related notions, distinguishing among: (i) design elements, (ii) design components, (iii) physical system components, (iv) conventional system components.

Keywords: architecture, design, ontology, creativity

1 Introduction

Talking of architectural design is not a simple matter. This is because every design process has an artistic/creative component, which often introduces communication problems due to difficulties in understanding the architect's language [1]. During the design process, architects create images of entities that are supposed to become real, things that may never exist but, once conceived, they do exist in some way. But in which sense do such design objects exist? How can we analyze their ontological status?

Indeed, while talking of their projects, architects tend to ascribe a genuine ontological status to their 'creatures', even if they do not have a physical presence: they refer to special imaginary, conventional entities that are specifically dependent on a design specification. Such entities have three main modes of existence: they can exist just as mental prototypes in the architect's mind, they can be (partially) realized in the real world by means of physical objects, or they can be "projected" features of physical objects, as in the case where a house is still in construction and the architect talks of the kitchen as if it existed already in the way it was designed. According to Guarino [2] these conventional design entities presuppose a non-standard ontological behavior. In particular, the same design element may undergo a complete replacement of the physical object that realizes it. In this paper we intend to elaborate on Guarino's analysis, focusing in particular on architectural design.

In the next Section we report some reflections about the architects' designing process, while in Section 3 we develop an ontological analysis of the different kinds of design entities. In Section 4 we draw some brief conclusions and discuss the follow ups of this research.

2 Design process and design objects

Probably no unified and unique definition of 'design' exists. Ralph and Wand [3] tried to give one. The noun 'design', for them, is a specification of an object, manifested by some agent, intended to accomplish goals, in a particular environment, using a set of primitive components, satisfying a set of requirements, subject to some constraints; the verb 'design', therefore, denotes the process of creating a design, in a particular environment (where the designer operates).

The aim of exploring design in a scientific way can be traced back to ideas in the 20th-Century modern movement of design [4]. What designers and architects know about especially is the ‘artificial world’ - the human-made world of artifacts. Their knowledge, skills and values lie especially in the techniques of the artificial [5].

Just as the other intellectual cultures in the sciences and in the arts concentrate on the underlying forms of knowledge peculiar to the scientist or the artist [4], designers' or architects', knowledge is inherent to the artifacts of the artificial world, gained through using and reflecting upon the use of those artifacts.

There is a lot of literature [6] [7] [4] that focuses on the knowledge lying behind design actions and in general on the design process; here we want focus on the *epistemic objects* this knowledge is about, i.e. on the nature of designers' *domain of discourse*.

Focusing on the designers' domain of discourse shifts the attention to the *language* used during the design process. Drawing and talking are actions occurring at the same time during the design process. Together, they constitute what Schon [8] defines a 'designing language' where verbal and non verbal dimensions are strictly connected. In one of Schon's observations the designer was designing and at the same time talking to another designer. His speech was full of expressions that were intelligible only by observing his hands moving on the paper, and his arms around himself. The two architects were using a meta-language to discuss their own projectual intentions.

According to the results obtained from experiments and reported in the literature, during the design process we can recognize different phases: an approach phase, a definition phase, a redefinition phase, and a refinement phase, where details are expanded at a different scale [9]. In all these phases, architects deal with design objects that will be finally realized, objects that will be modified, objects that will not be chosen for the final design (and subsequent realization). Note that these objects are not drawings, but physical objects existing in the architects' mind, i.e., mental prototypes existing in a possible world. Drawings are just representations (possibly at different levels of details) of such mental prototypes.

We are interested in understanding the ontological status of these design objects, in order to analyze and make explicit –as much as possible– the main characteristics of ontological assumptions behind the design process. Indeed, although our focus here is on the universe of discourse underlying the design process, our ultimate goal is to understand the whole design process, and especially design creativity. In this perspective, Purini [10] focuses on the intentional actions that he labels 'techniques for invention'. During the implementation of all these 'intentional actions' the designer feels himself as being in a certain oneiric dimension where objects slide from reality into an 'unreal dimension' where every object can be read according to many different points of view.

In general, all cognitive processes that lead to a design, or to the creation of an 'architectural thing' implicitly use techniques for enhancing creativity [11], [12]. These techniques are based on procedures whose linguistic descriptions unavoidably have two refer to geometrical entities that acquire their own ontological status, being conceived/projected as objects on their own.

3 Artefactual systems and architectural design

The result of a design process is an artifact, or more in general an artifactual system (composed of several system components playing different roles). In a previous paper [2], it was argued that the way people deal with artifactual systems presupposes a non-standard ontological behavior, which allows for the mere virtual presence of components expected to be in a certain position, and for the complete replacement of their physical realization.

While ascribing an ontological status to what engineers and technicians have in mind when they speak of artifacts and their components in a technical discourse, we clearly subscribe to the perspective of descriptive metaphysics [13], which ‘is content to describe the actual structure of our thought about the world’. By its very nature, descriptive metaphysics takes a liberal view concerning the introduction of new ontological categories as long as they are motivated by cognitive distinctions, often reflected by the surface structure of natural language. The focus of our analysis is therefore not the world as such, but rather our way to look at it, i.e., some kind of 'Weltanschauung'. This term is sometimes used in social sciences to indicate a set of high-level beliefs through which an individual or group experiences and interprets the world. A precise definition of this concept seems however to be elusive [14], while the approach of descriptive metaphysics seems more useful to understand the hidden assumptions associated to the design process.

Let us now discuss the status of the various entities involved in the design process. As we have seen, this process includes a number of activities involving intentional selection of design elements from a mental repository, composition and arrangement of these components, physical realization, and so on. From the ontological point of view, this means we can distinguish the following cases:

a. *design element*: a certain object that can appear more than one time (with variations) in the same design or in different designs; an architectural element extrapolated out from its context and considered as independent (note that we are talking here of a genuine object in the designer’s mind, not of a physical drawing).

b. *design component*: a design element that plays a specific role in a design, standing in a specific position

c. *physical system component*: a particular object, for instance a particular portico made with bricks posed according a peculiar texture.

d. *conventional system component*: what is expected to exist in a particular place, in the architect's mind.

Let's consider an example:

Suppose two architects are working on an urban garden. In its design, among other things, they include a portico. During the garden design process and the subsequent realization, different scenarios may open up:

(i) after they (painfully) agreed on a certain design solution for the portico, they discover later that this is infeasible due to technical constraints. They decide however to save this solution to reuse it in another project. Is that possible? What is the ontological status of that portico? What kind of mental mechanism makes this object so real that it can pass from a project to another? Maybe it will undergo some modifications, but will it be the same portico?

According to the classification suggested above, this portico is a design element. Of course, suitable rules will be associated to it in order to regulate the admissible changes while reusing this element in different projects.

(ii) it happens that the realized portico falls down because of natural or anthropic actions, and a decision is made in order to rebuild it. While describing the site to someone never been there before, keeping the design drawings in his hand, the architect says: "And here there we have the portico". And he starts describing it. What is he talking about?

In this case the portico is a conventional system component, this because here the portico is something previously designed, then realized, then destroyed, and now it will be there again, although different bricks will be used, so a different physical component will be built (here we have a sort of conventional solution to the Theseus's paradox).

As discussed in [2], conventional system components are like phantasms, where for phantasms we intend combinations of mental imagery constructed by embodied, distributed and situated cognitive process [14] that can materialise and disappear.

4 Conclusion

In this paper we have made a first preliminary attempt to analyze the way architects refer to their own design process under an ontological perspective. We intend to further develop these ideas in order to contribute to a better understanding of the design process, and especially the relationship between knowledge-in-practice and its 'objects' [6], [7]. One of the expected practical results of this work will hopefully be a more effective way to share design knowledge among the different agents involved in a complex design process. We are also thinking of an experimental setting to elicit data (including neural evidence) about linguistic and drawing actions occurring at the same time in the design process, to be used to evaluate the cognitive adequacy of our approach.

References

1. Tafuri, M., 1986, Storia dell'architettura italiana 1944-1985, Torino: Einaudi, 1986
2. Guarino, N., 2014, Artefactual Systems, Missing Components and Replaceability, in M. Franssen et al. (eds.), Artefact Kinds: Ontology and the Human-Made World, Zurich, Springer International Publishing
3. Ralph, P.; Wand, Y., 2009, A Proposal for a Formal Definition of the Design Concept, Sauder School of Business University of British Columbia Canada
4. Cross, N., 2001, Designerly Ways of Knowing: Design Discipline versus Design Science, Design Issues, Vol. 17, No. 3, pp. 49-55, Boston: MIT Press Journals
5. Cross, N, Developments in Design Methodology, Wiley, Chichester (1984)
6. Ewenstein, B.;Whyte, J., 2009, Knowledge practices in design: The role of visual representations as 'epistemic objects'. Organization Studies, 30 (1). pp. 7-30. ISSN 01708406, University of Reading, CentAUR: Sage Publications
7. Wang, D.; Ilhan, A. O., 2009, Holding Creativity Together, Design Issues, Vol.25, No.1, Boston: MIT Press Journals
8. Schon, D.A.,1983, The Reflexive Practitioner, New York, Basic Books (trad.it. a cura di Angela Barbanente, 1993, Il professionista Riflessivo, Bari, Dedalo)
9. Quaroni, L., 1988, Progettare un edificio. Otto lezioni di Architettura, Roma, Gangemi
10. Purini, F., 2002, Le opere, gli scritti, la critica, Milano, ElectaElemond Editori Associati
11. J. S. Gero, 1996, Creativity, emergence and evolution in design, ELSEVIER Knowledge-Based Systems 9 435-448
12. Hatchuel, A; B. Weil, 2002, Lathéorie C-K: Fondements et usages d'une théorie unifiée de la conception, Colloque Sciences de la conception, Lyon
13. Strawson, P. F. (1959). Individuals: An essay in descriptive metaphysics . London: Routledge
14. Harrell, D. F.jr, 2013, Phantasmal Media, Cambridge, The Mit Press

Will AI ever support Design Thinking?

Francesca A. Lisi

Dipartimento di Informatica, Università degli Studi di Bari “Aldo Moro”, Italy
francesca.lisi@uniba.it

Abstract. This paper addresses the question of whether AI will ever support Design Thinking, with a focus on Architecture and Urban Planning, by analyzing the current trends of research in AI and related fields.

1 Introduction

Design was first considered as a “way of thinking” by Herbert A. Simon [34]. The notion was then applied in engineering design by Robert McKim [27] although a significant early usage of the term *Design Thinking* in the design research literature is due to Peter Rowe [32]. Rolf Faste expanded on McKim’s work at Stanford University in the 1980s and 1990s [14], teaching “design thinking as a method of creative action.” Design thinking was later adapted for business purposes by David M. Kelley, a Faste’s Stanford colleague, who founded IDEO[1] in 1991 and the Hasso Plattner Institute of Design at Stanford (aka d.school)[2] in 2004. The term has been popularized through several initiatives of the d.school.

Design Thinking is especially useful when tackling so-called *wicked problems*, *i.e.* problems that are ill-defined or tricky [6]. In wicked problems, both the problem and the solution are unknown at the outset of the problem-solving exercise. This is as opposed to *tame problems* where the problem is well-defined, and the solution is available through some technical knowledge. For wicked problems, the general thrust of the problem may be clear. However, considerable time and effort is spent in order to clarify the requirements. Therefore, a large part of the problem solving activity in Design Thinking consists of problem definition [32].

Whereas Problem Solving is at the core of AI research [24] and the interplay between AI and design research has been widely investigated [17], the question of whether machines can design still remains little addressed [11,13]. This paper addresses the question with a particular reference to Design Thinking in the realms of Architecture and Urban Planning.

The paper is structured as follows. Section 2 presents Design Thinking as a creative process for Problem Solving. Section 3 clarifies the difficulties of Design Thinking in Architecture and Urban Planning. Section 4 reports recent advances from the field of Computational Creativity which could affect AI research on Design Thinking. Section 5 outlines possible directions of AI research on Design Thinking in Architecture and Urban Planning. Section 6 concludes the paper with final remarks that reflect my position on the question in hand.

[1] http://www.ideo.com/
[2] http://dschool.stanford.edu/

2 Creative Problem Solving with Design Thinking

Principles. Christoph Meinel and Larry Leifer [28] assert that there are four rules to Design Thinking:

The human rule all design activity is ultimately social in nature
The ambiguity rule design thinkers must preserve ambiguity
The re-design rule all design is re-design
The tangibility rule making ideas tangible always facilitates communication

Suitable process models for Design Thinking should follow these principles.

Process. According to the Stanford's d.school, the Design Thinking process consists of the following 5 steps:

1. Understanding users' needs [EMPATHIZE].
2. Framing problems as opportunities for creative solutions [DEFINE].
3. Generating a range of possible solutions [IDEATE].
4. Communicating the core elements of solutions to others [PROTOTYPE].
5. Learning from users' feedback to improve solutions [TEST].

These steps are compliant with the aforementioned rules.

At the core of this process is a bias towards *action* and *creation* which, if repeated iteratively, allow the design thinker to refine his/her initial ideas until they are considered satisfactory from the point of view of the intended users. Also, because of Design Thinking's parallel nature, there are many different paths through the phases. This is part of the reason Design Thinking may seem to be "ambiguous" when compared to more analytical, Cartesian methods of science and engineering. Design thinkers also use *divergent thinking* and *convergent thinking* to explore many possible solutions [12]. Divergent thinking is the ability to offer different, unique or variant ideas adherent to one theme while convergent thinking is the ability to find the "correct" solution to the given problem. Design thinking encourages divergent thinking to ideate many solutions (possible or impossible) and then uses convergent thinking to prefer and realize the best resolution. The "a-ha moment" is the moment where there is suddenly a clear forward path. It is the point in the cycle where synthesis and divergent thinking, analysis and convergent thinking, and the nature of the problem all come together and an appropriate resolution has been captured.

Methods. Although design is always influenced by individual preferences, Design Thinking methods share a common set of traits, mainly: *creativity*, ambidextrous thinking, teamwork, empathy, curiosity and optimism [14]. Methods include interviewing, defining user profiles, looking at other existing solutions, developing prototypes, mind mapping, asking questions like the five whys and situational analysis. Higher-order and obscure relationships, typically occurring in wicked problems, are usually addressed through the use of *analogies* [19]. An understanding of the ill-defined elements of the situation, or the expected results, or lack of domain-related knowledge for the task, may be developed by correlating different internal representations, such as *images* [27].

3 Design Thinking in Architecture and Urban Planning

Design Thinking has been deeply investigated in Architecture and Urban Planning since early 1970s. In particular, the work of psychologist, architect and design researcher Bryan Lawson has contributed to the understanding of the distinguishing features of Design Thinking with respect to other forms of Problem Solving [21]. The fame of Lawson arises from an empirical study conducted in 1972 to investigate the difference between *problem-focused* solvers and *solution-focused* solvers. He took two groups of students (final year students in architecture and post-graduate science students) and asked them to create one-layer structures from a set of colored blocks so that the perimeter of the structure had to optimize either the red or the blue color. However, there were unspecified rules governing the placement and relationship of some of the blocks (incomplete problem statement). By observing the solution approaches adopted by the two groups, Lawson found that the scientists are problem-focused solvers whereas designers are solution-focused solvers [21]. In 1973 Horst Rittel and Melvin Webber showed that design and planning problems are wicked problems as opposed to tame problems of science [31]. Later on, Nigel Cross concluded that Lawson's studies suggest that scientists problem solve by *analysis*, while designers problem solve by *synthesis* [9]. Actually, Design Thinking uses both analysis and synthesis. Every synthesis is built upon the results of a preceding analysis, and every analysis requires a subsequent synthesis in order to verify and correct its results. Pieter Pauwels, Ronald De Meyer, and Jan Van Campenhout [30] suggest that creative Design Thinking in Architecture rests on a cyclic combination of reasoning processes based on *abduction*, *deduction*, and *induction*.

Methods and approaches used by architects and urban planners were extensively described by Peter Rowe in his 1987 book on Design Thinking [32]. In recent years, as a consequence of a number of dramatic scientific discoveries (notably, the notion of *embodiment* from *neurosciences*), traditional arguments such as "nature versus nurture" [10] are rapidly disappearing because of the realization that just as we are affecting our environments, so too do these altered environments restructure our cognitive abilities and outlooks. If the biological and technological breakthroughs are promising benefits such as extended life expectancies, these same discoveries also have the potential to improve in significant ways the quality of our built environments. This poses a compelling challenge to conventional architectural theory. Drawing upon a wealth of research, Harry F. Mallgrave [26] argues that architects should turn their focus away from the objectification of architecture (*i.e.*, treating architectural design as the creation of objects) and redirect it back to those for whom they design: the people inhabiting their built environments. Mallgrave is the first to consider the "human rule" (see Section 2) in architectural terms and to question what implications the discussions taking place in philosophy, psychology, biology, anthropology, and neurosciences hold for architectural design.

In architectural design, *creativity* is highly valued. Although several methods for stimulating creativity are available in the literature, they are rarely formally present in the architectural design process. Also, the assessment of creativity is an open issue, mainly due to the lack of an unambiguous disciplinary definition of creativity. However, some progress has been done as shown in the next section.

4 Advances from Computational Creativity Research

Creativity is a phenomen typically characterized in terms of its product. In particular, it results in products that are (i) novel, (ii) useful or valuable, and (iii) non-obvious, unexpected or surprising. *Computational Creativity* (CC) concerns the use of computers to generate results that would be regarded as creative if performed by humans alone. More precisely, the goal of CC is to model, simulate or replicate creativity using a computer, to achieve one of several objectives:

- To construct a program or computer capable of human-level creativity.
- To better understand human creativity and to formulate an algorithmic perspective on creative behavior in humans.
- To design programs that can enhance human creativity without necessarily being creative themselves.

A prophecy of the advent of CC can be traced back to over 170 years ago, when Ada Lovelace said of Charles Babbage's Analytical Engine that it "might compose elaborate and scientific pieces of music of any degree of complexity or extent" [29]. However, CC includes not only the arts, but also, *e.g.*, innovative scientific theories and engineering design.

Creativity was identified as one of the primary goals of AI in the Dartmouth proposal. However, the pioneers of AI ignored the CC challenges because they were interested in modeling mental processes rather than building useful tools. The interest of the AI community in creative machines is now increasing [1]. Simon Colton, Ramon Lopez de Mantaras and Oliviero Stock [7] provide a review of more recent developments in AI research on CC. These developments build partially on psychological studies, socio-psychological studies, socio-cultural studies and phylosophical analysis of creativity. In particular, the work of Margaret Boden has been highly influential [3,4]. Boden's insights have guided work in CC at a very general level, providing more an inspirational touchstone for development work than a technical framework of algorithmic substance. However, notions such as *exploratory creativity* have been more recently formalized and operationalized, most notably in Geraint Wiggins' framework for description, analysis and comparison of creative systems [35].

Colton and Wiggins [8] have called CC the "final frontier" for AI research. However, fundamental research should be done to make AI able to face the challenges of CC. Selmer Bringsjord, Paul Bello, and David A. Ferrucci [5] have already pointed out the inadequacy of the Turing Test in the case of creative machines. A better test is one that insists on a certain restrictive epistemic relation between an artificial agent (or system) A, its output o, and the human architect H of A a relation which, roughly speaking, obtains when H cannot account for how A produced o. This test was called the *Lovelace Test* in honor of Ada Lovelace, who believed that only when computers originate things should they be believed to have minds.

5 Directions of AI Research on Design Thinking

In this Section I mention increasingly challenging directions for AI research on Design Thinking in Architecture and Urban Planning.

Developing intelligent Geographical Information Systems (GIS) and Computer-Aided Design (CAD) systems. An early work on AI in Urban Planning is INGENS, a prototypical GIS which integrates *machine learning* tools to assist planners in the task of topographic map interpretation [25]. It can be trained to learn operational definitions of geographical objects that are not explicitly modeled in the database. Carl Schultz and Mehul Bhatt [33] present a multimodal spatial data access framework designed to serve the informational and computational requirements of CAD systems that are intended to provide intelligent spatial *decision support* and analytical capabilities in Architecture. Bhatt *et al.* [2] interpret (structural) form and (artefactual) function by specifying modular *ontologies* and their interplay for the Architectural Design domain. They also demonstrate how their ontological modelling facilitates the conceptual modelling of requirement constraints in Architectural Design.

Pushing the "human rule". As mentioned in Section 3, Mallgrave promotes a *user-centered design* approach in Architecture. But how to make AI systems for Design Thinking compliant with the "human rule"? This can be achieved by applying AI techniques for *sentiment analysis*, *opinion mining* and *preference learning.* A recent attempt in this direction is the proposal of an approach to rank buildings through the automated analysis of Flickr metadata on the Web with the aim of measuring the public perception of particular building types (airports, bridges, churches, halls, and skyscrapers) [16]. Learning from user preferences is also central in a very recent AI application to a problem relevant in Urban Planning, *i.e.* the definition of an integrated touristic plan for urban areas [23].

Towards creative systems. The pioneer of CC in Architecture was John Frazer, whose work - as a student - on CAD and "intelligent environments won an Architectural Association prize as early as 1969. With his wife Julia Frazer, he went on to provide more elaborate computer-generated (and eventually interactively evolved) designs for buildings and urban centres [15]. He investigates the fundamental form-generating processes in Architecture, considering architecture as a form of artificial life, and proposing a genetic representation in a form of DNA-like code-script, which can then be subject to developmental and evolutionary processes in response to the user and the environment. After Frazer, other researchers have taken inspiration from nature in order to enhance CAD systems with some creative capabilities [18]. Creative CAD systems are intended to go beyond the abovementioned intelligent CAD systems since they aim at *machine creativity* rather than at *machine-supported human creativity* [22]. Ashok Goel *et al.* [20] envision that the next generation of knowledge-based CAD systems will be based on cognitive accounts of design, and will support collaborative design, conceptual design, and creative design.

6 Final remarks

In this paper I have addressed the question of whether machines can design. This capability is intertwined with the ability to autonomously pursue a Design Thinking process, *e.g.*, the one suggested by the d.school and briefly described in Section 2. So, the original question can be reformulated as to whether AI will ever support Design Thinking, meaning both the process and all the reasoning tasks involved in the process. Available AI techniques - notably those developed in the areas of sentiment analysis, opinion mining and preference learning as mentioned in Section 5 - can support the steps [EMPATHIZE] and [DEFINE] in the process. As for the last two steps in the Design Thinking process (*i.e.*, [PROTOTYPE] and [TEST]), rapid prototyping is now possible with 3D-printing technologies which have been developing at an amazing pace. Neri Oxman, architect and founder of the Mediated Matter group at the MIT Media Lab,[3] argues that *digital fabrication* is ushering in a third era of construction technology. There are still many limitations, such as the range of materials you can use, the maximum size you can print at and the speed of the process. However, as testified by the pioneering work of engineer Enrico Dini with his architectural-scale 3D-printer *D-Shape*,[4] in the near future we might print not only buildings, but entire urban sections. So, the communication of the design solutions to the users will become easier and faster than it is nowadays. Last but not least, in order to fulfill the requirements of a machine capable of Design Thinking, it is necessary to cover also the central step of the process, namely [IDEATE]. This implies that one such machine, besides being intelligent, should be also creative.

Current research in AI testifies a great effort towards the development of creative machines. So, as opposed to Pauwels *et al.* [30], I am not skeptical about the possibility that AI could support Design Thinking, even in challenging domains such as Architecture and Urban Planning. Good news come also from the field of CC. According to Boden [4], high levels of creativity result from the transformation of a conceptual space. In [35], Wiggins mentioned that Boden's *transformational creativity* could be achieved computationally by extending his framework so that search could range over the possible traversal and evaluation functions, as well as the conceptual spaces defined by each such choice. Such an extension would correspond to instatiating the design processes, more precisely the phases of divergent thinking. So it is very likely that next-generation AI systems will include more and more of the processes peculiar to Design Thinking, resulting also in an augmented perception of their creativity.

Summing up, I am enclined to think that creative machines for Design Thinking in Architecture and Urban Planning could be obtained by equipping upcoming mega-scale 3D-printers with software that combines the facilities of a CAD system with the automated inferences of AI-based reasoning engines and the generative capabilities of CC tools.

[3] `https://www.media.mit.edu/people/neri`
[4] `http://www.d-shape.com/index.htm`

References

1. Besold, T.R., Schorlemmer, M., Smaill, D. (eds.): Computational Creativity Research: Towards Creative Machines, Atlantis Thinking Machines, vol. 7. Springer (2015)
2. Bhatt, M., Hois, J., Kutz, O.: Ontological modelling of form and function for architectural design. Applied Ontology 7(3), 233–267 (2012), http://dx.doi.org/10.3233/AO-2012-0104
3. Boden, M.A.: Creativity and artificial intelligence. Artificial Intelligence 103(1-2), 347–356 (1998), http://dx.doi.org/10.1016/S0004-3702(98)00055-1
4. Boden, M.A.: The Creative Mind - Myths and Mechanisms (2. ed.). Routledge (2003)
5. Bringsjord, S., Bello, P., Ferrucci, D.A.: Creativity, the Turing Test, and the (better) Lovelace Test. Minds and Machines 11(1), 3–27 (2001), http://dx.doi.org/10.1023/A:1011206622741
6. Buchanan, R.: Wicked problems in design thinking. Design Issues 8(2), 5–21 (1992)
7. Colton, S., López de Mántaras, R., Stock, O.: Computational creativity: Coming of age. AI Magazine 30(3), 11–14 (2009), http://www.aaai.org/ojs/index.php/aimagazine/article/view/2257
8. Colton, S., Wiggins, G.A.: Computational creativity: The final frontier? In: De Raedt, L., Bessière, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., Lucas, P.J.F. (eds.) ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012. Frontiers in Artificial Intelligence and Applications, vol. 242, pp. 21–26. IOS Press (2012), http://dx.doi.org/10.3233/978-1-61499-098-7-21
9. Cross, N.: Designerly ways of knowing. Design Studies 3(4), 221–227 (1982)
10. Cross, N.: The nature and nurture of design ability. Design Studies 11(3), 127–140 (1990)
11. Cross, N.: Can a machine design? Design Issues 17(4), 44–50 (2001)
12. Cross, N.: Designerly Ways of Knowing. Springer (2006)
13. Cross, N.: Design Thinking: Understanding How Designers Think and Work. Oxford: Berg Publishers (2011)
14. Faste, R.A.: Ambidextrous thinking. In: Innovations in Mechanical Engineering Curricula for the 1990s. American Society of Mechanical Engineers (1994)
15. Frazer, J.H.: An Evolutionary Architecture. Architectural Association: London (1995)
16. Gadiraju, U., Dietze, S., Diaz-Aviles, E.: Ranking buildings and mining the web for popular architectural patterns. In: Proceedings of the ACM Web Science conference, Oxford, UK, June 28 - July 1, 2015. (2015)
17. Gero, J.S.: AI EDAM at 20: Artificial intelligence in designing. AI EDAM 21(1), 17–18 (2007), http://dx.doi.org/10.1017/S0890060407070084
18. Goel, A.K.: Design, analogy, and creativity. IEEE Expert 12(3), 62–70 (1997)
19. Goel, A.K., Shu, L.H.: Analogical thinking: An introduction in the context of design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 29(2), 133–134 (2015), http://dx.doi.org/10.1017/S0890060415000013
20. Goel, A.K., Vattam, S., Wiltgen, B., Helms, M.E.: Cognitive, collaborative, conceptual and creative - four characteristics of the next generation of knowledge-based CAD systems: A study in biologically inspired design. Computer-Aided Design 44(10), 879–900 (2012), http://dx.doi.org/10.1016/j.cad.2011.03.010

21. Lawson, B.R.: How Designers Think: The Design Process Demystified. Architectural Press, Oxford (1980)
22. Lawson, B.R.: CAD and creativity: does the computer really help? Leonardo 35(3), 327–331 (2002)
23. Lisi, F.A., Esposito, F.: An AI application to integrated tourism planning. In: Gavanelli, M., Lamma, E., Riguzzi, F. (eds.) AI*IA 2015, Advances in Artificial Intelligence - XIVth International Conference of the Italian Association for Artificial Intelligence, Ferrara, Italy, September 23-25, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9336, pp. 246–259. Springer (2015), http://dx.doi.org/10.1007/978-3-319-24309-2_19
24. Luger, G., Stubblefield, W.: Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Benjamin/Cummings (2004)
25. Malerba, D., Esposito, F., Lanza, A., Lisi, F.A., Appice, A.: Empowering a GIS with inductive learning capabilities: the case of INGENS. Computers, Environment and Urban Systems 27(3), 265–281 (2003)
26. Mallgrave, H.F.: Architecture and Embodiment: The Implications of the New Sciences and Humanities for Design. Routledge (2013)
27. McKim, R.: Experiences in Visual Thinking. Brooks/Cole Publishing Co. (1973)
28. Meinel, C., Leifer, L.: Introduction. In: Leifer, L., Plattner, H., Meinel, C. (eds.) Design Thinking Research - Building Innovation Eco-Systems. Springer (2014)
29. Menabrea, L.F.: Sketch of the Analytical Engine invented by Charles Babbage. Scientific Memoirs 3 (1843)
30. Pauwels, P., De Meyer, R., Van Campenhout, J.: Design thinking support: Information systems versus reasoning. Design Issues 29(2), 42–59 (2013)
31. Rittel, H.W., Webber, M.M.: Dilemmas in a general theory of planning. Policy Sciences 4(2), 155–169 (1973), http://dx.doi.org/10.1007/BF01405730
32. Rowe, G.P.: Design Thinking. The MIT Press (1987)
33. Schultz, C.P.L., Bhatt, M.: Multimodal spatial data access for architecture design assistance. AI EDAM 26(2), 177–203 (2012), http://dx.doi.org/10.1017/S0890060412000066
34. Simon, H.: The Sciences of the Artificial. The MIT Press (1969)
35. Wiggins, G.A.: A preliminary framework for description, analysis and comparison of creative systems. Knowedge-Based Systems 19(7), 449–458 (2006), http://dx.doi.org/10.1016/j.knosys.2006.04.009

Design knowledge representation: An ontological perspective

Emilio M.Sanfilippo[1,2], Claudio Masolo[1], and Daniele Porello[1]

[1] Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy
[2] Ph.D. School in ICT, University of Trento, Trento, Italy

Abstract. We present a preliminary high-level formal theory, grounded on knowledge representation techniques and foundational ontologies, for the uniform and integrated representation of the different kinds of (qualitative and quantitative) knowledge involved in the designing process. We discuss the conceptual nature of engineering design by individuating and analyzing the involved notions. These notions are then formally characterized by extending the DOLCE foundational ontology. Our ultimate purpose is twofold: (*i*) to contribute to foundational issues of design; and (*ii*) to support the development of advanced modelling systems for (qualitative and quantitative) representation of design knowledge.

Keywords: Design model, integration, conceptual space, DOLCE

1 Engineering design: from requirements to models

Despite the variety of definitions and theories for engineering design, it is common to understand it as the activity of producing a full description of the technical product to be realised that satisfies some (market) requirements [1] (cf. Figure 1). In this sense designing is a creative activity of *conceptualization* and *modelling*: designers have to think out (possibly innovative) practical solutions to satisfy customer needs and to release technical specifications[1] for the production of physical goods [2]. The designing process comprises different tasks, from the elicitation of customers' requirements to conceptual embodiment and detail design [3]. During each phase cross-functional expert teams work on different modelling aspects of the same design project. Consequently, various models are firstly obtained with respect to the required level of detail and content, and secondly integrated into an all-comprising representation of the product [4].

The specification of the requirement list about the properties that the future product has to satisfy plays a fundamental role, since it represents the document on which both the overall designing process is firstly based, and secondly evaluated [1]. According to Pahl and colleagues [3], (product's) properties in a requirement list can be meant either as *demands*, if they have to be necessarily met, or as *wishes*, if they should be taken into account whenever possible.

[1] 'Technical specification', 'product model' and 'design model' will be interchangeably used throughout the paper.

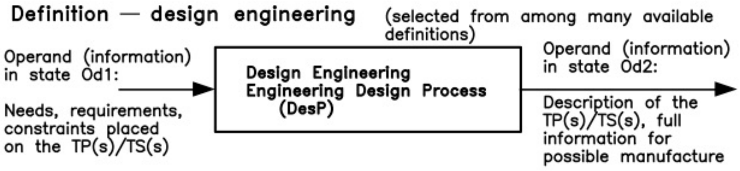


Fig. 1. Design engineering definition [1]. TP and TS are interchangeably used for "technical product" and "technical system".

After the requirement list has been completed, experts start specifying the design solution by means of various technical specifications. Examples include:

- *Functional model*: the main functionality of the product under design is represented and decomposed into sub-functionalities (e.g. [3, 169-181]);
- *Component model*: (also called *part model*) describes what constructional parts are required [5, 6]. In some theories of design, *organ models* are used to represent structural elements carrying a functionality [7];
- *Assembly model*: specifies spatial relationships holding among components [8]. The most common assembly relationship is *part-of* holding between components; *connection* is also used to represent physical connection [9];
- *Material model*: represents the type of material used for each component in a product. It can also be used to specify material's properties like stress resistance, or malleability [7];
- *Geometry model*: describes the product shape, typically together with its nominal dimensions and tolerances [5].

These are however only a few cases; at the current state of the art there is no complete, nor standardised list of models used in engineering design [6].

The importance of technical specifications is due to the fact that they determine the final properties that particular objects have to satisfy to be considered as products of a certain type. If a particular object satisfies the properties specified by its corresponding model, it is said *to conform to* the model [4].

This brief introduction suggests that the main core of designing is the development of (design) *concepts*, rather than their codified *description*, as suggested by Figure 1. It also shows that various quantitative but also qualitative knowledge aspects have to be considered. For instance, the functional aspects—a sort of teleological information about the product—have often a qualitative nature. Furthermore, at the early phases of the development of a product, the requirements and the characteristics of the product, including the geometrical ones, are not precise. The designer still has a quite general idea of the definitive form of the product and vague tolerances are accepted. The need to enrich quantitative product models with qualitative specifications about the design intents has been advocated for more than 20 years now [2]. However, computer-based modelling systems are mainly focused on quantitative knowledge, whereas qualitative aspects are mainly expressed by text annotation for human reading. As a consequence, design relevant models are not (computationally) represented [2].

The ultimate purpose of this work is twofold: (i) to contribute to foundational issues of engineering design; and (ii) to support the development of advanced modelling systems for design knowledge. However, in order to achieve these goals, it is needed beforehand a clear understanding of what a product model is and how it relates to its corresponding physical products. We thus present in this paper an high level approach, based on foundational ontologies and knowledge representation techniques, concerned with the individuation and analysis of the general notions needed for an integrated model of design. Our proposal offers just a conceptual base that needs to be specialized and instantiated to be useful for the practical purposes of knowledge representation in design.

The paper is structured as follows: in Section 2 we introduce the proposed formal model and we discuss some shortcomings and problematic issues related to design knowledge representation. In section 3 we briefly compare our approach with similar research initiatives.

2 Representing design product knowledge

We sketch a *general and high-level* theory that is capable of representing in a uniform and integrated way both the qualitative and the quantitative aspects of design. Manufacturing aspects are excluded from this analysis, even though our model is compatible with a future extension that addresses these aspects. First-order logic (FOL) is employed as representational language. This has clearly some impacts on the technics, in particular *reification* technics, used to overcome some limitations in the expressive power.

2.1 The conceptual nature of design

We want to distinguish the design of a product from both its specifications, such as a Computer-Aided Design (CAD) file or a printed drawing on a piece of paper, and from the set of objects that are possibly produced. For this purpose, it is worth presenting our point by means of the *semiotic triangle* developed in semantics. For instance, the word 'red' in its predicative sense is related to the concept of *being red* (the *intension*) and to the class to things that are red (the *extension*). In the case of a design, we distinguish three aspects of it (cf. Figure 2): its *specifications* as the physical supports of the design, the *product type* as its intension, and the class of *physical products*, or simply products, as the extension of the product type. For example, John's car (particular physical product) satisfies the technical specification of Ferrari Testa Rossa 512 TR (product type).

Firstly, the product type is distinguished from the specification, as the same product type may be represented by different supports, e.g., a CAD model and a pencil-made sketch. The specification is useful to represent and communicate aspects of the product type, but the design cannot be reduced to it.

Secondly, the product type is not identifiable with the class of products that can be realized by means of the design. This view copes with the fact that the

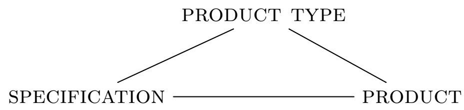


Fig. 2. Semiotic triangle for design

design may be about not yet realized products. If the design reduces to a class of products, then, before their realizations, one could only talk about the design in terms of future or possible objects. Although this view may be formalized by means of modal logic, we believe that it does not capture the nature of design. In principle, a design may exist even if the objects that are its realizations are never produced. This does not mean that a design may be about impossible objects: the point is that a design still exists, we can understand it, and practitioners can interact about it, even in the case that no concrete product is ever going to be realized. By focusing on the product type, we indeed construe the design as a *type* rather than a set of *tokens* (physical objects). Accordingly, we shall treat the product type as a *concept* in the sense made precise by the axioms we shall introduce in the next sections. From this perspective, our interpretation of the designing process departs from the one depicted in Figure 1. The outcome of the process there is understood as the "description of the technical product (TP(s))" plus "full information for possible manufacture" of a class of objects. By contrast, even though we recognize the importance of the specification (cf. Figure 2), we stress here the conceptual nature of the design. In the following sections, we do not discuss specification languages for engineering purposes, nor we will approach the analysis of manufacturing. We will rather formalize the conceptual nature of a design.

For this purpose, we rely on the DOLCE ontology [10]. More specifically, we consider an extension of DOLCE for representing roles [11] and the evolution of its core, called DOLCE-CORE [12]. These extensions are significant to our aim because they explicitly address the problem of representing the *intension* of concepts or properties. However, these theories are not enough to represent some important aspects of the design; hence, in what follows, we modify and extend them. In a first-order (and non-modal) setting, properties (and concepts) are usually represented by means of *predicates*. The *method of temporal arguments* [13] is a standard technique to account for (i) the dependence on time of the classification and (ii) the representation of change of objects through time. It consists of the introduction of time as an additional argument of the predicates and functions in the vocabulary that depend on time. For instance, $\texttt{Red}(x)$ becomes $\texttt{Red}(x, t)$. However, as noted in [12], this technique is not adequate to represent neither the contextual, social, or constructive aspects of concepts, nor their intensions.

To address these crucial aspects in FOL, we *reify* concepts into the domain of quantification, i.e., we introduce a new kind of entities: $\textsc{cn}(x)$ stands for "x is a *concept*" (or more generally a property). In this way, we can predicate on concepts, but we loose the possibility of representing classification via predication. It is then necessary to introduce a new primitive to relate the concepts to the entities they classify, a sort of (possibly intensional) 'instance-of' relation, that here we call *classification*: $\mathtt{CF}(x, y, t)$ stands for "the concept x classifies the entity y *as it is at time* t". From a design perspective $\mathtt{CF}(x, y, t)$ can be interpreted in a more restricted way as "the *product* y, as it is at time t, conforms to (the specification of) the *product type* x", or, more lengthily, "at time t, the *physical product* y has all the properties required to satisfy (the specification of) the *product type* x". The classification relation is temporally qualified: the entity y may change through time, thus the classification under a certain concept is, in general, contingent. For instance, a product y that at time t conforms with the (specifications of the) type x, could not conform anymore with x at time t' because of the loss of some properties necessary to be classified under x. That means that t does not individuate the time at which the classification is done but the time at which y is considered (measured, perceived, etc.). Consequently, the classification at time t does not necessarily implies the existence at t of the concept, see (a1) where $\mathtt{EX}(y, t)$ stands for "the entity y exists at time t". However, concepts are in time (a2), they can be created or destroyed (for instance, when no specification, including mental ones, exists any longer). Concepts created at a given time t can then classify, according to past information, entities that do not exist anymore at t.[2]

A similar modelling approach, even though (change through) time is not considered, is employed in the Industry Foundation Classes[3] (IFC)—a data modelling standard supported by most of the major CAD vendors—in which classification is called `IfcRelDefinesByType`. This relationship is used to represent the fact that the instances of a `IfcTypeProduct` (a product type) satisfy the properties defined by a `IfcProduct` (a physical product).

Standard extensional relations between concepts can be introduced relying on `CF`, e.g., the *extensional subsumption* relation (d1). However, concepts have an intensional nature, they are not reducible to their extensions, different concepts may classify the same entities: the extensional subsumption is not antisymmetric, i.e., $x \subseteq y \wedge y \subseteq x \rightarrow x = y$ does not hold in our theory.

a1 $\mathtt{CF}(x, y, t) \rightarrow \mathrm{CN}(x) \wedge \mathtt{EX}(y, t)$
a2 $\mathrm{CN}(x) \rightarrow \exists t(\mathtt{EX}(x, t))$
d1 $x \subseteq y \triangleq \forall zt(\mathtt{CF}(x, z, t) \rightarrow \mathtt{CF}(y, z, t))$

2.2 The intensional dimension of design

For the sake of example, assume that in our domain all the round entities are also red and vice versa. In our framework, this does not imply the identity of the concepts (properties) *being red* and *being round*. The distinction between the extensional and the intensional level of design is also used in design models and

[2] For our goal, the time at which the classification is done is not relevant. However it is easy to add a second temporal argument to `CF` to account for that.
[3] http://www.buildingsmart-tech.org/ifc/IFC2x4/rc4/html/

data modelling standards. In the IFC, for example, `IfcTypeProduct` is understood as the information common to all instances of `IfcProduct`. Borgo and colleagues [14] propose to understand the former in an intensional sense, i.e., as the properties characterising the instances of the latter.

After Carnap, the intensionality of properties is traditionally approached in logic from a modal perspective: the co-extensionality of *being red* and *being round* is contingent, it holds in the *actual world* but there are other *possible worlds* where *being red* and *being round* have different extensions. In our theory, the reference to possible worlds is not necessary although, indeed, one can still have an informal modal understanding of intensionality. Here, the intension is captured in a different way that is especially effective for products types.

The idea is that concepts cannot be characterized "in isolation", they always refer to other concepts. For instance, product types are usually characterized in terms of simpler concepts that are typically shared by the designers involved in a given phase, the common background of designers. This idea is quite similar to the one followed in [11], where the *identity criteria* for concepts are based on their *definitions*. However, in [11] the definitions of concepts are a sort of "placeholders" of their intensions, they are not structured and they are very weakly interlinked (by the notion 'used'). Here we use the DOLCE-CORE *quality spaces*—a formal variation of *conceptual spaces* [15]. The idea is to make (partially) explicit what characterizes a concept in intensional terms by linking it to some quality spaces.

Objects can be compared and characterized in terms of a variety of aspects such as weight, shape, size, color, function, etc. Such aspects are represented via (quality) spaces composed by *basic properties* (called *regions* in DOLCE-CORE).[4] For instance, the color space contains several basic properties, e.g., *being red*, *being orange*, *being scarlet*, etc. Basic properties in the same space can be organized in taxonomies or in more sophisticated ways: from ordering (weight, size) to complex topological or geometrical relations (color splinter).

We assume a finite number N of spaces SP_i that partition the basic properties BCN (a3)–(a5).[5] The idea is that basic properties—that still have an intensional nature—are used, but not created, during the design process, i.e., they represent the conceptual knowledge in the background, the conceptual knowledge that allows the product type to be characterized throughout the design phases.[6] Consequently, the intensional subsumption relation $\sqsubseteq$, assumed to be a classical extensional mereology closed under sum [16], is also local to spaces (a6). Ba-

[4] Quality spaces correspond to the *dimensions* of conceptual spaces in [15].

[5] This implies that basic properties are local, private, to spaces. This choice can be debated if one assumes that colors can be organized in different ways, or that spaces are associated to instruments with different resolution. In this case, it seems reasonable to assume that spaces share basic properties. Here we prefer to duplicate the basic properties, given their quite clear conceptual nature in these cases, and add *correspondence* links between them.

[6] As in the case of correspondence links, one can think that there are other intensional (logical) links among these basic properties (in the same or in different spaces). This is a very interesting aspect that, for reasons of space, we do not consider here.

sic properties are concepts, therefore they classify entities; (a7) establishes the link between the intension and the extension of concepts (the vice versa does not hold). Firstly, note that, differently from DOLCE-CORE, the *individual qualities*—a sort of tropes, e.g., the redness of my car—are not in the domain of quantification anymore.[7] Secondly, and more importantly, in DOLCE-CORE when an object has a quality, then this quality is completely determined.[8] For instance, a *uniformly* colored object is always mapped to an *atomic* basic property, a maximally specified property, the maximal information one disposes of (according to the resolution assumed in the space). A *multi-colored* object is mapped to a non-atomic property. However, this property is just the sum of the colors of all its uniformly colored parts. Differently, to account for underspecification and tolerances, a uniformly colored object could here be mapped to a *non*-atomic basic property; its color is just one of the atomic parts of the non-atomic property. This *disjunctive* reading is more useful in the design process, especially during the first phases when one has only a qualitative and rough characterization of the product type. This means that the basic properties need to be interpreted as the properties of the whole object under classification. The properties of the parts and of the structural aspects of a product type are quite problematic and will be briefly discussed in Section 2.3.

As we saw, the product types are complex concepts cCN (a8) characterized in terms of basic properties. We need then to link complex concepts to their basic properties: $\mathtt{CH}(x, y)$ stands for "the complex concept x is characterized by the basic property y" (a9). For example, Ferrari Testarossa 512 TR (product type) is characterised by the basic properties *being red*, *being 1500kg heavy*, among others. Complex concepts are characterized at least by two, but usually several, basic properties, i.e., they have a multi-dimensional nature (a10).[9] This is similar to composition of spaces in more complex ones, at least if the geometry of the complex space can be defined in terms of the geometries of its components.

a3 $\text{bCN}(x) \to \text{CN}(x)$
a4 $\text{bCN}(x) \leftrightarrow \bigvee_{i\in\{1,\dots,\text{N}\}} \text{SP}_i(x)$
a5 $\bigwedge_{i\neq j\in\{1,\dots,\text{N}\}}(\text{SP}_i(x) \to \neg\text{SP}_j(x))$
a6 $x \sqsubseteq y \to \bigwedge_{i\in\{1,\dots,\text{N}\}}(\text{SP}_i(x) \leftrightarrow \text{SP}_i(y))$
a7 $x \sqsubseteq y \to x \subseteq y$
a8 $\text{cCN}(x) \to \text{CN}(x) \wedge \neg\text{bCN}(x)$
a9 $\mathtt{CH}(x, y) \to \text{cCN}(x) \wedge \text{bCN}(y)$
a10 $\text{cCN}(x) \to \exists yz(\mathtt{CH}(x, y) \wedge \mathtt{CH}(x, z) \wedge \bigvee_{i\in\{1,\dots,\text{N}\}}(\text{SP}_i(y) \wedge \neg\text{SP}_i(z)))$
a11 $\text{cCN}(x) \to (\mathtt{CF}(x, y, t) \leftrightarrow \forall z(\mathtt{CH}(x, z) \to \mathtt{CF}(z, y, t)))$

[7] This option has already been considered in [17].
[8] In [15] objects are points in multi-dimensional spaces. Objects are then fully characterized with respect to all the possible qualities.
[9] Actually this is also the case of some basic properties, e.g., the color space has three dimensions: hue, saturation, and brightness. One could also think that colors (or, better, shapes) may be *designed*, i.e., there exist some original or proprietary color-properties. We do not consider these aspects that, however, could be modeled by extending CH or by assuming hue-, saturation-, and brightness-properties as basic.

Firstly, we need to guarantee that the classification under a complex concept x reduces to the classification under all the basic properties that characterize x (a11), i.e., the extension of x is the intersection of the extensions of all the basic properties that characterize x. This seems to authorize to interpret CH as a sort of intensional subsumption. However, the antisymmetry of $\sqsubseteq$ would imply the identity of complex concepts with the same characterizing basic properties. This is acceptable only if we assume that the characterization of complex concepts is always complete. For the moment, we prefer a weaker approach that allows also for partial characterizations of concepts, i.e., two different concepts can have the same partial characterization. Note that these partial characterizations are particularly interesting for hiding very specific or practical properties.

Secondly, while it seems quite reasonable to have a static view on the background knowledge, the product type under design could be intended as an evolving concept. This can be modeled by adding a temporal parameter to both CH and CF, i.e., what characterizes a concept can vary through time, thus the classification depends also on the time at which the concept is considered: $\mathtt{CF}(x,t,y,t')$ stands for "the complex concept x, *as it is at* t, classifies the object y, *as it is at* t'" (a12). The double temporal qualification allows to reclassify an object as it is at t' across the evolution through time of a given concept.[10] (a11) needs then to be substituted by (a13) where we assume the basic properties to be static. Consequently, the extensional subsumption relations between complex concepts may be temporally qualified as in (d2). The identity of concepts can be intended in terms of the 'trajectory' across time of its characterizing properties, i.e., $(\mathtt{CH}(x,z,t) \leftrightarrow \mathtt{CH}(x',z,t)) \rightarrow x = x'$, but weaker options that consider the designers and/or the design process can be considered. For this reason, we do not commit to this identity criterion.

a12 $\mathtt{CF}(x,t,y,t') \rightarrow \mathrm{cCN}(x) \wedge \mathtt{EX}(x,t) \wedge \mathtt{EX}(y,t')$
a13 $\mathrm{cCN}(x) \rightarrow (\mathtt{CF}(x,t,y,t') \leftrightarrow \forall z(\mathtt{CH}(x,z,t) \rightarrow \mathtt{CF}(z,y,t')))$
d2 $x\,{}^{t_1}\!\subseteq^{t_2} y \triangleq \mathrm{cCN}(x) \wedge \mathrm{cCN}(y) \wedge \forall zt(\mathtt{CF}(x,t_1,z,t) \rightarrow \mathtt{CF}(y,t_2,z,t))$

At this point we can also address the notion of requirement that is usually defined as the "[p]roperty that is required to be fulfilled during the origination phase of the object to satisfy the [customer] requirements" [1, p.317]. Customer requirements can then be seen as the (maybe rough) idea of product that the customers, as opposed to designers, have. Note that also the requirements, i.e., the customer concept, can evolve in time during the design process, maybe because of market change, or maybe because of the interaction with engineers that discovered some unrealizable constraints. We have then two complex concepts characterized in terms of intensions (their characterizing basic properties) and extensions (the objects that they classify). This allows for both an inten-

[10] Firstly, note that t is not the time at which the classification is done, it just 'freezes' the concept (while t' freezes the object). Secondly, to avoid evolving concepts, one could follow [11] and introduce a 'revision' relation between static concepts. Formally the two approaches are equivalent, we preferred the first approach because it seems more adequate for capturing the design practice.

sional and an extensional comparison. One can say that, at t, the requirements, represented by the concept c_r, are satisfied by a product type c_p if ${c_p}^t \subseteq^t c_r$. Requirements are then a sort of necessary properties of the product. However, this could mean that the product type is too specific, i.e., the matching exists only when both ${c_p}^t \subseteq^t c_r$ and ${c_r}^t \subseteq^t c_p$ hold. In any case, the way in which we realized the products can be very different from what the customer requirements report. An intensional matching at t may be defined in terms of CH as $\mathtt{CH}(c_r, x, t) \rightarrow \exists y(y \sqsubseteq x \wedge \mathtt{CH}(c_p, y, t))$, i.e., the basic properties that characterize c_p are intensionally subsumed by the ones that characterize c_r. Again we can strengthen this notion to a perfect matching. Finally, note that one could distinguish two CH relations, one for necessary and one for optional properties of product. This would allow for distinguishing, at the level of requirements, *demands* from *wishes*. In addition, because intensions are expressed in terms of basic properties and any space may have a metric, the level of (mis)matching between requirements and design can be measured.

As an illustrative example, let us consider a customer asking for a product to prepare Italian coffee. In addition to the function, her requirements regard the height (between $16cm$ and $20cm$) and the material (aluminium) of the product. Assuming to dispose of the function, height, and material spaces, the required product type can be represented by a complex concept rpt that, at the starting time $\mathtt{t}_0$ is characterized by three basic properties: $\mathtt{CH}(\mathtt{rpt}, \mathtt{prep\_it\_coffee}, \mathtt{t}_0)$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{16\text{-}20cm}, \mathtt{t}_0)$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{alu}, \mathtt{t}_0)$. Assume that 16-20cm is a non-atomic basic property while alu an atomic one, even though, because properties apply to the whole product, this does not necessarily imply the product to be exclusively made of aluminium. More controversial is the case of prep_it_coffee because it is unclear, for instance, whether the kind of energy used to prepare the coffee is part of the function or pertains a separate space. For our illustrative purposes we assume the first hypothesis and that there are at least two (intensional) subfunctions ($\sqsubseteq$) of prep_it_coffee, namely prep_it_coffee_methane and prep_it_coffee_elect. The designers start to consider the requirements (specified by the customer in some way) by assuming a designed product type dpt (a complex concept) such that $\mathtt{CH}(\mathtt{dpt}, \mathtt{prep\_it\_coffee\_elect}, \mathtt{t}_0)$, $\mathtt{CH}(\mathtt{dpt}, \mathtt{16\text{-}20cm}, \mathtt{t}_0)$, $\mathtt{CH}(\mathtt{dpt}, \mathtt{steel}, \mathtt{t}_0)$. This choice is partially based on the expertise of the company in developing products for induction hobs. Note that the designers know that (at $\mathtt{t}_0$) dpt does not match rpt because steel is not a subconcept of alu. Therefore they interact with the customer to explain the strategical importance of having a product to prepare italian coffee to work with induction hobs, and this constraints the material to steel. The customer agrees on that but change the size-constraints, in this case she wants a very small, less than 12cm high, product. The designers agree on that and at time $\mathtt{t}_1$ the requirements are matched: $\mathtt{CH}(\mathtt{rpt}, \mathtt{prep\_it\_coffee}, \mathtt{t}_1)$, $\mathtt{CH}(\mathtt{rpt}, \leq\mathtt{12cm}, \mathtt{t}_1)$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{steel}, \mathtt{t}_1)$ and $\mathtt{CH}(\mathtt{dpt}, \mathtt{prep\_it\_coffee\_elect}, \mathtt{t}_1)$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{10\text{-}11cm}, \mathtt{t}_1)$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{steel}, \mathtt{t}_1)$.[11]

[11] Here we are totally liberal with respect to how concepts can change through time. However, constraints on the way concepts "move" inside the spaces can be added.

2.3 Advanced and problematic aspects

Dependencies between spaces One interesting aspect of spaces is that it is quite easy to add dependence links between basic properties, i.e., to shape the composed space. This is the case of the color splinter: not all the combinations of hue, saturation, and brightness correspond to a color, a given hue constraints the possible values of brightness and saturation. In this case the consistency of the different characteristics of a model can be guaranteed from the beginning. The proposed framework can be extended to take into account these dependencies. From the designing perspective, this is interesting because it makes possible to represent the dependencies between different designing phases and design specifications by means of the dependencies between the spaces that are used at these phases. For instance, functional modeling is particularly relevant during the first stages of the design process, where more emphasis is given on *what* the product is designed to do, rather than on *how* this purpose will be achieved. Vice versa, at the design embodiment stage, designers focus more on the physical properties of the product, i.e., on how the functionality is realized. Dependencies between functional and physical properties can help the designer in (*i*) understanding how a required function constrains the product's physical layout—a top-down perspective: from the abstract functional view to the concrete physical one; (*ii*) verifying whether the chosen physical layout can, at least in principle, fulfill the required functionality (a bottom-up perspective); (*iii*) making explicit possible *accidental functions*, as opposed to *proper functions*, that could result from (improper) usages of the product. For instance, a screwdriver has the proper function of, it has been designed for, driving screws, while it has the accidental function of opening cans. According to de Vries [18], this analysis is very important to individuate dangerous improper uses of products to be avoided.

Structural knowledge We have already observed the importance of the component and assembly models. Together they allow to represent the *structure* of the product, i.e., how it can be decomposed into simpler products and which assembly constraints hold among these components. The "recursive" decomposition usually stops at standard (at least for a given company) units that are reused in several products. Structural knowledge is then really central to any modern design of complex products, e.g., cars or planes.[12] This structure is usually represented by means of a parthood relation between product types, i.e., by a set of necessary (and sufficient) conditions about the parts of an object. Both UML and ER languages have primitives to represent different kinds of part-whole relations and, in knowledge representation, mereology has now a quite long tradition (see

[12] One could say that the design itself reduces to reuse components already designed. However, as already said some properties like colors or shapes can be designed (actually, structural information seems fundamental to design new shapes). In these cases, it is not clear to us if the designed properties are reducible to original ways of putting together more basic properties or if an *extension* of a space is needed. It would be interesting to analyze creativity in terms of spaces.

[16, 19]). However, in this way, the structure is not intended as a space of properties of the product. Consequently, one lacks a structural similarity or distance that would offer quite precious information to guide the product development, test, and refinement. Recently, there has been a number of proposals to build a conceptual, and cognitively based, space for parthood [20, 21]. These approaches are quite complex and do not really represent the assembly knowledge. More promising, at least in our view, is the idea of representing structures as *patterns* or, more precisely, as *structural graphs*, i.e., labelled and directed graphs which nodes stand for product types and arcs for assembly relations. The construction of these graphs is complex, but it allows to fully capture the structural knowledge and to import or adapt standard measures of similarity between graphs to introduce a metric in the space of structures. In addition, similar graphs could also be useful to represent *relational* properties, i.e., properties that hold because the product has some relations with the external world, e.g., ergonomic properties or affordances. This would be quite relevant for *user-centered* design.

Without structure-spaces, to minimally represent structural constraints, we extend our framework with a temporally qualified classical extensional mereology defined on objects: $\mathtt{P}(x, y, t)$ stands for "the object x is part of the object y at time t". Consider the previous example of the Italian coffee and assume that at time $\mathtt{t_2}$ the designers subdivide the main functionality *prepare Italian coffee* in sub-functionalities *to boil water*, *to store powdered coffee*, *to filter coffee by boiled water*, *to store brewed coffee* and *to serve coffee.* Amongst the various solutions for the embodiment of these functionalities, they decide to develop a moka pot product type (`moka`) with, among the others, three components (that are themselves product types): boiler pot (`pot`), coffee container (`container`), and filter (`filter`) that are characterized (in terms of `CH`) by the first three functions discussed above (plus additional heigh and material properties). In the proposed theory, this structural constraint is represented by (f1), but because a structure space does not exist, it is not possible to understand if this constraint matches the original one about the function of the whole object.

$$\textbf{f1}\ \ \mathtt{CF}(\mathtt{moka}, \mathtt{t_2}, x, t) \rightarrow \exists yzw(\mathtt{CF}(\mathtt{pot}, \mathtt{t_2}, y, t) \wedge \mathtt{CF}(\mathtt{container}, \mathtt{t_2}, z, t) \wedge \mathtt{CF}(\mathtt{filter}, \mathtt{t_2}, w, t) \wedge \mathtt{P}(y, x, t) \wedge \mathtt{P}(z, x, t) \wedge \mathtt{P}(w, x, t))$$

3 Discussion and related work

We presented a high-level modelling approach based on conceptual spaces and ontology engineering methods for the formal representation and integration of (qualitative and quantitative) aspects of design. Foundational issues related to engineering design and the integration of different aspects of design knowledge have been discussed in various research areas, from knowledge representation approaches, to theories of design and philosophy of technology.

The conceptual view proposed in this paper is mostly based on the engineering literature. According to Pahl and colleagues, the goal of designing is "the mental creation of a new product" [3, p.1]. Along the same lines, Lindemann [6] distinguishes between the *content* of a design model (e.g. geometric content) and its specification by means of a representational language (2D geometry). Despite the emphasis on the "mental" side of design given in [3], a design does not have to be confused with its mental representation in our theory. Kroes stresses the difficulty to capture this conceptual nature of design: "When referring to a car design [model], for instance, what is meant is not usually its production plan but something that has more to do with the properties of the car itself [...]. It is not easy to grasp what this 'something' is" [22, p.146]. In section 2.1 concepts are intensionally understood as properties that their corresponding instances have to satisfy. In this sense, a (complex) concept (product type) is a bundle of basic properties defined by designers to satisfy customers' requirements.

In the philosophical analysis of engineering design proposed by Vermaas and colleagues [23], designing tasks are mainly aimed at providing a description for a technical product: "[...] the *core* of technical designing lies in finding a description D_S of an artefact with a physical structure S that is able to effectively and efficiently fulfil a certain function F" [23, p.28] (emphasis ours). In this perspective, D_S is a document specified in a modelling language suited for engineering purposes. The authors thus focus on specification as the core dimension of design, while we have rather focused on its conceptual nature. The two approaches, however, are not contrasting but rather orthogonal, since concepts have to be specified for representational and communication purposes.

Knowledge representation has been primarily focused on the physical makeup of products, while little (if any) attention has been given to design models. For instance, the Core Product Model (CPM) [24] associates the class `artifact` to (requirement) `specification`, meaning that the former has to satisfy the latter's properties. However, neither the difference between requirements properties and design specifications properties is discussed, nor it is clear whether `specification` refers to an encoded description, or its content.

In ontology engineering, the Information Artifact Ontology (IAO) [25] has been explicitly developed to represent information entities, that is, the content of encoded descriptions. The IAO thus shares in some degree the same purpose of the work hereby presented. There are however some relevant distinctions to be noted: (*i*) the IAO directly links a representation to the physical object that it is about. In our approach, concepts mediate this relationship (cf. Figure 2), since they cannot be reduced to their extensions, as argued in Section 2.1. (*ii*) `information content entity` (`concept` in our terminology) is only weakly characterised as an entity that existentially depends on some representation and is about something else. In our theory, there is no such a dependence, since a product type is not necessarily specified in a physical medium. Additionally, we have provided a more detailed characterisation of concepts by taking into account the theories of quality spaces in DOLCE-CORE.

Acknowledgements: This work was partially funded by the VISCOSO project financed by the Autonomous Province of Trento through the "Team 2011" funding programme, and the FourByThree project funded by the European Horizon 2020 program (grant agreement 637095).

References

1. W.E. Eder and S. Hosnedl, *Design Engineering. A Manual for Enhanced Creativity.* CRC Press, 2008.
2. S.K. Chandrasegaran, K. Ramani, R.D. Sriram, I. Horvath, A. Bernard, R.F. Harik, and W. Gao, "The evolution, challenges and future of knowledge representation in product design systems," *Computer-Aided Designd*, vol. 45, pp. 204–228, 2013.
3. G. Pahl, W. Beitz, J. Feldhusen, and K.H. Grote, *Engineering Design. A Systematic Approach.* Springer-Verlag Berlin Heidelberg, 2007.
4. C. Bock, X. Zha, H. Suh, and J. Lee, "Ontological product modeling for collaborative design," *Advanced Engineering Information*, vol. 24, pp. 510–524, 2010.
5. F.L. Krause, F. Kimura, T. Kjelberg, and S.C.Y. Lu, "Product modelling," *Annals of the CIRP*, vol. 42, no. 2, pp. 695–706, 1993.
6. U. Lindemann, "Models of design," in *An Anthology of Theories and Models of Design. Philosophy, Approaches and Empirical Explorations* (A. Chakrabarti and L.T.M. Blessing, Eds.), ch. 6, pp. 100–121, Springer-Verlag London, 2014.
7. M.S. Erden, H. Komoto, T. Beek, V. D'Amelio, E. Echavarria, and T. Tomiyama, "A review of function modeling: Approaches and applications," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 22, pp. 147–169, 2008.
8. D. Deneux, "Introduction to assembly features: an illustred synthesis methodology," *Journal of Intelligent Manufacuring*, vol. 10, pp. 29–39, 1999.
9. S. Rachuri, Y-H. Han, S. Foufou, S.C. Feng, U. Roy, F. Wang, R.D. Sriram, and K.W. Lyons, "A model for capturing product assembly information," *Journal of Computing and Information Science in Engineering*, vol. 6, no. 11, pp. 12–21, 2006.
10. C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari, "Wonderweb deliverable D18," tech. rep. CNR, 2003.
11. C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino, "Social roles and their descriptions," in *Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR 04)* (D. Dubois, C. Welty, and M. A. Williams, Eds.), (Whistler Canada), pp. 267–277, 2004.
12. S. Borgo and C. Masolo, "Foundational choices in DOLCE," in *Handbook on Ontologies* (S. Staab and R. Studer, Eds.), pp. 361–381, Springer Verlag, 2nd ed., 2009.
13. L. Vila and H. Reichgelt, "The token reification approach to temporal reasoning," *Artifical Intelligence*, vol. 83, pp. 59–74, 1996.
14. S. Borgo, E.M. Sanfilippo, A. Sojic, and W. Terkaj, "Ontological analysis and engineering standards: An initial study of ifc," in *Ontology Modeling in Physical Asset Integrity Management*, Springer Verlag Berlin Heidelberg, 2015.
15. P. Gärdenfors, *Conceptual Spaces: the Geometry of Thought.* Cambridge, Massachussetts: MIT Press, 2000.
16. R. Casati and A. Varzi, *Parts and Places. The Structure of Spatial Representation.* Cambridge, MA: MIT Press, 1999.
17. C. Masolo and S. Borgo, "Qualities in formal ontology," in *Foundational Aspects of Ontologies (FOnt 2005) Workshop at KI 2005* (P. Hitzler, C. Lutz, and G. Stumme, Eds.), (Koblenz, Germany), pp. 2–16, 2005.
18. M. Vries, "Translating customer requirements into technical specifications," in *Philosophy of Technology and Engineering Sciences. Handbook of the Philosophy of Science.*, pp. 488–512, Elsevier, 2009.

19. P. Simons, *Parts: a Study in Ontology.* Oxford: Clarendon Press, 1987.
20. S. Rama Fiorini and M. Abel, "Part-whole relations as products of metric spaces," in *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pp. 55–62, 2013.
21. S. Rama Fiorini, P. Gärdenfors, and M. Abel, "Representing part-whole relations in conceptual spaces," *Cognitive Processing*, vol. 15, no. 2, pp. 127–142, 2014.
22. P.Kroes, *Technical Artefacts: Creations of Mind and Matter. A Philosophy of Engineering Design.* Springer, 2012.
23. P. Vermaas, P.Kroes, I. de Poel, and M.Franseen, *A Philosophy of Technology. From Technical Artefacts to Sociotechnical Systems.* Morgan and Claypool Publishers, 2011.
24. S. Fenves., S. Foufou, C. Bock, and R.D. Sriram, "CPM: a core model for product data," *Journal of Computing and Information Science in Engineering*, vol. 8, 2008.
25. B. Smith and W. Ceusters, "Aboutness: Towards foundations for the information artifact ontology," in *Proceedings of the Sixth International Conference on Biomedical Ontology (ICBO)*, 2015.

Towards a novel method for Architectural Design through μ-Concepts and Computational Intelligence

Nikolaos Vlavianos[1], Stavros Vassos[2], and Takehiko Nagakura[1]

[1] Department of Architecture
Massachusetts Institute of Technology, USA
{nv2247,takehiko}@mit.edu
[2] Department of Computer, Control, and Management Engineering
Sapienza University of Rome, Italy
vassos@dis.uniroma1.it

Abstract. Architects typically tell stories about their design intentions and then translate the verbal descriptions into action-based commands in Computer Aided Design tools. In other words, during the creative process the designer works in a "concepts space" that is different than the "properties space" available in the software virtual world of CAD tools. In this paper we propose to study the vocabulary of concepts used in architectural design by employing machine learning methods over large data-sets of architectural drawings and their storytelling descriptions. The intention is to computationally characterize the meaning of a set of high-level concepts in architectural design, which we call them as *μ-concepts*, in terms of a set of low-level properties of drawings. With such a correlation, new opportunities can be explored for radically different design tools that allow architects to design by operating over such high-level architectural concepts. Eventually, this will can also provide a novel way of understanding the mental process of architectural design through verbal concepts.

1 Introduction

We are motivated by the linguistic depth with which architects think and communicate while developing their designs. Considering then that an architect conceptualizes design thoughts primarily through Computer Aided Design (CAD) software, we identify a gap between the linguistics of exchanging ideas at a conceptual level and visualizing them in the computer setting. If an architect is able to *use the linguistic schemata of conceptual thinking directly in the CAD software*, then the gap can be significantly bridged allowing for the *emergence of novel design tools*.

A major challenge is that the concepts and words that architects use are neither limited to a certain number, nor inscribed by any means of literature of design theory. Therefore, any approach that studies some specific cases over a "fixed" vocabulary is limited and biased towards the subjective choice of concepts at the particular time and space of the analysis. Employing modern computing paradigms that reveal the "collective wisdom" of communities, we aim at introducing an intelligent system that can learn any set of concepts from given examples and provide an automated form of understanding and detecting those concepts on designs and drawings.

The greater vision of the proposed line of work is to incorporate a computational understanding of conceptual thinking in the design process of architects. One can imagine a new generation of CAD software (or plug-ins for existing ones) that will feature tools related to high-level concepts rather than tools that perform low-level actions. For instance, instead of applying the basic command "break" of a line or a solid object in Rhino software, we can ask the program to "make the object smaller". What this project proposes is a rich set of concepts or ideas that are currently used in the design world as a "specialists slang" language: "make it larger", "make this volume sharper" or "create an introverted composition around the courtyard", to become a central integral part of design architectural tools.

2 Studying μ-Concepts in Architectural Design

What are we trying to achieve and why will it make a difference? We believe that currently there is a gap between the thought process of architects and the actions they can take in a CAD environment. In reality this gap refers to the separation of human intelligence from computer intelligence [10], in the sense that during the creative process the human designer works in a different "concepts space" than the "properties space" available in the software virtual world. The goals of this line of research revolve around bridging this gap and enabling novel forms of collaborative design between designers and computational machines in a new generation of CAD software tools, as follows.

Goals. A primary goal is to identify a set of design concepts currently used by architects in the form of verbal communication. Architects regardless of national or linguistic differences often utilize a set of special words in order to express design intentions. Although the actual meaning of those words might be slightly different in different contexts, architects do inscribe certain design concepts to them. A technical goal then is to create a system that is programmed to understand automatically those concepts by reading architectural drawings and text descriptions and correlating formal (low-level) design properties to casual (high-level) architectural concepts. Every single architectural drawing operates as a source of quantifiable data that describe space and architectural form. Simultaneously, verbal or written descriptions express design information. Thus, both architectural drawings and text descriptions operate as repositories of architectural information. The final goal is to embed the ability of understanding concepts, through the developed system, to novel architectural design tools and methodologies.

How does the proposed line of work go beyond the state of the art? The introduction of Computer Aided Design (CAD) tools has revolutionized a number of creative industries, and substantially changed the pipeline for creating new products, with architectural design being one of the most striking examples. CAD tools are used in creative tasks in order to minimize development time and cost, reduce human effort, and support collaboration among members of a design team. In practice CAD tools operate as "the designers slave" [7, 9] following the tasks instructed by the human designer and often carrying out labor-intensive low-level operations, e.g., simulations, calculations, etc. The design process becomes more interesting when the computer takes the role of a "colleague" [4] and can contribute to the design by proposing different alternatives and allow the co-creation of less conventional ideas. Classic work such as [2] as well as recent work in the so-called mixed-initiative design paradigms, e.g., [3, 13], show that this direction can be very helpful in the similar design setting for virtual spaces in videogames. The novelty of the research project is that it not only envisions the architectural CAD software as a collaborator to the human designer, but also incorporates essential high-level design concepts human-computer discussion, which are missing from all currently existing tools.

What is the expected impact of this line of work to architectural design? We expect that the realization of the goals and objectives will enable a new perspective in the methodology of architectural design and will open up novel opportunities in all aspects of architecture. These include the vocabulary used in the design process and the story-telling for architecture results, the CAD software tools capabilities, as well as the understanding of the cognitive processes carried out by designers. In particular, this line of research offers a revolutionary way for looking into the designers internal thoughts and representations by correlating the concepts they use verbally with formal elements that can be identified in the drawings.

3 A Five-Phase Research Plan

In order to realize the identified goals, we will follow a 4-phase technical plan as follows. In Phase 1, we will investigate and specify the set of μ-concepts that we want to incorporate into a new generation of design tools. In Phase 2, we will look into the low-level properties that can be identified directly from architectural drawings, such as dimensions, local or global area, thickness of walls, length of openings, number of columns or structural elements, relations between elements, etc that represent specific values and quantitative data. In Phase 3, we will use the output of Phase 2 along with user-generated content for existing architecture designs in order to populate a large dataset of information that demonstrate organic examples for the correlation of low-level properties and the high-level emotive concepts. In Phase 4, we will employ standard machine learning tools in order to train an intelligent system using the dataset from Phase 3. Finally, in Phase 5 we will look into how this new automated form of understanding for architectural drawings can be incorporated into a new CAD tool and motivate a novel design methodology.

3.1 Phase 1

This research will specify μ-concepts in architectural design. Right now, architects tell stories about designs metaphorically using multiple words enabling multiple design possibilities. However, the storytelling process for designers does not consist of a set of predefined words, but rather as a selection of words with relatively close meaning. At this point, we will study the literature of architecture, design and computation, as well as linguistics, in order to find similar case studies. Simultaneously, we will use methods from earlier work in the Plethora Project [11] in order to identify a variety of concepts that architects and non-architects use when they narrate a story of a 2D or 3D representation. In the context of Plethora subjects were asked to produce sketches of existing models and then map models to existing sketches. In our case the subjects will be asked not only to tell a story, but also to identify concepts as keywords for their stories. Acquiring knowledge from architects and non architects will allow us to map concepts that are repeatedly used by one of the groups or by both architects and non-architects.

3.2 Phase 2

Through this research, we will identify low-level properties of drawings, and develop a tool that performs automated extraction of the properties from a given diagram. Right now, low-level properties of drawings such as dimensions, area, the ratio of volume and void, etc. are defined by CAD software. CAD tools understand designs as local and global topologies of points and lines in the design space, with specific values. Considering that the current low-level properties in CAD have particular values, we will be able to extract values for properties that are not visible by the user, but they have hidden values. For instance, the length of the openings on a plan can be measured by the length of the lines with specific width or the length of the exterior walls by the thickest lines. In this way, the proposed apparatus can extract data from any architectural drawing in an automated way, and provide a mathematical representation of the properties and relations of the elements in the design.

3.3 Phase 3

The system will build a dataset to be used for training a machine learning system that identifies μ-concepts in drawings. At this step, we will create a dataset of properties and μ-concepts that are extracted by a large set of existing drawings and descriptions. The system will extract values from drawings for the given set of properties using the automated extraction tool of Phase 2. Then for the extraction of μ-concepts we will use a small collection of commonsense rules and higher-level reflection patterns similar to the ideas developed in Genesis project [12] in order to extract the μ-concepts from the stories or descriptions that are available for the drawings in the dataset. The engagement with Genesis project will give us the chance to explore how groups of words or intentions are recognized under the influence of particular words that describe an architectural project, even though certain concepts are not mentioned. An alternative way to "extract" the μ-concepts for architectural design in the dataset is by engaging architects to provide *user-generated content*. With a simple smartphone application that demonstrates drawings and asks the user to flip left if it is "fragmented" or "flip right" if not, architects can provide the information missing about μ-concepts such as "fragmented" or "extroverted", and generate a complete dataset in the form of the following figure.

3.4 Phase 4

The proposed system will develop a tool for the automated evaluation of architectural drawings with respect to the specified μ-concepts. This phase utilizes the dataset that

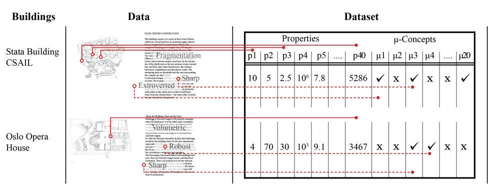


Fig. 1. The dataset for Phase 3: properties will be extracted from drawings and μ-concepts will be extracted from accompanying text.

Fig. 2. Architects can select through an app whether a μ-concept is supported in an architectural drawing.

comes out as output from Phase 3 in order to train a machine learning system that can understand μ-concepts and evaluate drawings with respect to those. More formally, the system developed in this phase will take as input an architectural drawing and will give as output the support level as a percentage for each of the μ-concepts specified in Phase 1. For realizing this system we will rely on existing successful methods for "supervised learning" from the academic field of artificial intelligence, that allows to automatically infer a function from labeled training data [6]. In this context, the system receives positive and negative examples of concepts along with the values for the properties (i.e., the dataset from Phase 3) and is able to train itself to identify μ-concepts from the properties that are extracted from the drawing.

3.5 Phase 5

Having a trained system that can evaluate architectural drawings with respect to μ-concepts, in this phase we will investigate how this can be embedded in existing CAD tools and motivate novel design processes. One direction is to explore how a computer designer-collaborator can offer a list of possibilities for the evolution of the current design according to actions that bring the design closer to supporting the different μ-concepts, following a mixed-initiative design approach similar to [3, 13]. Another direction is to look into inverting the internal machine learning interpretations of the system in order to apply the understanding of the μ-concepts in the ongoing works of a designer, similar to earlier work done for images, e.g., in [5]. Finally, we will explore how the μ-concepts can be used to specify a novel design methodology and a new generation of CAD tools that allows the designer to operate on high-level concepts instead of low level actions.

4 Conclusions

The proposed line of research is a multi-faceted project challenging the current ways of producing and communicating architectural design. By identifying μ-concepts emanating from architectural design process and providing automated tools for detecting those concepts through machine learning, we can eventually propose novel design methods based on the quantification of concepts. We envision that through work towards this direction, in the future architects will be able to design by a means of visual conversation that allows the designer to ask the next generation of CAD systems to alter the design along the lines of making it "more fragmented" or "extroverted".

References

1. Alexander, C.: Notes on the Synthesis of Form. Harvard paperback, Harvard University Press (1964)
2. Friedman, Y.: Information processes for participatory design. In: Cross, N. (ed.) Design Participation: Proceedings of the Design Research Societys Conference. pp. 45–50. Academy Editions (1972)
3. Liapis, A., Yannakakis, G.N., Togelius, J.: Sentient sketchbook: Computer-aided game level authoring. In: Proceedings of the 8th Conference on the Foundations of Digital Games. pp. 213–220 (2013)
4. Lubart, T.: How can computers be partners in the creative process: Classification and commentary on the special issue. Int. J. Hum.-Comput. Stud. 63(4-5), 365–369 (Oct 2005)
5. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. CoRR abs/1412.0035 (2014), `http://arxiv.org/abs/1412.0035`
6. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. The MIT Press (2012)
7. Negroponte, N.: Soft architecture machines. The MIT Press (1975)
8. Pickering, A.: Cybernetics and the mangle: Ashby, beer and pask. Social Studies of Science pp. 413–437 (2002)
9. Reintjes, J.: Numerical Control: Making a New Technology. Oxford science publications, Oxford University Press (1991)
10. Simon, H.A.: The Sciences of the Artificial (3rd Ed.). MIT Press, Cambridge, MA, USA (1996)
11. Vlavianos, N.: Plethora project. Tech. rep., MIT 6.843 The Human Intelligence Enterprise class (2014)
12. Winston, P.H.: The strong story hypothesis and the directed perception hypothesis. In: Advances in Cognitive Systems, Papers from the 2011 AAAI Fall Symposium, Arlington, Virginia, USA, November 4-6, 2011. AAAI Technical Report, vol. FS-11-01. AAAI (2011)
13. Yannakakis, G.N., Liapis, A., Alexopoulos, C.: Mixed-initiative co-creativity. In: Proceedings of the 9th Conference on the Foundations of Digital Games (2014)

Author Index