ENVIRONMENT BASED DESIGN (EBD) VS. X DEVELOPMENT: A DIALOG BETWEEN THEORY AND RETROSPECTION

Yong Zeng[1], Jean Vareille[2]
[1]Concordia University, Montreal, Canada, [2]Université Européenne de Bretagne, France

ABSTRACT

This paper presents two independently developed design methodologies: Environment-Based Design (EBD) and X-Development. These two methodologies share exactly the same foundation: design is based on environment. The environment is where the designed product is to work. The environment was there, is there, and will still be there. Any design action changes only the environment. This paper thus conducted a preliminary comparison of these two methodologies. This coincidental convergence of two design methodologies provides one kind of support to the each other. Future work will be focused on the formalization and refinement of X-Development through the mathematical operations included in the EBD.

Keywords: Environment-Based Design (EBD), X-Development, Comparison, Validation

The research represented in this paper stems from a coincidental meeting between the two authors in a workshop in November 2010. The two authors happened to find that they share so much about design research, ranging from the philosophy/logic of design to the process of design. Nevertheless, the paths leading to the same understanding are different. One was axiomatic whereas the other was retrospective.

The core of the common understanding is: design is based on environment. The environment is where the designed product is to work. The environment was there, is there, and will still be there. Any design action changes only the environment. Therefore, design comes from the environment, serves for the environment, and goes back to the environment.

In the rest of this paper, we will first review briefly the existing literature in design theory and methodology. In section 2, our respective work: Environment-Based Design (EBD) and X-Development, will be introduced. Section 3 shows a preliminary dialog between the two methodologies. The last section gives remarks on the implication of the work and the future research issues.

1 LITERATURE REVIEW

Compared to the long history of design practices, the study of design theory as a scientific discipline is quite young. This study has become more and more important because of an increasing need of “Best Design Practice” in optimizing the available yet limited resources for the benefit of mankind. As a result, a variety of design theories and methodologies have been proposed in the last several decades, such as the Systematic Design Methodology [1, 2], Decision-Based Design theory [3], Theory of Inventive Problem-Solving (TRIZ) [4], Axiomatic Design [5], General Design Theory [6], Formal Design Theory [7], Total Design of Pugh [8], Adaptable Design [9], C-K Theory [10], Affordance Based Design [11] and Axiomatic Theory of Design Modeling [12].

Early design methodologies are mostly based on experience and practices. The systematic design methodology was developed in the 1970’s by German professors Pahl and Beitz [2]. This methodology divides the design process into four phases: product planning and clarifying the task, conceptual design, embodiment design and detail design. The methodology of total design provides a design framework for a structured design process model, which is the application of design methodology in design practice [8]. It uses Concept Selection Process to choose the best concept from a series of candidates according to some criteria using a Pugh Matrix. It can be used not only in overall conceptual design process, but also for concept decision of system components.

Researchers have also attempted to develop design methodologies based on axioms. By viewing engineering design as a decision-making process, Decision-Based design aims to maximize the value of a designed artifact through applying the rules of decision theory to the decision-making process in design [3]. The axiomatic approach to design provides a general theoretical framework for all design fields. The theory is applicable to many different kinds of systems, including machines, large systems, software systems, organizations, and systems consisting of a combination of hardware and software [13]. The Theory of Inventive Problem Solving (TRIZ) [4] is developed by G.S. Altshuller and his school in the former USSR, based on an extensive study of the World Patent Database. The central concepts of TRIZ are contradiction and ideation. Though TRIZ provides possibilities for effective solutions of difficult problems, it needs intensive training in order to have the specialized knowledge to use it properly.

Mathematical approach to design, though deemed as an impossible mission by many researchers in design community, has also been made some progresses. General Design Theory (GDT) is a mathematical theory of design, which is proposed by Yoshikawa and extended by Tomiyama for the development of advanced CAD (computer-aided design) and for innovative design from the research results of a group [14]. GDT tries to explain how design is conceptually performed with knowledge manipulation. It formalizes design knowledge based on axiomatic set theory. In further developing the GDT, the Formal Design Theory (FDT) tries to develop a domain independent core model of the design process. The FDT explores issues such as the algebraic representation of design artifacts, idealized design process cycle, and computational analysis and measurement of design process complexity and quality [7]. A science-based approach to design provides a formalized design representation and design process that captures the dynamic, evolving, and ill-structured nature of design [15, 16]. This science based approach was further developed into the Axiomatic Theory of Design Modeling which can be used to derive formal models of design that represent the syntactic structure of hierarchical evolving design objects and the dynamic design process [12].

Apart from the mathematical and methodological research of design, the nature of design process has also been studied by many researchers. Simon attempted to transform design into a search based AI problem by decomposing an ill-structured design problem into a set of well defined problems [17]. Zeng and Cheng proposed that the logic of design is recursion where the conclusion (design solution or concept) is recursively dependent on the major premise (design knowledge) [18]. Design is thus a process to simultaneously produce both the artefact (design solution or concept) and its behavior system (design knowledge) [18]. This logic form was later further discussed by Roozenburg [19]. Dorst and Cross examined this solution-problem co-evolutionary process through a protocol analysis experiment [20]. Hatchuel and Weil introduced the similar concept in C-K theory where the core is also the recursive dependency between the space of “Concepts” (C) and the space of “Knowledge” (K) [10]. Based on the intuition that design aims to change an existing environment to a desired one by creating a new artifact into the existing environment, Environment-Based Design (EBD) [21-23] was logically derived following the axiomatic theory of design modeling [12]. Environment-Based Design (EBD) provides step-by-step procedures to guide a designer throughout this environment change process to resolve the recursive nature of design [18, 23, 24].

A major problem faced by all the design methodologies is how to validate them. Experimental validation and case studies are two most widely accepted approaches for testing a design methodology [25]. EBD has been under those tests over the years (for example [26, 27], among many others). Another interesting validation could be the convergence of independent research efforts. This paper presents such a case for both the EBD and X-Development.

2 THE TWO METHODOLOGIES

2.1 Environment Based Design (EBD)

Intuitively, design is a human activity that aims to change an existing environment to a desired one by introducing a new artifact into the existing environment. Environment-Based Design (EBD) is such a design methodology that provides step-by-step procedures to guide a designer throughout this environment change process. The underlying principles behind the EBD are that design comes from the environment, serves for the environment, and goes back to the environment.

Environment-Based Design (EBD) [21-23] was logically derived from the observation above following the axiomatic theory of design modeling [12] and based on the recursive nature of design

[18, 24]. As illustrated in Figure 1, the Environment-Based Design includes three main activities: environment analysis, conflict identification, and solution generation. These three activities work together progressively and simultaneously to generate and refine the design specifications and design solutions.

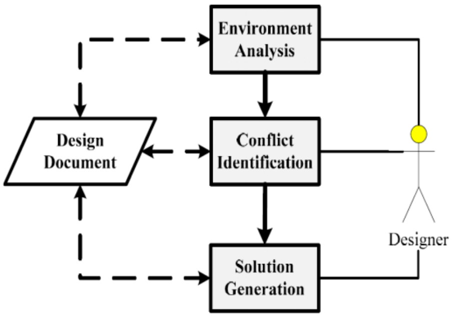

Figure 1 EBD: process flow [22].

The objective of environment analysis is to understand the real meaning of the design problem by identifying the real intent of the design and the complete requirements that can be made available based on the current knowledge of the design. The basic tools include a roadmap for design requirements defined by the environment-based classification of product requirements [28] and a question asking algorithm [29]. Recursive Object Model (ROM) [30] is the foundation and result of the environment analysis.

In some cases, the ROM diagram generated by environment analysis already included partial design solutions. Under all situations, three basic domain-independent rules can be applied to identify potential conflicts between relationships in the ROM diagram [22]. The three rules will identify the conflicts between constraint and predicate relations. In the case of multiple conflicts being identified, the structure of ROM diagram can be used to prioritize the conflicts so that the major/critical conflict will be dealt with first [23].

Instead of producing a comprise for an identified conflict, the EBD will generate a design solution by first looking into the ROM objects that are associated with the conflict and then adding, removing, or separating an object implied in the conflicting relation. The design solution will thus redefine the ROM diagram, which triggers another round of design.

The EBD is a logical and recursive process that aims to provide designers the right direction for solving a design problem. The three activities can be carried out simultaneously for an experienced designer.

Mathematically, the EBD process can be represented by structure operation, denoted by ⊕. Structure operation can be defined as the union (∪) of an object O and the interaction (⊗) of the object with itself [12].

$$\oplus O = O \cup (O \otimes O), \quad (1)$$

where ⊕O is the structure of the object O. Everything in the universe can be seen as an object. Interactions between objects are also objects. Examples of interaction include force, movement, and system input and output. Structure operation provides a means to represent a hierarchical system with a single mathematical expression. The application of structure operation can be found in the representation of sketches [31] and linguistic information in design [30].

Due to the capacity of human congnition and the scope of an application, a group of primitive objects can always be defined as [12, 30]

$$\oplus O_i^a = O_i^a. \quad (2)$$

Equation (2) means that a primitive object is an object that cannot or need not to be further decomposed.

In the design process, any previously generated design concept can be indeed seen as an environment component for the succeeding design. As a result, a new state of design can be defined as the structure of the old environment (E_i) and the newly generated design concept (S_i), which is a partial design solution.

$$\oplus E_{i+1} = \oplus (E_i \cup S_i). \quad (3)$$

It has been shown that the environment structure, which is $\oplus E$, includes the description of the design solution at design stage i and the design requirements for the design stage i+1 [21].

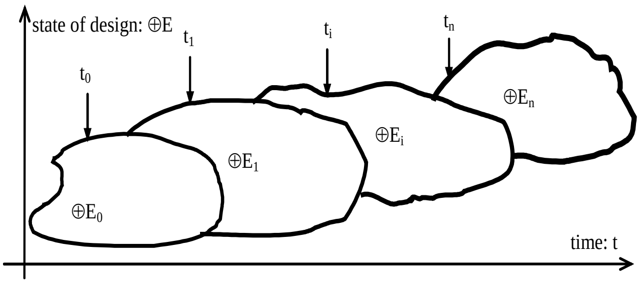

Figure 2 Environment Based Design: process [21]

Therefore, the EBD process can also be graphically illustrated in Figure 2, which implies the recursive evolution of design requirements and design solution [18, 24].

The following explains the three steps included in EBD [22, 32]:

Step 1: Environment analysis: define the current environment system $\oplus E_i$.

$$\oplus E_i = \oplus\left(\bigcup_{j=1}^{n_e} E_{ij}\right) = \bigcup_{j=1}^{n_e}(\oplus E_{ij}) \cup \bigcup_{j_1=1}^{n_e} \bigcup_{\substack{j_2=2 \\ j_2 \neq j_1}}^{n_e} \left(E_{ij_1} \otimes E_{ij_2}\right), \quad (4)$$

where n_e is the number of components included in the environment E_i at the i^{th} design state; E_{ij} is an environment component at the same design state. It should be noted that decisions on how many (n_e) and what environment components (E_{ij}) are included in E_i depend on designer's experience and other factors relevant to the concerned design problem.

Step 2: Conflict identification: identify undesired conflicts C_i between environment components by using evaluation operator K_i^e, which depends on the interested environment components.

$$C_i \subset K_i^e\left(\bigcup_{j_1=1}^{n_e} \bigcup_{\substack{j_2=1 \\ j_2 \neq j_1}}^{n_e} \left(E_{ij_1} \otimes E_{ij_2}\right)\right). \quad (5)$$

Step 3: Solution generation: generate a design solution s_i by resolving a group of chosen conflicts through a synthesis operator K_i^s. The generated solution becomes a part of the new product environment for the succeeding design.

$$\exists C_{ik} \subset C_i, K_i^s : c_{ik} \rightarrow s_i, \oplus E_{i+1} = \oplus\left(E_i \cup s_i\right). \quad (6)$$

The design process above continues with new environment analysis until no more undesired conflicts exist, i.e., $C_i = \Phi$.

2.2 X-Development

The X-development method is resultant of a long, deep and slow analysis performed during about 20 years. The actual formulation was designed in collaboration with Mehdi Tahan (during his Master's project) [33], Amara Touil (a PhD student), and Prof. Philippe Le Parc (through his helpful participation and comments).

The fundamental assumptions are:

- each artifact system is made of material extracted from the environment, which includes the earth, fluids or other materials. The trash produced during the usage an artifact system is put back to the environment. Finally, the system itself is put back to the environment after use;
- the environment has existed before the creation of the system;
- the environment will remain after the usage of the system; and
- each system is first designed, then produced, and used.

Starting from these ideas we have searched how the different existing methods used for the development of technical systems could be associated and combined. We have adopted a retrospective or synthetic point of view as is shown in Figure 3.

We have considered that each system could be produced only when its definition is completed. The materials needed to produce the system should be available when the production starts. By using the ¯_ shaped representation of the well-known waterfall method, we have added a left leg to represent the fact that the bulk materials and the standard elements have to be collected before the production. The resulting representation is λ shaped. At the bottom of the graph we have added a rectangle to represent the environment existing before the beginning of the process and still remaining after the whole lifecycle.

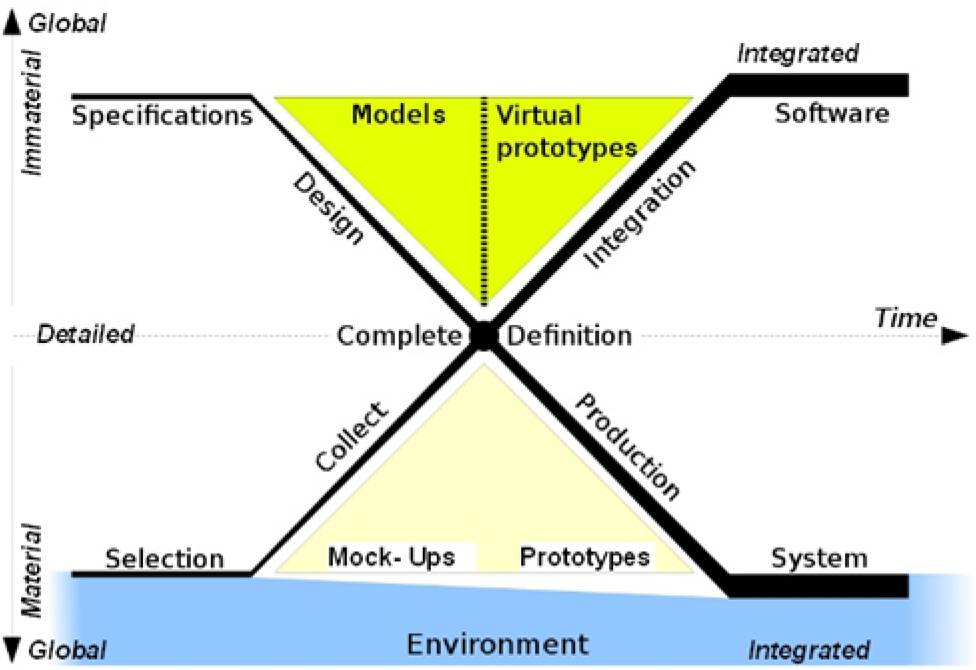

Figure 3 Scheme of the X-development method.

Furthermore, we have added the right arm to the graph to include the famous V-cycle [34], which leads to the X shape: $\lambda \cup {}^{v} \rightarrow X$. We consider that the crucial point of the graph is the meeting point of the branches. Above this point, it is the field of the information, data, software and others whereas below this point it is the physical and material field. We have drawn a horizontal arrow through the central point to represent time. The whole scheme could be understood as a PERT - each line as a task and each node as a transition between tasks. The process of the design begins on the upper left branch by the definition of the specifications. Early during this stage the designer should specify the environment in which the system will work and the borders between this environment and the system. Then the exchanges between the system and its environment will be mostly known and so is the geometry of the interaction. This knowledge is sufficient for choosing the materials that should be used to build the structure of the system [35]. So it is possible to start the task of the selection of the materials and real elements to be used in parallel to the engineering design process.

We denote P as the set of the performances of the system, G the set of the geometrical parameters, F the constraints, and M(p) the combinations of the p properties of the materials. P is resulted from the constraints F, the choices of G and M(p).

$$P \leftarrow f\{F, G, M(p)\} \quad (7)$$

Obviously in the set of F constraints there are parameters with physical dimensions like mass, energy consumption, the lifespan of the system. The physical dimensions used to define these parameters are not only length. We have exactly the same situation in the cases of P and M(p). It appears that the system is evolving in the environment between the causes and the effects, transforming constraints and consumptions into performances. In the case of systems changing the nature of the physical dimension of the causes into other dimensions of the effects, e.g. electrical power into light or heat into mechanical work, there are equations and inequality where M(p) and F are mixed with the G variables and parameters. In these equations and inequalities G is separable of F and M(p). Then we can write the separable character of the equations and inequalities:

$$P \leftarrow f1\{F,M(p)\}f2\{G\}. \quad (8)$$

The problem of the designer is to start the process. He has an alternative choice, solving first f1 and then f2 or the opposite order solving f2 then f1. The great advantage of the first order is that the choice of the materials is made earlier. Then another team could take care of the choice of the provider, prices and contracts. This last team could give to the design team the deadline to deliver the complete definition of the system. The manager of the design team could separate the work and evaluate the workload of each designer. There are vertical arrows not drawn in Figure 3 matching on these information exchanges. These quantities are physically dimensioned below the middle horizontal line, and above they can be transformed as dimensionless variables and parameters.

The shape of the rectangle of the environment means that its size is constant and all the materials used to build the system and its consumption will go back to the environment at the end of the lifespan or will be recycled or reused. The result is that the system modifies the environment during its usage and after. If these aspects are not taken into account at the beginning of the design we can have bad consequences. One solution is to compare these modifications during the first stage of the design when several solutions are possible. When the materials are chosen first, before the geometrical details of the parts and the assemblies we can do it [36].

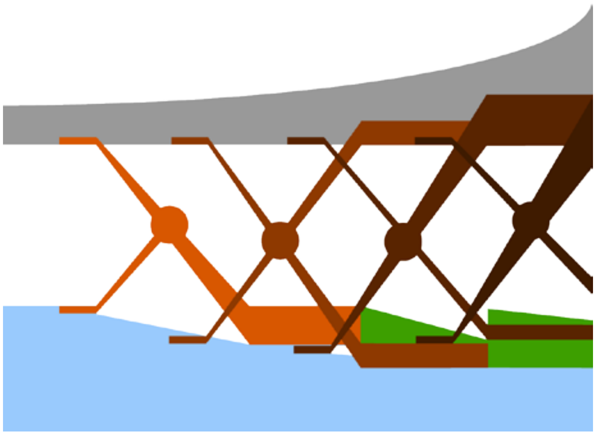

Figure 4 Iteration process of the X-development for same usage system.

On the right of the central node of the scheme, the production process of the parts and the assemblies starts below the middle line, beyond the integration process of the software is performed. At both ends each result forms an integrated set. The software is commonly embedded inside the whole system.

The bold line at the end of the upper right line means that the size of the data and software is not limited. The size of the knowledge is not limited, but the knowledge has to be well organized to be reusable for further systems. The ideal reuse of immaterial knowledge, data, and software is sketched in Figure 4.

3 THE DIALOG

3.1 X-Development from EBD Perspective

It can be seen from the fundamental assumptions underlying X-Development that environment is a constant element behind the evolution of an artifact. An artifact comes from environment and will become a part of environment. The entire lifecycle of an artifact is associated with the environment.

In X-Development, the causes and the effects are the driven force for generating solutions whereas conflicts are considered in EBD. Environment is divided into the material environment and immaterial environment in X-Development; however, three components in EBD – natural, built and human environments are taken into consideration. These are two different representations of the environment.

For solution generations, both methods share the same idea that a product/system being designed changes the environment during the life cycle; but differences exist. In X-Development, it is emphasized that the materials are chosen first in order to solve f1 and then f2 in Eq. (8). A problem may arise: how to choose the materials? Since the material selection may be constrained by the both functional and structural requirements, M could be a function of F and/or G. Indeed, in many cases, G, F and M(p) are highly coupled. For example, nickel-titanium alloy has a property of “shape-memory” (M(p)), its geometrical parameters (G) may change under certain condition (F), say temperatures. As a result, the performances of the system may be affected. Sometimes customers do not know exactly what they really need for a product, and their requirements are often described in plain language. How to find out the specifications shown in Figure 3 is not known in X-Development. In other words, the designer should have a clear idea of “what kind of criteria/criterion should be followed” in order to find the specifications and choose the “right” solution among potential candidates. In their own words, “the problem of the designer is to start the process”, but X-develop does not provide any systematic guidance on how to start. While in EBD, a designer focuses on the analysis of objects and their relations, based on which choosing of the material is a part of solution generation. The selected materials will be put back to the environment for the proceeding design. Since design solutions are constrained by many environment components, some of which are implicit and must be identified through a systematic approach [28, 29].

In addition, both methodologies consider the search and organization of unlimited knowledge. However, X-Development in its current form lacks a guideline or method on how to achieve the objective whereas EBD provides a question asking based guideline with a software tool for this purpose [28, 29].

In summary, EBD and X-Development do share a common base, which is that design changes environment and comes from the environment. This common base provides a mutual support between the two methodologies. Though X-development is still in its preliminary stage, which is in demand of systematic procedures to support the activities included in the development, it does include the manufacturing activities as its component. EBD focuses on only the design stage of the product development with manufacturing as one of its environment components. The actual manufacturing of an artifact is not the concern of the EBD. For this same reason, X-development appears to be a method for materials based engineering product development whereas EBD is a generic design methodology. A feasible development is to use EBD as the theoretical tool to refine the activities in X-Development.

3.2 EBD from X-Development Perspective

The EBD is a formalization of the design activities, it seems that it could be used along the two first lines on the upper-left side of the X to improve the rate, the evaluation of the quantity, the quality and at last but not least, to reduce the environmental impact of the system along all the following phases. When he starts a study using the X-development method, the designer has to compose a "supersystem" centered on the interaction to be created or to be modified, and the elements of the environment involved in it. The "supersystem" is not closed, nor isolated nor in equilibrium with its environment. After this stage he checks all the known variables suspected to play a role, physical, chemical, biological, economical, etc. This process is close to the process described above in the EBD theory and illustrated inFigure 1. The designer is completely free to modify the border of the "supersystem" and the exchanges through, taking elements from the environment or putting back to it. This freedom corresponds to the iterative process described in the EBD theory and illustrated in Figure 2. This last one could be understood as an oblique projection on a layer orthogonal to the arrow of the time in Figure 3. Hence, during the design process the solution is dramatically influenced by the environment, and this last one is modified by the solution, because the designer is working with a "supersystem". To associate an immaterial expression of the design problem to the material one, we use the dimensional analysis which can facilitate application of the EBD theory to X-Development.

Though we tried to use the tools and methods of the “Model Driven Engineering” (MDE) [37], the theoretical and systematic approach implied in the EBD could give us a new formalism and paradigm to model the transformations among activities in the X-Development. On the right hand side of Figure 5, we present the waterfall method [34] while the MDE is on the left hand side. The basis of the MDE is that during the development of a system we should use the maximum of models and software to produce and perform the maximum of documents, codes, drawings, test, proofs, simulations, before the production and the real usage. Today all of these models and related data are stored on computers and usable with these. In fact all the software and data-base could be modeled using the same universal modeling language because all the computers use algorithms and run similarly.

The recursive evolution of design requirements and design solutions described by Zeng and Cheng [18] can be a foundation for the recursive development process shown in Figure 5, where several arrows are going from the information field on the left to the real field on the right and several others are coming back from the right to the left. In particular, the dashed arrow reaching the know-how indicates that the process is dynamically modified and updated.

Because the X-development method has two poles of concern, one centered on the material aspects and the other centered on the immaterial aspects, it could be seen as more suitable for mechanical or civil engineering.

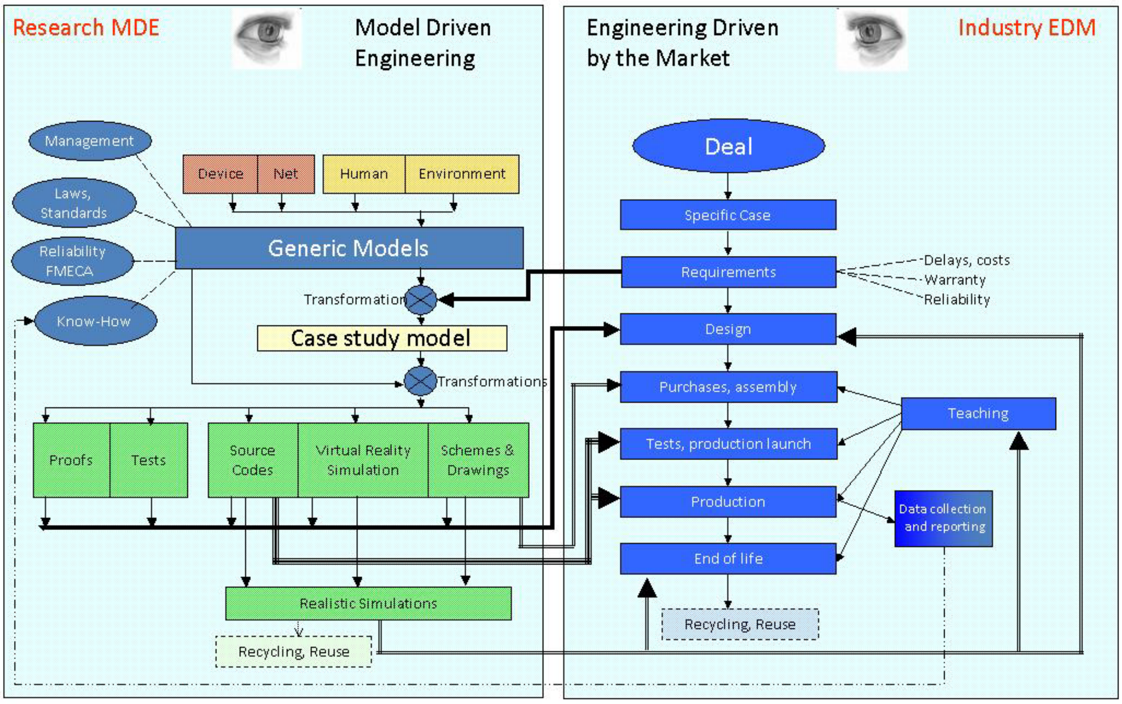

Figure 5 MDE versus EDM [38].

4 CONCLUDING REMARKS

This paper presents two independently developed design methodologies: Environment-Based Design (EBD) and X-Development. These two methodologies share a common foundation: design is based on environment. The environment is where the designed product is to work. The environment was there, is there, and will still be there. Any design action changes only the environment. Based on the description of each methodology, the similarities and differences between the two methodologies were analyzed. The EBD can be taken as a theoretical foundation for the further refinement of X-Development whereas X-Development can be used as a test bed to validate EBD.

The dialog and comparison presented in this paper is far from conclusive and it is even to certain extent superficial. However, the coincidence of the independent development of similar ideas does provide an interesting insight to both methodologies.

Future work will include the formalization of the preliminary ideas implied in X-Development by using the formal mathematical approach included in the EBD and conducting case studies.

ACKNOWLEDGEMENTS

We thank warmly Eric Coatanea and Jean-Pierre Micaelli for their excellent organization the Crecos seminar, which has brought the two authors together. The first author’s work is partially supported by the NSRERC Discovery Grant.

REFERENCES

[1] G.Pahl, W.Beitz, and K. Wallace, *Engineering design: A systematic approach*. 1988: Springer-Verlag.

[2] Hubka, V. and W. Eder, *Theory of technical systems*. 1988: Springer-Verlag.

[3] Hazelrigg, G.A., *Systems engineering: An approach to information-based design*. 1996: Prentice Hall.

[4] Altshuller, G.S., *Creativity as an exact science: The theory of the solutions of inventive problems*. 1984: Gordon and Breach Science Publishers.

[5] Suh, N., *The principles of design*. 1990: Oxford University Press.

[6] Yoshikawa, H., *General design theory and CAD system*. Man-Machine Communication in CAD/CAM, 1981.

[7] Braha, D. and O. Maimon, *A mathematical theory of design: Foundations, algorithms, and applications*. 1998, Netherlands: Kluwer.
[8] Stuart, P., *Total design: Integrated methods for successful product engineering.* 1991, New York: Prentice Hall.
[9] Gu, P., M. Hashemian, and A.Y.C. Nee, *Adaptable design.* CIRP Annals - Manufacturing Technology, 2004. **53**(2): p. 539-557.
[10] Hatchuel, A. and B. Weil, *A new approach of innovative design: An introduction to c-k theory*, in *International Conference On Engineering Design ICED 03,* 2003: Stockholm.
[11] Maier, J.R.A. and G.M. Fadel, *Affordance based design: A relational theory for design.* Research in Engineering Design, 2009. **20**(1): p. 13-27.
[12] Zeng, Y., *Axiomatic theory of design modeling.* Transaction of SDPS: Journal of Integrated Design and Process Science, 2002. **6**(3): p. 1-28.
[13] Suh, N.P., *Axiomatic design theory for systems.* Research in Engineering Design, 1998. **10**(4): p. 189-209.
[14] Tomiyama, T., *From general design theory to knowledge-intensive engineering.* Artificial Intelligence for Engineering Design, Analysis and Manufacturing 1994. **8**(4): p. 319-333.
[15] Zeng, Y. and P. Gu, *A science-based approach to product design theory part I: Formulation and formalization of design process.* Robotics and Computer-Integrated Manufacturing, 1999. **15**(4): p. 331-339.
[16] Zeng, Y. and P. Gu, *A science-based approach to product design theory part II: Formulation of design requirements and products.* Robotics and Computer Integrated Manufacturing, 1999. **15**(4): p. 341-352.
[17] Simon, H.A., *The structure of ill structured problems.* Artificial Intelligence, 1973. **4**(3-4): p. 181-201.
[18] Zeng, Y. and G.D. Cheng, *On the logic of design.* Design Studies, 1991. **12**(3): p. 137-141.
[19] Roozenburg, N.F.M., *On the logic of innovative design*, in *Research in design thinking,* N. Cross, K. Dorst, and N.F.M. Roozenburg, Editors. 1992, Delft University Press. p. 127-138.
[20] Dorst, K. and N. Cross, *Creativity in the design process: Co-evolution of problem-solution.* Design Studies, 2001. **22**(5): p. 425-437.
[21] Zeng, Y., *Environment-based formulation of design problem.* Transaction of SDPS: Journal of Integrated Design and Process Science, 2004. **8**(4): p. 45-63.
[22] Zeng, Y., *Environment based design: Process model*. 2004, Concordia Institute for Information Systems Engineering, Concordia University: Montreal. p. 40.
[23] Zeng, Y. *Environment Based Design*. in *ASME IDETC/CIE*. DETC2011-48263. 2011. Washington D.C., USA.
[24] Zeng, Y. and J. Jing. *Computational model for design*. in *Fourth International Conference on Computer-Aided Design and Computer Graphics*.638-643. 1996. Wuhan, China: SPIE.
[25] Blessing, L.T.S. and A. Chakrabarti, *DRM, a design research methodology*. 2009, London: Springer.
[26] Feng, D. and Y. Zeng. *Validation of environment-based design (EBD) through protocol analysis experiment*. in *Proceedings of International Conference on Engineering Design* 2009. Stanford.
[27] Tan, S., Y. Zeng, and A. Montazami, *Environment-based design of medical devices: A case study*, in *Biomedical engineering: Health care systems, technologies and techniques*, S. Suh and M. Tanik, Editors. 2011, Springer: New York.
[28] Chen, Z.Y. and Y. Zeng, *Classification of product requirements based on product environment.* Concurrent Engineering: Research and Applications, An International Journal, 2006. **14**(3): p. 219-230.
[29] Wang, M. and Y. Zeng, *Asking the right questions to elicit product requirements.* International Journal of Computer Integrated Manufacturing, 2009. **22**(4): p. 283 - 298.
[30] Zeng, Y., *Recursive object model (ROM) - modeling of linguistic information in engineering design.* Computers in Industry, 2008. **59**(6): p. 612-625.
[31] Zeng, Y., A. Pardasani, H. Antunes, Z. Li, J. Dickinson, V. Gupta, and D. Baulier, *Mathematical foundation for modeling conceptual design sketches.* Transactions of the ASME: Journal of Computing and Information Science in Engineering, 2004. **4**(2): p. 150-159.

[32] Zeng, Y. and S.J. Yao, *Understanding design activities through computer simulation.* Advanced Engineering Informatics, 2009. **23**: p. 294-308.
[33] Tahan, M., *Prototypage virtuel et modélisation*. 2009.
[34] Royce, W.W. *Managing the development of large software systems* in *Proceedings of IEEE WESCON*.1-9. 1970.
[35] Ashby, M.F., *Materials selection in conceptual design.* Mater Sci Tech, 1989. **5**(6): p. 517–525.
[36] Tahan, M., A. Touil, J. Vareille, and P. Le Parc. *The X development method, a new viewpoint about the product lifecycle management*. in *CRECOS 2010* 2010. Helsinki.
[37] Schmidt, D.C., *Model-driven engineering.* IEEE Computer, 2006. **39**(2).
[38] Le Parc, P., J. Vareille, and A. Touil. *Models for remote action*. in *BE2011* 2010. Artemis.

Contact: Yong Zeng
Concordia University
Concordia Institute for Information Systems Engineering
1455 de Maisonneuve Blvd. West, EV.07.633
H3G 1M8, Montreal, Canada
Phone: +1-514-848-2424 #5801
Fax: +1-514-848-3171
E-mail: zeng@ciise.concordia.ca
URL: http://users.encs.concordia.ca/~zeng/index.html
Dr. Yong Zeng is a Professor in the Concordia Institute for Information Systems Engineering at Concordia University, Montreal, Canada. His research interests include design theory and methodology, requirements engineering, and cognitive science.

Contact: Jean Vareille
Université Européenne de Bretagne, France
University of Brest
EA3883 LISyC Laboratoire d'Informatique des Systèmes Complexes
20 av. Victor Le Gorgeu, BP 809, F-29285 Brest
Tel: Int +33 2 98 01 69 59
Fax: Int+33 2 98 01 80 11
Email: jean.vareille@univ-brest.fr
URL: http://www.lisyc.univ-brest.fr/
Jean Vareille is an Associated Professor of Mechanical Engineering at the IUT of Brest, and he is researcher in the Computer Science Department of the Faculty of Science at the University of Brest. He is interested in many aspects of design and models, especially of systems containing mechanical elements and embedded systems, connected to a communication network, being able to be remotely controlled.