BEHAVIOUR-ATTENTIVE PROTOTYPING OF A DESIGN AND SIMULATION SYSTEM FOR IC CHAMBERS

Hou, Yuemin (1,2); Horvath, Imre (3); Rusak, Zoltan (3); Ji, Linhong (1); Sun, Yunchun (1); Lin, Jia (1)
1: Tsinghua University, China; 2: Beijing Information Science & Technology, China; 3: Delft University of Technology, The Netherthands

Abstract
This paper proposes a method of behaviour-attentive prototyping (BAP). BAP differentiates an approach of prototyping that provides a comprehensive description of the main features of the evolving system based on the conjoint behaviours of users, computers, and the designed system. The goal of BAP is to produce a rough working model of the designed system. The method has been developed with a view to and is intended to be used primarily, but not exclusively, in the early development of design systems of processing chambers of integrated circuit (IC) equipment. The method uses a behaviours tracking strategy. It involves five steps: (1) construction of behaviour space; (2) reasoning with the sequence of behaviours; (3) behaviour decomposition; (4) behaviour prototyping; (5) validation on computer. The BAP approach is interested not only in how the system is configured and manifest, but also in what it operates and how it behaves.

Keywords: Design methods, Computer aided design (CAD), Early design phases, prototyping, IC chamber

Contact:
Dr. Yuemin Hou
Tsinghua University
Mechanical Engineering Department
China, People's Republic of
hym01@mails.tsinghua.edu.cn

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

Computer support design systems are necessary for faster, better and cheaper designs of novel products. To develop such systems requires close collaboration of product design engineers, knowledge engineers and computer programming experts. Stakeholders like managers and potential users are also involved in the early stage so that the system can be operated in a way that all stakeholders are satisfied with. In the initial stage, a big picture of the system is expected by each group as a starting point. One solution is early prototyping. The concept of prototyping has different interpretations and definitions. In general, a prototype is a draft version or a simulation of a real thing (UMSL, 2014). A prototype can be a physical manifestation or a virtual manifestation. In systems analysis, it means an overall model of the system (or subsystem) (Davis and Yen, 1998). In systems design, it means an evolving model of the system (or subsystem) being designed. The model can be constructed using real data or virtual data sample data (UMSL, 2014). Prototyping is the process of developing prototypes (Janson and Smith, 1985). In a broad sense, prototyping may involve any method that visualizes or describes an idea (Kraf, 2012).

Many methods have been proposed for making various types of prototypes. Most of these methods involve purpose-driven selection of tools and activities, while some involve detailed design procedures. There is one thing in common among these methods: they require specialists to execute prototyping. However, such specialists are typically in the lack of knowledge of product design and simulation. On the other hand, product engineers have domain knowledge but they often lack knowledge and skills to develop a computer system. 'A big picture' over the designed system is important to all stakeholders before the development of the future system. For instance, potential end-users may want to see what input information is required from them, what output the designed system provides for them, and how they may use the system. Programmers and knowledge engineers may need to know what tasks should be completed by computers and what pieces of knowledge are involved in fulfilling these tasks. To address all these aspects, an easy way to follow by all stakeholders is to demonstrate behaviours of the designed system.

Towards the end mentioned above, this paper proposes a method of *behaviour-attentive prototyping* (BAP). BAP differentiates an approach of prototyping that provides a comprehensive description of the main features of the evolving system based on the conjoint behaviours of users, computers, and the designed system. The goal of BAP is to produce a rough working model of the designed system. The method has been developed with a view to and is intended to be used primarily, but not exclusively, in the early development of design systems of processing chambers of integrated circuit (IC) equipment. Smart products provide better service for medical, work and social life. The brains of smart products are IC chips. Processing chambers are key equipment in IC manufacturing. To increase the design quality and reduce the production time, design and simulation integrated system is required. BAP *is used to develop a rough working model of* the design support system.

The next section of the paper presents the brief review of the recently developed and applied prototyping methods. The third section discusses the major theoretical considerations and the essence of the proposed prototyping methodology. The fourth section demonstrates the application of the BAP method to behavioural prototyping of a design support system for processing chambers of IC equipment. The fifth section discusses the work and the results, offers some implicative conclusions, and discusses future research and development opportunities.

2 REVIEW OF CURRENT PROTOTYPING APPROACHES

The concept of prototyping is known for a rather long time, and its current scope of applications extends from prototyping of mechanical parts to prototyping of complex systems, in both the physical and the virtual realms. Many methodological approaches (types of prototyping) can be identified according to the objectives and manifestations. The various types, tools and objectives of the known prototyping approaches are summarized in Figure 1.

Abstract prototyping uses sticky notes or other simple tricks to represent contents of a user interface to avoid the deduction of attractive prototypes that may disguise weak designs (Constantine, 1998). By now, it has been developed into a computer based workflow. Among the computer-supported technologies, *abstract prototyping* has three important features: (i) it presents the real life processes established by artefact-service combinations, (ii) it is an instantiation of an information model having both content- and context-related information constructs, and (iii) it is implemented using various multi-media resources the construction of information structure and a workflow as a basis of demonstration to stakeholders. The information structures are created in the form of narration and enactment (Horváth, 2011). The workflow consists of thirty five steps in four phases. The first phase is dedicated to requirement engineering and concept development, the second phase concentrates on the contents development, the third phase sets up a complete scenario of system operation, human actions, human-system interactions, and environment effects, and the fourth 4 deals with the design, media-enabled implementation, recording and integration of the elements of the narration and enactment into a complete demonstration material. *Paper prototyping* uses sketches, illustrated story-boards, cardboard mock-ups, and videos to represent the design ideas (Beaudouin-Lafon and Mackay, 2014). *Paper prototyping* is cheap and can be created quickly, but it cannot illustrate how the systems work in real world. *Virtual prototyping* has grown out from the traditional CAD modelling and has been extended by various virtual reality (VR) and other augmented realty technologies (Gowda, Jayaram and Jayaram, 1999). Rapid prototyping is appropriate for physical products but not for prototyping of processes and services. *Digital prototyping* produces mathematical algorithms-based models, and simulates and validates the real-world performance of a product design digitally. On the other hand, *on-line digital prototyping* is a distributed form of digital prototyping, which facilitates the collaborative development of prototypes in a multi-disciplinary matter. In addition to media materials, it can produce animations and simulations of future systems/products with interactive interfaces. Since on-line digital prototyping is powered by program codes and advanced computer technologies, it requires good programming skills, as well as costly software and hardware resources.

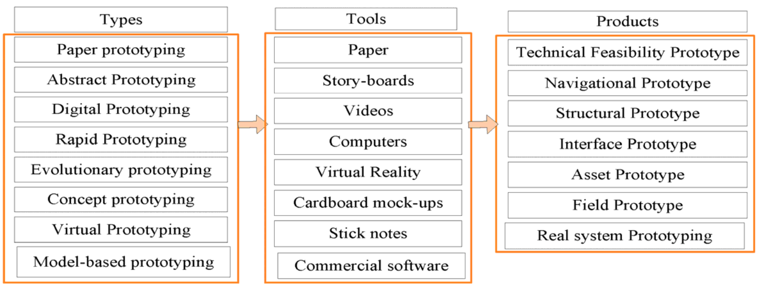


Figure 1. Types, tools and objectives of prototyping

Some other prototyping concepts have also been developed with focuses on specific contents, functions and precision, such as *evolutionary prototyping, concept prototyping* (Wood and Kang, 1992), and *model-based prototyping*. *Evolutionary prototyping* uses an evolutionary approach to develop a mature system through a series of prototype iterations. This type of prototyping involves the creation of a series of field prototypes. A field prototype is a high-fidelity real system that can be applied in a particular application field, and performs the intended function. *Concept prototyping* takes an idea from the mind to create physical replicas in a quickly manner, by using professional techniques and tools, and by offering the opportunity of improving design in the very early design stage (Arcindy, 2014). *Model-based prototyping* builds visual on-line simulation of embedded software systems and cyber-physical systems by integrating disparate models including function deployment, analysis, verification and testing etc. using commercial software tools, such as Simulink, Modelica, SystemC, etc. (Porter etc. 2009; Holden, 2014). Besides these, various types of prototypes, such as *technical feasibility prototype*, *navigational prototype*, *structural prototype*, *interface prototype*, *asset prototype,* etc. have also been reported in the literature. *Technical feasibility prototype* explores the technical operation and implementation constraints of complex design features. *Navigational prototype* seeks to determine whether or not links/connections between screens or web pages work as expected. *Structural prototype* is orientated to software design and implementation. A structural prototype focuses on composition and navigation rather than the intricacies of the user interface. On the other hand, *interface prototype* is detailed screens representations of proposed contents and arrangements. *Asset prototype* demonstrates the final production quality of assets, such as graphics, video and sound. Both the concept and the implementation technologies of each mentioned type of prototyping evolves. For example, the concept of abstract prototyping has different meaning: using simple tricks to represent user interface in the 1980's; low-fidelity prototypes in the 1990's (Fay, Hurwitz and Teare, 1990); a computer based pre-implementation testing methodology in the 2000's (Opiyo, Horváth and Vergeest, 2002), and a self-contained, digitally recorded, multi-medium enabled information structure which represents real life processes in the 2010's. These types of prototyping mainly involve tools and activity oriented procedures and need specialists to build them. Most important, they are constructed from a view of software development. In this paper, we develop BAP method that uses behaviour tracking to prototype a future system in the early development stage. Product engineers, specialists in prototyping, knowledge experts, skilled programmers and other stakeholders can contribute to the behaviour description of one part of the system respectively and collaborate to make a big picture of the design system

3 METHODOLOGY

Behaviours are key aspect to evaluate a system. It is the behaviour that product engineers are concerned with and want to optimize by taking the underpinning physical principles, operations, configuration, architecture, materialization and implementation as variable. That is the reason why we propose that capturing behaviour is at the core of prototyping in system development. The paper uses *behaviour tracking strategy to prototype design support systems.* The term 'behaviour' has multiple connotations and is used to describe somewhat different phenomena in the literature. The online dictionary explains behaviour as 'the way things act in various situations' (www.vocabulary.com). The behaviour is defined by the Oxford Dictionary as 'the way in which an animal or person behaves in response to a particular situation or stimulus' and as 'the way in which a machine or natural phenomenon works or functions' (Oxford, 2015). For this reason we have to elaborate on this term and inaugurate our particular interpretation. There are intrinsic relationships among all abovementioned terms. For instance, in the FBS framework, behaviours connect functions and structures (Gero, 1990). In this context, the function is an ascription, while the behaviour is a derivation. In the formulation of Horváth, the behaviour is the physical operation of a system observable under the influences of its stakeholders, internal situations, and embedding environments.
Based on these definitions, we used the term of 'behaviour' as a kind of blend of the above non-diverging interpretations. At the same time, we considered it as a compound of three types of behaviour, namely the behaviour of (i) the user, (ii) the engineered system, and (iii) empowering computers. For the time being, the behaviour of the surrounding environments has not been considered in our research. The *behaviour of users* refers to the way how users behave in response to the different stages of the design process. The *behaviour of a system* refers to the way how the system behaves in execution of its operations in response to the interaction of the user and the control provided by computers. The *behaviour of computer* refers to the way how computer performs the algorithmically coded tasks in varying operational situations. In BAP, we intend to give proper consideration to the conjoint behaviours of the users, the empowering computers, and the system. The procedure entailed by BAP method involves five steps: (i) construction of a behaviour space; (ii) reasoning about the sequence of behaviours; (iii) decomposition of the behaviours; (IV) behaviour-attentive prototyping and (5) validation. The BAP approach is interested not only in how the system is configured and manifest, but also in what it operates and how it behaves.

3.1 Behaviour space

The behaviour of a design system depends on the design process and the communication between the users and the system. A six stage design framework is used as the guideline of the method because it represents both states of the design process and the state transformation process. The six stages have been named as: (i) function, (ii) surrogate, (iii) property, (iv) specification, (v) feature, and (vi) parameter stages (Hou and Ji, 2011), see Figure 2. In the *function* stage*, main functions at system level are defined*. In the *Surrogate* stage, functions are interpreted as properties that are used to evaluate functions, such as the stiffness k and the temperature T. In the *Property* stage, the basic configuration of a product in terms of properties is induced from initial properties in *Surrogate* stage. In the *Specification* stage working principles and working structures as well as known parameter values are specified. In the *Feature* stage*,* the topological and material features are produced. In the *Parameter* stage*,* the drawing and documents of detailed description of the products are produced. The transformation from one stage to another one is enabled by transfer functions and evaluated by measurement functions. The six-stage design framework encompasses both design and simulation sub-processes. The design sub-process starts with *function* and outputs *product drawing*, while the simulation starts with *parameters* and outputs property evaluation in terms of graphs, figures, tables and documents.

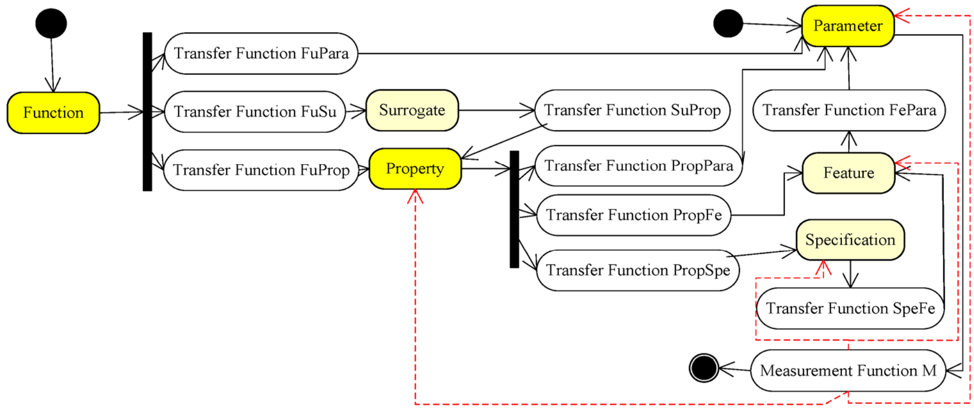


Figure 2. State space of the six stage design framework

The sources of input of the system are the users and the output is conveyed by the system to the users. This domain of activities concerning the users lends itself to the user's behaviour space. The interaction and communication between the users and the computers define the system behaviour that is visible to user. Data processing and information generation upon requests are conducted by computers, and this defines the computer behaviour space. The system state is transferred from one state to another by transfer functions and measurement functions, and these operations define the state transfer behaviour. Transfer functions and measurement functions can be executed by either computers or users. The product state is changed from one state to the next by various transfer functions. Figure 3 illustrates the basic behaviour space.

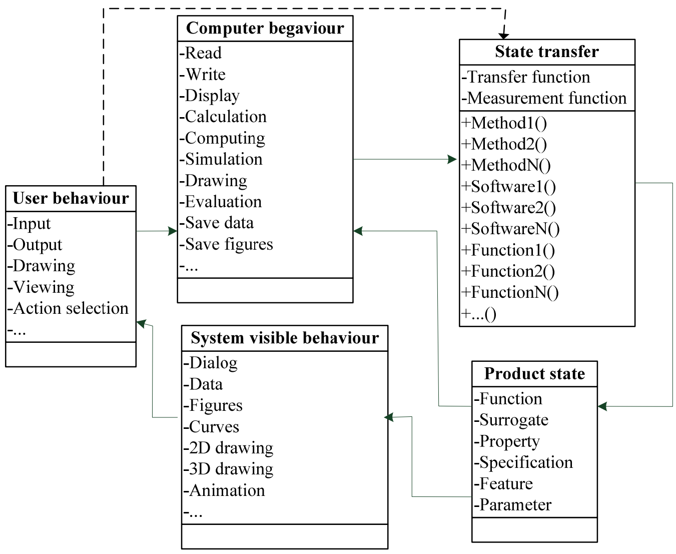


Figure 3. Basic behaviour space

As shown in Figure 3, the set of user activities and the set of computer operations define and change the state of the system, which in turn define the state of the designed product. Some operations of the system are not visible for the users, while others (related to the visible behaviour) create the platform for human-system interaction. Both the user actions and the system operations are logically and temporally arranged and dependent. This implies that the behavioural spaces reflect some procedural order (structure) that is typically sequencing.

3.2 Sequence of behaviours

The sequence of system behaviours is determined by the manifestation of the design and simulation processes. Figure 4 shows a generic design process, whose elements are arranged according to the natural execution logic. Figure 5 shows a combined application of analysis and simulation with the objective of optimizing the working parameters. These processes define the sequence of behaviours in the behaviour space.

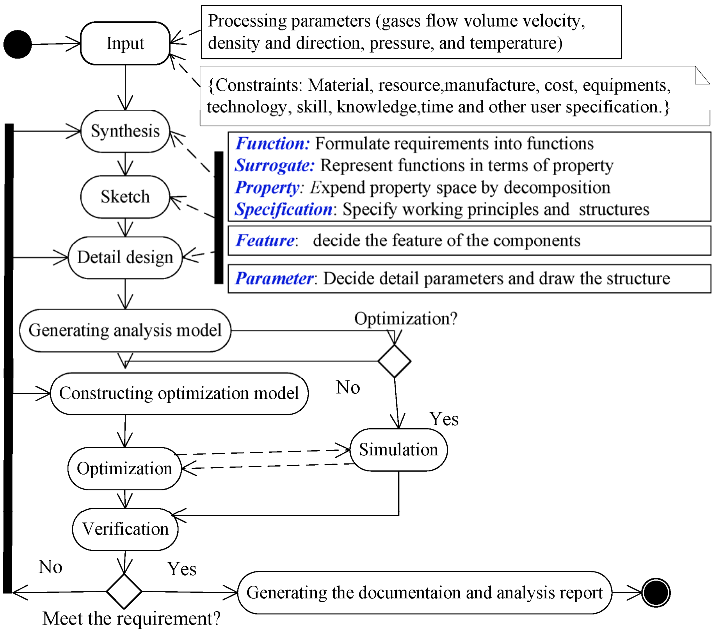


Figure 4. Generic design process

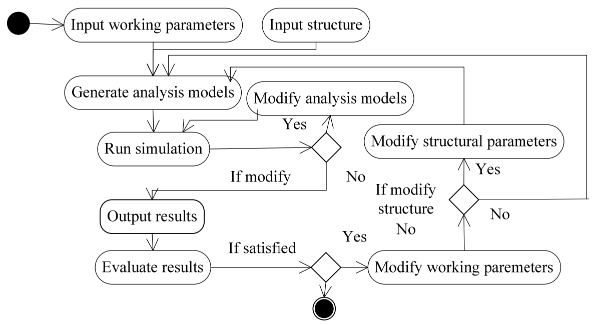


Figure 5. Analysis and simulation supported optimization

3.3 Behaviour tracking and decomposition

The *behaviour* space needs to be decomposed into *Operational Behaviours* so that they can be operated by either human users or computers. This can be done by *behaviour tracking,* starting out from the specified technical criteria, which are derived from the set of parameters (product state). The technical criteria are evaluated by measurement functions and transformed by transfer functions. Transfer functions transfer the product state through state transfer behaviours. The system state is changed by either computer behaviours or user's behaviour, or by both. Computer behaviours are reflected by system behaviours. The usage of the system by users starts by defining the requirements and is terminated when the expected output is available. During the design, analysis and simulation processes, additional parameters may be specified by users. Figure 6 shows the behaviour tracking path.

Behaviour decomposition is required due to different pathways of the design and simulation processes. Generally, more than one technical criterion needs to be satisfied. This entails that different transfer functions are required and different set of product data are required. Users may use different design strategies, *e.g.*, reusing an existing product structure, using existing components to configure a new product structure, using patents as references as basis of synthesis, sketching up a novel product structure, etc. These different design strategies have an influence on the behaviours of the user too.

Figure 7 shows decomposition of the criteria-related behaviours and the decomposition of the user behaviours.

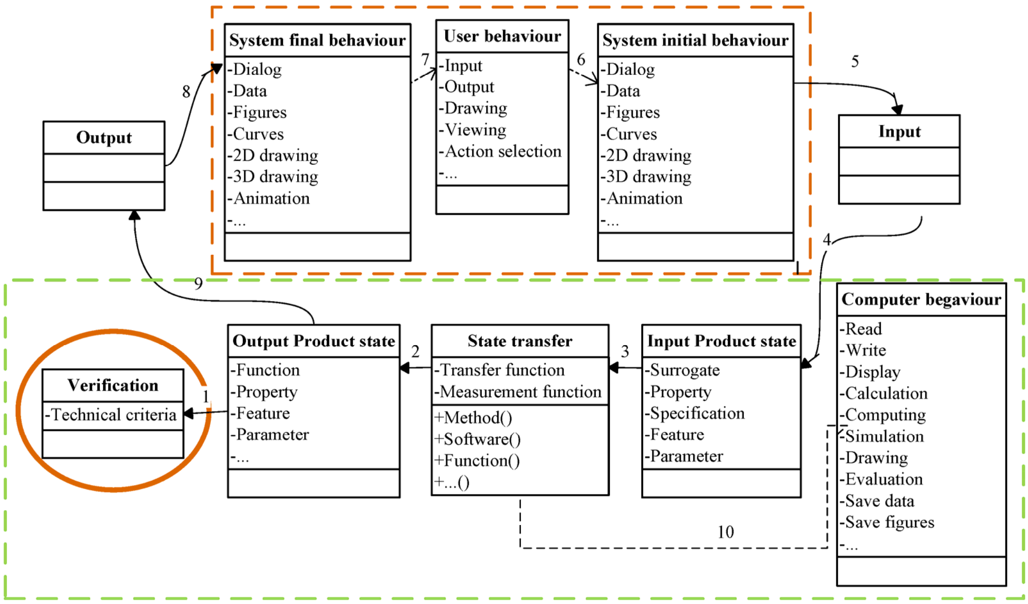
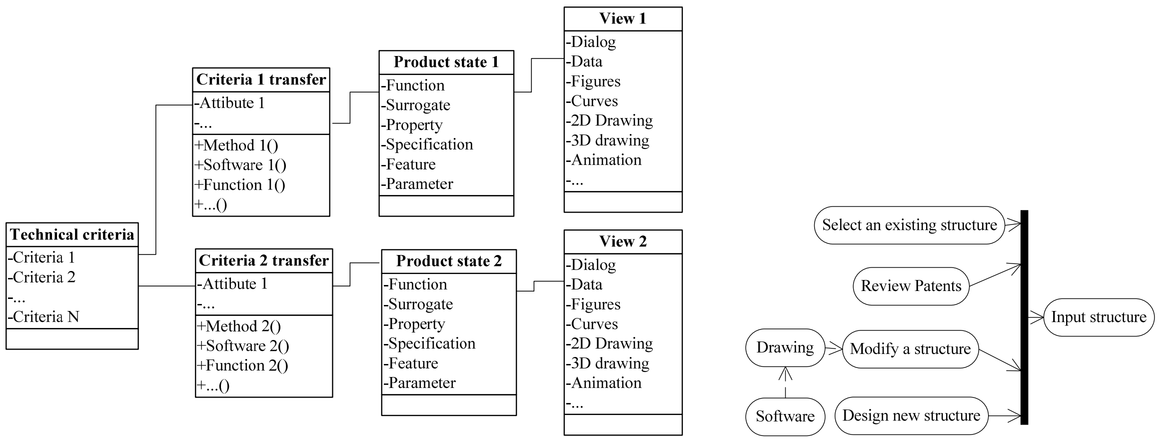


Figure 6. Behaviour tracking through specifying input and output data and related operations



(a) (b)

Figure 7. (a) Decomposition of criteria-related behaviours to computer behaviours, and (b decomposition of input behaviours to user behaviours

3.4 Behaviour prototyping and validation

Once the operational behaviour space is established, the next step is representing the system behaviours and computer behaviours. This process resembles taking snapshots of a running system and establishing a control flow over the set of behaviours. In the case of GUI, system behaviours are represented as layout components. In the case of script input and output, they are provided by the functions of software modules. The computer behaviours are represented as UML models including only basic properties (or attributes) and method names. The act of connecting the behaviour representations according to the sequence of behaviours results in BAP. BAP demonstrates the main features of the designed system. It uses simplified equations and codes to substitute complex and time consuming computing. It uses existing curves, figures, and drawing from literatures and websites to substitute sketch, curves and figures that otherwise would be results of complex computing unavailable in this stage. BAP creates a working model with low precision and low fidelity, but is able to evolve into a real system by enabling the behaviours by real computing. The final step is validation of the BAP through on-line simulation and based on the feedback of the potential customers.

4 BEHAVIOR PROTOTYPING OF A DESIGN SUPPORT SYSTEM FOR PROCESSING CHAMBERS OF IC EQUIPMENT

In this section, we present a demonstrative application example of the design and simulation system for IC processing chambers to show the practical implementation and utility of BAP.

4.1 System requirements for the design and processing simulation of chambers

Processing chambers are used to produce thin films on wafers. The film deposition is a process by which molecules or atoms are transported from the target material to the deposited substrate. The target material may be a metal in a solid state, as in the case of physical vapour deposition (PVD), or materials in gaseous states, as in the case of chemical vapour deposition (CVD). . Typically, a vacuum chamber system is therefore required to deposit thin films. The architecture of a vacuum chamber system is not complex, but the physical and chemical phenomena inside the processing chambers during the deposition process are. A vacuum chamber system consists of: a chamber shell for confining the transportation of the molecules and atoms, a stage on which the substrate is placed, a support to hold the target material in the case of PVD or a shower head in the case of CVD, a gas inlet and distribution system (e.g., a showerhead) to introduce the gasses into the chamber, a pump or a pump system to exhaust the gasses out of the chamber, a power source to drive the gas flow or to make the molecules and atoms to move, a heater to control the substrate temperature so that required deposition film is produced on the substrate surface, a cooler system to control the temperature of the walls of the chamber or of the target material, a transport robot to transfer the wafer into and out of the chamber, a magnetron to control the sputtering of the target material in the case of PECVD, and other accessory components for maintaining the required state inside the vacuum chamber at a specified pressure, temperature, flow density, and velocity. Typically, the first step of designing a chamber system includes the investigation and comparison of similar existing chamber systems. Based on this, a sample implementation is selected. Depending on the difference of the film characteristics, the processing parameters and structural dimensions are modified. Then, the modified parameters are verified by simulations and experiments. The simulation involves multi-field phenomena (Sriram, 2011). When the produced film meets the requirements, the design is accepted. Figure 8 illustrates the overall process.

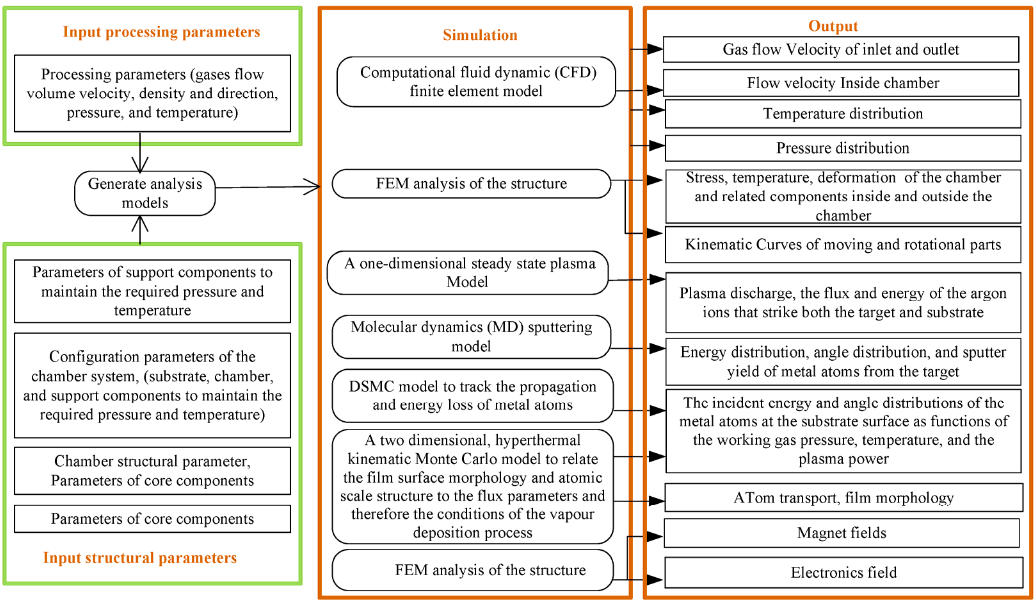


Figure 8. Input data, transfer functions, and output data of simulation

All of these analyses, simulations and experiments are time consuming. Even a minor modification requires various simulations to find optimum values of structural parameters and processing parameters. Hence, parameter optimization is required. Other parts of the chamber system may also need to be modified in consequence. A quick response to the change is preferable for making the production decision. On the other hand, minor modifications may occasionally fail to meet required film quality standards and major modifications or novel design is needed. Therefore, a computer system is necessary to support the design and processing simulation. The main difficulty to develop this computer system comes from the diversity of the domains of research groups and disparate models for various design, analysis and simulation tasks. In the case of our project, more than 10 research institutes and companies have been involved and each group basically know little or nothing about the domain knowledge of other groups. BAP is used to demonstrate the overall and specific features of the design and simulation system and to produce a rough working model on how the system works.

4.2 Criteria behaviour decomposition

The technical criteria to evaluate a thin film depend on the requirement of the film quality. Generally, the film uniformity, film thickness, optical constants and surface roughness are basic requirement. Other optical, electronic and mechanical properties may be required depending on the custom's special requirements. Beside the quality of the film, the deposition rate is also a criterion. Based on the criteria, the basic *behaviour* space and an *operational behaviour* space are produced.

4.3 Validation of prototyping

A rough working model of the future system is developed by the BAP method. Figure 9 illustrates its main features including system/user's behaviours and computer behaviours.

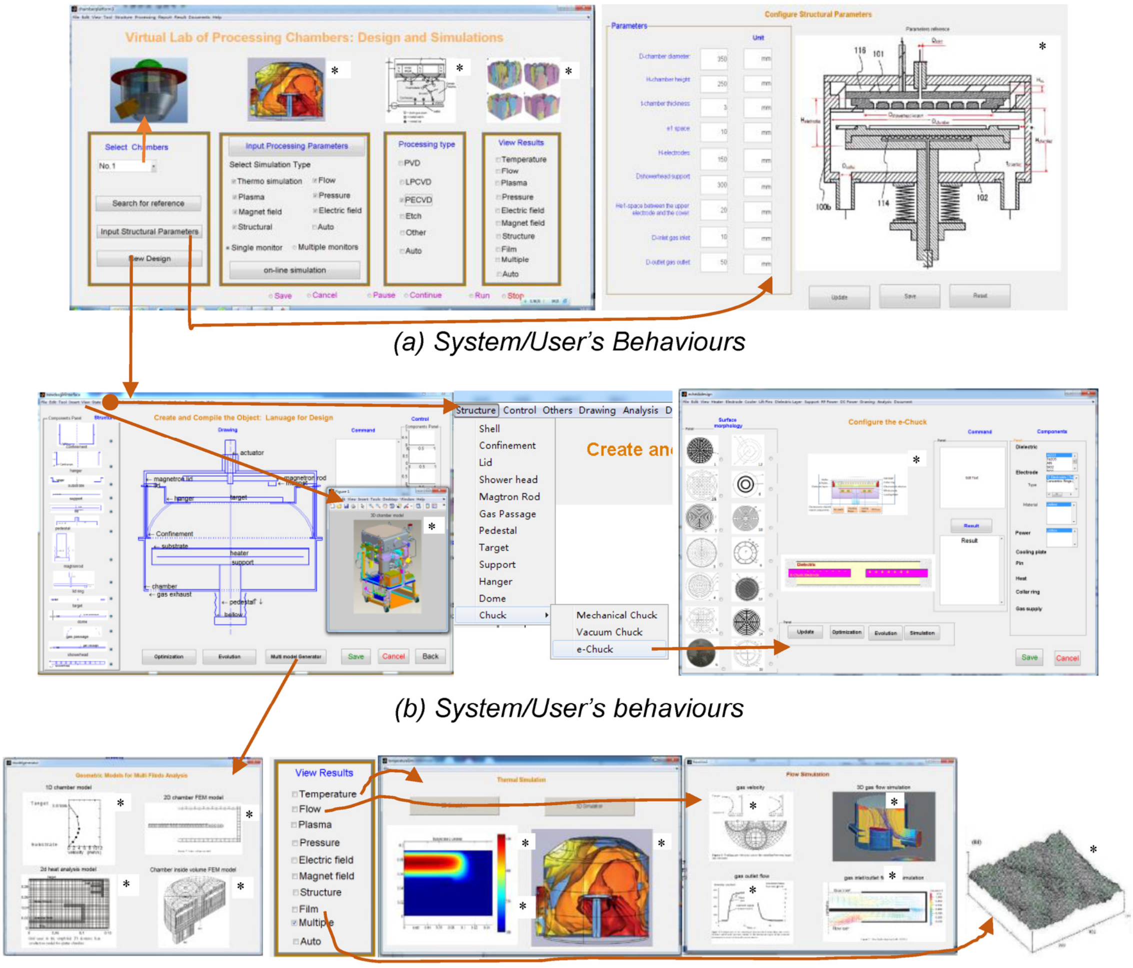


*Figure 9. Behaviour prototyping of: (a) system input/output; (b) design and configuring components of chamber and e-chuck; (c) generating analysis model and thermal flow, and simulation of the film surface morphology. (graphs indicated by * are taken from literature)*

5 DISCUSSION AND CONCLUSIONS

Current prototyping practice is based on advanced computer software tools, and action oriented information flows and workflows, which require specialists in prototyping. We proposed a behaviour attentive approach, which addresses the main issues of system design with limited programming skill and basic knowledge with respect to the design and simulation process. The objective of BAP is to generate a rough working model. The method uses a behaviours tracking strategy and involves five steps: (1) construction of behaviour space; (2) reasoning with the sequence of behaviours; (3) behaviour decomposition; (4) behaviour prototyping; (5) validation on computer. The applicability of BAP to the development of the design and processing simulation system for IC chambers has been shown. Our future work intends to develop mathematics-based behaviour prototyping to improve the fidelity and adaptability of the approach.

REFERENCES

Arcindy (2014) http://www.arcindy.com/concept-prototyping.html.

Beaudouin-Lafon, M., Mackay, W.E. (2014) Prototyping Tools and Techniques, https://www.lri.fr/~mackay/pdffiles/ Prototype.chapter.pdf.

Consantine, L. (1998) Rapid Abstract Prototyping, Software Development, USA. http://www.dtic.upf.edu / ~jblat/material/dissinterf/notes/nidia/ abstractprototypes.pdf.

Davis, W.S and Yen, D.C. (1998) The Information System Consultant's Handbook, Boca Raton, Florida, USA, CRC Press.

Fay, D., Hurwitz, J., Teare, S. (1990) The use of low-fidelity, prototypes in user interface design, Proceedings of the 13th International Symposium of Human Factors in Telecommunications, Torino, Italy, Information Gatekeepers Inc.

Gero, JS (1990) Design prototypes: a knowledge representation schema for design, AI Magazine, Vol. 11, No. 4, pp. 26-36.

Gowda,S., Jayaram S., Jayaram, U. (1999) Architectures for Internet-based Collaborative Virtual, Prototyping, Proceedings of the 1999 ASME Design Technical Conference and Computers in Engineering Conference, Nevada, USA Sept. pp. 11-15.

Horváth, I. (2011) Theoretical Framework for Comprehensive, Abstract Prototyping Methodology, International Conference On Engineering Design, pp. 15-18.

Holden T, Dickerson C., Luff R. (2014) Preliminary Report: A Proposed Model Based Systems Engineering Approach to A Virtual Vehicle Architecture Model (V2AM) For Live-Virtual Testing and Prototyping, IEEE 17th International Symposium on Object/Component-Oriented Real-Time Distributed Computing. Reno, USA.

Hou, Y., Ji, L (2011) Partially autonomous conceptual development of multifunctional structures, International Journal of Computer Applications in Technology, Vol. 40, No. 1-2, pp.13-22.

Janson, M.A and Smith, L.D. (1985) Prototyping for Systems Development: A Critical Appraisal, MIS Quarterly, Vol. 9, No. 4, pp. 305-316.

Porter J., Volgyesi P., Kottenstette N., Nine H., Karsai G., Sztipanovits J. (2009) An Experimental Model-Based Rapid Prototyping Environment for High-Confidence Embedded Software, IEEE/IFIP International Symposium on Rapid System Prototyping, Paris, France.

Kraf C. (2012) Chapter 18, Prototyping and verifying solutions, User experience innovation: User Centered Design that Works, http://link.springer.com/content/ pdf/10.1007%2F978-1-4302-4150-8_18.pdf#page-2

Opiyo, E.Z, Horváth, I., Vergeest, J.S.M (2002) Case Studies On The Application of The Abstract Prototyping Strategy In The Development of Design Support Software, Proceedings of DETC'02/ASME 2002 Design Engineering Technical Conferences, Montreal, Canada, September 29-October 2.

Oxford, (2015) Oxford Dictionaries, Oxford University Press, http://www.oxforddictionaries.com)

Sriram, S. (2011) Surface Morphology Induced Localized Electric Field and Piezoresponse Enhancement in Nanostructured Thin Films, ACS Nano, Vol. 5, No. 2, pp. 1067-1072.

Uckun, S., Kurtoglu, T., Bunus, P., Tumer, I., Hoyle, C., Musliner, D.(2011) Model-based systems engineering for the design and development of complex aerospace systems, SAE Aerotech Congress and Exposition, 2011 October 18-21; Toulouse, France.

UNSL, 2014, Prototyping in Systems Analysis, http://www.umsl.edu/~sauterv/analysis/488_f01_papers/Hammer/term_paper_body.htm#Introduction.

Wood, D.P., Kang, K.C. (1992) A classification and bibliography of software prototyping, CMU/SEI-92-TR-13, ESD-92-TR-013, Software Engineering Institute, Carnegie Mellon University, pp. 1-86.

ACKNOWLEDGMENTS

The presented work is jointly supported by National Natural Science Foundation of China (51175284), the Ministry of Science and Technology of the People´s Republic of China (2011ZX02403) and the Oversea Collaboration Innovation Team Project of Tsinghua University.