Journal of Modern Education Review, ISSN 2155-7993, USA
July 2016, Volume 6, No. 7, pp. 470–477
Doi: 10.15341/jmer(2155-7993)/07.06.2016/005

http://www.academicstar.us

Modeling and Simulating Persuasion Strategies: Domain Specific Language for Persuasive Systems' Design Productivity

Saad Abdessettar[1], Mickaël Gardoni[1,2], Bessam Abdulrazak[3]
(1. École de Technologie Supérieure, Montréal, Canada; 2. INSA Strasbourg, France;
3. Sherbrooke University, Sherbrooke, Canada)

Abstract: The exponential growth of Information and Communication Technologies (ICT) has changed our way of performing tasks in diverse domains such as economy, health, and higher education. Still the effectiveness of the ICT is not maximal due to lack in efficient Human Machine Interaction (HMI) design. Hence, several studies have emerged in the persuasive technologies science to address the limitation of the HMI design. Nevertheless, there are still lack of tools and models that enable better productivity management. Indeed, all persuasive system design models highlight the need for tests before implementing persuasive strategies. These tests are challenging as they are often time and resource consuming. Our research work addresses the question: At which extent Domain-Specific Languages can enhance productivity in Persuasive Design processes. These Languages can provide the expressiveness power needed either for designing and testing persuasion strategies. In this paper, we discuss our analysis of existing tools and models and we explore the possibilities and limitations of a Domain-Specific Language to be applied and/or adapted to persuasive technologies.

Key words: persuasive technology, Captolgy, behavior engineering, DSL, domain-specific language, persuasion strategy, behavior change, behavior design

1. Introduction

Assisting designers throughout the Persuasive Design processes is key productivity success. Though, there are still lack of tools and models that enable better productivity management. Indeed, all persuasive system design models highlight the need for tests before implementing persuasive strategies.

Persuasive systems can be seen as those designed to change user attitude or behavior without coercion or deception (B. J. Fogg, 2002). In this domain, Fogg proposed an eight step methodology to design persuasive systems, which was refined later in a Behavior Wizard (B. Fogg & Hreha, 2010). Numerous researchers proposed Persuasive Design models, inspired from Fogg's research activities. Lockton, Harrison, & Stanton (2008)

Saad Abdessettar, Ph.D. Candidate, École de Technologie Supérieure (ÉTS) (Montreal, Canada); research areas/interests: persuasive technologies & strategies, innovative design, smart-environments, human factor & innovation. E-mail: saad.abdessettar.1@etsmtl.ca.

Mickaël Gardoni, Ph.D., Professor, Department of Automated Production Management (ÉTS), INSA; research areas/interests: knowledge management, innovation management, innovative design, human factor & innovation; E-mail: mickael.gardoni@etsmtl.ca.

Bessam Abdulrazak, Ph.D., Professor, Department of Computer Engineering, University of Sherbrooke; research areas/interests: persuasive computing, ubiquitous and pervasive computing, ambient-intelligence, smart-environments, assistive living technologies, context-awareness and software engineering; E-mail: bessam.abdulrazak@usherbrooke.ca.

proposed the Design with Intent (DwI) method, which target a wider field than Persuasive Technology. The method is based on a "suggestion tool", inspiring design solutions by proposing techniques and examples applicable to particular target behaviors. Oinas-Kukkonen & Harjumaa (2009) presented a new framework for designing and evaluating persuasive systems called "Persuasive System Design (PSD) Model". The framework underlined seven postulates behind a three phases PSD model that focus on the use of contexts. Later, Oinas-Kukkonen (2010) proposed the "Behavior Change Support Systems (BCSS)" as an information system designed to form, alter or reinforce attitudes, behaviors or an act of complying without using deception, coercion or inducements. All these four existing Persuasive Design models have significantly contributed to the advance of knowledge in persuasive technology science. Nevertheless, productivity limitations sill need to be addressed (Torning, 2013). We focus in this paper, on the productivity aspect related to the design process, and we stress the need for more efficient techniques to design persuasive strategies. We then, explore the opportunities of a Domain-Specific Languages (DSL) to facilitate persuasion strategies design. Indeed, DSLs offer focused expressive power that can facilitate designing and testing of persuasion strategies.

In this paper, we will present an analysis of existing persuasive system design, and evaluation methods and models, after introducing the CAPTOLOGY science field. Then, we will discuss our finding from the analysis. We will then explore more DSLs and their capacity to bring expressiveness that can facilitate Persuasive Design processes. We will end our paper with a conclusion and future work perspectives.

2. Persuasive Technologies and Captology

Brian J. Fogg (1998) introduced a new area of inquiry called Captology referring to the study of computers as persuasive technologies. The definition of "Captology" was introduced by a group of researcher interested in exploring the domain of computers and persuasion in the CHI annual international conference in 1997. The definition of "Persuasion" was introduced by Fogg as "an attempt to change attitudes or behaviors or both (without using coercion or deception)". Thus persuasive technologies refer to computing product, application or systems that are designed with the "intention" to change behavior or attitude or both without coercion or deception. Later, B. J. Fogg (2002) published his famous book: "Persuasive technology: using computers to change what we think and do." The book emphasized on computer functions throughout what Fogg described as functional triad. He claimed that computer plays their role in three basic ways from the perspective of the user: as tools, as medium, and as social actors.

- Computer as persuasive tool provides humans with new ability or power, allowing people to do things they could not do before, or to do things more easily.
- Computer as a Medium can convey either symbolic content (e.g., text, data graphs, and icons) or sensory content (e.g., real-time video, simulations, and virtual worlds). Then, computer can be persuasive by assisting users in exploring experiences that can motivate or persuade them.
- Computer as Social Actor, can adopt animate characteristics, play animate roles, or follow social rules or dynamics. As Social Actor computing products can be persuasive by modeling target behavior and providing social support to attain it.

The work of Fogg was an inspiration for a number of researchers around the world that tried to advance the sciences of persuasive technology. Following, we present our analysis of the existing Persuasive solutions, process and methods.

3. Analysis of Persuasive System Design & Evaluation

3.1 Fogg's Eight Step Design Model

Fogg was the first to (Brian J Fogg, 1998; B. J. Fogg, 2002) present a model that can guide designers to design a persuasive system. He proposed an eight step model : 1) Choose a simple Behavior to target, 2) Choose a Receptive Audience, 3) Find Out What is Preventing the Audience from Performing the Target Behavior, 4) Choose a Familiar Technology channel, 5) Find Suitable Examples of Persuasive Technology, 6) Imitate Successful , 7) Test and Iterate Quickly , 8) Expand on Success. The last four steps are more challenging, as they prerequisite extensive knowledge in persuasion and behavior change Field. For non-domain-expert, finding the right examples, and testing is often not a trivial process.

3.2 Design with Intent (DwI)

(Lockton, Harrison, & Stanton, 2008) proposed their vision on designing persuasive system by adopting the DwI method. According to them, Persuasive Technology is a part of DwI. They argue that DwI don't exclude coercion to change to target behavior. They claim that forcing functions can be used as tactics for Behavior Shaping. They also claim that the design is persuasive anyway, and propose a more complete model that emphasized on a design intended to influence or to impact user behavior. Lockton et al. also proposed two main modes in their model: (1) Inspiration mode and (2) prescription mode (Lockton, Harrison, & Stanton, 2010).

(1) In the **inspiration** mode, designers take inspiration from a set of "headline" design patterns, which are applicable to a wide range of target behaviors. These patterns are grouped into six different "lenses" as follow: Architectural, Error proofing, Persuasive (this group of design patterns, represent the emerging field of persuasive technology defined by Fogg (B. J. Fogg, 2002), Visual, Cognitive and Security lenses. These lenses represent particular disciplinary perspectives on using design to influence behavior.

(2) In the **prescription** mode, designers formulate a range of target behaviors describing interactions for each target behavior. The total number of patterns varies depending on the chosen target behavior(s) (typically 15–25 applicable patterns). This mode effectively 'prescribes' a set of patterns, which are deemed especially applicable or have already been applied to similar problems in other contexts.

By classifying behavior patterns, DwI offers more clarity by guiding the design process throughout the proposed six lenses. Nevertheless, there is no tool or method to build persuasion strategy in order to bring the final user to the target behavior. Inspiration and prescription dimensions of DwI, are like the last four steps proposed by Fogg in the section above, they lack tools and methods for building persuasion strategies.

3.3 Persuasive System Design (PSD) Process Model

Oinas-Kukkonen & Harjumaa (2009) presented PSD as a systematic framework to design and evaluate persuasive systems. The PSD process model is based on seven postulates that frame the design process: 1) Information Technology is never neutral, 2) people like their views about the world to be organized and consistent, 3) direct and indirect routes are key persuasion strategies, 4) persuasion is often incremental, 5) persuasion through persuasive systems should always be open, 6) persuasive systems should aim to unobtrusiveness, 7) persuasive systems should aim at being both useful and easy to use.

The PSD model is comprised of three distinct phases: (1) First, designer must understand key issues behind persuasive systems; the seven postulates can frame this understanding. (2) Then in the second phase, the design have to Analyze the persuasion context which requires a thorough understanding of what happens in the information processing event, namely understanding the roles of persuader, persuade, message, channel, and the larger context (Oinas-Kukkonen & Harjumaa, 2008). (3) Finally, designers would be ready to select and implement the system qualities. The optimal result of this linear model is Behavior and/or attitude change. In this final phase designers can select from an extensive catalogue listing 28 design principles for persuasive system content and functionality.

Oinas-Kukkonen & Harjumaa (2009) insisted on the relevance of the context analysis to efficiency and effectiveness of the designed persuasive system. They claimed that it would be hard or even impossible to recognize inconsistencies in a user's thinking, discern opportune and/or inopportune moments for delivering messages, and effectively persuade. Oinas-Kukkonen & Harjumaa (2009) proposed seven core elements to create a better understanding of the context of persuasion: Persuader, Change Type, Use Context, User Context, Technology Context, Message and the Route (persuasion can be direct, indirect or both).

The PSD model proposes a more systematic approach to design persuasive systems. The approach focuses on the context of use, by guiding the design process to consider all actors involved in user context awareness. Though, when it comes to build or test persuasion strategies, there is still a big challenge in time and resource management.

3.4 Behavior Change Support System (BCSS)

Oinas-Kukkonen (2010) introduced the Behavior Change Support System as an extension to hi work on PSD process model. He presented BCSS as an information system to form, alter or reinforce attitudes, behaviors or an act of complying without using deception, coercion or inducements. He claimed that BCSS constitute an object of study in the field of Persuasive Technology.

Oinas-Kukkonen (2010) also defined an Outcome/Change design matrix that filled from the intended outcomes and the types of change. Designers can use the matrix as a referential to evaluate which of these nine different goals the application will be built for. Oinas-Kukkonen (2010) defined three types of change: A-Change (a change in an act of complying), B-Change (a behavior change), C-Change (an attitude change). He also defined three types of outcome: F-Outcome (a forming outcome: means the formulation of a pattern for a situation where one did not exist beforehand), A-Outcome (an altering outcome, means changes in a person's response to an issue), R-Outcome (a reinforcing outcome, means the reinforcement of current attitudes or behaviors, making them more resistant to change).

BCSS is adapted to behavior shaping, and guide designers to build behavior change tactics or strategies based on the matrix proposed. Nevertheless, there is still a lack in productivity that can be mitigated by bringing expressiveness and visualizing modeling which can facilitate behavior shaping strategies' design and evaluation.

3.5 Behavior Wizard

B. Fogg & Hreha (2010) proposed the Behavior Wizard as an outcome-based method for classifying research and design related to persuasive technology in 2010. The aim of this method is to match types of target behaviors with solutions for achieving those targeted behaviors. This matching is to be done in a matrix of Behavior change, qualified by Fogg as the foundation of the Behavior Wizard's first phase. The Behavior Wizard is a method of identifying specific types of behavior targets, and matching those targets to relevant solutions. This method encompasses three steps: 1) identify target behavior, 2) identify triggers and frequencies, and 3) map behavior strategy to the 15 cells grid of Behavior Wizard.

This method is more rich and concise as it summarizes findings in psychology and behavior change field in a fifteen cells matrix of behavior types and change types. Still, similar to the methods discussed above, there is a big challenge for designers to build the efficient behavior change strategy and test it quickly. Especially, in designing strategies for complex target behavior that need several combinations of broken down behaviors either in sequence or in parallel.

3.6 Persuasion Evaluation

Andrews & Manandhar (2009) assess the validity of the ranking task in theory of persuasion. They also describe a formalization of the ranking task that provides an evaluation metric and a standard measure. The formulation highlights the difference between perception (attitude and/or beliefs) on the initial point, and on a final point of a persuasion session. Andrews & Manandhar (2009) based their formula on the Theory of Reasoned Action (Ajzen & Fishbein, 1980). They claim that their ranking formula does not provide a detailed view on every internal belief that user holds about situations. However, if user changes this ranking, this change represents a measurable change in the internal beliefs. They also argue that according to the Theory of Reasoned Action, this change in beliefs has an impact on user's intention towards the targeted behavior, and they can assume therefore that the measured persuasion has an influence on the behavior too.

3.7 Discussion on Methods and Models

The persuasive system design PSD methods and models (presented above) are based or inspired from the Fogg Behavior Model (FBM) (Brian J. Fogg, 2009). In his FBM, Fogg stressed that in order to drive user (persuaded person) to a target Behavior, three conditions have to be fulfilled: 1) Motivation to do the Behavior, 2) Ability or capacity to do it easily, and 3) the trigger that would fit the right moment to make the Behavior happen. Another model has been proposed by Michie et al. (2011) to capture a range of mechanisms and rules, that may be involved to ensure behavior change effectively and efficiently. The model was named The Capability, Opportunity and Motivation (COM-B). The later was based on existing theories of behavior and a consensus meeting of behavioral theorists. COM-B hypothesizes that in order to get the target behavior performed, there must be interactions between Capability, Opportunity and Motivation. (Michie et al., 2013) proposed then her Behavior Change Technique (BCT) with a defined taxonomy, to guide behavior change designers in their persuasion strategy or behavior change design.

All Persuasive Design methods discussed above highlight the need for iterative and quick tests, in order to give designers the ability to evaluate and adapt his persuasion strategy while implementing it. To our knowledge, there is no solution on tools or frameworks for designing and testing persuasive strategies. Existing solutions focus mainly on methods and methodologies to guide the design process of the persuasive system. There is less attention given to facilitate persuasion strategy design, test, simulation and implementation. Moreover, the majority of the presented Persuasive Design methods and models do not offer solutions to measure the efficiency of their Persuasive Design outputs, or test and evaluate the persuasiveness or the feasibility of the persuasion strategy embedded in the designed product. From another point of view, designers of persuasive systems lack a common vocabulary of expressing behavior change tactics and persuasion strategies, which complicate specification, communication, collaboration and agreement between domain experts.

We believe that the DSLs can represent an opportunity to overcome the limitations of the existing Persuasive Design methods/process. DSLs have been successfully applied in numerous domains such as financial engineering and mathematics (FORTRAN, COBOL), artificial intelligence (LISP) etc. Those applications resulted in substantial improvements in quality and productivity. Still, DSLs have not been explored in the domain of Persuasive Technology. Therefore, our research aims at introducing a DSL for Persuasive technologies. The main goal of this introduction is to bring the expressiveness power of DSL, to let designers visualize the persuasion strategy and the behaviors sequence that lead to the target behavior. Hence they can even test their designed sequences or persuasion strategies throughout simulations.

4. Domain-Specific Language (DSL) for Persuasive Technology

DSLs were introduced in computer sciences to resolve problems related to specific application domains. For examples, FORTRAN is a language used for scientific and financial calculus; LISP is another language that was originally created as a practical mathematic notation, and now used as a programming language for artificial intelligence.

Definitions of DSL are similar. Van Deursen, Klint, & Visser (2000) defined "DSL" as a programming language or executable specification language that offers, through appropriate notations and abstractions, expressive power focused on, and usually restricted to, a particular problem domain. Fowler (2010) defines DSL as a computer programming language of limited expressiveness focused on a particular domain. He categorizes DSLs into three: 1) External DSLs (a language separate from the main language of the application it works with), 2) internal DSLs (a particular way of using a general-purpose language), and 3) language workbench (a specialized IDE for defining and building DSLs).

Fowler (2010) stated that DSLs are a tool with limited focus. They aren't similar to object orientation or agile processes which introduce a fundamental shift into the way we think about software development. Instead, DSLs are a very specific tool for very particular conditions. In fact, a typical project might use half a dozen of DSLs for various specific needs.

Van Deursen et al. (2000) stress the risks and opportunities of using a DSL. They state that the benefits list includes the ability to express solutions in idioms and at abstraction levels (of the problem domain). The list also includes the conciseness and reusability of the developed programs, and an enhanced productivity, reliability, maintainability and portability. Moreover, Van Deursen et al. (2000) and Fowler (2010) state the expressiveness power of DSLs. This expressiveness lead to better communication with domain expert in order to embody the domain knowledge. Fowler (2010) also discussed the advantage of the ability to adapt the design specifications to the execution context.

Van Deursen et al. (2000) enumerate disadvantages of the use of DSLs: design, implementation and maintenance costs, education and training costs, limited availability of DSLs, difficulty of finding the proper scope for a DSL, difficulty of balancing between domain-specificity and general-purpose programming language constructs, the potential loss of efficiency when compared with hand-coded software.

Numerous research studies (Cleaveland, 1988; P. Klint & R. Van Rozen, 2013; Van Deursen & Klint, 1998; Van Deursen et al., 2000) have been undertaken on how to design and implement DSLs. The majority shares the following steps:

- Analysis: (1) Identify the problem domain. (2) Gather all relevant knowledge in this domain. (3) Cluster this knowledge in a handful of semantic notions and operations on them. (4) Design a DSL that concisely describes applications in the domain.
- Implementation: (5) Construct a library that implements the semantic notions. (6) Design and implement a compiler that translates DSL programs to a sequence of library calls.
- Use: (7) Write DSL programs for all desired applications and compile them.

We believe that bringing the expressiveness power of DSLs to persuasive design processes will enable bring agility and flexibility, as well as reduce the time to finish the design process and improve productivity. Therefore, our goal is to build a DSL for Persuasive technologies. In the next section, we discuss our analysis of the closest relevant activities to our research study.

5. Relevant Works

An analysis of the DSL led lead as to conclude that three languages are relevant to our study: 1) Behavior Specification Language, 2) machination and micro-machination, and Behavioral Description Language.

(Loetzsch, Risler, & Jungel, 2006) proposed the Extensible Agent Behavior Specification Language (XABSL) as a pragmatic and formal approach to the design of agent behavior. XABSL is based on hierarchies of Finite State Machines for action selection and supports the design of long-term and deliberative decision processes as well as of short-term and reactive behaviors. The proposed language was dedicated to robotic industry, in order to offer a platform-independent execution engine that makes the language applicable. XABSL is a convenient and powerful system for the development of complex behaviors, as it includes a variety of visualization, editing and debugging tools.

Paul Klint & Riemer van Rozen (2013); Van Rozen & Dormans (2014) brought a new dimension that can enhance design effectiveness and efficiency in game design. They presented the concept of machination and micro-machination as a domain specific language, and how it can help well design and evaluate game mechanics. Van Rozen & Dormans emphasized that machination starting point is the notion of internal economy for games that describes game dynamics in terms of distribution and flow of game resources. Game resources include tangible resources such as money, property, and food. The Micro-Machination concept also applies to abstract notions such as hit points, experience points, and strategic advantage. It uses a diagrammatic language to visualize a game's internal economy.

A similar attempt was realized by Bertrand & Augeraud (1999). They proposed a domain-specific language called BDL Behavioral Description Language. They designed BDL to express the control of the behavior of simple objects in object-oriented programming, or group of objects based on a reactive language.

6. Conclusion & Future Work

Persuasive technology is now a relatively mature science domain compared to what it was since its initiation by (Brian J Fogg, 1998). We discussed in this paper the existing process and methods in the field, and we highlighted the need to bring agility and flexibility in the persuasive systems design.

As a solution to the existing limitations, we proposed and discussed the use of domain-specific language in order to improve Persuasive Design processes productivity, and facilitate communication and collaboration between domain experts. Currently, we are working on the design and implementation of the proposed DSL for persuasion strategy design. We plan to evaluate the language in a real education case in collaboration with an industrial partner. Our partner is a multinational leader in mobile devices and telecommunication: DataWind inc. The application case will be a persuasive self-learning environment for Kindergarten to K12 curriculum, that will be embedded in DataWind tablets and distributed to rural kids who have limited or no access to schools and teachers in developing countries such as India.

References

Ajzen I. and Fishbein M. (1980). *Understanding Attitudes and Predicting Social Behavior*, Prentice-Hall, Englewood Cliffs, NJ.

Andrews P. and Manandhar S. (2009). "Measure of belief change as an evaluation of persuasion", in: *The Persuasive Technology and Digital Behaviour Intervention Symposium*.

Bertrand F. and Augeraud M. (1999). "BDL: A specialized language for per-object reactive control", *Software Engineering*, Vol. 25, No. 3, pp. 347–362.

Cleaveland J. C. (1988). "Building application generators", *IEEE Software*, No. 4, pp. 25–33.

Fogg B. and Hreha J. (2010). *Behavior Wizard: A Method for Matching Target Behaviors with Solutions Persuasive Technology*, Springer, pp. 117–131.

Fogg B. J. (1998). "Persuasive computers: Perspectives and research directions", in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Fogg B. J. (2002). "Persuasive technology: Using computers to change what we think and do", *Ubiquity*, Vol. 2, doi: 10.1145/764008.763957.

Fogg B. J. (2009). "A behavior model for persuasive design", in: *Proceedings of the 4th international Conference on Persuasive Technology*.

Fowler M. (2010). *Domain-specific Languages*, Pearson Education.

Klint P. and Van Rozen R. (2013). *Micro-Machinations Software Language Engineering*, Springer, pp. 36–55.

Klint P. and Van Rozen R. (2013). "Micro-machinations: A DSL for game economies", Vol. 8225 LNCS, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 36–55.

Lockton D., Harrison D. and Stanton N. (2008). *Design with Intent: Persuasive Technology in a Wider Context Persuasive Technology*, Springer, pp. 274–278.

Lockton D., Harrison D. and Stanton N. A. (2010). "The design with intent method: A design tool for influencing user behavior", *Applied Ergonomics*, Vol. 41, No. 3, pp. 382–392.

Loetzsch M., Risler M. and Jungel M. (2006). "XABSL-a pragmatic approach to behavior engineering", in: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Michie S., Abraham C., Eccles M. P., Francis J. J., Hardeman W. and Johnston M. (2011). "Strengthening evaluation and implementation by specifying components of behaviour change interventions: A study protocol", *Implement Sci.*, Vol. 6, No. 10, p. 10.

Michie S., Richardson M., Johnston M., Abraham C., Francis J., Hardeman W. and Wood C. E. (2013). "The behavior change technique taxonomy (v1) of 93 hierarchically clustered techniques: building an international consensus for the reporting of behavior change interventions", *Annals of Behavioral Medicine*, Vol. 46, No. 1, pp. 81–95.

Oinas-Kukkonen H. (2010). *Behavior Change Support Systems: A Research Model and Agenda Persuasive Technology*, Springer, pp. 4–14.

Oinas-Kukkonen H. and Harjumaa M. (2008). *A Systematic Framework for Designing and Evaluating Persuasive Systems Persuasive Technology*, Springer, pp. 164-176.

Oinas-Kukkonen H. and Harjumaa M. (2009). "Persuasive systems design: Key issues, process model, and system features", *Communications of the Association for Information Systems*, Vol. 24, pp. 485–500.

Torning K. (2013). "A review of four persuasive design models", *International Journal of Conceptual Structures and Smart Applications*, Vol. 1, No. 2, pp. 17–27, doi:10.4018/ijcssa.2013070103.

Van Deursen A. and Klint P. (1998). "Little languages: Little maintenance?", *Journal of Software Maintenance*, Vol. 10, No. 2, pp. 75–92.

Van Deursen A., Klint P. and Visser J. (2000). "Domain-specific languages: An annotated bibliography", *Sigplan Notices*, Vol. 35, No. 6, pp. 26–36.

Van Rozen R. and Dormans J. (2014). "Adapting game mechanics with micro-machinations", paper presented at the Foundations of Digital Games.