Volume 18, Number 2, 2016

Global Journal of Engineering Education

Improving the learnability of system dynamics simulation tools: a new design for a tablet device

Ricardo Sotaquirá-Gutiérrez, Francisco Eduardo Ramírez-Rosales, Juan Pablo Velásquez-Díaz & Jenny Constanza Robayo-Gómez

Universidad de La Sabana
Chía, Colombia

ABSTRACT: A common problem for novice users of several engineering simulation tools is that they perceive a great gap between the mathematical model and how it will be represented in software applications. Available packages, particularly for system dynamics simulation methodology, do not overcome this gap, because they were not designed within the framework of user interface and user experience design. Therefore, the simulation tool itself becomes a barrier to learning this methodology. In this article, the authors present a new visual and interactive design for system dynamics modelling, which could foster the experience for the user. To test this design, prototype software, *Flux*, was implemented on the iPad tablet platform. A learnability test of the proposed solution against a well-known software package showed an improvement of the initial performance for engineering students and of their understanding of system dynamics concepts. This research also revealed great opportunities to apply user-centred design and multi-touch interfaces to improve the user experience of simulation software in engineering, sciences and management.

Keywords: User interfaces, user experience, mobile computing, system dynamics, user-centred design, interactive design

INTRODUCTION

Even well prepared engineering students have great difficulties when they try to learn a simulation methodology using available software tools. To master any of these methodologies, the learner requires highly developed skills for mathematical modelling and abstract thinking.

Current simulation tools for engineering offer graphics and animation that illustrate these abstract models and ease to the user the understanding of the analysed system. However, these visual elements do not solve the gap between the formal/mathematical model and the clear representation of such models in the software package. There is a new field of exploration, experimentation and innovation of multimedia software based on user experience design to improve the learning process of engineering simulation methodologies.

This research was focused on examining this general problem for a particular simulation methodology: system dynamics [1-3]. System dynamics is a widely used approach to build quantitative models that help decision-making problems or complex systems analysis. It is a generalist framework with a lot of relevant applications on fields like engineering, management and science.

The visual language of system dynamics models was established 50 years ago [1] and it has been the basis for all available software tools. This approach uses an abstract diagram, named stocks and flows diagram, which represents a mathematical model. The four most used software tools in system dynamics [4-6] share the following common features: 1) they are software tools for desktop computers; 2) the key element of their graphical user interface (GUI) was a customary icon bar with a set of elements and functions for building system dynamics models. They did not support gesture interaction or multi-touch user interfaces (Figure 1); 3) the GUI shows two main types of static graphics elements: a stock and flow diagram, and time charts of simulation results. This kind of static visual interface is one of the main barriers for novice users; and 4) the GUI is generally crowded with elements, charts and diagrams in a confusing way.

This design approach has been fruitful for expert users, but it has imposed learning barriers for novice users of the system dynamics methodology; they feel even *intimidated* by these tools [7]. There are opportunities to explore and propose new visual and interactive design alternatives clearly oriented to improve user experience, especially, for beginners.

The main purpose of this research was to design an alternative visual and interactive representation, which could solve the aforementioned problems, improve the learnability of these engineering simulation techniques and, thus, fostering their diffusion and application in real world scenarios. The authors present a visual design, an interaction design and a software prototype of this solution. User experience and interaction design principles were applied to the design of the system. The software prototype was developed for a multi-touch user interface and a mobile device. The last section of the article presents the results of a learnability user study.

RELATED WORK

Various projects have led to the design of software for non-expert users in the field of mathematical modelling [8-10]. One of these research directions has examined how to improve the interface to help the user to understand abstract concepts and models. The explored solutions implied a need to design new and enriched forms of scientific visualisations. Bret Victor, formerly an apple engineer, has developed some inspiring projects from this perspective. He said, …*a person should not have to imagine the interpretation of abstract symbols* [11], instead of this, a software platform must offer him/her simpler and more direct visual dynamic elements. He speaks about the *kinetic dimension* of modelling: in order to understand certain symbolic expressions, a human being needs to see and experience the meaning of them. His solution is not based on static diagrams; he develops some specific animated or dynamics visual representations of specific mathematical models. His solutions are not general, they offer a hint of possible visual and software strategies for different mathematical models. The authors propose a general design for any mathematical model in system dynamics.

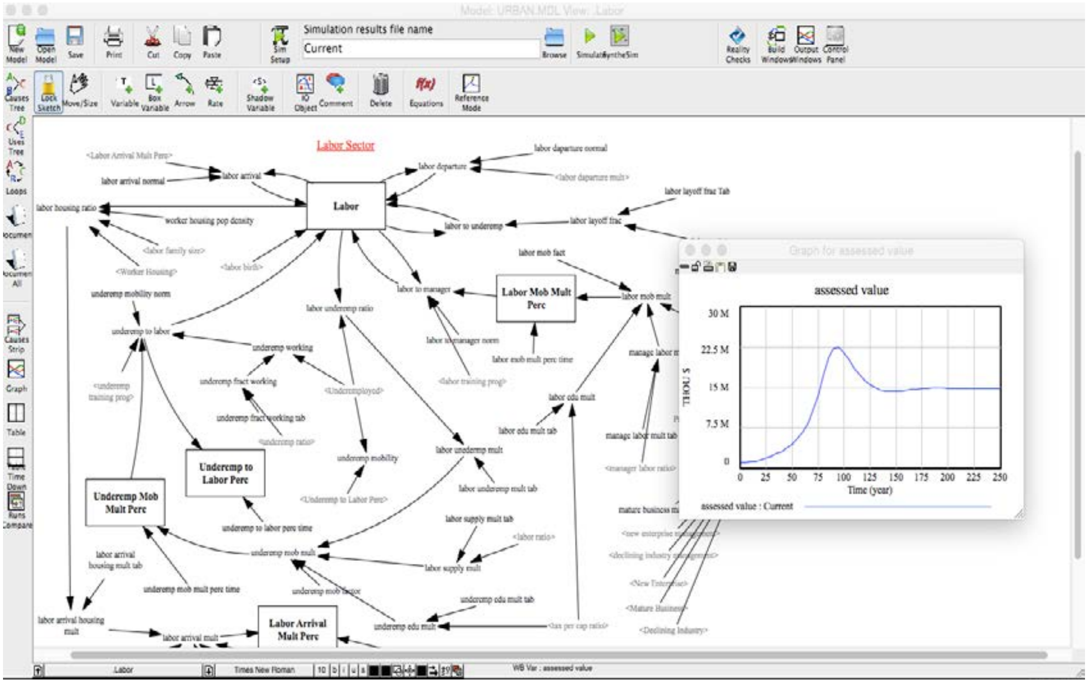

Figure 1: A conventional GUI of system dynamics software - *Vensim*.

In the field of system dynamics, three previous works have emphasised the key role of visualisation and a proper interface design in order to overcome the learning barriers of users [12-14]. All of them explored solutions for specific cases. The first two used static visualisations; however, the last one presented a dynamic visualisation not of the mathematical model, but of the simulation results. The authors were interested in general solutions for a dynamics visualisation of the system dynamics model.

However, improving visualisation and the GUI is not the only direction explored in recent research projects. Another way to make the model understanding easier for novice users is to introduce new technologies and new kinds of user interfaces. System blocks is a good reference for the system dynamics case: a …*set of computationally enhanced children's blocks, made of wood and electronics, the system blocks can assist K-12 educators to teach the complex concepts of system dynamics and causalities* [15]. Experiments with these system blocks showed a fun and effective option concerning learnability for youngsters.

This project explores an alternative to the conventional GUI for modelling, involving the design of a physical interface. One of the purposes of this interface is to overcome a learning barrier: *concretising the abstract* [15]. The interface of the system dynamics modelling tools must enable the user to connect the mathematical model to the real world. Current software tools just use static diagrams to represents variables and relationships of a model.

In the system block design, wooden-electronic blocks with *LED* displays replaced these variables and the relationships were replaced with electric cables. But, the authors of that project concluded that this physical interface was not enough for concretising the abstract model. It is understood that the platform also must develop a *kinetic* dimension, that is, a way to expose the dynamic behaviour of the system. The authors observed here a very important insight to be explored.

The system dynamics methodology aims to capture, through a mathematical model, the dynamic behaviour of a real life situation. However, conventional software tools do not offer a dynamic representation of this real life problem; they do not offer the user a kinetic dimension. This would be a relevant learning barrier, especially, for novice users who have not yet developed a mature mental model of this approach. System blocks just hinted at this insight, but offer a limited way to improve the kinetic dimension of the platform: they experimented with a *musical instrument digital interface* (MIDI) sound module implemented on certain blocks to give sound feedback about the values of the variables. They test this experimental platform with children and they give positive feedback. The authors aimed to design a tool for a wide population of users: novice users, particularly engineering students.

In summary, previous studies have reported that a common problem with software tools for mathematical modelling is the user experience gap between the abstract model and its symbolic representation in software. These research works have explored several alternatives oriented towards improving the software with a kinetic dimension: dynamic visualisations, physical or tangible interfaces. None of these innovative alternatives was completely orchestrated in a solution in the field of engineering simulation. In particular, for system dynamics methodology, earlier research projects have used static conventional GUI, dynamic visualisations of results or physical interfaces. Next, the authors present *Flux*: a new visual and interactive design for system dynamics modelling and simulation.

IMPLEMENTATION

Visual Design

The core visual element of the GUI in the conventional software is a static diagram: stocks and flow diagram [1][3] (Figure 1). As in many other mathematical symbols, this static diagram presented in software is just an inherited version of the pen and paper original praxis. Therefore, the central visual concept for *Flux* design was to present to the user a version of a model with an animated dimension, a dimension that will improve perceivability and predictability.

The design was created through metaphors in order to offer users a more concrete experience. Initially, it was based on the elemental water metaphor and later it was adapted towards a micro-organismic approach. *Warm signal* [16], provided inspiration for developing the visual experience to an enriched dynamic visualisation. Similarly, considering Bertalanffy's general systems theory sprung from biology, embracing a micro-organismic metaphor for the final design results intuitively and correspondingly accurate [17][18]. Movement and colour were used to better describe the behaviour of feedback loops. Positive cycles are composed through lime-yellow tessellated expanding nuclei. Their behaviours reflect the initial decision of communicating an incremental behaviour by amplifying their nuclei dimension. Negative cycles are composed through an organelle-like structure that collapses the sphere-like nuclei within. This collapse behaviour is used, again, to keep the communication's initial idea of negative cycle's imploding (Figure 2).

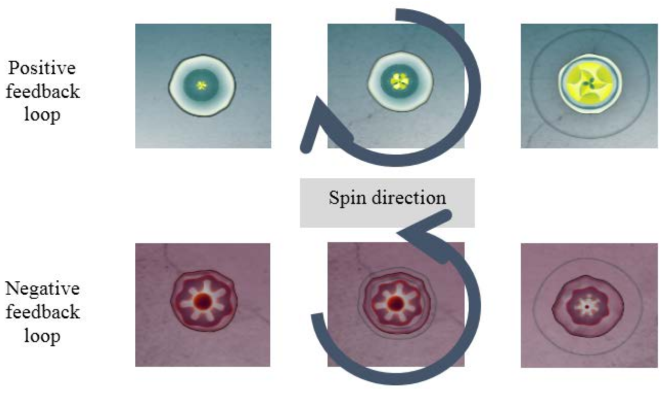

Figure 2: Sequential screenshots of positive and negative feedback loops.

Stocks were represented with rectangles like in the traditional stocks and flows diagram. Contours of all objects are slightly and irregularly distorted and sharp corners rounded to represent more organic-like shapes (Figure 2). In order for stocks to communicate receptiveness to its environment, the notion of an inner voided nucleus is used through a gradient from grey edges to a translucent centre (Figure 3). This reinforces the idea of stocks as neutral objects affected by their environment. The background was created through semi-translucent overlays resembling an aqueous medium.

Figure 3 presents several states of the dynamic diagram built with *Flux*. The user could see something that looks like a living microorganism by means of the continuous animation of the different elements. All of these are the animated dimensions of this new simulation tool's visual component. This is how *Flux* implemented an enhanced and intuitive visual experience, and the main reason behind the micro-organismic metaphor: a pragmatic way of bridging users' mind-sets onto a unified perceivable interface.

Interactive Design

Flux is controlled through tapping interactions (see Figure 4). By one-tap selecting the stock, people can either decide to name the stock or add a numeric (natural) value. If the person taps anywhere in the top half of the user interface - outside the stock - *Flux* automatically adds a positive cycle, and if the person taps anywhere in the remaining bottom half the application adds a negative cycle. There is a scenario where there can be up to two stocks, a case in which the application adds a mediating cycle by default between both stocks. Any of the cycles could be modified in its corresponding pane by selecting the cycle with a simple tap. The name and value attributes of each component are modified through standard tablet use, and the user could choose between linear or exponential behaviour. Finally, to close the pane, the person simply taps outside it, and to erase any of the objects the person taps the corresponding minus icon beside it. It is a straightforward use.

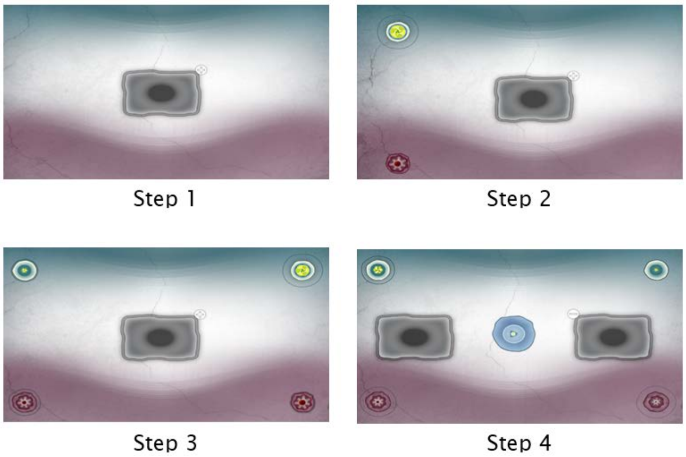

Figure 3: Four sequential screenshots of the *Flux* interface.

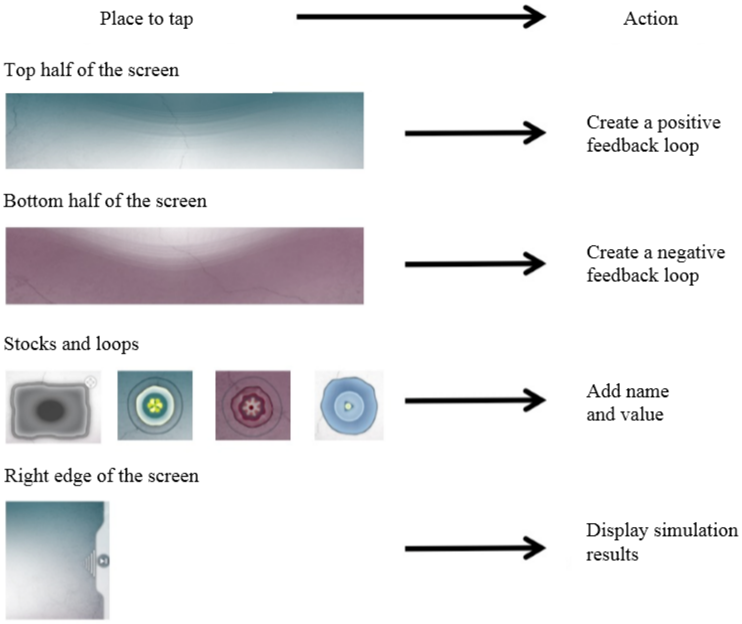

Figure 4: Touch interactions in *Flux*.

Software Prototype

The authors developed *Flux*, an iOS app that implements the proposed visual and interactive design. This software prototype supports system dynamic models with one or two stock variables and a maximum of five feedback loops. *Flux* is a native iOS app developed with Xcode and Objective-C and its visual layout was designed for iPad (versions 1 to 3 or iPad mini).

LEARNABILITY TEST

Nineteen students of Informatics Engineering participated in the learnability test. Their ages ranged from 18 to 23 years. All of them have basic knowledge of system dynamics; they took an introductory 10-hour course on this subject. Participants were assigned to one of two groups: one group (10 participants) used *Flux* simulation tool, the other group (9 students) used *Vensim* version 6.3, a classical software package for system dynamics modelling regularly used in a university context.

Each participant had 28 minutes to read the case and to build a causal loop diagram with pen and paper. Later he/she modelled a case with the software, *Vensim* or *Flux* in each case, for 35 minutes. The case represented a typical modelling challenge that a beginner learner of system dynamics could deal with. Also, it sampled a representative set of the features of the tested design. It was based on a commonly used instructional material designed at MIT [19]. The situation to be modelled was about the dynamics of a nursery tree.

The authors compared the learnability of their designed solution versus the one of a conventional system dynamics tool by means of a summative evaluation of the initial learnability for novice users [20]. They applied two metrics for measuring and comparing learnability [20-22]: ability to complete a basic modelling task and influence of the software tool on user understanding of key modelling concepts. They applied a post-test survey of learnability-related questions. A five-point Likert scale was used.

Users of the proposed design (*Flux*) achieved better task completion results than participants who used the traditional software tool (*Vensim*). Only one student could complete the task optimally, a participant of the *Flux*'s group. Table 1 shows that only 44% of the students with *Vensim* arrived to the final stage of the modelling task (results visualisation). In the group with *Flux*, all users (100%) arrived at this final point of the experiment. Creation of a stock variable was the only one of the four subtasks achieved by all the participants using both software tools. The percentage of users who could develop the last three stages of the task was noticeably superior in the proposed design than in the traditional one.

Table 1: Task completion with traditional and proposed designs.

Stages of the modelling task	Traditional (*Vensim*)			Proposed (*Flux*)		
	Users who achieved this stage (%)	Mean grade	SD	Users who achieved this stage (%)	Mean grade	SD
Stage 1: create a stock variable	100%	4.8	0.67	100%	4.9	0.32
Stage 2: build a sub-model	78%	2.2	1.56	89%	3.6	1.51
Stage 3: introduce data and equations	78%	1.6	1.33	89%	2.9	1.29
Stage 4: simulation results visualisation	44%	1.6	1.88	100%	4.1	0.99

Each participant grades his or her own performance for each of the subtasks or stages of the experiment (Table 1). A zero grade means that he/she could not perform this subtask, and a grade of five corresponds to *easy*. The users of *Flux* graded their performance in all stages better than the users of Vensim did. The difference increases in the final stages. In the last stage, the users of the traditional software graded their achievement as 1.6, while the participants who use the proposed design graded it as 4.1.

Figure 5 shows for the traditional software (blue line) that as the user advanced from one modelling subtask to the next, his/her performance got worse. Even in the last two modelling stages, the *Vensim* users self-graded it as 1.6, indicating that for this group, the user experience became harder. On the contrary, the learning curve for the proposed design (yellow line) decreased less. This result supports the idea that *Flux* effectively improves the initial performance of novice users.

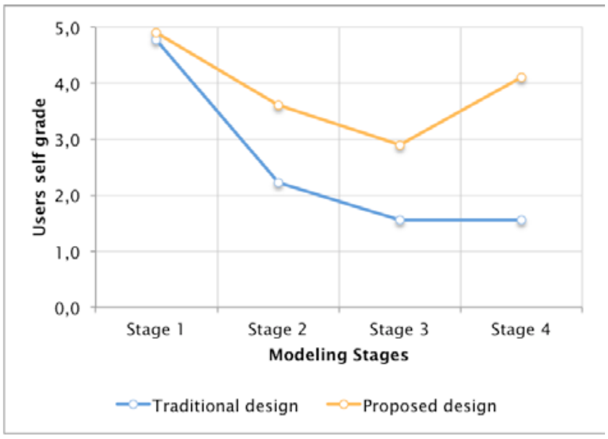

Figure 5: Comparison of user initial performance with both software tools.

The post-test questionnaire also measured the perception of the participants about their own understanding of two the key concepts in system dynamics modelling and simulation applied to the case presented in the experiment. The authors asked the users to establish if the software fostered their understanding, and they graded this from 1 (*no*) to 5 (*clearly*). Table 2 shows that the users of *Flux* perceived a better understanding of the dynamic nature and the fundamental structure of the model, than the participants who used the traditional software. This experimental evidence is consistent with one of the main orientation ideas behind the proposed design: software must offer a *kinetic* dimension of the model through an animated and more visually engaging interface. These findings confirmed that stocks and flows diagrams traditionally used in available system dynamics tools tend to obscure the loops that compound the model structure. Therefore, the design of the software, its user interface and interaction design, and the emergent user experience, are all of them the key factors to improve the learning possibilities for a novice user of a system dynamics simulation tool.

Table 2: Influence of software tool on the understanding of key modelling concepts.

Key modelling concepts	Traditional (*Vensim*)		Proposed (*Flux*)	
	Mean grade	SD	Mean grade	SD
Understanding the dynamic nature of the model	3.0	0.87	3.4	1.07
Understanding the fundamental relationship between stocks and loops	3.0	1.41	3.7	1.06

CONCLUSIONS

In this article, the authors showed that available software tools in system dynamics create barriers to learning for novice users. They have to confront a gap between the abstract model and its symbolic representation in the software. The learnability test shows how hard this challenge is for an engineering student. To overcome this limitation, the authors developed a new visual and interactive design and a prototype software, *Flux*, based on principles of user experience and interaction design.

In summary, the principles of the proposed design were: a simpler user-interface with fewer visual elements than traditional ones; a new diagram for models with a kinetic dimension implemented through a set of animated organismic elements; a small group of touch interactions for basic operations; and a direct form of interaction to move between model building and model simulation. The results of the learnability test with engineering students confirmed the proposed design increased the initial performance of novice users with this new simulation tool. Students even improved their understanding of key concepts of system dynamics when they used *Flux*. This proof of concept will need more experimental studies and software development resources to achieve a final product. However, in general, this research also revealed that there is a broad range of opportunities to apply user-centred design and multi-touch user interfaces to improve the user experience of simulation tools in management, engineering and sciences.

ACKNOWLEDGMENTS

This research was fully supported by the Engineering Faculty and the Research Direction of Universidad de La Sabana, Colombia.

REFERENCES

1. Forrester, J., *Industrial Dynamics*. Cambridge: MIT Press (1961).
2. Richmond, B., Systems thinking: critical thinking skills for the 1990s and beyond. *System Dynamics Review*, 9, **2**, 113-133 (1993).
3. Sterman, J., *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Boston: Irwin/McGraw-Hill (2000).
4. Richmond, B., Peterson, S., Vescuso, P. and Maville, N., *An Academic User's Guide to Stella Software*. New York: High Performance Systems (1987).
5. Eberlein, R. and Peterson, D., Understanding models with Vensim™. *J. Oper. Research*., 59, **1**, 216-219 (1992).
6. Byrknes, A., *Run-Time User's Guide and Reference Manual*. Bergen: Powersim (1996).
7. Meijer, K., Altamirano, M. and Goossens, H., System Dynamics with the Touch Table? An Exploration of Desirability and Desired Functionalities of a System Analysis Touch Table Application (2012), 10 January 2016, http://dtvirt35.deltares.nl/products/30135
8. Liang, H. and Sedig, K., Can interactive visualization tools engage and support pre-university students in exploring non-trivial mathematical concepts?. *J. Computer Educ.*, 54, **4**, 972-991 (2010).
9. Abrahamson, D. and Lindgren, R., *Embodiment and Embodied Design*. In: Sawyer, K., The Cambridge Handbook of the Learning Sciences. (2nd Ed), Cambridge, UK: Cambridge University Press, 358-376 (2014).
10. Isbister, K., Karlesky, M., Frye, J. and Rao, R., Scoop!: a movement-based math game designed to reduce math anxiety. *Chi'12 Human Factors in Computing Systems*. Austin, USA (2012).
11. Victor, B., Kill Math (2011), 17 September 2015, http://worrydream.com/KillMath/
12. Howie, E., Sy, S., Ford, L. and Vicente, J., Human-computer interface design can reduce misperceptions of feedback. *System Dynamics Review*, 16, **3**, 151 (2000).
13. Qudrat-Ullah, H., Perceptions of the effectiveness of system dynamics-based interactive learning environments: an empirical study. *J. Computer Educ.*, 55, **3**, 1277-1286 (2010).
14. Sotaquirá-Gutiérrez, R., A new dynamic visualization technique for system dynamics simulations. *Dyna*, 81, **188**, 229-236 (2014).

15. Zuckerman, O. and Resnick, M., System blocks: a physical interface for system dynamics learning. *Proc. 21st Inter. System Dynamics Conf.*, New York, USA, 810-811 (2003).
16. Sieler, S., Warm Signal (2011), 17 September 2015, http://vimeo.com/13395858
17. Bertalanffy, L., *General Systems Theory: Foundations, Development, Applications*. New York: Braziller (1968).
18. Andrade, H., Dyner, I., Espinosa, A., López, H. and Sotaquirá-Gutiérrez, R., *Pensamiento Sistémico: Diversidad en Búsqueda de Unidad*. Bucaramanga: Universidad Industrial de Santander (2001).
19. Forrester, J., Road Maps: a Guide to Learning System Dynamics (2005), 17 September 2015, http://www.clexchange.org/curriculum/roadmaps/
20. Grossman, T., Fitzmaurice, G. and Attar, R., A survey of software learnability. *Proc. 27th Inter. Conf. on Human Factors in Computing Systems*, Boston, USA (2009).
21. Paymans, T., Lindenberg, J. and Neerincx, M., Usability trade-offs for adaptive user interfaces. *Proc. 9th Inter. Conf. on Intelligent User Interface*, Funchal, Portugal (2004).
22. Linja-aho, M., Evaluating and Improving the Learnability of a Building Modeling System. Helsinki: Helsinki University of Technology (2005).

BIOGRAPHIES

Ricardo Sotaquirá-Gutiérrez graduated with a Doctorate in Applied Sciences from Universidad de los Andes (Mérida, Venezuela). He received a Master's degree in Informatics and Systems Engineering from Universidad Industrial de Santander (Bucaramanga, Colombia). He has been Head of the Informatics Engineering Department since 2010, and Professor-Researcher in the Human-Centred Design Group of Universidad de La Sabana, Chía, Colombia. He has been the principal investigator on research projects in computer simulation, systems thinking, human-computer interaction and user-experience design.

Francisco Ramírez-Rosales is a complexity interpreter and simplicity advocate as a designer, researcher and creative technologist. As an explorer of emergent technologies, he seeks to figure out how we should approach these, and how they can potentially amplify our abilities and multiply our joy. He has featured his work at events, such as NYC Media Lab's *Future Interfaces*, NYU Tisch's *GSO Interdisciplinary Grant* and Microsoft's *Design Expo*. Currently, Francisco is a graduate student-researcher at NYU's Interactive Telecommunications Programme seeking to understand how conversational user interfaces and machine learning can express cultural diversity and foster ethical behaviour.

Juan Pablo Garzón is a mobile developer for iOS with many years of experience. He has a strong background and experience in Objective-C, Swift, Java, C#, Cocos2D, Unity3D, processing and object oriented programming. He is also a student of informatics engineering at the Universidad de La Sabana and a current entrepreneur. Juan likes to play sports and helps other people with everything he can.

Jenny Robayo has a Bachelor's degree in Informatics Engineering from the University of La Sabana received in 2011. She has been teaching programming subjects in the Department of Engineering and has been working as a developer-leader at the Laboratory of User Experience (LUX).