EvoSpaces:
Multi-dimensional Navigation Spaces for Software Evolution

Michele Lanza
REVEAL@Faculty of Informatics
Univ. of Lugano, Switzerland

Harald Gall
s.e.a.l.@Dept. of Informatics
Univ. of Zurich, Switzerland

Philippe Dugerdil
HEG
UAS Geneva, Switzerland

Abstract

EvoSpaces is a Swiss-wide research project sponsored by the Hasler foundation. It involves three partners: University of Zurich, University of Lugano and the University of Applied Sciences in Geneva. The overall goal of the project is to explore novel ways to visualize and navigate evolving software systems in a 3D environment. In this paper we briefly describe the particularities of the project and the results obtained so far.

1 Project Summary

The EvoSpaces project (lasting from Jan 2006 to Dec 2009) aims to exploit multi-dimensional navigation spaces to visualize evolving software systems to ease their analysis and maintenance. Specifically we address the following questions:

1. How can we visualize evolving software systems in 3D by finding the right visual metaphor and by preserving the usefulness (from a reverse engineering point of view) of the obtained views?

2. How can we interact with and navigate such multi-dimensional spaces without incurring into navigational complexity (i.e., getting "lost") or friction (i.e., being "too slow")?

3. How can the discovered visual means contribute to the already existing body of reverse engineering methodology to understand and evolve industrial size software systems?

One of the outcomes of the project is a prototype implemented under Eclipse, in which large software systems can be visualized and navigated. This virtual world has been named "Software City" because we use the city metaphor to represent the virtual software entities: classes or files are represented as buildings, packages or directories form districts; functionality such as methods or functions are workers inside the buildings. The size and dimension of the buildings are set according to the value of user-selected metrics so that the city gets a different shape depending on the selected metric. Finally, the relationships between the classes or files are represented as solid pipes with an animation to show the flow of dynamic information.

In this virtual world, not only the static relationships, but also the dynamic behavior can be visually represented thanks to a technique to display the execution trace.

2 Project Details

Funding. The funding is provided by the Hasler foundation, that selects *"research projects in the area of information and communications technology, and by so doing contributes to the development of Switzerland as an intellectual and economic area."*[1]

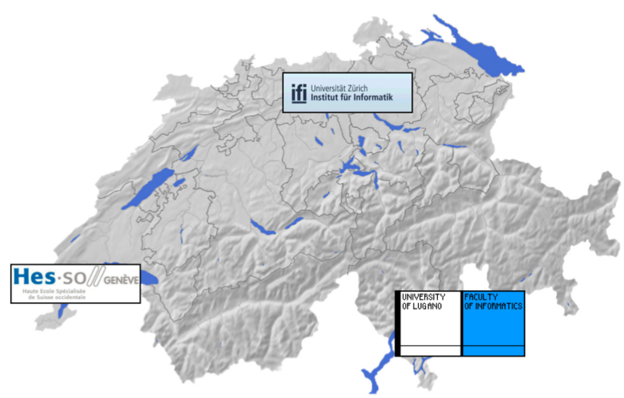

Figure 1. The EvoSpaces project spans three linguistic regions of Switzerland.

[1] `http://www.haslerstiftung.ch`

3 Participants

The Evospaces project involves three institutions located in three different regions of Switzerland (see Figure 1). The project is coordinated by the University of applied sciences in Geneva. The main research activities are performed at the Universities of Zurich and Lugano. In the following table we have listed the people participating in the project.

Name	Location	Role
Philippe Dugerdil	HEG Geneva	Project coordinator
Harald Gall	U. of Zurich	Research leader
Michele Lanza	U. of Lugano	Research leader
Sazzadul Alam	HEG Geneva	Software developer
Sandro Boccuzzo	U. of Zurich	PhD student
Richard Wettel	U. of Lugano	PhD student

Table 1. The EvoSpaces project participants.

The project hinges on the complementary research domains of the two research groups in Zurich (software evolution analysis, repository mining) and Lugano (visualization), whose outcomes are condensed in a pragmatic fashion by the university of applied sciences in Geneva, exemplified in a professional tool being developed.

4 Mission Statement

In the current information technology society we are relying more and more on software systems. Maintenance has been identified to be the primary factor of the total cost of large software systems (more than 90% of the total cost). One of the key difficulties encountered in software maintenance is given by the intrinsic complexity of such systems, and indeed, software complexity is recognized as one of the major challenges to the development and maintenance of industrial size software projects.

In that respect, the key aspect of software is that it is a virtual product. In other words, it is difficult to grasp the full complexity of a system that one cannot see or touch. However, many attempts to visually represent the structure of software systems have been proposed, essentially through flat representations. However, the quantity of information that can be represented in 2D drawings is limited. Handling the complexity and observing the evolution of very large software systems needs the analysis of large complex data models and the creation of condensed views of the system. In the context of visualization, software metrics have been used to compute and enrich such condensed views. However, current techniques concentrate on visualizing data of one particular software release, providing insufficient support for visualizing data of several releases. *The goal of this project is to exploit multi-dimensional navigation spaces to visualize evolving software systems.*

We tackle this challenge by exploiting a unique feature of the human brain, the representation of objects in 3D space. For example, this feature is used efficiently in the domain of civil architecture where CAD/CAM software lets architects navigate virtual buildings. Using the architectural metaphor we want to investigate the multiplicity of views given by construction plans and apply these techniques for software visualization in the form of software blueprints.

The central theme of our project is the development of a prototype implementing such visualizations and its application on large-scale systems both from the open source and proprietary domains.

The project consists of the following tracks:

- **Visualization models** - Investigate models to visualize software systems using metaphors stemming from civil architecture in order to obtain multi-dimensional spaces.
- **Interaction and navigation models** - Explore ways to interact with and navigate the multi-dimensional spaces.
- **Software models** - Prepare the "raw data" being large-scale software systems and produce various models of them from the point of view of the structure, architecture, design, etc.
- **Prototyping and Integration** - An overall parallel track involving the development of a prototype system to integrate the previously mentioned models.

5 Achievements

The achievements of the project can be divided in two categories: the tools that we have developed and the scientific works that we have published in the past few years.

CodeCity. CodeCity [12, 11], developed at the University of Lugano by Richard Wettel, is an integrated environment for software analysis, in which software systems are visualized as interactive, navigable 3D cities [10]. The classes are represented as buildings in the city, while the packages are depicted as the districts in which the buildings reside. The visible properties of the city artifacts depict a set of chosen software metrics, as in the polymetric views[7] of CodeCrawler [6].

In Figure 2 we see a visualization of the system ArgoUML produced by CodeCity. With CodeCity we have developed and explored a number of approaches to comprehend software systems [9], to analyze and understand their evolution [13], and to display design problems [14, 8].

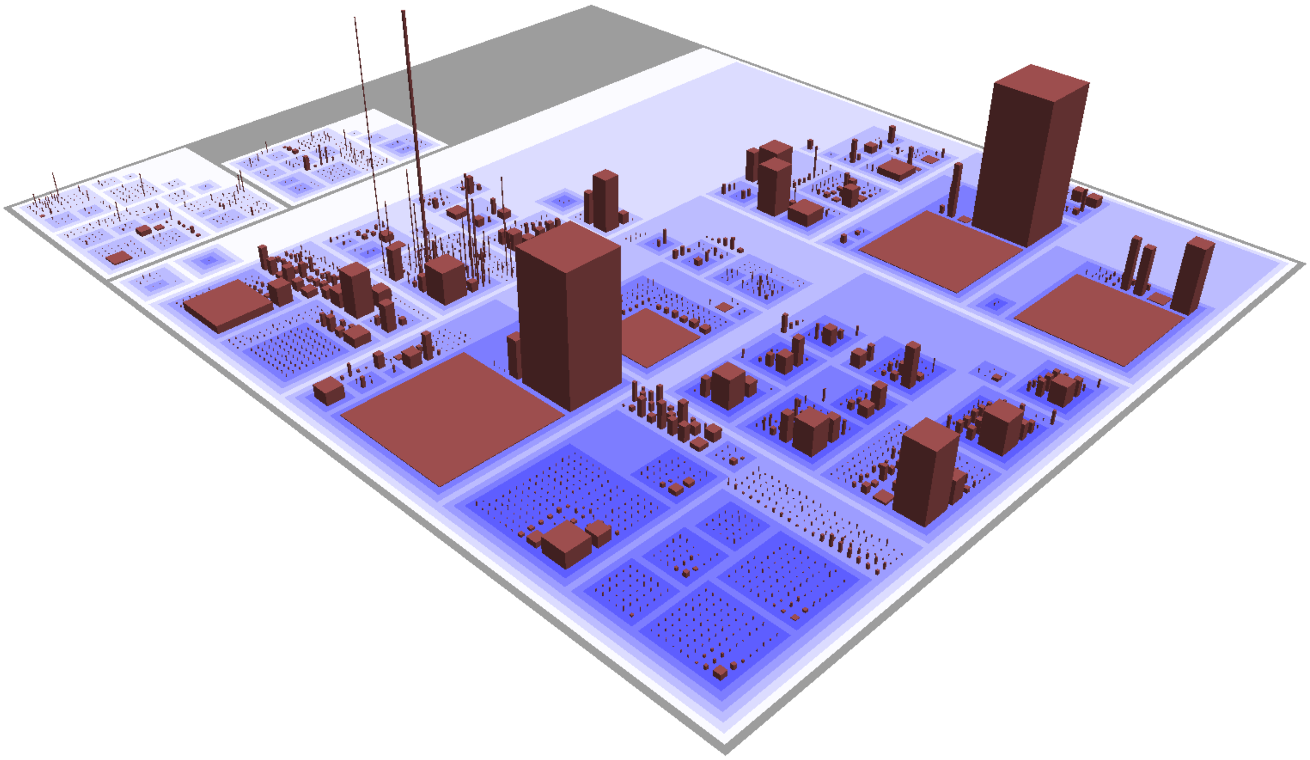

Figure 2. CodeCity displaying ArgoUML

Apart from having a remarkable impact in terms of scientific publications, CodeCity has been downloaded more than 1400 times since its release in March 2008 as free and open software. For more information we refer the interested reader to `http://www.inf.unisi.ch/phd/wettel/codecity.html`. CodeCity has also been partially ported to Eclipse in the form of a plugin [1].

CocoViz. CocoViz [2] (see Figure 3), developed at the University of Zurich by Sandro Boccuzzo, addresses software comprehension by a combination of visualization and audio. It uses common place metaphors (like houses) for an intuitive understanding of software structures and evolution. For each source code entity, evolution and structural aspects are mapped to metaphors to represent concepts such as design erosion, code smells or evolution metrics.

Cocoviz was later extended to also include audio as a supporting means to comprehend software systems [3]. For more information we refer to `http://seal.ifi.uzh.ch/cocoviz/`.

The EvoSpaces Tool. The conceptual insights stemming from both CodeCity and CocoViz flow into a unifying EvoSpaces tool [4], that is being developed at the HEG Geneva by a dedicated software developer.

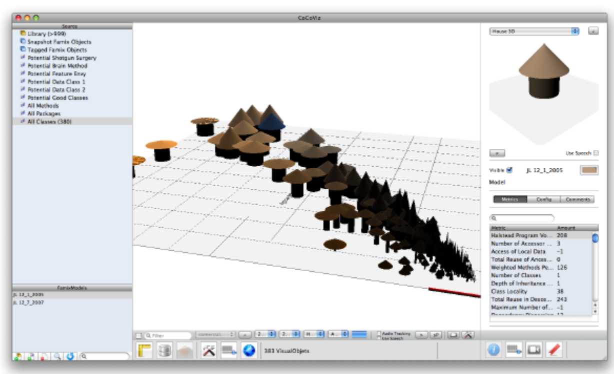

Figure 3. CocoViz at work.

The Evospaces tool is a distillate of the concepts that proved to be successful either in CodeCity or in CocoViz. The fact that it is being implemented by a dedicated developer also implies that it can go beyond the prototypical stage. In Figure 4 we see it as it visualizes the Mozilla system. The EvoSpaces tool also implements some additional features, such as the visualization of execution traces [5].

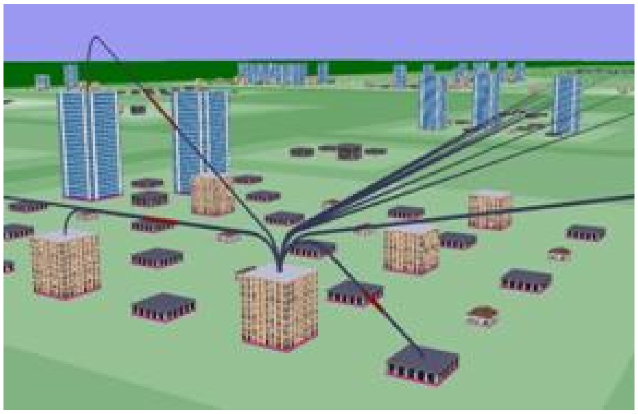

Figure 4. The EvoSpaces tool at work: Visualizing Mozilla.

For more information we refer to `http://www.inf.unisi.ch/projects/evospaces/tool/`.

6 Relevance

The relevance of the EvoSpaces project for the software maintenance community is mutli-faceted:

It is a living proof that multiple research institutions can indeed pursue close and complementary research directions not only without stepping into each others' fields, but actually positively influencing each other.

Moreover, the number and quality of the scientific publications illustrates the relevance of the research domain.

Lastly, a true first timer for the two university partners was the involvement of an actual developer. Most tools built by academic researchers are not mature and actually often do not need to be mature, but serve more as a proof-of-concept. In the specific case of this project we argue that the development of a professional tool led to a number of insights that could not have been obtained otherwise, but which go beyond the scope of this paper.

Acknowledgements

We gratefully acknowledge the financial support of the Hasler Foundation for the project "EvoSpaces" (MMI Project No. 1976).

References

[1] A. Biaggi. Citylyzer - a 3d visualization plug-in for eclipse. Bachelor's thesis, University of Lugano, June 2008.

[2] S. Boccuzzo and H. Gall. CocoViz: Towards cognitive software visualizations. In *Proceedings of VISSOFT 2007 (4th IEEE International Workshop on Visualizing Software For Understanding and Analysis)*, pages 72 – 79. IEEE CS Press, 2007.

[3] S. Boccuzzo and H. Gall. Software visualization with audio supported cognitive glyphs. In *Proceedings of ICSM 2008 (24th IEEE International Conference on Software Maintenance)*. IEEE CS Press, 2008.

[4] P. Dugerdil and S. Alam. Evospaces: 3d visualization of software architecture. In *Proceedings of SEKE 2007 (19th IEEE International Conference on Software Engineering and Knowledge Engineering)*, pages 500 – 505. IEEE CS Press, 2007.

[5] P. Dugerdil and S. Alam. Execution trace visualization in a 3d space. In *Proceedings of ITNG 2008 (5th International Conference on Information Technology: New Generations)*, pages 38 – 43. IEEE CS Press, 2008.

[6] M. Lanza. Codecrawler — lessons learned in building a software visualization tool. In *Proceedings of CSMR 2003 (7th European Conference on Software Maintenance and Reengineering)*, pages 409–418. IEEE CS Press, 2003.

[7] M. Lanza and S. Ducasse. Polymetric views — a lightweight visual approach to reverse engineering. *IEEE Transactions on Software Engineering (TSE)*, 29(9):782–795, Sept. 2003.

[8] M. Lanza and R. Marinescu. *Object-Oriented Metrics in Practice*. Springer-Verlag, 2006.

[9] R. Wettel and M. Lanza. Program comprehension through software habitability. In *Proceedings of ICPC 2007 (15th International Conference on Program Comprehension)*, pages 231–240. IEEE CS Press, 2007.

[10] R. Wettel and M. Lanza. Visualizing software systems as cities. In *Proceedings of VISSOFT 2007 (4th IEEE International Workshop on Visualizing Software For Understanding and Analysis)*, pages 92–99. IEEE CS Press, 2007.

[11] R. Wettel and M. Lanza. CodeCity. In *Proceedings of WASDeTT 2008 (1st International Workshop on Advanced Software Development Tools and Techniques)*, 2008.

[12] R. Wettel and M. Lanza. Codecity: 3d visualization of large-scale software. In *ICSE Companion '08: Companion of the 30th International Conference on Software Engineering*, pages 921–922. ACM, 2008.

[13] R. Wettel and M. Lanza. Visual exploration of large-scale system evolution. In *Proceedings of WCRE 2008 (15th Working Conference on Reverse Engineering)*, pages 219–228. IEEE CS Press, 2008.

[14] R. Wettel and M. Lanza. Visually localizing design problems with disharmony maps. In *Proceedings of Softvis 2008 (4th International ACM Symposium on Software Visualization)*, pages 155–164. ACM Press, 2008.