



# 1st Workshop on Exploring the Fitness and Evolvability of Personal Learning Environments (EFEPLE'11)

2nd STELLAR Alpine Rendez-Vous (ARV), the French Alps, 30-31 March, 2011

## Co-chairs

Effie L-C. Law<sup>1</sup>, Felix Mödritscher<sup>2</sup>, Martin Wolpers<sup>3</sup>, Denis Gillet<sup>4</sup>

<sup>1</sup>University of Leicester, UK

<sup>2</sup>Vienna University of Economics and Business, Austria

<sup>3</sup>Fraunhofer FIT, Germany

<sup>4</sup>EPFL, Switzerland

## Participants (in alphabetic order):

*Sandy El Helou*, EPFL, Switzerland

*Carlo Giovannella*, University of Rome Tor Vergata, Italy

*Martin Memmel*, DFKI, Germany

*Maryam Najafian-Razavi*, EPFL, Switzerland

*Christopher Nehaniv*, University of Hertfordshire, UK (Keynote speaker)

*Christian Prause*, Fraunhofer FIT, Germany

*Jose L. Santos*, Katholieke Universiteit Leuven, Belgium

*Benham Taraghi*, TU Graz, Austria

*Fridolin Wild*, Open University, UK

*Special thanks to H. L. Cornish, the Graphic Designer of Open University UK, for the aesthetically pleasing cover of the proceedings*

## 1. INTRODUCTION

### 1.1 Motivation

In the recent decade a plethora of interactive software tools, be they open source or proprietary, have emerged and perished in the realm of technology-enhanced learning (TEL). Concomitantly, there have also been surge and demise of contents, social networks, and activities associated with the use of these TEL tools. It is intriguing to understand what factors contribute to their rises and falls, and how. While controversies on the viability of making an analogy between the evolution of natural and artificial objects prevail, it is deemed worthwhile to explore its potential for analysing the changes in TEL and charting the future.

In accordance with evolutionary theory, the fitness of an environment or tool can be defined with respect to its purpose and depends on the ‘genes’ from former generations. In context of TEL, these genes can be understood as features of existing tools and functionality being reused from software libraries or developed over multiple lifecycles thus leading to new generations of software artefacts. Personal learning environments (PLEs) aggregate these functionalities to enable learners to connect to peers and shared artefacts along their learning activities. Consequently, the success of a PLE can be measured by its uptake and usage within different communities of practice, its perceived effectiveness and efficiency in supporting the attainment of learning goals, its application beyond pre-defined purposes, its distribution and outreach beyond single communities, and its evolution to new PLE generations through active developers. Moreover, data mining of so-called *variables of evolvability* (e.g., perceived pragmatic/learning and hedonic/fun value) will enable the derivation of specific guidelines for designing and developing PLEs. Such empirically grounded guidelines, supplementary to those for generic IT applications, are currently lacking and much desired.

Overall, the main aim of the workshop is to explore the fitness and evolvability of PLEs in order to identify and understand characteristics and mechanisms for successfully evolving PLEs.

### 1.2 Related Work

In principle, for a software system to be sustainable, it needs to be able to adapt to the changing requirements [1] in terms of use contexts, user goals, organizational cultures and technological opportunities. Specifically, in the field of TEL, there has been a shift from the pioneer work on designing and implementing full-featured, organisation-driven learning management systems (LMSs) to the emerging trend of developing specialised tools, which then can be assembled by users to extend/create personal learning environments (PLEs, Attwell, 2007) [2]. Not least due to the Internet, users have access to a seemingly innumerable amount of content and software tools, which are useful and partially even necessary to achieve the learning goals driven by the demands of job tasks, higher, and further education, or even private activities.

In the context of PLEs, the selection of tools is at the discretion of individual users, their organisations and the communities of practice (CoP) where users engage in a variety of collaborative activities. It is observed that some software tools, after being used for a few typical tasks by a few people only, unexpectedly spread out within a CoP widely as well as wildly through good practice sharing, convincing peers of the benefits of these tools for particular lifelong learning activities. In a very short period of time such tools can become as must-have infrastructure for collaborative work (e.g. various Google services). These tools and the environments built on them are not only intensively used but are also modified and sustained by active developer communities. On the other hand, some tools are endangered to be rejected by end-users and to die out after a few successful cases of application, even though they have undergone several iterations of redesign. Apparently, these observations manifest the notions of *descent with modification, heritable variation and selection, sensitivity to changing environmental or contextual requirements, and “control of and types of variability”* (Nehaniv, 2003 [3]; Wernick et al. 2004 [4]) that characterize Darwinian evolution. In the context of PLEs, it is relevant to understand the processes leading to successful tool uses, create respective models and learn how to control respective processes to increase the efficiency and effectiveness of modern individual learning environments.

The assumption that changes in PLEs can be modelled by Darwinism underpins this proposed workshop, which aims to explore several pertinent issues:

- Nahaniv et al [5] (2006) define the notion of *evolvability* as “*the capacity to vary robustly and adaptively over time or generations in digital and natural systems*”. This definition leads to a basic question: **What is evolvable?** Is it a matter of the complexity of a system that is quantifiable such as lines of codes, number of modules? Or is it more a matter of quality-in-use manifests in terms of user experience [6] (i.e. a non-functional requirement)? Another key question: **Why does a system evolve?** It can be instigated by changes in a system’s environment, user requirements, usage, implementation methodologies and technologies. Answers to these *what* and *why* questions can shed some light onto the question **How to effectively and reliably evolve a system** (Ciraci & van den Broek, 2006; footnote 3)? Addressing these questions in the context of PLEs will instigate stimulating discussions.
- Fitness for survival is a widely known but poorly understood concept of Darwinian evolution. Paradoxically, the idea of heritable variation and selection is necessary but not sufficient to explain inherent phenotypic expression of fitness (Nehaniv et al. 2006; footnote 5). It hinges on the rigidity (or flexibility) of the genotype-phenotype mappings. The main difficulties lie in drawing analogies between biological concepts and artificial artifacts (e.g. What constitutes an “individual”, a “species”, or “interbreeding”). Insights can be gained from the

notion of *fit-for-purpose* in the field of HCI (e.g. Wong et al., 2005) [7] and the fitness model of nodes in the science of (social) networks (Barabasi, 2002) [8]. Nonetheless, it remains an open question on how to define and measure the fitness of PLE tools

## 2. WORKSHOP DESCRIPTION

There were 10 presentations, including a keynote speech. In addition, plenary discussions on specific topics were held. Section 2.1 reports the main ideas addressed by individual presentations. Section 2.2 highlights the ideas explored by the workshop participants.

### 2.1 Report on Presentations

In this section, we highlight the ideas discussed in each of the presentations and present them in the form of notes that may inspire further thoughts along the related inquiries. These notes can serve as pointers to the tenets of the respective workshop papers.

#### 2.1.1 Keynote by Prof. Chrystopher Nehaniv

- Core concepts addressed: individual, reproduction, population, robustness, variability, phenotypic plasticity, autopoiesis, self-replication and repair, and evolvability
- The notion 'replicating individual' is difficult to define in the realm of software evolution – Is it a behaviour, an artifact or software release?
- Self-replication is a key notion in evolution (cf. computer viruses, cancer cells, self-reproducing automata); replicators entail external support;
- Constraints of evolution: finite resources, heredity, variability, differing reproductive success, turn-over of generations;
- Increasing complexity through successive inheritable mutation; a measure of complexity in biological sciences can be number of cell types and in software can be level of embeddedness, lines of code, number of loops, etc. Adaptive changes in population over generations (genotype-phenotype map)
- Artificial selection vs. natural selection;
- Variability: neutral mutation (no harm, no benefit) is important: similar fitness in the same environment; mutation that is neutral in such an environment is beneficial as a resource;
- Neutral mutation such as user interfaces – a variety of choice for selection;
- Fitness landscape: inheritable fitness to flourish
- Open-ended evolution is unbounded increase of complexity over time;
- External fitness function imposed on agriculture (can we learn from this domain?); number of offspring and living long enough to reproduce (fitness measures);
- Symbiogenesis: dynamic user-synthesis of PLE from components; combinations from the lower level units;
- Evolvability for artefacts: capacity for producers to rise to adaptive variants for flexibly meeting

- changing requirements; lineage, different fitness between offspring and parents
- Properties of evolvable systems: robustness to genetic variability, phenotypic robustness, redundancy, conservation of core mechanisms/features; robustness to environment change (resilience), self-monitoring, compartmentalization (modularity), symbiogenesis
- Software evolution: re-use, modularity, information hiding, encapsulation, OO inheritance, coupling and cohesion;
- PLE: system as fielded (instance: individual)
- Persist over time, descent with modification
- Lines of code, modules can be considered as genes (re-usable)
- Variation: customization of generic software product via parameterization, copying and sharing
- Iteratively adapted by users to context and changing requirements;
- Immediate fitness is very different from capacity to support possible evolvability;
- Variational capacity (vary/be varied robustly and adaptively) is crucial to evolvability

#### 2.1.2 Discussion on the Keynote

Notion of energy/resources in the context of software;

- Areas of tension:
  - immediate fitness vs. variability
  - simplicity: usability vs. complexity
  - genotype (design: functionality) vs. phenotype (affordances: practices)
- Complexity: base is interaction, energy comes from interaction, non predictable
- Consciousness/Intentionality (or awareness): comes from interaction, collaboration
- Is evolvability kind of higher level creativity
- Success: performance improvement of learners; "form follows failures"
- Complexity: maximise contact with environment subject to being able to understand and manipulate: complexity needs to be close to contact
- Educational technology so far has failed: because there are no solutions of scale (past: LMS have been successful, but not 'real' learning support tools)
- Capacity for variability: Learning is development of potential for action: competence, but we can only assess performance
- Capacity relates to complexity through adaptation through exchange of modules and over time!
- Freedom of adaptation vs. ethical concerns experimenting with bad combinations of software
- Sharing of successful practices/arrangements/etc. is hereditary replicability
- Problem: It's not the PLEs surviving and being fit, it's the widgets
- Problem: PLE: Lifespan of generations is not controlled

- But: Behaviour vs. artefacts: patterns of practices vs. widgets
- Behaviour: duplication and divergence; behaviour patterns can be very far away from genetics; active copying vs. environment driven auto discovery
- Controlling of behaviour: we can (to a part) control the environment to recreate 'situations'
- Translation of behaviour (phenotype) into genotype? No convergence in other areas.
- Would be helpful to very clearly define concepts such as genotype, phenotype in the PLE context
- Groundbreaking works in e.g. evolutionary algorithms: e.g. von Neumann: theory about life; e.g. evolutionary algos: were designed as optimisation techniques (example: designing nozzles, aircraft wings)

### 2.1.3 Presentation by Benham Taraghi

- Success measurement:
  - Complexity: number of widgets in an environment
  - Change: rate of change: number of replacements, new widgets
  - Number of users
- Selection types: stabilising selection, disruptive selection, directed selection
- Selection strategies: r-strategy (short lifespan, unknown environments) vs. K-strategy (long lifespan, known environments)
- Mutation: slight variation of existing functionality or UI
- Recombination: combining code of different widgets to build new ones: code sex
- Tracking of use: frequency of activated widgets, frequency of interactions with widgets that can be tracked in the system
- TUG system: 1000 users, 30% active users
- Competition not between widgets, but between PLE system and competing websites
- Code complexity of the PLEs: PLE as a whole (of one user) or widgets? How did it change over time? Lines of code? Level of embeddedness? Modularisation? Interwidget communication? Service orientation?
- Affordances (= in a certain cultural context)?
- Other factors (besides fitness): usability, usefulness (e.g. indirect via level of the learners)?
- Need to look at overall PLE system, not only at single widget; still: number of contexts, number of functions, number of other widgets it has been used with (degree centrality, betweenness, prestige): indicator of complexity
- Symbiotic relations: themingWidget: cannot exist on its own
- Coevolution of development and users

### 2.1.4 Presentation by Carlo Giovanella

- Evolution: strong focus on learning analytics: e.g. activity graphs, emotions, social networks, emotion in social networks

- Use **traces of user activity** to observe evolution
- Arrival of facebook changed the use of the system
- New journal: Interaction Design & Architecture

### 2.1.5 Presentation by Felix Moedritscher

- Environment: socio-technical system: activities, purposes, patterns, interaction, features, functionality, implementation
- Evolvability: versioning, copying/reusing, interoperability
- Fitness: usefulness & usability, user feedback, technological compliance
- Distribution approximation
- Fitness depends on the usage context (e.g. publication impact)
- Impact of papers very strongly relates on experience of the researcher (years of experience in a field). What about production of widgets? Are widgets produced by more experienced users more successful?

### 2.1.6 Presentation by Martin Memmel

- Sustainability
- Interoperability: using and offering APIs, following standards
- Number of application scenarios: very many application scenarios for PLEs
- Low technical and low conceptual barriers to system use
- Resources are finite: people, time, infrastructure, money
- Repurposing and re-theming/branding of systems
- Solve a specific problem, but do it in a generic way
- Support tools for setup and deployment
- Refactor
- Fitness is plasticity with respect to user requirements

### 2.1.7 Presentation by Sandy El Helou

- Viability:
  - flexible representation of interaction and contents
  - adopt social media paradigms (encouraging participation)
  - elastic community and CMS services
  - automate/openness: recommender systems: open corpus environments
- Use of Graasp
- Flexible representation: not necessarily dependant on number of users

### 2.1.8 Presentation by Jose L. Santos

- CAM dashboard
- Activity – actions executed in widgets
- Capturing communication data from interwidget communication
- Specialisation to styles?
- Active use of the dashboard to change behaviour?
- Evolution: Awareness > Social Behaviour > ...

- How to support awareness between developer and user?
- Representation of context to make use of the activity monitoring
- Fitness: take care of environment
- Visual quality
- Trust relationship between developers and user

### 2.1.9 Presentation by Fridolin Wild

- Acceptance: expectancies, social influence, facilitating conditions etc.
- Longer term

### 2.1.10 Presentation by Christian Prause

- “Walking on water and developing software from a specification are easy if both are frozen.” (Edward V. Berard)
- high costs of change lead to extinction
- evolvability: internal quality
- software quality: ISO 9126: functionality, reliability, usability, efficiency, maintainability, portability
- developers learn software: documentation! Code!
- Fitness = external quality + quality in use = Tool in environment in its context
- Case-based tools

### 2.1.11 Presentation by Maryam Najafian-Razavi

- Barriers to adoption (of gleanr)
  - Lack of simplicity
  - Slow ROI: differed benefit
  - Need for training
  - Usability problems: memorability, error rate, portability
  - Success factors: clear value prop, awareness, ease of integration
  - Interesting: big and fluid sites show up earlier in google
  - Suggestions: anonymity, prepopulation, network effects
- Success factors: could be fitness factors
- Fitness leads to adoption
- Prepopulation: problematic and difficult
- Prepopulating vs. survival?
- Ecosystem: has to be created, needs a context

## 2.2 Report on Plenary Discussions

### 2.2.1 Contextual Issues

- Flexibilisation of technology support for any kind of educational process
- Culture of certification: assessment and accreditation;
- Fitness: Integration of environments: mobile, web, all
- Fitness of users: critical design skills, measure experience / styles
- Context: capture context of learners holistically, make this context description available to sound applications;
- Plasticity: Support change in pedagogical approaches

### 2.2.2 Teachers as Target Groups

- Find a way to prove to the teacher that relying on a specific technology will help them be more effective
- Tackle danger for teachers: environments disappear: but environments change with their needs
- How to sell technology to the teachers?
- Show that with the help of any technology, the learners in the classroom/course became 10% better: works only with criterion-referenced testing (no norm referenced testing): skills assessment: increase by 10%
- Emergence of new widgets coming from the teacher and learner community
- Living community: Increased sharing of best practices: 1 million teachers / million learner using a PLE; There are enough teachers in Europe
- Digital literacy of teachers is a problem
- Technology is seen as an amplifier
- Combine agents and human tutors to provide high quality tutoring to every child

### 2.2.3 Invisible PLE

- very low entry barrier
- Sharing a curriculum in 15 minutes
- No good idea: it is rather about reconfiguration, not sharing: more about the adoption than that it is fast
- Extremely complex issue
- Widgets: 1000 widgets: which one is better and how do we measure that? Through the community
- Testing: could include teacher has to be able to re-use a PLE in 15 minutes; but: it's not about time, it's about the return on investment
- Identifying the scores that someone gets based on the traces that someone leaves in the system
- Pedagogically sound user interfaces

### 2.2.4 Predictive Modelling

- Predictive models: Predicting performance based on traces
- Testing of predictive models in competitions: accuracy vs. satisfaction
- Learning analytics: graphical user interfaces that foster quick understanding of performance and aesthetic display, streaming feedback
- Learning analytics, traces, context capturing; Privacy-ensured, anonymised; Streaming analysis
- Open requirements elicitation: Implicit requirement modelling, helpdesk monitoring, Implementation competitions in the bartering platforms for software development

### 3. EMERGING RESEARCH QUESTIONS

- *Find a way to prove to the teacher that relying on a specific technology will help them be more effective*
  - The million practices & million teacher challenge: ad hoc formation of large scale learning networks: Reach a certain level of scale in variability and build capacity for variability of practices of technology use in learning and teaching.
  - This includes: sharing of context information such as attention meta data, interoperability, practice capturing and sharing facilities such as scripts or learning designs or activity streams
  - This is not about showing that a certain template is used by a million people, but that 1 million people have differing, adapted to their needs practices in technology support
  - Ad hoc formation of large scale learning networks
- *Fitness of learning environments is plasticity with respect to user requirements:*
  - Variation: Adaptation or mutation: construction set widget-based PLE, coding according to changing user requirements, mash-ups
  - Speed of change:
    - Evidence that a trajectory is followed that a system has been adapted: evidence of plasticity
    - Knowledge management for teachers
    - Dissolving of communities of practices: problem solved, community dissolved
- *Invisible PLE*
  - Low entry barriers
  - Flexibility with respect to pedagogical and andragogical approaches
  - fitness of widgets: create an open market for widgets; then we can use the market mechanisms; show that there are widgets from each of the European countries; differing learning contexts (school, university, Ill) and stakeholders (providers, learners, teachers, educational institutions)

### 4. CONCLUDING REMARKS

Evolutionary or Darwinist theories are inherently controversial; applying them to explain and predict the trajectory of the development of Personal Learning Environments (PLE) is particularly challenging. PLE is still at its infancy stage, and a consensual definition is still lacking. Amongst others, the task of defining fitness models for predicting the rise and demise of specific widgets (which are commonly seen as the building blocks of PLE) and a specific configuration of PLE per se is daunting. The workshop is seen as the first step moving in

the direction, though there are still many steps to be taken to achieve this seemingly insurmountable task. The initial step is seen as successful with intriguing ideas being conceived. Future work includes organizing a series of related workshops/seminars that involve participants with diverse backgrounds. Project proposals addressing the emergent topics are seen as a promising way to explore them in depth over a relatively long period of time. In the meantime several meetings amongst the workshop participants have been held to explore these possibilities.

### ACKNOWLEDGEMENTS

We are obliged to the two EU FP7 projects on technology-enhanced learning: ROLE (<http://www.role-project.eu/>) and STELLAR (<http://www.stellarnet.eu/>) for enabling the realisation of this stimulating workshop. We would also like to express our appreciation of the organisers of the 2<sup>nd</sup> Alpine Rendez-Vous (ARV) 2011 whose efforts have made the event enjoyable and successful. Last but not least, thanks should go to authors of the workshop papers.

### REFERENCES

- [1] Ciraci, S. and van den Broek, P. M. (2006) Evolvability as a Quality Attribute of Software Architectures. In: The International ERCIM Workshop on Software Evolution 2006, 6-7 Apr 2006, LIFL et l'INRIA, Universite des Sciences et Technologies de Lille, France, pp. 29–31.
- [2] Attwell, G. (2007). Personal learning environments: The future of eLearning, *eLearning Papers*, January 2007, 2(1), [www.elearningpapers.eu](http://www.elearningpapers.eu). ISSN 1887-1542
- [3] Nehaniv, C. (2003). Evolvability, *Biosystems: Journal of Biological and Information Processing Systems*, 69(2-3), 77-81.
- [4] Wernick, P., Hall, T., Nehaniv, C. (2006). Software evolutionary dynamics modeled as the activity of an actor-network. Proceedings of 2<sup>nd</sup> Intl. Workshop on Software Evolvability. IEEE computer society press.
- [5] Nehaniv, C., Hewitt, H., Christianson, B., & Wernick, P. (2006). What software evolution and biological evolution don't have in common. In Proc. Of 2<sup>nd</sup> Int'l IEEE Workshop on Software Evolvability (SE'06).
- [6] Law, E. L-C. & van Schaik, P. (2010). Modeling of user experience: An agenda for research and practice. *Interacting with Computers*, 22, 312-322.
- [7] Wong B. L. W, Keith S. & Springett M. (2005) Fit for Purpose Evaluation: The case of a public information kiosk for the socially disadvantaged, People and Computers XIX – Proceedings of HCI 2005, Edinburgh Sept. 5-9. Springer.
- [8] Barabasi, A-L. (2002). The Linked: The new science of networks. Cambridge, MA: Perseus.
- [9] Star S.L. & Griesemer, J.R. (1989). "Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39". *Social Studies of Science* 19 (4): 387–420. [doi:10.1177/030631289019003001](https://doi.org/10.1177/030631289019003001).