Biological Life Cycle:

A New Interpretation for the Evolution of ICT Applications

Davide Bolchini[2], Nicoletta Di Blas[1], Paolo Paolini[1], Caterina Poggi[1]

Politecnico di Milano, Department of Electronics and Information, Hypermedia Open Center

Indiana University, School of Informatics, Indianapolis, User Experience and Simulation Lab

ABSTRACT

We address Design-Based Research from the perspective of engineering for interactive applications. In particular, we investigate the lifecycle of Design-Based Research projects in the realm of advanced educational technologies in opposition to more "rational" development models, commonly used in software engineering. We codify a metaphor based on biological evolution to account for the characteristic complexity, unpredictability and growth of ICT lifecycles and to ground the different kind of rationality underlying the development of these applications. Like living organisms, they continuously adapt to their environment, by changing, evolving new features, sometimes retaining old ones, the rationale of which becomes opaque and difficult to explain (remember). We base our discussion on a 8 years experience with programs in multi-users virtual environments for education (and entertainment) that have involved so far more than 9,000 students from 20 countries (3 continents).

1. INTRODUCTION

For many man-made artifacts (e.g. buildings, trains, bridges, a piece of software controlling an elevator, etc.), we can detect a lifecycle pattern based on the following principles:

1. *Engineering Requirements and Design*. The activity of "design" follows requirements elicitation; the bulk of what is called "requirements" is for the greatest part related to engineering quality (e.g. performance, robustness, safety). Higher-order requirements about usefulness, purpose, etc., are not explicitly included in the process. Design engineering is in essence a rational process: engineering requirements means often to "refine" them into high-level design decisions, to be further refined into lower level design components.
2. *Engineering Implementation*. Some well-grounded theories or models allow computing, quite precisely, the relevant features in order to make the artifact corresponding to the expected quality in terms of "expected behavior", performance, reliability, or safeness. The quality of the design and implementation is intended as "predictable behavior" and performance can be often estimated (if not proved) at design time. The "implementation" (i.e. "building" the artifact) can be thus relatively straightforward, following a rational plan from the design specifications.
3. *Engineering Traceability and Iterative Correction*. The results need to be tested, to identify possible mistakes, but, unless mistakes were made either during requirement elicitation, design or implementation, the overall quality of the result is guaranteed. If problems emerge during the testing, they can be quite precisely traced back to certain requirements, design or implementation faults, which can be subsequently corrected. The same traceability-correction model can be applied if new needs emerge, iteratively restarting the lifecycle.

We can characterize the above pattern with terms such as a “rational lifecycle”, including “rational requirements elicitation”, “rational design and design process”, and “rational traceability”. If we look now at the field of ICT applications, these are the key characteristics underlying all “rational” development lifecycles in various ICT domains, spanning from Software Engineering (Ghezzi, Jazayeri, & Mandrioli, 2002) to Human-Computer Interaction (Sharp, Rogers, & Preece, 2007).

There are many situations, however, where rational lifecycles have almost always eluded engineers: large corporate information systems are one example; large web applications are another, and games for learning are one additional example that we examine in this paper.

Experience shows that in most of these situations a different, apparently “irrational” lifecycle is at work. Software engineers have often considered this lifecycle a sort of accident (mainly due to irrationality of the organization, or to lack of solid scientific background in developers, or to random circumstances), striving to bring it back to the “rational” paradigm. The thesis of this paper, however, is that *in many cases these “anomalous” lifecycles are not irrational at all, but correspond to a different kind of rationality*. We call this particular development approach **biological lifecycle**.

Let us consider, to introduce the specific nature of this new type of rationality, how a form of life develops. A new “prototype” of a species appears (either by chance or by a plan, according to the philosophical-religious beliefs of the reader); at this stage, it is very difficult to rationally predict whether it will survive and proliferate. The prototype lives “in the field” (not in a lab) and faces real-life situations. Environmental factors and hostile forms of life attack the prototype. In time, changes and new features appear (e.g. hair, lungs, blood vessels…) that may affect the prototype’s chances of survival as individual or as species. The prototype is again tested for quality: those with good, reliable new features survive, the others disappear. The principles governing this biological lifecycle are clear:

1. *Trial and Error.* Mistakes are done often: they just disappear, since they do not survive. In fact, it is very difficult to predict which features will be effective in order to cope with the environment and the enemies: the only approach seems to be trial-and-error.
2. *"Incomputable" Design Solutions.* For the same "goal" (e.g. getting oxygen inside the body) different strategies, designs, and implementations are carried out. Living creatures are chaotic and apparently irrationally built; still, almost everything of them can be accounted for through some reasonable motivation. Consider the nose, for example: we can explain the role of each of its features, but it is hard to argue that, given the goal of inhaling, this was the only or the best possible "design solution". At the extreme, living creatures can develop features even dangerous for themselves to satisfy fundamental requirements (say reproduction): e.g. the tail of male peacocks, impressive for females, makes them easier to be caught by predators.
3. *Fragile Redundancy*. Living creatures are at the same time fragile (they easily break down) and redundant (we can cut a lot out of them and they still survive).
4. *Irrational Feature Persistence*. Living creatures retain obsolete features (arising from requirements of previous ages) that are useless and/or sometimes harmful. In other words, features that had been developed for coping with environmental factors or enemies in a certain time, remain for a long while after those factors or enemies have disappeared.
5. *Irreducible Wholeness*. Survival is ensured by the entire organism as a whole, not by a single part; and traceability is difficult. If something is wrong with the organism (e.g. it gets sick), we can conjecture through induction or "statistically" determine which part is responsible. However, we cannot go back to a rational design to pinpoint exactly where the source of the problem is, and how to fix it (as we normally do with rational artifacts).

Based on these principles, living creatures are chaotic, apparently irrational, but effective and successful: they can survive and sometimes thrive in their environment. Complex ICT artifacts (say a large Information System or a complex interactive installation) sometimes are better conceived, approached, developed and understood by means of a biological lifecycle (rather than a rational one), since they have strong similarities with living creatures, rather than with technological artifacts: they are chaotic, apparently difficult to explain rationally, but effective and successful (if they survive).

One of the consequences is that for each of their components (take the analogy with the nose of human beings), we can find a motivation, explaining the purpose and he way it works, but we can't explain the design strictly rationally or algorithmically (why it has been shaped and designed exactly in that way).

There is an additional analogy: even when a living creature feels well, there could still be something wrong (i.e. some components may not be working properly), but this dysfunctional element is compensated by other parts or other features. We can observe an analogous behavior in some "biological" ICT artifacts.

Since 1999, the HOC laboratory of Politecnico di Milano has been developing educational experiences based on shared online 3D environments, involving over 9,000 teenage users and their educators from 20 countries and 3 continents, on topics ranging from history to sports. These programs constitute a complex, multifaceted example of Design-Based Research, since their evolution, discussed in the paper, highly resemble that of a living organism retaining old features, developing new ones etc. The "biologic" metaphor we codify is an interpretation paradigm of the development lifecycle that emphasizes the need for a new understanding of traditional models of interactive application development. Our approach can thus help better understanding Design-Based Research both in the field of educational technology and in the field of software engineering.

2. COLLABORATIVE EDUTAINMENT APPLICATIONS AS "BIOLOGICAL CREATURES"

In this section we show an example of "biological" development making reference to a concrete set of case-studies: real-life collaborative edutainment applications developed by HOC of Politecnico di Milano. Edutainment is the blending of education and entertainment. Collaborative edutainment applications involve multiple users in engaging and enjoyable experiences that provide also a learning value. Therefore, the result of the development effort is not just an application (or a set of systems) but a complex **experience.** Let us relate now empirical evidence (Di Blas, Garzotto, & Poggi, 2009; Di Blas, Paolini, & Poggi, 2003; Di Blas, Poggi, & Reeves, 2006; Learning@Europe 2004-07: Final Report) of successful collaborative edutainment with the concepts of biological lifecycle.

2.1. Requirements

Requirements in collaborative edutainment applications belong to very different realms: technology, pedagogy, social behavior, computer-supported cooperative work, Human Computer Interaction, entertainment, and many others. Some factors are out of direct control: in collaborative edutainment environments there are several players whose social interactions are largely outside the designer's reach, and yet with a strong influence over the quality of the experience.

The requirements are (by their very nature) ill-defined and chaotic, and not amenable, in general, to a formalization and/or an algorithm, since they concern engagement, compelling experiences, expectation creation, attractiveness, pedagogical impact, and other widely heterogeneous factors. There is **only one real overall requirement**: the **experience**

as a whole (i.e. the living creature in the analogy) must be **effective**; the effectiveness of individual parts is, in a sense, less relevant.

2.2. Design

It is impossible to bring the design to a rational and/or formal and/or algorithmically provable correct process, because the overall design comes as **"emergent property"**, not planned, and unexpectedly resulting from a continuous process of ongoing adaptation and reaction to external and (sometimes) unforeseen factors. Induction, ingenuity, readiness to metabolize unexpected external stimuli, and intuition (possibly based on previous experiences) are the main tools guiding design. One small example: in the very first year of deployment, a bug in the software was discovered that allowed avatars in the virtual environment to look "through the eyes" of someone else's avatar. At first we were tempted to fix the bug, but then we thought it could be a fun feature and we introduced a new rule in a game (the "treasure hunt") that compelled players to use the view from their teammate's eyes to double check items before selecting them. Thus a totally unpredictable feature was turned into an asset.

2.3. Evaluation and Traceability:

What can be really evaluated is the quality of the experience as a whole, not the quality of each individual part (with the exception of trivial manufacturing/implementation faults that can be easily identified and removed). If something goes wrong and there is a generic symptom of sickness, it is not possible to prove formally, or algorithmically, or strictly rationally where the cause is. Again, intuition and induction must be used. Sometimes it is easier to cure the symptom, rather than identifying the cause.

Even with the above difficulties, traceability (finding *a posteriori* the motivation of existence for a given feature) is the only way to remove problems or to increase the effectiveness, since formal deduction or algorithms do not work. In a number of cycles like the one above, the resulting shape of the application (the living creature) can look chaotic, complex, hard to explain, but effective, in the sense that users like the experience and the pedagogical impact is good (or very good). As a result, reverse engineering, i.e. rationalizing the complex resulting design, is difficult to do, since the exact correlation between individual components and global effectiveness is not well understood.

3. LEARNING@EUROPE: A BIOLOGICAL CASE STUDY

Learning@Europe (Di Blas et al., 2009; Di Blas & Poggi, 2007; Di Blas et al., 2006; Paolini & Di Blas, 2006; Learning@Europe 2004-07: Final Report; Poggi and Di Blas, 2006) is a collaborative edutainment experience based on a multi-user virtual 3D environment, where students from different European countries meet in real-time to chat, play and learn about European history. More than 9,000 students, from 18 different European countries, Israel and the US, have successfully completed a L@E experience, with high pedagogical impact, as reported by their teachers.

The design of Learning@Europe is based on our previous experience with SEE – Shrine Educational Experience (Di Blas et al., 2003), and Virtual Leonardo (Barbieri & Paolini, 2001), two previous projects also based on collaborative virtual environments, developed by HOC – Politecnico di Milano.

Virtual Leonardo (developed in 1999-2001) mimicked a shared virtual visit to the Museum of Science and Technology in Milan, in order to enhance collaboration and learning.

SEE (Shrine Educational Experience) – a virtual adventure in the Israel Museum, Jerusalem – was a much more structured experience, including scheduled meetings in the 3D world among high-school students from 4 geographical areas, reading of educational materials in the intervals between meetings, team games, and a quiz-based cultural competition. Between 2002 and 2004 SEE was tested with over 1500 teachers and students (aged 12 to 19) from Italy, Belgium and Israel. Observations in schools, surveys and focus groups with teachers helped refining the experience and demonstrating its educational value.

Learning@Europe (L@E), developed in 2004 in cooperation with Accenture International Foundation, somehow represents a biological evolution over the two previous projects: good features were retained and emphasized, less essential features were removed, other improvements were attempted. L@E storyboard is a typical result of a biological lifecycle: apparently chaotic, difficult to explain through a rational design, but extraordinarily effective (98.4% of the teachers rated the overall educational impact good or higher, with 60.7% finding it very good or excellent – data from year 2006-07, available on the project's website). Also, as it typically happens with biological applications, it is very difficult (nearly impossible) to synthesize what it is about (interested readers can look at the website - www.learningateurope.net- or at the references mentioned above), while the way it works becomes clear when the experience is lived, and it is possible to give a detailed account for each of the main features.

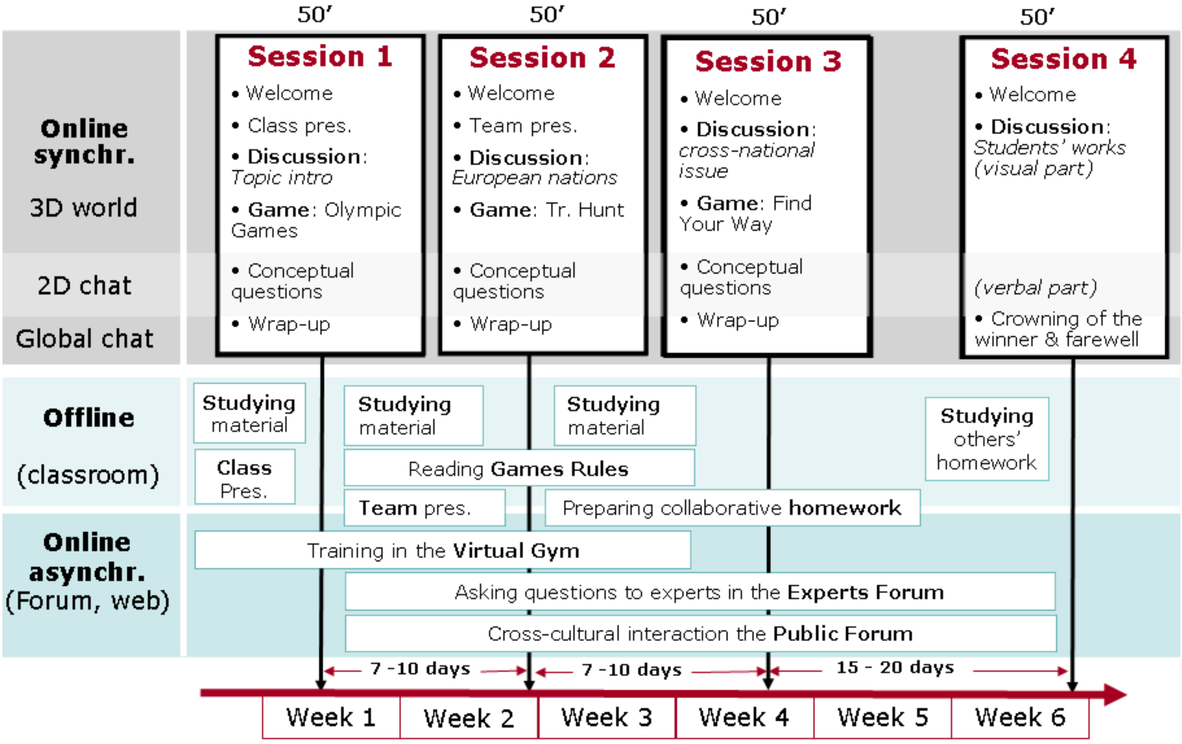

Figure 1. The storyboard of a Learning@Europe experience

Figure 1 shows the complexity of a Learning@Europe experience, which involves 4 groups of students from different schools and countries collaborating synchronously and asynchronously through different channels. They meet 4 times in an online 3D world, where they introduce themselves and their countries, form teams, discuss history-related topics, present their own research projects and compete in collaborative games (such as a treasure hunt) and cultural quizzes, also in a separate (2D) chat panel without the 3D graphics. In the intervals between online meetings, they study background material and collaborate with their remote peers on online forums, preparing shared research projects. All online interactions take place in English and are supervised and coordinated by two human tutors, who guide the students through the various tasks and games.

Stori@Lombardia (S@L), developed in 2004 in cooperation with the Regional Government of Lombardy, is an Italian-only version of Learning@Europe, focusing on Medieval history; it has involved between 2004 and 2006 over 1100 students from 29 different towns in Northern Italy.

Learning@SocialSports (L@SS) is the youngest member of the family: developed in collaboration with Fondazione Italiana Accenture and Verde Sport (of the Benetton group) within a Master in Sport Business Strategies, it targets teenage athletes of sport associations all over Italy, encouraging them to reflect and compare experiences about the social value of sports. Between 2006 and 2009 it has involved approximately 440 young athletes from over 60 sport associations in 50 Italian towns (Di Blas, Paolini, Poggi, & Torrebruno, 2008).

3.1. An example of biological evolution: the boards

In order to clarify our point with an example, we explain how one specific feature evolved, from its origin to the current version: the boards, i.e. panels displaying information in our shared 3D environments. The reader must bear in mind that the same path could be traced for most of the experience's features.

The point that we are trying to make is twofold:

- The current version of the boards is the result of a biological evolution, reacting to successes and failures of previous versions.
- Although we can explain why the boards have the current shape and functionality (and they fulfil their role), we can't come up with one synthetic requirement, and gracefully and rationally refine it into the current design.

Virtual Leonardo: the boards' "ancestors"

"Virtual Leonardo" was initially based on the idea that visitors to the virtual museum would spontaneously interact with each other (since they were given the possibility to do so, by means of the chat and the simultaneous visualization of their avatars in the 3D world), and that the only requirement would be to provide "cultural objects" as topics for the discussion.

Therefore, 3D models of Leonardo's machines (exhibited in the real museum) were scattered around the virtual environment, and some were linked to pages from the museum website describing the machines in detail. However, the expectation was not fulfilled: people, though being together in the virtual world, did not spontaneously engage into meaningful cultural discussions about Leonardo's machines. Something different was needed.

SEE: displaying objects

The Israel Museum (our partner in the SEE project) wanted users to understand and "experience" that the Shrine of the Book was a real place in Jerusalem, and (being a museum) they considered very important to introduce *objects* in the application, to "start from the objects". A new challenge also emerged: the environment had to be easily and quickly customizable in terms of content, since participating schools (our new target) were allowed to choose which topic they wanted to deal with (e.g. religious traditions, history, etc.); putting 3D reproductions of objects with their labels in the 3D environment would have been definitely too costly and cumbersome.

The devised solution was therefore the following. Boards (physical hotspots in the 3D environment), once clicked, would activate pop-up windows, showing a **picture of an object** (e.g. an archeological finding) plus an explanatory text (Fig. 2). Boards were activated by the **online tutor**, so that all the users would see the same content at the same time. The underlying motivation was the following: if users shared the same piece of information at the same time, they were more likely to **engage into a discussion** on that topic.

The solution did work: virtual museum objects (and associated descriptions) became relatively easy to create, to show, and to replace basing on the schools' requests; nonetheless, one weak point emerged: students closed the boards before reading the whole text (evidently **too verbose**). In addition, cultural discussions still would not start spontaneously.

A board of a different kind, introduced in the environment almost accidentally, turned out to be tremendously engaging; it displayed a set of **"yes/no" questions** about "being admitted in the Qumran community", to which students could answer and get a response in the end ("you are ready to be admitted" or "you need a long preparation"). This board had been introduced with the sheer purpose of keeping people busy while waiting for everybody to log in; the questions were so trivial that no scores were associated to them. Nonetheless, as observers in the classes reported, students got really excited about them, because they were asked to do something specific together as a response. This gave us the hint for improving the board feature in the next project: L@E.

Figure 2: Students (represented by avatars) in front of two of the 3D hotspots that activate a Board (left), and a verbose "proto" board from SEE 2002 (right)

Learning@Europe: play and learn!

In L@E we see the latest evolutionary step of the boards. As in SEE, boards in L@E are also pop-up windows activated by the tutor, so that all users see the same content at the same time. However, **boards are shown in pairs**. The first is a "suggestive" or documentary image of a topic, and the second is a "question" which comes in different kinds: multiple choice, "right or wrong", "what's missing", and so on (Fig. 3). Students love competing, therefore they are eager to answer. Thanks to this solution, students are more stimulated to learn (they have to study in order to answer), they are engaged, and they discuss about the same topic (for example, they want to know why the answer is incorrect); finally, this solution allows to test students' understanding.

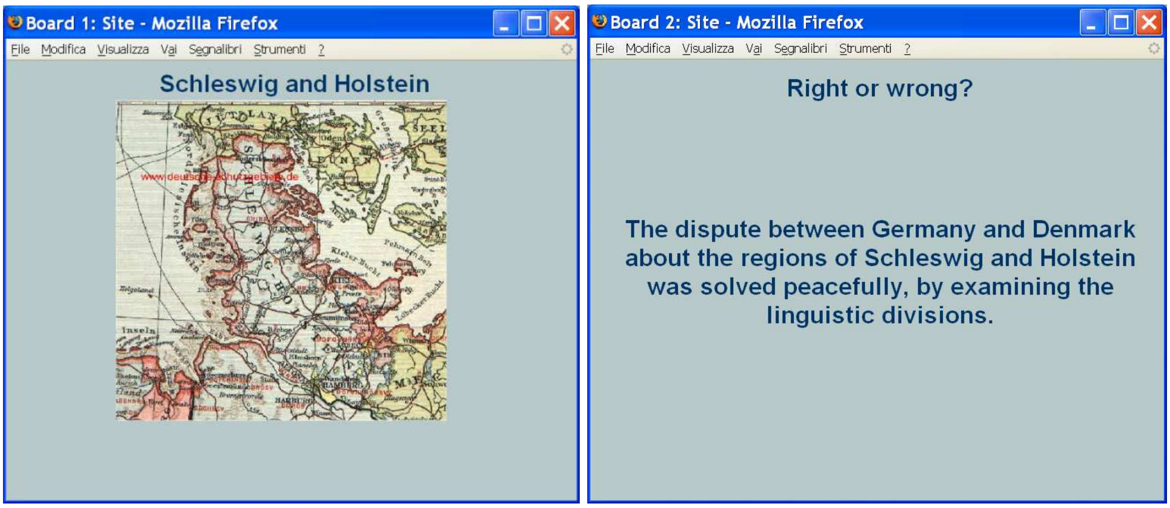

Figure 3: A picture-board (left) and a question-board (right)

This example shows how the successful characteristics of a feature (the boards) were not planned or designed in advance: they emerged as the experience with the application evolved in response to environmental factors (e.g. users did not have time to read verbose boards) and after the “trial” introduction of a new feature (Yes/No questions), which turned out to be effective.

4. RATIONALITY AS READINESS TO METABOLIZE CHANGE

In a biological perspective, the rational development is not the execution of a well-reasoned and pre-defined plan; the biological rationality of the lifecycle lies in the fact that some reasonable motivation can be investigated a posteriori for a given feature (or set of features), and this is possible only while the “organism” is living and evolving in its environment, not in advance.

Continuous feedback from the environment, reaction to external factors, responsiveness to changes and adaptation is favored in place of “planning in advance”, because the complexity of the application, of the environmental factors, and of their mutual relationships cannot be anticipated and described in a unified and synthetic view.

Therefore, rationality plays a fundamental role in this process, but assumes a different meaning with respect to the traditional “rational development”: rationality becomes the capability of being constantly **open to intelligently receive the input and the challenges** coming from the environment, and metabolizing them for changing the system in order to meet such challenges.

From this standpoint, **agile methods** (The Agile Manifesto, http://agilemanifesto.org) have already advocated that lightweight, trial-and-error, highly iterative approaches to development are much more effective than heavyweight models in specific contexts, such Human-Computer Interaction Design (Sharp et al., 2007).

However, with respect to agile approaches, a biological lifecycle has the following distinctive features:

- While agile methods focus on **software** development, a biological approach copes with complex human-intensive **experiences, with requirements of different nature, and often not strictly related to technological solutions;**
- While the agile movement suggests achieving “customer satisfaction by rapid, continuous delivery of useful software” (The Agile Manifesto), a biological application often responds to the diversified needs of many (hundreds or thousands), heterogeneous and changing **stakeholders** (e.g. students, teachers, tutors, staff, project managers, opinion makers, financing partners) and not a single client.
- While in the agile development, “working software is the principal measure of progress” (The Agile Manifesto), in biological applications the overall effectiveness of the experience is the only measure of progress, far more important than the technical soundness of the application.

- While in the agile lifecycle, “even late changes in requirements are welcomed” (The Agile Manifesto), life-like applications live with the assumption that requirements are always in flux and changes impose a wide variety of uncontrolled factors.
- Agile development works around the philosophy of “**simplicity**”, whereas biological applications are intrinsically complex and articulated by their very nature. The simplification of the development - and, consequently, of the application - will likely lead to its death.
- The agile philosophy assumes that a clear and unified picture of the artifacts can be - eventually – defined and shared (even if at the end of many iterations). Life-like, biological applications cannot be modeled, captured and explained by traditional design primitives. At every instant, we have a multiform prototype of a species which strives to fight with the environment to survive.

Following the biological metaphor, we can conceptualize the difference between a biological approach to development and the traditional models explaining the lifecycle as shown in Figure 4.

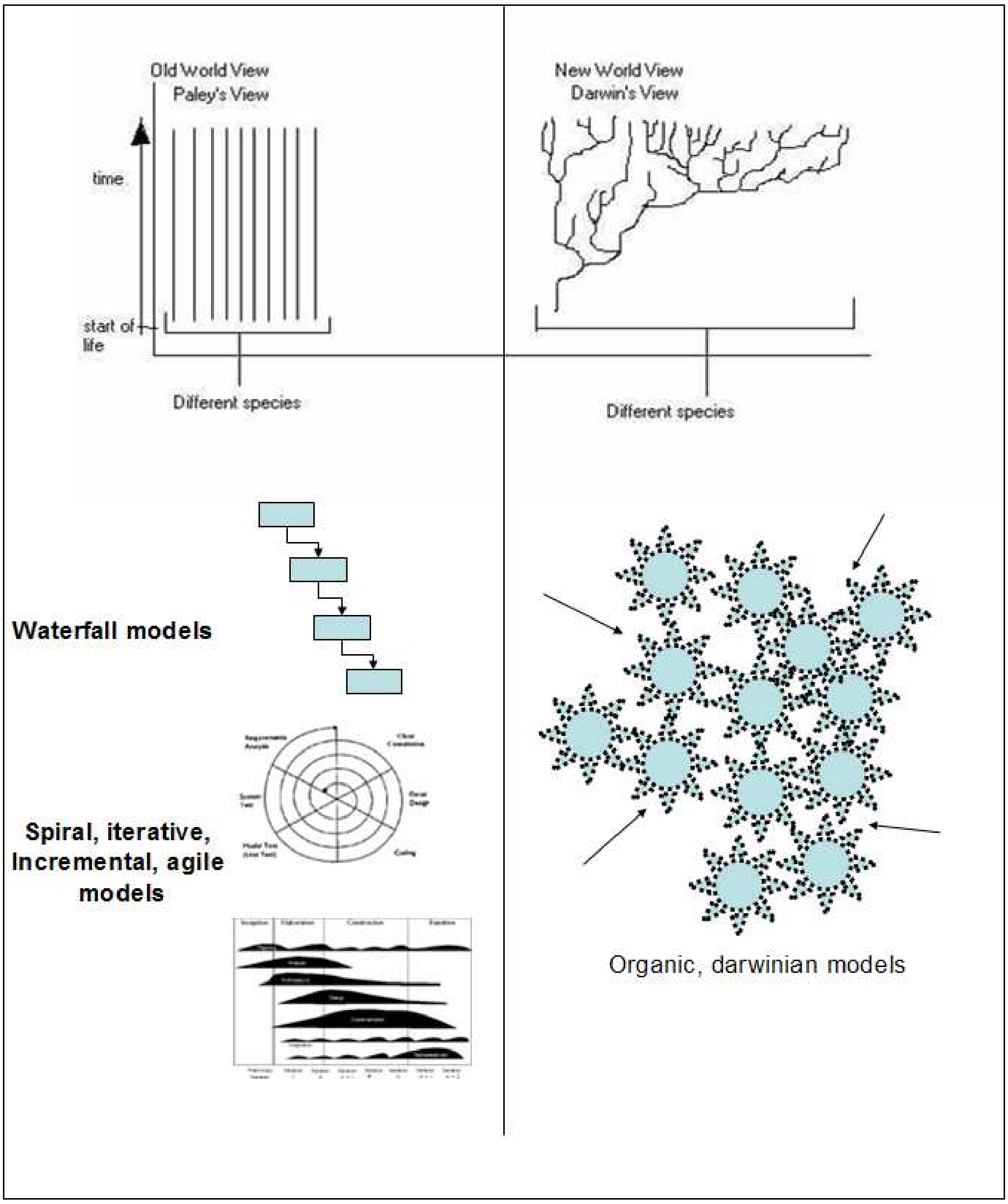

Figure 4. Models of biological evolution and of software evolution

While an old, naïve, view assumes the existence of the different species from the beginning of life until now, the main tenet of Darwinian evolution is that change occurs in species over time, through population variability and natural selection, with the eventual result being the appearance of new species.

Similarly, in traditional models explaining the lifecycle of applications (being them the waterfall model, or iterative, spiral, agile ones) there is a predefined number of discrete steps to be carried out from the conception of the application, and the key initial design ideas are iteratively refined, but not substantially changed. Prototypes are delivered, tested and refined according to the given methods.

The biological approach to the lifecycle assumes that the application lives in an environment which challenges its features and imposes a continuous, substantial evolution, adaptation, as well as the emergence of new, unexpected characteristics, building up a non "perfectly" shaped organism, whose only condition for survival is effectiveness. There is no distinction between development, prototype, final delivery and deployment: everything is in a flux of ongoing improvement.

Of course, in the case of software applications, changes occur because the designers carefully evaluate the prototype in its environment and intentionally modify it in order to better adapt to environmental conditions; this aspect of agency is not necessarily present in nature. Nonetheless we believe that the metaphor of biological evolution is useful to describe the different kind of rationale behind the lifecycle of these applications.

5. CONCLUSIONS

A biological lifecycle can model complex situations in which interactive software applications are designed, adapted and evolve organically without a clear, rational and formally describable process. In this perspective, requirements are just a few, very high level, non functional indicators or expectations upon the final results, such as the educational impact, the engagement, and the content to be learned.

The biological nature of the design stems from the fact that the factors at play are several, very complex, and belonging to very different dimensions (technology, content, social behavior, game rules, instructions to guides…). No general rules for design can be found. This is the opposite of “rational design”, where a few design principles are at the basis of complex artifacts.

Traceability follows an organic pattern, although it is very detailed and complex: several design elements of different categories are associated to each requirement; effectiveness of the global result is the only way to justify them; there is no simple, rational, formally describable correlation between requirements and design.

6. REFERENCES

Barbieri, T., and Paolini, P. (2001). Cooperation Metaphors for Virtual Museums. In D. Bearman & J. Trant (Eds.), *Museums and the Web 2001. Selected papers from an international conference*. Pittsburgh, PA: Archives & Museum Informatics, 115-126.

Di Blas, N., and Poggi, C. (2007). European Virtual Classrooms: how to build effective ‘virtual’ educational experiences. *Virtual Reality: Special Issue on Virtual Reality in the e-Society*. Springer London. ISSN 1359-4338 (Print) 1434-9957 (Online). Accessed February 26, 2010, at http://www.springerlink.com/content/r264167mt8654q81/

Di Blas, N., Garzotto, F., and Poggi, C. (2009). Web Engineering at the frontiers of the Web 2.0: Design Patterns for online 3D Multiuser Spaces. In *World Wide Web Journal, 12*(4), 345-379, Springer. DOI 10.1007/s11280-009-0065-5.

Di Blas, N., Paolini, P., and Poggi, C. (2003). Shared 3D Internet environments for education: usability, educational, psychological and cognitive issues, in J. Jacko & C. Stephanidis (Eds.), *Human - Computer Interaction: Theory and Practice*. Proc. of HCI International 2003, Vol. I, Mahwah (NJ), U.S.A: Lawrence Erlbaum.

Di Blas, N., Paolini, P., Poggi, C., and Torrebruno, A. (2008). 3D Worlds to Learn and Play: 6 Years of Projects with an Engaging, Pedagogically Effective, and Versatile Educational Format. In G. Richards (Ed.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2008* (pp. 738-745). Chesapeake, VA: AACE. Awarded paper.

Di Blas, N., Poggi, C., and Reeves, T. (2006). 3D for Education: Design and Evaluation of Educational Benefits. *Proc. 7th International Conference of the Learning Sciences* (ICLS), June 27 - July 1 2006, Indiana University, Bloomington, IN.

Ghezzi, C., Jazayeri, M., Mandrioli, D. (2002). Fundamentals of Software Engineering (2nd edition). Upper Saddle River, NJ: Prentice Hall.

Learning@Europe 2004-07: Final Report on Results. Available online at: http://www.learningateurope.net

Paolini, P., and Di Blas, N. (2006). Multi-User Virtual Environments for Education: A European Experience. *Proc*. E-Learn 2006, Honolulu, Hawaii (USA), October 2006; 1383-1394.

Poggi, C., and Di Blas, N. (2006). Visual Communication in Virtual 3D Environments, in *Proceedings of VL/HCC06. IEEE Symposium on Visual Languages and Human-Centric Computing,* September 4-8, 2006, Brighton, UK.

Sharp, E., Rogers, Y., Preece, J. (2007). Interaction Design: beyond human-computer interaction (2nd edition). Chichester, UK: John Wiley & Sons.

The Agile Manifesto, http://agilemanifesto.org/.