Ralf D. Fabian

Validating Non-Algorithmic Software in Service-Oriented Engineering Research

Addendum to

Bounded Rationality in Agent Orientation – "Just-In-Time" Visual Pattern Recognition

(PhD Thesis in Computer Science and Information Technology)

"Lucian Blaga" University of Sibiu,
"Hermann Oberth" Faculty of Engineering
Sibiu, 2013

In the discussion of ecological or social systems, it is not enough simply to say that everything depends on everything else, and so we must look at the whole system (OECD, *Science, Growth and Society: A New Perspective*, 1971: 57) [16].

Table of Contents

FIRST CHAPTER: INTRODUCTION

From Requirement to Opportunity

After showing its *raison d'être* (1.1), the *addendum organization* is presented (1.2).

1.1. About the Topic

The addendum *context* (*thematic*, 1.1.1 and *historical*, 1.1.2) is followed by (and subjectively filtered through) the *author's stance* (1.1.3).

1.1.1. Thematic Context

The Addendum (*ADT*) was necessary to obey to [1]. Referring to [2], the decision to validate the thesis was postponed ("*Rezoluţia*: Se amână luarea deciziei privind validarea tezei de doctorat" [1]), based on the following reasons: "*Argumentarea rezoluţiei*: Teza este insuficient susţinută din punctul de vedere al validării contribuţiilor. Recomandăm extinderea tezei cu un capitol substanţial de validare, bazat pe metodologii ştiinţifice şi inginereşti general acceptate/utilizate, prin care contribuţiile tezei să fie clar reliefate. Comisia se va pronunţa asupra acordării titlului după completarea tezei cu observaţiile de mai sus" [1].

The key phrases are: "The thesis is insufficiently supported as regards validating its contributions", "based on scientific methodologies and engineering methods", "that are generally accepted/used", and "emphasise clearly the thesis contributions".

To comply with [1] in both letter (validation must be based on "generally accepted engineering methods") and spirit (an irrelevant and poorly implemented experimental model cannot "emphasise clearly the thesis contributions"), its requirements are dissected and interpreted (3.1), restated from a descriptive form into a prescriptive one (3.2) and followed in the rest of *ADT*. However, because the analysis in Chapter 2 revealed deeper causes of failing to validate [2], the "New Course in New Context" (3.3), involved a wholly new approach in a very different research domain, changing drastically the profile of [2].

Consequently, the application research domain where the thesis contributions should be proved, was changed from "Visual Pattern Recognition" to "Non-Algorithmic Cybernetic Modelling of Living Systems". Within this domain, the non-algorithmic software mechanisms to be validated were applied in a research toolkit aimed at *exploring homeostasis* (with hysteretic delay caused by intense anthropogenic disturbance) *in benthic communities* (in Transylvanian lotic systems).

1.1.2. Historical Context

Although *ADT*'s history begins at September 12^{-th} 2012 (the date of the issuance of [1]), its prehistory (after defending [2], December 2nd, 2011) was marked by extending the existing cooperation with the research in [7], to memetic engineering (mainly applied to

ecolinguistics [4] [5]) and by starting to reorient [2] from "Visual Pattern Recognition" - already perceived as blind alley - to a "user-pulled" and more complex transdisciplinary research area more suitable for applying non-algorithmic software.

At first (October 2012, "*Plan A*") it was intended to design only one application ("*Preservation of Ecological Systems Modelled Using GST Applied to Benthic Communities in Transylvanian Lotic Systems*") for both validating [2] and continuing [7]. However, possible delays in meeting research deadlines and in completing the experimental model of [7] (now modified, to apply the lessons learned from failing to validate [2]) required to separate the application in [2] from [7]. Likewise, it was necessary to speed up and reduce the scope of the "Proof-of-Concept" (PoC) application for [2] to (new title): "*Homeostasis in Benthic Communities Modelled Via GST*".

At the begin of 2013, because of institutional reasons, both cooperations had problems. First was discontinued common research with the author of [7] (restarted only at the begin of July, much too late to be reflected in *ADT*). As a result, it has been necessary to set up a "*Plan B*" – but from the joint post-thesis work, only [5] and partially [4] could be used in *ADT*.

Plan B had two phases:

- *B1* was short-lived (January – February, until transdisciplinary cooperation was much slowed down; fortunately, it restarted with full power in April). *Plan B1* aimed to save the downscaled application mentioned above, by compensating – at least partially – the lack of temporal information and, above all, the lack of time to get it gathered. An urgency solution trying to "metamorphose" valuable spatial data into badly needed temporal ones was proposed but it proved to be too late (though, the failed solution was kept as – undesired – relevant example of applying "*JIT"*, 4.3.3).

- *B2* draw all the inferences from the new situation and set up the current framework, based on proposing a new approach to CybMd ("*Non-algorithmic cybernetic modelling of living systems*", 3.3).

The last update of this addendum took place on *July 16^{-th}, 2013*.

1.1.3. Author's Stance

Since the author's stance is implied in Chapter 1 and Chapter 2, here I add just the thanks for people who helped me most in the last 10 month.

Besides God and my parents, I have to pay tribute to the myriad contributions of my advisors, collaborators, friends and many others that I won't be able to mention here.

My sincere thank goes to Boldur Bărbat for advising me in so many matters of research and life. His tremendous support and incredible energy have had a major influence on this Addendum.

I am grateful to Angele Bănăduc for guiding me along the challenge of applying thesis ideas in ecological/biological systems and validating results as end-user. Together with people of her research group she initialized thought-provoking collaboration opportunities to support Cybernetic Modelling for middle and long range transdisciplinary endeavours.

A great debt of gratitude I owe to my friends: Adriana Deac – for her medical advice and support, keeping me in working parameters throughout the last few months; Cristina Brumar – for her great “Just-in-Time” situation management and always being there when help was needed, even at the cost of timeless discussions; Lucian Răhăian – for his help on demand and brave patience in the final testing phase.

Thank you all for your love, encouragement, patience, inspiration and friendship.

1.2. Addendum Configuration

After an updated list of the *abbreviations* (1.2.1), the *contents organization* as self-contained *Addendum* (*ADT*) to [2] is outlined in (1.2.2). General information about the *appendices* is given in 1.2.3.

1.2.1. Abbreviations

A:	Amplification factor
ADT:	Addendum to Thesis
AI:	Artificial Intelligence
API:	Application Programming Interface
β:	transfer factor of the feedback network
BeTi:	Bergsonian time
BR:	Bounded Rationality
CSIT:	Computer Science and Information Technologies
CybMd:	Cybernetic Modelling/Model(s)
Δt:	time granule (interval between two successive discrete time moments)
DOMINO:	Decision-*O*riented *M*echanism for "*IF*" as *N*on-deterministic *O*perator
DIU:	Acronym for *D*elay, *I*rreversibility, *U*ncertainty (delay time span)
GST:	General System Theory
IDE:	Integrated Development Environment
JIT:	“Just-in-Time”
MEng:	Memetic Engineering
NeTi:	Newtonian time
PoC:	“Proof-of-Concept”
PrsADT:	*ADT perspective*: validating [2] (software mechanisms) in line with [1]
PrsCSIT	*CSIT perspective*: from one (transdisciplinary research) bridge pillar
PrsU:	*User perspective*: from the other (transdisciplinary research) bridge pillar
SE:	Software Engineering
SOE	Service-Oriented Engineering
WISC:	“*W*hat-*I*f” *Sc*enarios
WiTi:	Wienerian time

1.2.2. Organization

Since an *ADT* is unusual, to enhance clarity, each chapter has a supra-title concentrating the chapter's *aim*, while the title focuses on the *contents*. Thus, bringing together all supra-titles from Chapter 2 to Chapter 7, the *ADT* target is distinctly outlined:

Starting from the *lessons learned (diagnosis)*, the *lessons* are *applied (treatment)*, proposing a *new modelling subbranch*, illustrated by *nested mechanisms* in a *research toolkit appliance*, able to ensure *validation*.

Because the syndrome revealed by the diagnosis in Chapter 2 is serious, the treatment prescribed in Chapter 3 is drastic too, thoroughly revisiting *validation of transdisciplinary service-oriented research*, and setting up a *new course in new context* (as a result, it is, totally reshaping [2], 1.1.1). The lifeblood of the new autonomous research, as substantial transdisciplinary extension of [2], aiming at *modelling homeostasis in ecologic systems*, is in the multifaceted Chapters 4, 5, and 6, answering the questions:

"*Why*" (Chapter 4): top-down rationale and bottom-up approach for *Non-Algorithmic Cybernetic Modelling for Biology*.

"*What*" (Chapter 5): key non-algorithmic *Mechanisms and Time Species for Modelling Stability* of living systems.

"*How*" (Chapter 6): research *Tool for Exploring Homeostasis in Benthic Communities* aimed at "What-if" scenarios (WISC) in Transylvanian lotic systems.

In Chapter 7, titled "*Validating Process-Oriented Modelling Mechanisms*", the engineering perspective regains pre-eminence: based on PoC validation in SOE, the concepts, approaches, and mechanisms asserted in [2] are systematically *tested quantitatively* in line with the industrial paradigm.

To be convincing, this thoroughly redesigning required ten multifocal "State of the Art" sections, six of them explicitly labelled as such: Chaoplexity in biologic modelling (4.2.1), Limits of current biologic modelling (4.2.2), Ecolinguistics blocked by conflicting memes (6.1.3), WISC in managing living systems (6.1.4), Variants of applying the Lotka-Volterra Model (6.4.2), Validation through PoC applications (7.1.2). Therefore, 52 from the 65 references, are published in 2012 or 2013 (the older 13 had significant role in reshaping *ADT*).

1.2.3. Appendices

There are two appendices for validation, two for scenarios, one with the toolkit code, and one with recent papers, downloaded but not referred to:

- *AV1*. *Code of threads relevant to validation*. The excerpt of code fragments from *AC5* (below) is necessary to show: a) compatibility between non-algorithmic software and quantitative testing; b) lack of suitable IDE for post-industrial engineering exploratory research; c) caducity of Java syntax for SOE [39], [34]; d) scarcity of API functions for real-time programming (mainly, heavy multithreading, [19], [32]); e) extended comments that reduce the "user-unfriendly-C++-family-spawned-unclearness"; f) programming techniques applied to overcome the Java language and the Windows 32 platform

weaknesses; g) introductory information for the next appendix (*AV2*, where relevant information is drowned in the deluge of data); h) compliance to the letter of [1].

- *AV2*. *Quantitative testing results*. Presents the test results for the four magnitudes (**A**, **β**, **Δt**, and **DIU**) necessary and sufficient to validate the non-algorithmic mechanisms incorporated in the research toolkit (6.2, 6.4) and used in WISC (7.3.3, 7.3.4) the toolkit is designed for.

- *AS3. Copy of the decision to postpone thesis validation*

- *AS4*. "*Proof-of-Concept*" *User-Validation Assessment*

- *AC5*. *Integral code of the toolkit*. The version with comments in Romanian is a future work middle range intention (8.2.1) to be provided together with the Romanian version of the interface, aimed at easing communication in field experiments (7.3.2).

- *AB6*. *Bibliography not referred to in ADT*. Aimed at helping transdisciplinary research, it includes about 150 recent scholarly papers downloaded (at least as abstracts) when investigating the ten "State of the Art" sections (1.2.2). For the sake of briefness, they were not included in the sections but could be useful in seminars (or laboratory work) in ecology. However, the appendix is not yet included in *ADT*, lacking a suitable organisation of the topics (hence, it is postponed as future work short range intention).

SECOND CHAPTER: LESSONS LEARNED (DIAGNOSIS)

Root Causes of Failing to Validate New Software

Working out what was going wrong when failing to validate the contributions of [2], three kinds of aspects emerged, with cumulative negative effect: the very *rationale* of *non-algorithmic software* was uncared for, leading to an inappropriate *application area* (2.1); *the experimental model* was *incongruous* (2.2); perhaps the most important factor: the *application* was *deficient* (2.3). Without convincing engineering facts, validation was impossible.

2.1. Non-Algorithmic Software? Rationale, Application Area, Misuse

Can *software* be *non-algorithmic*? If yes, is it *oxymoron, gadget, luxury, or slogan*? (2.1.1). Next, its rationale and application area are outlined in opposition to algorithms: *(Non-)algorithms* are good *to model entities that behave (non-)mechanistically* (2.1.2). A bewildering software species tries to bridge the gap, *pseudo-algorithms: variants, drawbacks and, above all, confusion* (2.1.3).

2.1.1. (Non-)Algorithms: Oxymoron, Gadget, Luxury, or Just Slogan?

A Google search (June, 23, 2013) for "everyone who taps at a keyboard... is working on an incarnation of a Turing machine," yields about 4,240 results. Mixing two antithetic phrases in the same sentence and accepting that the sentence is true, proves that any tapping at a keyboard is perceived as algorithmic, ignoring the essential difference between two wholly opposed species of data input.

Indeed, if the input occurs as *reaction* to a *READ* instruction, the stepwise functioning of the Turing machine is *not interrupted*. On the contrary, if the input is initiated by “everyone who taps”, the (kernel-mode) thread generated by the (hardware) *interrupt* is inherently *asynchronous* with respect to *all* other threads running in that moment. Hence, any data entering by users (from *own initiative*) is non-algorithmic *par excellence* because it happens in an *unpredictable* moment. Regarding this species of uncertainty as “epistemic facet of nondeterminism” in [2], perhaps exceeded the scope and competence of both thesis and author. On the other hand, denying the very existence of non-algorithmic software because it is sometimes labelled as “nondeterministic”, implying the (non-existing) “free will” of “everyone who taps”, is debatable too.

A most closed link issue appears when validating qualitative aspects with quantitative testing (7.2.1), where the dispute “future contingent versus determinism”, is replaced by the likewise unnecessary metaphysical choice between holism and reductionism.

The stance here is that it is counterproductive to take such stances. Indeed, in computer science as a whole, and mainly in software engineering, there is no need to choose – in a Manicheistic manner – between Emil du Bois-Reymond and David Hilbert, because a boundedly rational approach is to relativize the debate shifting from standpoints

(either '*ignorabimus*" or "*wir werden wissen*") to circumstances ("*ignoramus*", here and now).

Moreover, in science there are syntagma used for such simplifications: "macroscopically it appears that"; for instance: a) cognition is holistic; b) there is no algorithmic method for learning the mother tongue; c) robots and humans reason and act in distinct temporal frameworks. Thus, pragmatically, since this software species can be legitimately referred to, its entities should be labelled. A convenient label at hand could be: *non-algorithms*.

In short, non-algorithms should not be regarded as:

- *Oxymoron*. Product-oriented engineering, control engineering, robotics, need predictability, not chaotic asynchronicity (when applied only to be similar to humans).

- *Gadget*. Nobody cares about the (non)algorithmic nature of the software used for virtual weapons in a Nintendo-like game world.

- *Luxury*. A bodiless interface agent – the very incentive, apex, and even ethos of non-algorithms – is clearly still luxury. An avatar, even more.

2.1.2. (Non-)Algorithms to Model Entities That Behave (Non-)Mechanistically

The (too) concise title needs explaining, since the brackets suggest two assertions: The first refers to *non-algorithm*s (without brackets). The second states the role of *algorithms* versus their negative counterparts (no brackets, contents included): "Algorithms are good for modelling any entity that behaves mechanistically", i.e., not just *objects* but also *processes* as usual in domains cited above. The lesson learned is twofold, namely a PhD thesis in computer science should never: a) use disputable concepts, or mechanisms, or procedures, without a comprehensive, undisputable explanation of the intended meaning, target, reason, and approach; b) apply such entities outside the framework set up by the explanation.

To avoid repeating the mistakes of [2], both aspects are abridged after [6], where the issue is looked at from larger – both software and memetic engineering – perspectives:

- "*Non-algorithmic" means asynchronous*. Explaining the "non-scientific" name through memetic factors, [6] precedes the scrutiny of non-algorithms with a telling history of real-time programming, showing that the first *exogenous* and *asynchronous* intervention in programs was called "*unsolicited* input". Then, the defining feature of this software species is focused on: "*Unexpectedness* is mirrored in software through *asynchronicity*. In turn, asynchronicity involves *two threads*: the thread (perhaps kernel thread) that *signals* the interrupt or event and the thread that *handles* the event as exception. The major problem is to obtain a suitable trade-off between *asynchronicity* (required by the very exception nature) and a minimal *synchronicity* (required by communication coherence)" [6].

- *Non-algorithms should reflect "Non-Turing machine" behaviour*. Defined by opposition to algorithms, non-algorithms are applicable to the set of "(sub)systems to be modelled considering uncertainty caused by future contingency". The "computer-science-legitimacy" of using "future contingency" without any philosophic connotation, is defended by the authority of *The Stanford Encyclopedia of Philosophy*: "To

qualify as contingent the predicted event, state, action or whatever is at stake must neither be impossible nor inevitable" [20]. Thus, to illustrate convincingly "what they are and where to apply them", it is fundamental to realise that this predicted "event, state, action or whatever", "occurs in *irreversible* time. Here *irreversible* has the old and simple Augustinian meaning: (even for God) *Undo* is excluded. [...] Any kind of outside stimulus must be modelled as *unexpected* in irreversible time (if "*expected*"**,** the stimulus is trivial to model, since certainty is the desired limit of uncertainty). [...] In short, non-algorithms should be applied *iff* unexpectedness is architectonically inevitable" [6].

2.1.3. Pseudo-Algorithms: Variants, Surrogates and, Above All, Confusion

If non-algorithmic software is both conceptually rejected and practically necessary, how *are* currently modelled processes that involve unexpectedness caused by future contingency, i.e., that progress in irreversible time? There are four kinds of situations, described in [6] "in order of increased connection to unexpectedness:

- *Online interaction*. [...] When using script or mark-up languages, there is neither model, nor program, [...] nor unexpectedness. [...] Because of the "all software is algorithmic" meme, most users of such languages *believe* that these are programming languages. [...]

- *Service-oriented applications*. [...] In SOE any service provider should inform the user about the service evolution, mainly when something unexpected occurred. [...] Considering – as usual in product-oriented engineering – that unexpectedness is excluded (all possible situations being dealt with in the long time ago frozen "Product specification", or even in obsolete expert systems, via the "Closed-World Assumption") means oversimplifying even for the few non-trivial services that can be provided in canned form. [...] Informing the user only through some rare, synchronous messages, means ignoring the basic requirement that *services* are *processes* that have to evolve in user irreversible time.

- *Replacing exceptions through polling*. [...] Due to Moore's Law, time granularity is so fine that "unexpectedness can be looked for" through so fast polling, that interrupts can be treated immediately, needing no mirroring in exceptions. [...] Events occur so quickly that *macrochronically* they appear as *instantaneous* and *synchronous* (thus waiting for exceptions in irreversible time, can be substituted by looping in reversible time). [...] It is a delicate risk-management issue to assess whether polling or exceptions are less expensive and more robust.

- *Simulating BIC (bio-inspired computing)*. [...] This is the only situation when unexpectedness is accepted – and even welcome. However, this "unexpected" was unknown before, not because it did *not occur yet*, but because it was not revealed by discovery" [6].

2.2. Visual Pattern Recognition: A Threefold Unsuitable Research Area

An experimental model can be rather *inadequate* than *incorrect*. Thus, in [2] the model incongruity had three facets: the *domain* was *unfit for applying non-algorithms to prove the thesis contributions* (2.2.1); there was/is *no "user-pulled" interest in exploratory research (why* should users take part in investigation based on *other memes?)* (2.2.2); likewise, there was/is *no "technology-pushed" research interest "Outside the (Moore's) Law"* (2.2.3).

2.2.1. Domain Unfit for Applying Non-Algorithms to Prove the Thesis Ideas

"Thesis Ideas" is used as shortcut for the contributions of [2] selected as "concepts and approaches to be emphasised in relevant modelling" (3.2.2). The main thesis ideas – and above all the synergistic effect of the two thesis pillars, *"BR + JIT"* – cannot be proved without non-algorithmic software mechanisms. Thus, the first wrong decision was to choose "Visual pattern recognition" as domain for the experimental model aimed at validating [2] since it was unfit for illustrating the main feature of non-algorithms: *asynchronicity* expressing *unexpectedness* from entities with *non-mechanistic behaviour* (2.1.2).

Indeed, how behaves a pattern? As any object, a pattern is *atemporal* and *does not behave* at all. Only (too) metaphorically – even according to Dennet's intentional stance – it can be deemed that it "waits to be discovered". However, as it was shown above, such kind of problems are suitably solved by pseudo-algorithms (mainly by artificial neural networks, that seem to be the best for this task).

Hence, the working – and proving – power of non-algorithmic mechanisms aimed at supporting the temporal dimension of SOE could not be put into action. The most frustrating example is the powerlessness in exploiting the trivalent *IF* in DOMINO, where the third value was devised to deal with uncertainty caused by future contingents. Thus, the third *IF* exit, realised as "blend of a Łukasiewicz "**i**" interpreted as "unknowable" or "problematical" and a Kleene "**u**" interpreted as "temporary lack of knowledge"" [12], the blend being understood pragmatically as "*UNDECIDABLE in the time span given*" [12], remained unexploited.

2.2.2. No "User-Pulled" Interest in Exploratory Research (Why *Other* Memes?)

Of course, any dedicated researcher is (firstly) interested in exploratory research. Though, to reach a critical mass, the interest must be user-pulled by economic or social motivation (3.1.1). In the case of [2], the lessons learned through failing to find transdisciplinary research support are telling, because – besides the usual financial reasons – they highlight the cardinal influence of (ignoring) memetic factors, both general and specific ones. "Specific" refers here to much stronger variants of the general memes encountered at psychologists and sociologists, as potential transdisciplinary research collaborators (mainly in carrying out a relevant experimental model for validating [2]):

- *Widespread memes*. a) *Qualitative research* is held in discredit, because it seems to disregard mathematics (from optimisation methods, to differential equations, to numeric

computation in general, [2], 5.3.2, 6.1.2, 6.4.1). A (careless) reference to real-world facts is stigmatised as “empirical research” and could downgrade a PhD thesis – no matter what topic, domain, or relation with CSIT is involved. (Referring to “field data” is just a bit less imprudent.) b) *Doxastic logic* is preferred to credal logic, when interpreting/disputing any term having (even far away) relation to a philosophic stance (e.g., “nondeterministic”, “reductionist”, “future contingent”) or seeming to confine the humanistic ideal of perfection (e.g., “bounded rationality”, “Just-in-Time”, “AI”).

- *Psychologist/sociologist memes*. To impair bias in assessing the causes of lacking transdisciplinary support (namely, the memetic background of the paradigmatic discrepancy), just a quote about their main IT workhorse tool *GPSS* (General Purpose Simulation System), “developed around 1960 [...], an old tool, but it is still used and works perfectly” [56]: “It seems that the commercial simulation tools are being developed in order to prevent the user from any coding and to make all the mathematics (statistical considerations). [...] To say that this is a correct tendency, first of all we must know who the simulationist is. If he/she is a plant engineer, a sociologist, a warehouse manager or just an amateur, the tool should be "fast and easy", and the simulationist need not be a programmer or mathematician (user of kind 1). [T]he users of kind 1 frequently commit a fundamental errors, like using the Poisson enter-arrival time distribution for the Poisson arrival process” [56]!.

In short, *why should* users (of kind 1 or not) take part in investigation based on *other memes* (or simply ignoring the deep-rooted own ones)? Above all, when “recent trends” “enable the user to create intelligent objects in discrete event simulation” [56]!.

2.2.3. No “Technology-Pushed” Research Interest Outside the (Moore’s) Law

“Whether there is an ultimate limit to Moore’s Law is an open debate dependent upon future electronic innovations, material science, and physics. [T]he potential future nanotechnologies will enhance the current known barriers for Moore’s Law. [O]ur discussion of Moore’s Law in turn of density of discrete computing elements will become meaningless from quantum mechanical uncertainty and entanglement technologies point of view. As before those days, the economic limit will continue playing the key role, despite of the fact that we know we cannot break the fundamental limits of the atomic and nucleus nature of matter” [68]. Despite a thorough scrutiny of nanotechnologies versus limits (3D semiconductor, Spintronics, Heisenberg uncertainty, CNT transistor, Single-atom transistor), even the pessimistic evaluation in [68] predicts a limitation somewhere at “log2 (number of transistors/1000)” = 30, between 2030 and 2040.

Relating to image processing, the lack of engineering interest in incremental research (specifically, in software mechanisms aimed at speeding up image processing) should have been visible from the begin of the century and obvious a few years ago. As regards the superior domain (visual pattern recognition), sound engineering risk management (“If it works don’t fix it”), entailed continuing to apply the (pseudo-algorithm based) methods to improve artificial vision in robotics. (Perhaps, robotics is the only field requiring exploratory

research in any kind of pattern recognition.) Anyhow, visual pattern recognition was ill placed for validating software mechanisms aimed at SOE.

In short, because of Moore's Law, illustrating new mechanisms outside a chaoplex user-validated application is obsolete too or – as in the case of image processing – even pointless. Lesson learned: exploratory research in (software) engineering is necessary *because of* Moore's Law, provided that a suitable application domain has been found.

2.3. Deficient Application

After the chosen domain proved to be a blind alley, a successful application was the last rescuer. Yet, as *(partial?) corollary* of previous choices it proved to be too hard to avoid an *unimpressive and old-fashioned application* (2.3.1). Thus, it was no surprise that *in product-oriented setting* the *new* non-algorithmic *mechanisms* were *poorly illustrated* (2.3.2). Next came the *final blow for validation: simulated simulation of* an *interface agent* (2.3.3).

2.3.1. (Partial?) Corollary: Unimpressive and Old-Fashioned Application

The question sign in the brackets suggests that, despite the limited capability of the "visual pattern recognition" domain to illustrate convincingly non-algorithmic software, there was still a "partial hope" that a successful application could save validation. The attempt failed from both perspectives: *significance* (unimpressive architecture) and *handling* (old-fashioned structure). They are dealt with separately only to highlight the lessons to be learned:

- *Unimpressive architecture*. The core idea was good: focus on uncertainty to put the new mechanisms at work, first of all *DOMINO*, despite all difficulties (2.2.1). Yet, the power of DOMINO lies in expressing unexpectedness caused by future contingency through asynchronicity (2.2.1). This power remained vain because – as in the case of pseudo-algorithms (2.1.3) – the "unexpected" was *unknown* but not *uncertain* (since it was frozen in the bits of Lena's images). The attempt to force the *unexpected* via a detective-like scenario was prone to failure (it was a blatant example of the "solution in search of a problem" syndrome). As a result, interesting architectonic features referring to patterns (designating and demarcating patterns instead of just recognising them) or to non-verbal communication (reducing both graphocentrism and logocentrism) were undefended.

- *Old-fashioned structure*. Despite conceived as clearly service-oriented, the experimental model was implemented in an obsolete product-oriented outward appearance. Indeed, the service was rather hidden since the tangible output was still an image, albeit with non-uniform processed fragments. What is more, this processing – if necessary at some moment – would have become futile very soon because of Moore's Law (2.2.3).

In short, the application, lacking a genuinely post-industrial real-world setting, was unable to illustrate non-algorithms in an industrial background, where, acting as mixed blessing, they are perceived (understandably) as oxymoron (2.1.1). The lesson is detailed below.

2.3.2. Poor Illustration of New Mechanisms in Product-Oriented Setting

“Exceptions convey to software the fundamental need for mirroring “Just in Time” *events* in the *system* and its (open, dynamic, and uncertain) *environment*). To be seen as exceptions, such events must be a) *unexpected*, b) *consequential*, and c) *rare enough*. If the available IDE lacks high-level primitives for the first two features, they should be *emulated* through software mechanisms able to *reflect interrupts* as non-algorithmic entities” [6]. Referring to chaoplex systems [6] adds three reasons why reaction is their key feature: “a) *reacting* to environment stimuli is a manifest sign of efficient intentional behaviour; b) it evokes *resistance* to environment actions (a chaoplex system is able to *respond* according to its own intentions); c) likewise, it evokes also *persistence*, *continuance* through *stability* (here the undertone is to react in line with system (preservation) strategy, not necessarily implying *antagonism*)” [6].

The paramount role of interacting with the environment via “duplex communication line”, namely both “*speaking*” (speech acts, directives) and “*listening*” (paying attention, reacting) was uncared for. Though, it was not ignored, as the following quotations from [2], in turn abridging quotations from previous author and teamwork, shows: “designing *services* instead of *products*, involves *parallelism*, hence a powerful temporal dimension”, “the main weaknesses of current IT systems [...] stem from inappropriate conceptualising, based on rigid, algorithmic (i.e., deterministic, almost sequential, "computational", and atemporal processing), meant for decision making as "step by step solving of arising subproblems", not for decision making as "continuous process of dealing with unexpected, potentially risky, fast changing situations requesting immediate albeit not optimal response” [2].

Moreover, the other important software mechanisms proposed in [2], were also lost when the task of interacting with the environment was delegated to a dubious interface agent: a) the multifunctional analog input bar lost its meaning being downgraded to a trivial slider illustrated by a toy *problem* (not a toy *process*); b) “RETURN -1”, the only properly implemented important service-oriented mechanism, was too lonely to be able to save validation. It was too little (as engineering fact), too late (to be convincing in an – albeit only in background – product-oriented setting).

2.3.3. Final Blow for Validation: Simulated Simulation of Interface Agent

“Simulated simulation” refers to a second degree simulation: a) The interface agent got the (hard to fulfil) task to simulate the real-world unexpectedness, unavoidable in any nontrivial service-oriented application. The task became impossible to perform when validation of non-algorithmic software was at stake. b) Simulating the simulator was just as to give “la suerte suprema” to an already heavily bleeding bull.

Thus, the lesson learned refers to the (il)legitimacy of simulation *per se* and is so strong that some redundancy with the pseudo-algorithm portrayal (2.1.3) is excusable:

It is easier – and not at all risky – to simulate rather non-algorithmic software replacing (in debatable “tacit knowledge”), *programs* (inherently and overtly *algorithmic*) by *scripts* (inherently and covertly *non-algorithmic*). Indeed, tags (and mostly mark-up languages as

a whole) are not perceived as outside, (macroscopically) nondeterministic – sometimes arbitrary – intervention, but as “normal program instructions” (forgetting that a script has nothing in common with a Turing machine). What is more, a decade after giving up flowcharts, most syllabi do not consider necessary to explain that “compiling” and “interpreting” are distinct IDE procedures – some do not explain fully the “D” in DLL, neither.

On the other hand, for research in software engineering – above all when uncertainty and chaoplexity are involved – it appears that “I” in IDE stays yet for “inadequate” rather than for “integrated”. Thus, except very improbable “avatars meeting in kiosks” (perhaps just a commercial label for a teleconference), software mechanisms offered by API functions currently available in 32-bits platforms, are unable to implement convincingly “avatars as alter ego”. Thus, why trying to develop agents beyond the level of soundly programmed interface agents (based on pseudo-algorithms), (inter)acting as a usual secretary?

In short, in engineering – above all in exploratory research – simulation is unacceptable at architectonic level (where features that cannot be emulated, should be given up) and inexorable at structural level (where deficient language syntax should be circumvented).

THIRD CHAPTER: LESSONS APPLIED (TREATMENT)

Validating Transdisciplinary Service-Oriented Research

"Treatment" is here in its both meanings: "*cure*" and "*action*": After trying to interpret [1] in depth, *dissecting the requirements* (3.1), and to set up what it means to *follow* the prescripted *cure* (3.2), the entailed *action* is outlined: *new course in the new context* (3.3.)

3.1. Dissecting the Requirements

To single out the requirements, three significant phrases in [1] are interpreted (in order of increased specificity), starting from the lessons learned and reshaping them into a framework to address the problems exposed by the diagnosis above. They refer to: "emphasise clearly the *thesis contributions*" through *validation* (3.1.1), based on "*scientific methodologies and engineering methods*" (3.1.2) that are "*generally accepted"* (3.1.3.)

3.1.1. "The thesis is insufficiently supported as regards validation"

The "increased specificity" has to be applied from the very beginning: the circumstance that the thesis pertains to computer science adds rigour to the claim "to emphasise clearly the thesis contributions" [1]. In fact, that means: a) [2] has engineering research contributions to be emphasised; b) these contributions are explicitly in service-oriented transdisciplinary research; c) implication: the contributions are embodied in software engineering mechanisms; d) the mechanisms essentials for proving the thesis ideas failed to be validated. The immediate causes were identified as: a) irrelevant application domain (2.1.3); b) disregarding the effects of Moore's Law (2.2.2).

Coalescing the factors above, to avoid repeating failure, a careful blending of "Models of Innovation" [16] is necessary: "Each model postulates one single explanatory factor ("scientific or technological discovery" versus "customer or management need"), [...] scientists and in-house R&D necessarily drive innovation (science or technology-push)" [16]. Since any PhD engineering research is assimilable to "in-house R&D", the contradiction between perspectives becomes obvious: to be relevant, transdisciplinary research must be *user-pulled*; to allow undisputable validation, it must be *technology-pushed*. In practice, *user-pulled* means that the research domain should: a) be of definite economic concern or social priority; b) involve systems and environments to be dealt with, too chaoplex for existing modelling approaches (current models are either too simple or too complicated); c) require urgent exploratory research. Likewise, *technology-pushed* means that the mechanisms to be validated should be expressly developed for proving: a) soundness of thesis ideas; b) relevance to the systems to be modelled.

In short, *too* chaoplex and chaoplex *enough* means, from the "*user-pulled*" stance, challenging chaoplexity and avoiding the "solution in search of a problem" syndrome. From the "*technology-pushed*" stance it means more: what kind of *method(ologie)s* can be used.

3.1.2. “Scientific Methodologies and Engineering Methods”

Simplifying the meaning of “methodology” to “meta-method”, in short, “scientific methodologies” means *rigour*, and “engineering methods” means *facts*. Scientific rigour means – for [2], focused on “*technology-pushed*” – to consider the lessons learned from 2.2 and 2.3 (in brackets are very abridged hints to causes). Thus, the software mechanisms should be:

- *Applied in transdisciplinary exploratory research*. (Moore’s Law makes futile any attempt to develop software useful as research tool in other domains, above all when aimed at incremental research, 2.2.3.)

- *Incorporated in a service-oriented tool(kit)*. Moreover, the service should be certified by user researchers twice: at the beginning as being of non-trivial interest; at the end that the engineering embodiment of the service (previously agreed upon) is operational. (Any appliance shaped without user consultancy is prone to failure, 2.2.2, 2.3.1.)

Likewise, as engineering *facts*, the same software mechanisms should be:

- *Relevant to the ideas they have to illustrate*. No significant architectonic feature can be convincingly defended if simulated. (If no IDE suitable for non-algorithmic software is available, the solution is to *emulate* the feature, not to *simulate* it, 2.3.1, 2.3.3.)

- *Innovative for transdisciplinary research*. Since modelling is the main area were this requirement can be fulfilled (maybe, the only at hand), in practice that means that the mechanisms were crucial for modelling systems/processes/environments where existing models failed chronically to satisfy user research expectations. (If the mechanisms are new but marked no visible advance in user research, than they are not successful engineering facts, 2.3.2). In other words, the mechanisms have to *prove concepts*. The proof involves new validation methods, but subject to the restrictions below

3.1.3. “Generally Accepted/Used” Methods

Within the conventional paradigm, prevalent in the industrial era, quantitative validation (including quantitative testing) is perceived rather as pleonasm than as reference to a validation species. Moreover, qualitative validation is seen as a dilution (totally non-engineering in spirit) of the very validating process. Hence, only quantitative procedures are “generally accepted” as engineering methods. Apparently, the post-industrial era is not yet conceptually established, although most leading world economies entered this era, over a half a century ago. As a result, there is no theory of value for service-oriented economies, hence no metric to assess service quality, neither. Thus, for practical reasons, engineering has to keep on weighing *services* with validation methods developed for *products*. Corollary: the requirement to validate service-oriented software, via quantitative testing is justified.

On the contrary, avoiding quantitative testing would involve a “*petitio elenchi*”, the (informal but deceiving) fallacy of begging the question. Indeed, the implicit premise (“post-industrial engineering involves non-algorithmic software mechanisms *because* it is *service*-oriented”) would directly entail the conclusion (“*service*-oriented software mechanisms should not be validated by *product*-oriented methods – despite being still the

only generally accepted ones"). The fallacy must be avoided finding *characteristics* (expressing well defined *and* user-accepted semantics) that are able to be relevantly validated as *magnitudes* (i.e., tested *quantitatively*) in line with generally accepted product-oriented methods.

Even inherently holistic, highly subjective, elusive features as *anthropocentrism* should be validated by *quantitative testing*, albeit indirectly (for instance, testing a quantifiable significant aspect of *simplicity*, regarded as main manifestation of user-centeredness. Though, to get user acceptance, such testing can hardly replace negotiating every interface component with the end user (preferably, applying the Scandinavian method.

3.2. Following the Requirements

In line with the interpretation in 3.1, the requirements are restated to be fit for setting up the approach in 3.3. This requires *changing the perspective after failing to validate thesis contributions* (3.2.1). From this stance, the *concepts and approaches to be emphasised* (in modelling) as well as the *mechanisms to be demonstrated* are stated in 3.2.2 and 3.2.3, respectively. Next, *choosing a research domain fitting with "Proof-of-Concept" Validation* becomes rather easy (3.2.4).

3.2.1. Changing the Perspective after Failing to Validate Thesis Contributions

To prove that the lessons have been fully learnt, the thesis contributions to be unambiguously mirrored by a validated application should be selected carefully, avoiding ideas (in parentheses are given some of the most frustrating examples) that are: a) irrelevant (from a leading paradigm stance) for non-algorithmic software (semiotic-oriented software engineering, interface agents); b) relevant *per se* but are not supported by affordable programming environments and, hence, become irrelevant when they are simulated (agent-orientation, "precision is often needless", "precision is sometimes dangerous").

On the contrary, key ideas wiped out by the failed experimental model (chaoplex situations cannot be managed using deterministic software, semantic validation through analog /approximate data input, uncertainty expressed through trivalent *IF*, anthropocentric interfaces), should be considered from the very beginning in the application design space – even if lacking *API* entail their partially simulated reflection in the implemented software (exception handling, dynamic exception propagation, irreversibility expressed by Bergsonian time (BeTi)). Thus, the perspective has to be refined, avoiding any weak link in the following (informal) chain of inferences:

- Non-algorithmic software is believable only when *modelling chaoplex processes*.

- Chaoplex processes are worth to be modelled only in *transdisciplinary applicative research*, in a leading-edge domain, with high: a) economic significance; b) social priority.

- Transdisciplinary modelling requires *GST* as: a) *Lingua Franca*; b) metascience. *A fortiori*, *modelling living systems* requires *cybernetics*.

- CybMd are most relevant to living systems in *dynamic environments*.

- Sensibly, that means that *challenging biologic* problems require *CybMd*.

3.2.2. Concepts and Approaches to be Emphasised in Relevant Modelling

In line with the above, the selected contributions are grouped into two categories: the *sine qua non*, resumed below and the desirable ones, taken into account in the approach (3.3):

- *Bounded rationality*: as both *strategy* and *mechanism* for urgent decision-making in high-risk situations, based on severely incomplete information.

- *"Just in Time"*: as key requirement for any service providing (in the post-industrial era). Hence, *a fortiori* for (nontrivial) modelling as transdisciplinary service.

- *"BR + JIT"*. The synergistic effect due to merging the two thesis pillars is obvious when *simplicity* seen as *output* of bounded rationality meets *urgency* seen as *input* (mostly, even *raison d'être*) of "Just-in-Time". The effect is even more relevant when applied to modelling (e.g., the success of discrete-time modelling is irreducible to just one of them)

- *GST as*: a) *Lingua Franca* (living systems cannot be investigated without it). b) scientific infrastructure for Cybernetics (crucial processes – for instance, reaching/maintaining stability of living systems – cannot be modelled without it).

- *Transdisciplinarity*. Is essential since, otherwise, any research in Computer Science could be regarded as "solution in search of a problem" (3.1.1).

- *User-centredness*. Anthropocentric interfaces are vital in reducing cognitive chaoplexity and in hiding structural chaoplexity (otherwise, wide acceptance is lacking).

- *"Precision is useless"* and *"Precision can be harmful*". From validation stance they can be regarded as outcome of *"BR + JIT"* (shown in the slider for analog input).

3.2.3. Mechanisms to be Demonstrated via "Proof-of-Concept" Software

Being assessable only incorporated in functional applications, mechanisms are main prey of any kind of validation failure. Thus, the perspective has to be refined more carefully than above. The factors (including circumstantial ones) hindering mechanisms validation are:

- *User-validation is difficult* in transdisciplinary research because mechanisms are *structural components*, not *architectural features*. Moreover, anthropocentric design requires that they should be hidden by the interface. (Paradoxically, a mechanism noticed by the end-user signals unsuitable design, mainly a poor interface.)

- *Conceptual confusions* could distort seriously the validation process. The most dangerous is described in detail in 7.1.1 but there are others too (e.g., "pseudo-algorithmic programming", 2.1.3).

- *Software mechanisms* are the core of *engineering research* in [2]. Thus, refuting the utility or even the very existence of non-algorithmic software (2.1.1) could be bewildering for validation evaluators, since such mechanisms, *non-algorithmic par excellence* must be validated against their nature, namely using *algorithmic testing* (7.2).

- *Structural chaoplexity*. The reasons for chaoplexity (3.1.1, 3.2.2) refer mainly to its *cognitive* component. Though, its structural component is also necessary: to be relevant as test bench, the system to be modelled has to be inherently chaoplex. Accordingly, the approach (3.3) involves the development of new mechanisms – integrating those proposed in [2] – requiring validation for themselves.

Hence, the mechanisms to be validated (through quantitative testing, *AV2*) are those carried out for ecologic modelling (5.2, 5.3), implemented in the appliance (6.3, 6.4), *and assessed* according to PoC validation (7.1).

3.2.4. Choosing a Domain Fitting with "Proof-of-Concept" Validation

"A proof of concept […] is a realization of a certain method or idea to demonstrate its feasibility, […] or a demonstration in principle, whose purpose is to verify that some concept or theory has the potential of being used. A proof of concept is usually small and may or may not be complete […] In engineering and technology, a rough prototype of a new idea is often constructed as a "proof of concept". For example, a working concept of an electrical device may be constructed using a breadboard. A patent application often requires a demonstration of functionality prior to being filed. [wikipedia.org/wiki/ Proof_of_concept].

A recent (2012) example [33] from an authority (Los Alamos National Laboratory) highly professional branch (Engineering Advanced Studies Institute), in the spirit of SOE ("Potential Collaborative Research"), in a most sensitive domain ("with Korea's Agency for Defense Development") suffices to show the power of: this concept in action: "Invite innovative graduate student and postdoctoral researchers [...]. ~3 week duration. [...]. Develop a lecture series to catch bring researchers up to speed on multidisciplinary problems of interest to the laboratory. Research high-risk, forward thinking concepts. In some cases try to develop prototype, proof-of-concept demonstrations or models" [33].

In short, the power of the method is manifest *iff chaoplexity is the spinal cord of a robot-portrait* of the research domain looked for. The search area was narrowed further to a subfield involving chaoplex *biologic* problems, where modelling requires *cybernetics*. In such fields chaoplexity is both *cognitive* and *structural* fitting with the requirements. (Besides, in biology structural chaoplexity "seems macroscopically" to be irreducible; 4.2.1).

Thus the role of the (transdisciplinary research) user – important anyhow – becomes paramount. Here the circumstances where decisive in finding an on-going ecologic research aiming at *Preservation of Ecological Systems*. (Applied to benthic communities in Transylvanian lotic systems, the research was in the phase when gathering data about biodiversity was completed and a model for system stability was badly needed, 3.3.1.).

To keep the end-user role, as protagonist in SOE (even in the new situation of PoC validation), the appliance functionality should be concentrated in the interface. Moreover, the interface is designed as *workhorse tool* for ecologic research, focused on setting up WISC aimed at *preservation of* (oversimplified) *ecological* (sub)*systems* (within the research project mentioned above, 4.3.1, 6.2.4, 7.3.3, 7.3.4)

As a result, "*Preservation of Ecological Systems*" appeared as a research subfield that fits very well the traits of the robot-portrait. It was chosen for PoC validation.

3.3. New Course in New Context

The new approach resulted from *adapting to new conditions* (3.3.1) and *(re)adapting to the (widening) paradigm gap* (3.3.2). The inferences from lessons (Chapter 2) and requirements (3.1, 3.2) are stated in *defending the orientation of the next three chapters* (3.3.3). Changing the perspective, the same chapters could be regarded now not as a plain extension of [2], but as self-contained research, proposing a new subbranch of modelling, maybe entitled "*Non-algorithmic cybernetic modelling of living systems*" (3.3.4).

3.3.1. Adapting to New Conditions

Adapting was influenced by repeatedly changing circumstances and, mainly, by their effect on the approach. Two factors (commented upon in 1.1.2) had major impact in shaping the new course (3.3.2): a) Interrupting the common research with [7]. b) Suspending the application aiming at preservation of ecological systems.

Paradoxically, both factors had also substantial positive outcome:

- Learning and applying the now usual way of validating transdisciplinary research via PoC applications (3.2.4, 4.1.4).

- Realising that the application mentioned above was far too chaoplex, as well as over-sized in scope and aim, to allow suitable modelling, applicable in due time, with research resources affordable at a medium-sized East-European university, even with West-European sponsorship (State of the Art, 4.2.1, 4.2.2).

- Realising that even the downscaled application "*Homeostasis in Benthic Communities Modelled Via GST*" is too challenging to be developed as PoC application.

- Realising that living systems require CybMd (4.2.3) and its corollary: non-algorithmic software, despite unavoidable, must be implemented with outworn IDEs.

- Corollary: the new course is substantially more relevant to both illustrating the thesis concepts and approaches (Chapter 6) and to engineering research, via two original mechanisms that proved their utility in transdisciplinary research (Chapter 5).

As a result, from here on the whole *ADT* architectonic was changed, as shown below.

3.3.2. (Re)Adapting to the (Widening?) Paradigm Gap

"Interfacing science and policy raises challenging issues when large spatial-scale [...] environmental problems need transdisciplinary integration within a context of modelling complexity and multiple sources of uncertainty [...] Approaches (either of computational science or of policy-making) suitable at a given domain-specific scale may not be appropriate for wide-scale transdisciplinary modelling for environment (WSTMe) and corresponding policy-making [...] In WSTMe, the characteristic heterogeneity of available spatial information (a) and complexity of the required data-transformation modelling (DTM) appeal for a paradigm shift in how computational science supports such peculiarly extensive integration processes. [...] This challenging shift toward open data [...] and reproducible research [...] is also strongly suggested by the potential – sometimes

neglected – huge impact of cascading effects of errors [...] within the impressively growing interconnection among domain-specific computational models and frameworks" [59].

This relevant paper was not cited in the numerous "States of the Art" below, but here because it stands for the decisive motive to give up any hope that in the next years there could be accessible IDE for non-algorithmic software as a whole or just for its crux, exception handling. The polyvalent, multifaceted, and consequential reason follows:

- authored by highly professional transdisciplinary researchers,

- endorsed by authorities like the *European Commission, Joint Research Centre, Institute for Environment and Sustainability* or the *United Nations World Food Programme*.

- presenting "an awesome battery of free scientific software" to "exploit advanced geospatial modelling techniques"

- approving that "complexity [...] should not be intended as an excuse for obscurity"

- accepting that approaches "of computational science or of policy-making [...] may not be appropriate for wide-scale transdisciplinary modelling for environment (WSTMe)",

- referring new branches of science like "post-normal science" or "emerging engineering" (aiming at easing that kind of problems),

- calling for "paradigm shift in how computational science supports [...] integration"

- considering uncertainty "within a context of modelling complexity and multiple sources of uncertainty" and relating the "sources" to "post-normal science" from "a post-normal perspective"

[59] disregards all transdisciplinary assumptions for modelling living systems (4.2.3) as proved by what is ignored in the paper (for instance, neglecting the possibility that uncertainty could stem from "*lack of occurring*" in a dynamic environment).

Thus, three key concepts related to evolution appear only once in irrelevant context: "*process*" (in a Journal title), "*time*" (in "scattered time series", referring to "geospatial data"), "*temporal*" (in "temporal resolution" referring to "remote sensing"). Emphasis is only on "the characteristic heterogeneity of available spatial information".

Therefore the desideratum "to better communicate part of the policy-relevant knowledge, often difficult to transfer from technical WSTMe to the science-policy interface" is far to attain. Moreover, as regards non-algorithmic software – or just a suitable temporal dimension in modelling living systems – , there is no soon paradigm shift in sight.

3.3.3. Defending the New Orientation of the Next Three Chapters

As new *ADT* core and also as significant transdisciplinary extension of [2], the three following chapters are outlined from a fresh (threefold) perspective – albeit in an old *Matryoshka doll* shape. The three perspectives are hinted at in Figure 3.1 and are leitmotif of the next chapters. Thus, they are worthy of a thorough explanation at the beginning of Chapter 4 where the bridge (suggesting only transdisciplinary research perspectives) is replaced by three step-pyramids portraying in detail the role of each doll, from each perspective. Then, why this prolegomena here? Because the self-contained research in cybernetic modelling of living systems (3.3.4) condensed in the next chapters should be

judged upon with the bridge in mind: the Matryoshka *structure* is necessary for validation but the transdisciplinary architecture is paramount.

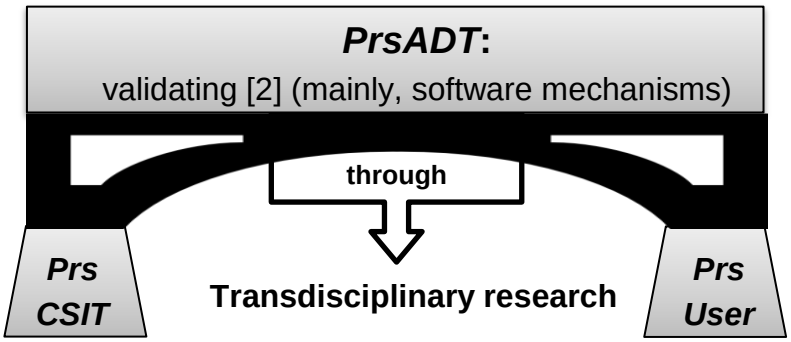


Figure 3.1. Threefold perspective of (non-algorithmic) cybernetic modelling.

For the next chapters "*U(ser)*" means researcher in biology/ecology (sub-perspectives are given in 6.3.3). "*Non-algorithmic*" is placeholder for "*macroscopically nondeterministic*" (being more palatable within old paradigms, 2.1.1, 2.2.2). It will be employed for pragmatic reasons, mainly in the syntagm "*non-algorithmic software*" (necessary from both *PrsADT* and *PrsCSIT*), and is put in brackets as it is useless from *PrsU* (users are interested neither in the intrinsic software nature, nor in philosophical debates about determinism). For similar (memetic) reasons, "*process-oriented*" will be employed instead of "*agent-oriented*".

3.3.4. Non-Algorithmic Cybernetic Modelling of Living Systems

Focusing on modelling homeostasis as transdisciplinary extension aiming to validate [2], the undertaking reached (through serendipity) another architectural status. Since the following chapters have to remain integrated in the initial architectonics, the new stance can be outlined only from outside them. Thus, below, each main motivation of *non-algorithmic CybMd* is followed by citing a most recent (2013), focused (process-system engineering), and effectual (successful CybMd with significant results in biology and biotechnology) reference [41]. Since the posture in [41] is one of mechanistic modelling the quotes cannot be supposed to be biased and show that the modelling proposed in the following chapters is a distinct new subfield of *CybMd* (itself a novel field of modelling).

- *CybMd is necessary in the case of living systems*. ("Considering the developments in analytical and molecular biology over the past decades brings [the] argument that the development of mathematically and computationally orientated research has failed to catch up with developments in biology. Mathematical biology today revolves around mathematical expressions developed a hundred years ago [...] Notable studies attempting to introduce a new approach to biological systems modelling include, but are not limited to, cybernetic modelling".)

- *Processes in living systems are chaoplex at any level*. (“The complexity of the regulatory network and the interactions that occur in the intracellular environment of microorganisms highlight the importance in developing tractable mechanistic models of cellular functions and systematic approaches for modelling biological systems.”)

- *Boundedly rational approaches are necessary because cognitive chaoplexity is vast*. (“with the field of biotechnology shifting from method development to application development [...], a systems biology approach of detailed, mechanistic modelling becomes problematic since modelling of complex biological systems inherently is an inverse problem that cannot be solved [...] and understanding of experimental information has lagged far behind data accumulation”.)

- *CybMd needs much simpler mathematical tools*. (“the sophisticated mathematical toolset that led to the explosive growth of manufacturing capacity in traditional chemical industries known as Process Systems Engineering (PSE), is not readily applicable to the bio-industry. Obstacles hindering the adaptation of traditional PSE approaches to bio-processing include the complexity of the biological systems, the limited understanding of the biological processes, and the resulting lack of adequate process models”.)

- *Atemporal models are unsuitable for living systems*. (”Various studies presented in the past few years have focused on dynamic modelling of genetic circuits [...]. As the function of a greater portion of the gene control network is clarified, it will be possible to apply mechanistic mathematical models that describe the dynamics in key regulatory systems for the design of optimal bioprocesses”).

FOURTH CHAPTER: NEW MODELLING SUBBRANCH

Non-Algorithmic Cybernetic Modelling for Biology

The framework, together with the *start vector* are presented first (4.1). Since the *top-down rationale* is based on *fighting cognitive chaoplexity* (4.2), it entails a practical *bottom-up approach*, trying to support researchers, "*Just-in-Time*", with effectual and user-friendly *workhorse research instruments* (4.3).

4.1. Matryoshka Doll Framework and Start Vector

To prove that the subfield of CybMd, chosen as applicative research domain for validating software mechanisms, is suitable also from a pragmatic SE viewpoint, it must lead easily to a convenient *Matryoshka doll* validation *framework*, adequate also to house a *research toolkit* (4.1.1) serving as PoC appliance. In this light, the *start vector* – that has to mirror from all perspectives the robot-portrait sketched in 3.2.4 – is based on three types of traits, labelled (in order of increasing *flexibility*) as *premises* (4.1.2), *criteria* (4.1.3) and *desiderata* (4.1.4). However, from a transdisciplinary viewpoint, the order is of decreasing *priority*.

4.1.1. Research Toolkit in Matryoshka Doll Shape. Three Perspectives

The reasons for this nested modular shape are given, below, the first being decisive:

- *PrsADT*. The very idea of Russian dolls *secures validation*: if the outermost Matryoshka is attested, all the others are attested too. The dolls are taken "out of the box" to identify clearly each mechanism to be validated.

- *PrsCSIT*. The CSIT language implied by the modular framework conveys better the *links between* the *mechanisms* and the thesis *contributions* they have to support.

- *PrsU*. Empathising with the end-users, by trying to utter the topics in a (quasi)ecologic language, it is to expect that researchers will devise more suitable WISC.

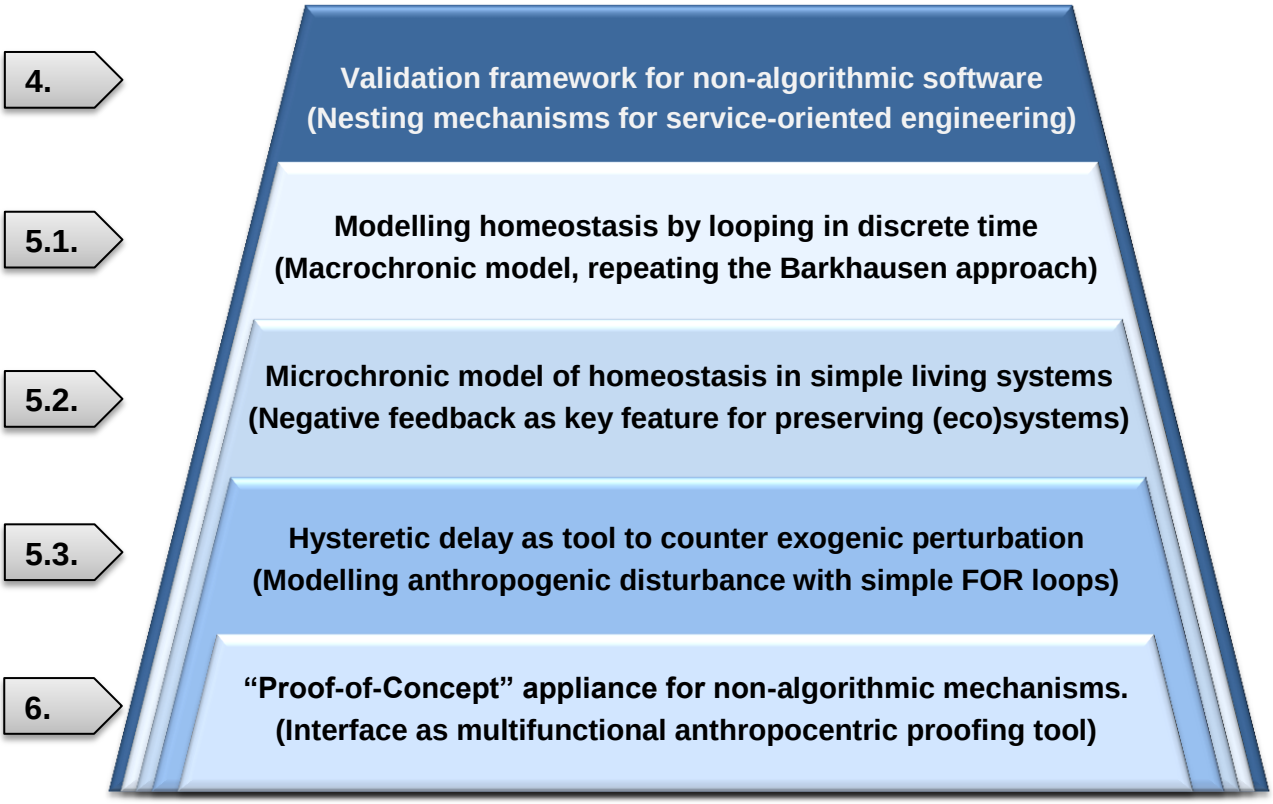


Figure 4.1. Exploring homeostasis (with hysteretic delay) in living systems.

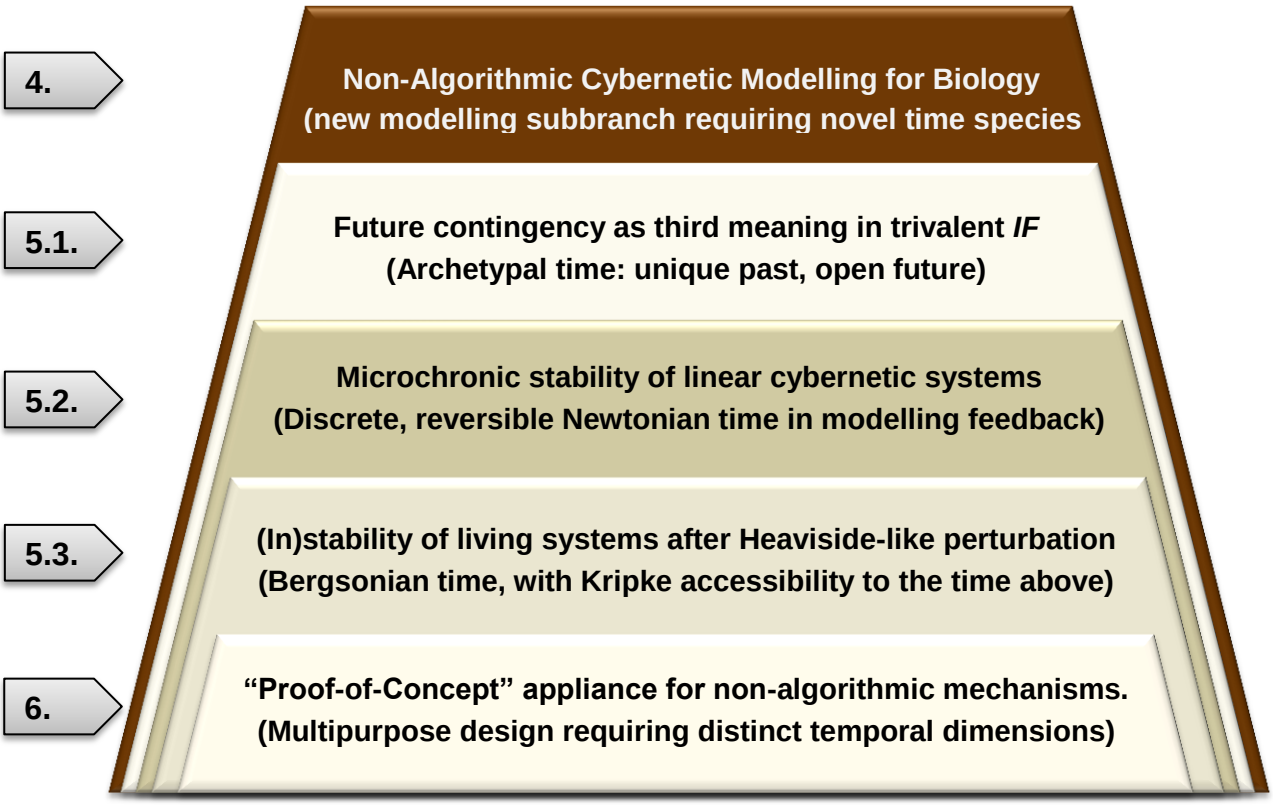


Figure 4.2. Exploring stability of chaoplex systems (in discrete time).

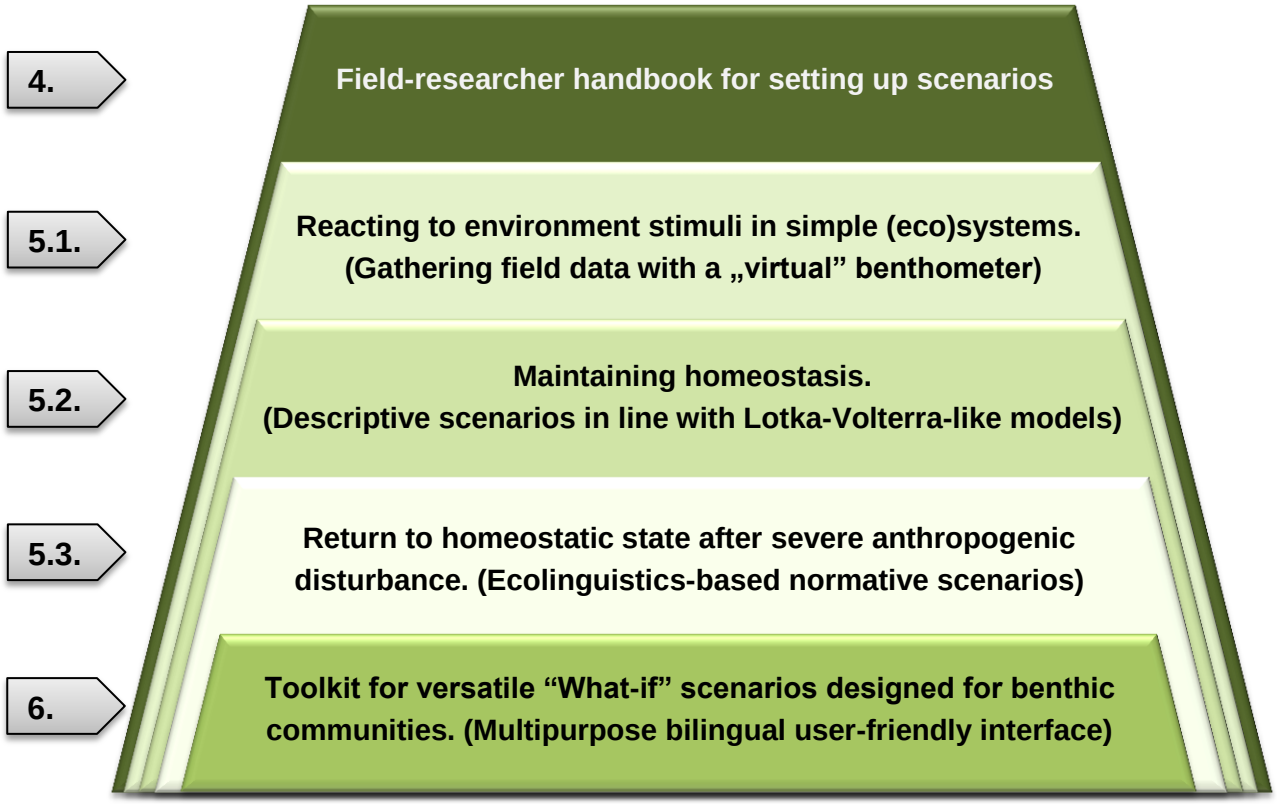


Figure 4.3. Exploring preservation of (simplified) ecologic systems.

The dolls are regarded from each perspective in Figures 4.1-4.3, but not as Matryoshkas because the step-pyramid arrangement: a) allows easier readable text; b) encourages to put the dolls on a postament (the appliance) and to cover them with a (conceptual) umbrella; c) suggests chronological order; d) for Figures 4.2 and 4.3 it suggests also X-raying the two transdisciplinary research bridge pillars in Figure 3.1.

4.1.2. Premises to Start from

As fundamental and set up in advance requirements, they are *sine qua non* in the very meaning of the phrase: without fulfilling these prerequisites, the undertaking is fruitless. Here, they refer to three underlying features of non-algorithmic software (i.e., non-algorithms) that must be compellingly reflected by the mechanisms. To impair stiffness, they are shrunk to three. To increase clarity they are stated in three terminological formulations, reflecting the stakeholder stance: *user* (*PrsU*), *designer* and *implementer* (*PrsADT*). (To defend the user position, the order is of decreasing importance from a *SOE* stance.)

- *Application Domain Language*. Here, this means the terminology familiar to *ecologists* as service beneficiaries (model end-users). Obviously, users consider that any model should reflect at least three basic features of biologic systems: a) they *evolve* in time; b) they *react* to environment stimuli; c) as a result, they have to face *uncertainty*.

- *Transdisciplinary Language*. This is the vital bridge between user requirements and model suitability (4.2.1, 4.2.2). It is the task of post-industrial engineers to set up a common linguistic denominator: some concepts are eliminated (e.g., after a scrutiny based on [5], the term "*biocenosis*" was replaced by "*ecologic system*"), others have their meaning (and use) revisited (e.g., "predictable model", 4.2.2). In order to act as essential design-space dimensions, they are stated as: a) the *evolution* of biologic systems must be modelled as *process* taking place in *BeTi* (4.2.4, 5.1); b) system *reactivity* to environmental *perturbations* should be modelled to ensure *homeostasis*; c) *uncertainty* cannot be modelled using time series (decision making based on *future contingency* should be considered – at least – to try to reduce major effects of unforeseen anthropogenic disturbance).

- *Computer Science Language*. In line with 3.2 and 3.3 the language must be that of *conventional* SE. Thus, despite being intrinsically non-algorithmic, the software described in Chapter 5 avoids any reference to new paradigms or to ambiguous concepts. As a result, the premises become: a) *temporal evolution* should be reflected by inputting data based on repeated measurements in the same places (e.g., of a lotic system) rather than expressing extended spatial distribution (4.3.3); b) *reactivity* should be implemented responding to interrupts via *exceptions*, not by frequent *polling* (6.3.2); c) *uncertainty* should be dealt with using only mechanisms implemented in *Windows32* API [19] (6.3.1, 6.3.2).

4.1.3. Negotiable and Adjustable Criteria

Unlike premises, they are both *negotiable* (with the end-user) and *adjustable* (during the design process). Here they embody the flexible traits of the robot-portrait (3.2.4) starting with the most concerning features for ecologists: system intrinsic *complexity* (4.2.1), *reflected by the model* (most models are disappointing, 4.2.2). However, the degree of chaoplexity is set up by three factors depending on user acceptance: a) manageable user-perceived model (cognitive) complexity; b) satisfactory reduced (structural) complexity (via bounded rationality); c) tolerable deadline for ("Just-in-Time") model implementation. (Of course, the last two are strongly related – a leitmotif of [2], 4.3.)

- A) At least the following facets of chaoplexity should be significantly illustrated by a CybMd: a) a *negative* feedback loop furthering *homeostasis* (4.2.3); b) corollary: since this loop cannot be conceived in atemporal space, a (Bergsonian) time dimension is unavoidable (4.2.4); c) intensely dynamic (uncertain) environment (4.2.3).

- B) As less simulation as possible when implementing system reactivity (6.3.2).

- C) As much "Scandinavian method" as possible when designing the interface (4.3.1).

- D) As simple time model as possible (5.1.2, 5.2.3).

- E) Circumstantial criterion: make the most of the available data warehouse. From *PrsU*, the criterion is essential since (because of poor modelling), several (eco)system parameters and relations – crucial for controlling its evolution – are either (at least partially) unknown or ineffectually reflected by the model (4.3.3).

4.1.4. Desiderata. Why “Proof-of-Concept” Applications?

From *PrsADT*, they are in fact (less important) criteria but they are in a separate category to avoid any possibility to interpret the choice made in 3.2.4 as “solution in search of a problem”. Moreover, the topic is unavoidable because of two desiderata: one that seems unfulfillable (but needs mentioning), and one that seems surprising (hence, needs justifying).

- *Suitable IDE*. Since non-algorithmic software is still neglected, an appropriate IDE is chronically non-existent. There is not yet even a non-integrated programming environment allowing *genuine exception* handling (3.3.2, 5.1.3, 6.3.2). However, a legitimate request should be evoked – at least – as “wishful thinking”. Likewise, the non-existence will be proved (again) before choosing alternative software solutions (6.3.1).

- *Why “PoC” Applications*? From *PrsADT*, the desideratum is sufficient for validation but not *necessary* (e.g., in “*Plan A*”, 1.1.2). On the other hand, from *PrsU*, it seems rather undesirable to reduce the scope of an application to prove concepts instead of a full scale, tailor-made application. The apparent paradox is solved by a straightforward exercise of resource management:

Some universities have proof of concept centres to "fill the 'funding gap" for "seed-stage investing" and "accelerate the commercialization of university innovations". PoC centres provide "seed funding to novel, early stage research that most often would not be funded by any other conventional source” [17]”.

In short, because of resource (un)availability it is mandatory to set up a demanding compromise between *improving* application *performance* and *deadlines*. Hence, for the sake of “Just-in-Time”, a PoC application is desirable from all perspectives.

4.2. Rationale: Fighting (Cognitive) Chaoplexity in Biologic Research

The main concerns stated in 4.1.3, are interrelated in biologic modelling: *chaoplexity* (4.2.1), *limitations of current models* (4.2.2), and *unavoidability of a* (Bergsonian) *time dimension* (because of impact, scope, novelty, and (mechanism) design consequences, it needs a separate subchapter, 5.1). Each is endorsed by a focused “*State of the Art*”. From here *assumptions for cybernetic modelling* (4.2.3) are inferred.

4.2.1. Chaoplexity, Main Hurdle in Biologic Modelling. State of the Art

In the subchapter title “*Cognitive*” is in parentheses to suggest that both other options are misleading: a) “*Fighting Cognitive Chaoplexity*” could imply that structural chaoplexity is of no concern. b) “*Fighting Chaoplexity*” could imply that structural and cognitive chaoplexity are similar (unlike in robotics, in biology structural chaoplexity is – here and now – irreducible *per se*; though, through boundedly rational modelling, it can be, at least partially, hidden, reducing thus the cognitive component).

Chaoplexity is understood as stated in [2] and employed in [5]. Thus, in the State of the Art it is referred to as “(intense) complexity”. In fact, complexity is the main motivation of CybMd: “the concept behind cybernetic modelling is the adaptation of a mathematically

simple description of a complex organism which is compensated for oversimplification by assigning an optimal control motive to its response" [41].

Though, "close examination reveals that commonly applied definitions of complexity fail to accommodate some key features of ecological systems, a fact that will limit the contribution of complex systems science to ecology. [F]eatures of ecological complexity – such as diversity, cross-scale interactions, memory and environmental variability – [...] continue to challenge classical complex systems science. Further advances in these areas will be necessary before complex systems science can be widely applied to understand the dynamics of ecological systems" [8].

As regards the chaoplexity inherent to ecologic systems, an excerpt from [5] is relevant: "ecological chaoplexity is still a challenge, despite the increasing computer power, versatile software instruments (above all for modelling and simulation), and unrelenting research efforts [...] [46], presenting a state-of-the-art of (explicit) ecological models, emphasizes "the need for developing mixed methods based on a combination of various reduction techniques to cope with the spatio-temporal complexity of real ecosystems including processes taking place on multiple time and space scales" because "the construction of analytically tractable mean-field models is becoming a key issue to provide an insight into the major mechanisms of ecosystem functioning." [46]. Telling is the focus on understanding the modelled phenomenon (implying that current modelling is far from satisfying the needs): [47] proposes a new theory: "Scale transition theory is an approach to understanding population and community dynamics in the presence of spatial or temporal variation in environmental factors or population densities. It focuses on changes in the equations for population dynamics as the scale enlarges. [...] Scale transition theory more generally, however, does not aim to have fully analytical solutions but partial analytical solutions applicable for circumstances too complex for full analytical solution. These partial solutions are intended to provide a framework for understanding of numerical solutions, simulations and field studies where key quantities can be estimated from empirical data." [47]. In [48] the criticism is even stronger: "Establishing a direct link between individual based models and the corresponding population description is a common challenge in theoretical ecology. Swarming is a prominent example, where collective effects arising from interactions of individuals are essential for the understanding of large-scale spatial population dynamics, [...] and where both levels of modeling have been often employed without establishing this connection" [48].

4.2.2. Major Limits of Current Biologic Modelling. State of the Art

Here are only topics regarding the limits of (cybernetic) modelling in biology as a whole, (caused by both chaoplexity components) focusing on *ecologic stability* (even more focused references are cited in 5.3.2):

A boundedly rational approach follows (recently, April 2013) [31]: "Ecological stability is touted as a complex and multifaceted concept, including components such as variability, resistance, resilience, persistence and robustness. Even though a complete appreciation of the effects of perturbations on ecosystems requires the simultaneous measurement of

these multiple components of stability, most ecological research has focused on one or a few of those components analysed in isolation. Here, we present a new view of ecological stability that recognises explicitly the non-independence of components of stability. This provides an approach for simplifying the concept of stability. We illustrate the concept and approach using results from a field experiment, and show that the effective dimensionality of ecological stability is considerably lower than if the various components of stability were unrelated. However, strong perturbations can modify, and even decouple, relationships among individual components of stability. Thus, perturbations not only increase the dimensionality of stability but they can also alter the relationships among components of stability in different ways. [T]he multidimensional stability framework that we propose gives a far richer understanding of how communities respond to perturbations" [31].

To put it in its larger ecologic context, the *Stanford Encyclopedia of Philosophy* [21] is the best source; hence, a long quote: "As with population ecology, what is of most interest [...] are the changes in a community over time. This brings us to one of the most interesting – and one of the most vexed – questions of ecology: the relationship between diversity and stability. A deeply rooted intuition among ecologists has been that diversity begets stability. If this claim is true, it has significant consequences for biodiversity conservation. [...] [A]lmost every community experiences significant disturbances. With this in mind, stability has been variously explicated using a system's response to disturbances or its tendency not to change beyond specified limits even in the absence of disturbance. [...] How do any of these measures of stability relate to diversity? The only honest answer is that no one is sure. If diversity is interpreted as richness, traditionally, it was commonly assumed that diversity is positively correlated with at least persistence. However, there was never much hard evidence supporting this assumption. If stability is interpreted as a return to equilibrium, mathematical models that should answer questions about stability are easy to construct but hard to analyze unless the system is already close to equilibrium. [...]. The most systematic analyses performed so far give no straightforward positive correlation. […] It was once believed that natural ecosystems are usually at equilibrium (the "balance of nature"). [...] But ample empirical data now suggests that this assumption is almost never correct: natural ecosystems are usually far from equilibrium […] Moreover, if natural selection between species occurs during the transition to equilibrium, equilibrium communities will be less rich than those that are yet to reach equilibrium. On short time scales (short enough to make speciation unlikely), selection between species that utilize the same resources (that is, they occupy the same "niche") will lead to the exclusion of the less fit by the more fit through "competitive exclusion." […] The eventual equilibrium community, one in which selection would no longer be acting, the (controversially) socalled "climax community," is necessarily less rich than those that temporally preceded it" [21].

4.2.3. Assumptions for Cybernetic Modelling

From the general principles of cybernetics [25] – and supported by the wide-ranging, twofold, "State of the Art" above as well as those in 6.1 and 6.4 – the transdisciplinary assumptions below are inferred, aiming at CybMd of living systems (some of them are adapted from [14]):

- A) Any *subsystem* is a *system* but the two concepts are not synonymous; in practice, when systems are too complex, subsystems should be taken into account.

- B) Ecologic systems are – at least macroscopically – *nondeterministic* and *open* (i.e., even statistical determinism is irrelevant).

- C) "Ecologic systems are chaoplex. Chaoplexity is both cognitive (main parameters, processes, and relations are unknown) and structural (there are very many species and environment features)" [14].

- D) "The key relation between diversity and stability is chaoplex and is not yet suitably modelled (ecologic stability can refer to several attributes: resilience, persistence, etc.)" [14]

-.E) Processes cannot be modelled *atemporally*. Processes within a living system require *irreversible* time – at most simulated through closed *reversible* time [14]. (As foundation of non-algorithmic CybMd, it is elaborated upon in 5.1.)

- F) Cybernetic perspective*:* Homeostasis (as key species of *stability*) is neither *fixity* nor *thermodynamic equilibrium*; it is outcome of a (negative *feedback*) *process*.

- G) Biologic perspective: Homeostasis (as key species of *preservation*) is the *state* of a *living system* resulting from an *adaptation process* triggered by a *perturbation* in its *dynamic* and *uncertain* environment [14].

- H) Uncertainty cannot be effectively modelled presuming that future evolution will merely extrapolate the past (for instance, using time series).

- I) Predictive models predict *synchronically* not *diachronically*. "Such models are useful for *diagnosis* not for *prognosis*" [14] (6.2.2).

- J) To *predict* evolution, *temporal information* is unavoidable [14] (4.3.3, 6.2.2).

In spite of this chain of restrictions, the assumptions above allowed setting up the operational CybMd of *homeostasis with hysteretic delay* in the next chapters.

4.3. "Just-in-Time" Approach: Modelling as Workhorse Tool in Research

Modelling homeostasis endorsed all three ways "Just-in-Time" is reflected in software design: a) beneficial side-effect of bounded rationality (*anthropocentric multifunctional analog (fuzzy) interface*, 4.3.1); b) crucial for the synergystic effect of merging it with bounded rationality "BR + JIT" (*microchronic CybMd* is *sine qua non in biologic research*, 4.3.2); c) in its own rights, as (post-industrial) approach to the Toyota production system [2] (*from spatial data to temporal information*, the *undesired example*, 4.3.3).

4.3.1. Anthropocentric Multifunctional Analog (Fuzzy) Interface

As underlined in [2], for the end user there is no application, there are only *features* accessible *through the interface*. Thus, here the user perspective is considered first:

- *PrsU*. The interface *is*, in fact, the very appliance. (Its importance grew after the failure of *Plan A*, since it became *workhorse tool.* That is why the "Scandinavian method" is so useful to realise user-centeredness, 4.1.3, 4.1.1.)

- *PrsADT*. The interface should validate the first mechanism proposed in [2], "multi-functional input bar, based on fuzziness, computing with words, cognitive psychology, non-algorithmic software and semantic validation". (An interface that is not perceived as "anthropocentric enough" is unable to hide structural chaoplexity.) By serendipity, this bar proves also another thesis idea: "Precision is useless".

- *PrsCSIT*. The meaning of anthropocentric interfacing is condensed in a chain of implications: reducing chaoplexity → proficient human-computer interaction → user-friendliness → human-oriented (symbolic) communication → (predominantly) analog input and numeric output → multimodality → dialog-oriented interface.

"*Dialog-oriented*" means supporting genuine dialog, where both interlocutors can intervene as needed *any time* during the interaction. That means *initiating interrupts*: a) *Users* must be able to reconfigure scenario executions, to fine-tune the performance of *any function(ality)* mirrored in the interface. b) *Environment* factors (at least the essential and/or the most dynamic ones) must be able to act as *stimuli* during software execution, to inform or alert the user in real time about the latest changes – no matter if they are real (e.g., ecologic alert) or virtual (e.g., system response to a modified scenario). In other words, to get in due time an answer to the "What-if" question.

In nuce, the interface conveys the inherent need for *"Just in Time" interactivity* – i.e., the environment *too* should have the possibility to *initiate dialog*. Because that can happen *at any moment* – mainly in emergencies – it requires *non-algorithmic* software. (How this dialog is reflected in the appliance, is shown for architecture in 6.2 and for the interface itself in 6.3 and 6.4.)

4.3.2. Microchronic Cybernetic Modelling, *Sine Qua Non* in Biologic Research

The syntagm "*microchronic modelling*" is still not at all used (Google search, April 9, 2013) while the term "*microchronic*" is yet very rare in scholarly papers; the last mention was in 2010 in the phrase "*micro-chronic delay*" in musical context. (Paradoxically, the concept was firstly applied very far from engineering, namely in Mircea Eliade's categories of – *microchronic*, mesochronic, and macrochronic – *time*.)

In CybMd a *microchronic approach* is unavoidable because cybernetics as science was built on the concept of *negative feedback*. In its simplest meaning, "*feedback*" describes the process of taking a fragment of a system output and *feeding* it *back* into the system input. "At a first glance it seems puzzling to look for a process consisting in taking a system's output magnitude, processing it [...], and feeding it back into the system together with the initial input so that the new output value should diminish. [...] The rationale emerges not in cybernetics, where it is fundamental, but – decades before

Wiener created this science [25] – in radio engineering/electronics (where it comes from), to reduce the fading effect in radio wave propagation, i.e., to counteract an external perturbation" [6].

"The strangeness or even illusoriness of a total instantaneity suggests a kind of time travelling" [14]. Indeed, if it is straightforward to neglect "macrochronically" the time needed by a input magnitude to be processed by the system – namely to consider that the system output is quasi-synchronous with its input –, it is weird to regard that a fraction of this *output* is combined with the system's *own input* and "is even capable to influence the amplification process as a whole (i.e., to consider that "signal *cause*" and "signal *effect*" can be mixed up instantaneously). This perspective becomes totally inacceptable in ecology (where **Δt** represents usually the duration of a generation of a species)" [14]. Expressions like "the same input value should be considered, both after and before feeding back" are unable to make atemporality more palatable.

In short, for modelling living systems: a) GST as *Lingua Franca* is necessary but not anymore sufficient: the conceptual framework of cybernetics itself becomes *sine qua non*. b) Atemporal models are out of question.

Though, the problem of what kind of time should be used – i.e., setting up a model of time itself –, is so consequential for modelling the stability of living systems, that it requires a detailed investigation, together with the mechanisms that implement it (5.1, 5.4).

4.3.3. From *Spatial* Data To *Temporal* Information. (Undesired) Example

The emergency solution trying to "metamorphose" (existing) valuable spatial data into (badly needed but yet lacking) temporal information (part of "*Plan B1*", 1.1.3) is outlined:

The central idea of the spatio-temporal information metamorphose is to transform the spatial data (acquired through *expanded* measurements) into temporal information (to be acquired taking *repeated* measurements in the same locations). This is possible when employing predictive models where they are aimed at, i.e. to predict measurement results in all locations taking measurements only in a (small yet statistically relevant) *s*ubset of them. The results in the complementary subset are obtained through statistic extrapolation, saving thus costly time. Next, measurements need to be taken only in the small subset (getting thus a minimal temporal information about system evolution during the time interval between successive measurements).

To carry out the metamorphose, the six milestones on the road from spatial data to temporal information are listed below as steps (the first three must be carried out sequentially, but the other can be performed in parallel):

1. Reassessing the data inventory of spatial distribution.
2. Taking out a *r*elevant *s*mall *s*ubset of *l*ocations (*RSSL*) with a relevant cardinal.
3. Taking the same measurements in *RSSL* as before in all locations.
4. Selecting a predictive model able to predict measurement results in the complementary subset of *RSSL*, based on step 3.

5. Setting up a reasonably repetition frequency *F* for step 3, depending on the results of steps 2 and 4. “Reasonably” means compromise between *N*, available personnel, and deadlines. (Within *Plan B*, *N = 6*, *F = 1/month*).

6. Repeating step 3, *N-1* times.

FIFTH CHAPTER: NESTED MECHANISMS

Mechanisms and Time Species for Modelling Stability

The chapter organization follows, from *PrsADT*, the steps in Figure 4.1, emphasising the increasing mechanism complexity – mainly as regards the key problem of modelling time itself (4.1.1, 4.3.2). The first engineering use of negative feedback was *to cut perturbation in simple systems*, without needing to care about time (5.1). Next, the key non-algorithmic mechanisms built particularly for homeostasis can be specified: *ARC*, for *microchronic modelling of simplest biologic subsystems* (5.2) and the outer mechanism *ARCH* (*H* from hysteresis) that adds a crucial *tool to counter anthropogenic disturbance* (5.3). Since hysteresis requires a second time model, *WiTi*, a *newcomer in microchronic CybMd*, could be tried to see if it helps correlating different times (5.4).

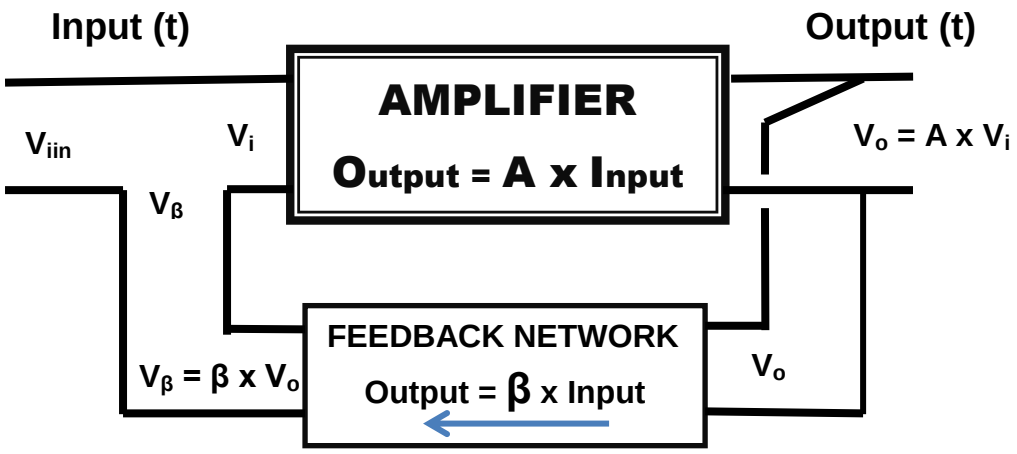
5.1. Macrochronic Modelling to Cut Perturbation in Simple Systems

The title needs explanation. The terms mean here: a) "*Macrochronic*": time can be ignored – even as NeTi – when speed is so high that instantaneity seems normal. b) "*Cut*": "reducing the effect" not "removing the cause". c) "*Simple*": formed by an amplifier and a reaction network, both linear. Since instantaneity is unacceptable, *Barkhausen's* approach is *revisited*, as foundation for shifting *from atemporality to discrete time* (5.1.1). To this end, *DOMINO* is used *as inner doll*, taking advantage of its *discretisation potential* to model *stability of* simple *linear systems in discrete time* (5.1.2). Thus, incorporated in more complex mechanisms (*ARC*, *ARCH*), *DOMINO* was *adapted* for scenarios, where the third *IF* value is used *to answer* key *"What-if" questions* (5.1.3).

5.1.1. Barkhausen Revisited: From Atemporality to Discrete Time

As start point for CybMd, it is useful to rediscover Barkhausen's relation for system (in)-stability based on (negative) feedback when radio took off in the '20ies (Figure 5.1. This figure, together with Figure 5.2, "tries to "recapitulate transdisciplinary" the conceptual progress from *technologic solution* to *focal principle* of *cybernetic modelling*" [14].)

Figure 5.1 shows the quadripolar structure of a typical negative feedback loop in early electronic equipment. At first it aimed just to counteract fading in early radio receivers; later emphasis was put on reducing noise and distortions in the output signal $\mathbf{V_o}$ – not to reduce perturbation in the input signal $\mathbf{V_i}$.



$A_{fb} = A/(1 - \beta*A)$ *iff* $\Delta t \to 0$ (macrochronic view).

Figure 5.1. Barkhausen: β < 0 to counteract perturbations (adapted from [14])

The initial **I**(nput) magnitude value, $\mathbf{V_{iin}}$ is amplified **A** times, where **A** denotes the *amplification factor*. From the **O**(utput) magnitude value $\mathbf{V_o}$ a fraction $\mathbf{V_{fb}}$ /$\mathbf{V_o}$ is transferred through the feedback network having a transfer factor **β** = $\mathbf{V_β}$ /$\mathbf{V_o}$. (To suggest that at first it was electric tension, advantage was taken of the fact that "*v*alue" and "*v*oltage" have the same initials.) Thus, in the presence of feedback, the input value becomes $\mathbf{V_{ifb} = V_{iin} + V_β}$. However, to assess the feedback effect, the same input value should be considered, both after and before feeding back, namely the *feedback amplification factor* $\mathbf{A_{fb}}$ refers to the initial input value, $\mathbf{A_{fb} = V_o/V_{iin}}$.

In this context, the feedback amplification is

$\mathbf{A_{fb} = V_o/V_{iin} = V_o /(V_{ifb} - V_β) = V_o /(V_{ifb} - β*V_o) = A* V_i/(V_i - β*A*V_i) = A/(1 - β*A)}$

Since the factor "**1 - β*A**" changes the amplification value – reflecting thus the *feedback effect* – it deserves more attention: the sign before **β*A** governs the whole process. As shown in Figure 5.1, the feedback tension is added to the initial input before entering the amplifier: $\mathbf{V_{ifb} = V_{iin} + β*V_o}$ (as shown); Thus, when feedback is *positive* (as considered above, **β > 0**), $\mathbf{A_{fb} > A}$. To apply *negative* feedback, $\mathbf{V_{fb}}$ should be in antiphase with $\mathbf{V_o}$. Then $\mathbf{A_{fb} = A/[1 – (-|β|)*A] = A/(1 + |β|*A),}$ hence less than **A**.

5.1.2. Stability of Linear Systems in Discrete Time. DOMINO as Inner Doll

Despite not being original contribution – neither in [2] nor in *ADT* – *DOMINO* [12] is of utmost significance for enabling a simple shift from *macro* to *micro*chronic modelling via *discrete time*. To this end, the third *IF* exit is used rather for its *discretisation potential* due to the "*time span given"*, than to its ability to handle uncertainty through "*UNDECIDABLE*" in *that time span*. (This rich semantics is employed in WISC, 5.1.3.) Hence, why *DOMINO* as innermost doll?

As shown (3.2, 4.1.3, 4.2, 4.3.1) chaoplexity is a major drawback in modelling living systems through both its components. Since a temporal dimension is essential, to reduce

cognitive chaoplexity, the time model should be the simplest possible able to deal with the – hardly reducible – *structural* chaoplexity. This is obviously *discrete* time. (As attested by the history of cinematography, the first *movies* were just a sequence of *pictures*.)

However, not any sequence of distinct moments in conventional (Newtonian) time, can be used in CybMd because it cannot mirror *irreversibility* – even, in an oversimplified manner. Perhaps, the simplest common form of irreversibility is: "something happened". (Unfortunately, in biology the possibility to "*Undo*" – i.e., to reduce the effects of the happening to nonexistence – is extremely rare in untrivial situations, 2.1.2.)

DOMINO satisfies also this minimal requirement for irreversibility through its *non-algorithmic TIMEOUT*: the *WAIT* is terminated either by an *exogenic environment intervention* or by an *exception*. Both are irreversible "*in the time span given"* (i.e., in **DIU**; the acronym marks its main connotations: *D*elay, *I*rreversibility, *U*ncertainty). Thus, incorporated in a loop with a suitable number of iterations (for instance, in the example in 4.3.3 there are six iterations), *DOMINO* allows "taking snapshots", that give a rough idea about the complex process going on (for instance, how far is the system from a stable state).

In advanced research, *DOMINO* could be employed in sampling nontrivial processes in line with the Nyquist–Shannon–Kotelnikov theorem (e.g., effects of severe pollution). Yet that kind of problem is totally outside the scope of *ADT* since it does not require non-algorithmic software – unlike most processes in ecologic systems, that are neither stationary, nor stochastic.

5.1.3. DOMINO: Using Trivalent *IF* to answer questions in "What-if" scenarios

Since the prospect to employ *DOMINO* in WISC – changing its architectural role from supporting procrastination to prompting reactivity – emerged (almost) through serendipity, it should be carefully scrutinised from all perspectives:

- *PrsADT*. As based on *exception handling*, it embodies the very idea of *non-algorithmic software*. Thus, its validation is crucial for complying with [1]. Moreover, if it is validated within the outermost Matryoshka (namely incorporated, with significantly enriched semantics, in the *ARCH* mechanism), then, *a fortiori*, are validated both its initial meaning (trivalent logic semantics implemented via the bivalent logic of *Windows API*, [12]) and its use in [2] (explicitly decision-oriented procrastination mechanism).

- *PrsCSIT*. The semantic enrichment refers to the possible interpretations of the third output value of *IF*. Although keeping its basic meaning (2.2.1), there are (at least) two other possible employments, distinct from the procrastination-oriented "Decision can(not) be postponed": a) "*Measure X is mandatory*" in a boundedly rational (very) simplified deontic logic (moreover, rather a kind of "*obligatoriness*" in von Wright's initial system). b) "*Risk surpasses the limit value*" in the directions proposed in [27] "to quantify the risk management effort necessary [...] to address the uncertain, imprecise, and multi-dimensional nature of the interactions between system elements" [27].

- *PrsU*. The new semantics of the trivalent *IF*, can be deemed as useful only if their potential is reflected in the interface allowing user to devise WISC, where the third *IF* value

is used to answer "What-if" questions. For instance, "*Measure X is mandatory*" is relevant in *normative* scenarios (for reducing pollution), while "*Risk surpasses the limit value*" is vital in *descriptive* scenarios (for ecologic warning).

5.2. ***ARC*: Microchronic Modelling of Simplest Biologic Subsystems**

As elemental entity for modelling negative feedback, the mechanism has two parts: ***A**mplifier* and ***R**eaction network* that must be ***C**oordinated*. Thus: ***amplification** is natural, yet nature is chaoplex* (5.2 1); ***reaction**: negative feedback is vital for stability in living systems* (5.2.2); ***coordination**: homeostasis of oversimplified living systems* (5.2.3).

5.2.1. ***A**mplification*: Is Natural. Yet, Nature is Chaoplex**

Here "*natural*" has almost all its undertones: the amplification process is *as usual*: the system outputs a number of entities (for instance, *individuals* of a benthic community) **A** times greater than the number of entities at input. When "*amplification*" is perceived as a transdisciplinary concept applicable to ecology, namely quasi-synonymous to "*growth*", it is also *organic*, *biologic*, *inborn*, *instinctive*, and *unprocessed*. Nevertheless, to be operational in a CybMd mechanism for stability (not particularised to ecologic stability), it must be investigated not only in its – artificial? – radio engineering hypostasis (5.1.1) but also in its biologic meaning – still keeping a "CybMd mindset" (4.3.2) and based on its widespread use in electronics and electroacoustics.

From *PrsADT*, it suffices to regard an amplifier (electronic, biologic, ecologic, etc.) as a quadripole having associated a real number **A**, meaning that the output value y of the amplified magnitude will be **A** times greater than its input value x. (For the sake of simplicity, potentiometers, i.e., simple passive electric circuits composed by a resistor and a cursor, are considered amplifiers with **A** ≤ 1.)

***Theorem*.** The set of *cybernetic* amplifiers $\boldsymbol{CA}$ together with the operation of connecting in *s*eries $\boldsymbol{S}$, forms a group.

Proof. To prove that ($\boldsymbol{CA}$, $\boldsymbol{S}$) has a group structure, i.e.,.$\forall\, x, y \in \boldsymbol{CA} : x\boldsymbol{S}y \in \boldsymbol{CA}$, it suffices to show that the four group axioms are satisfied:

- *Closure*. The result of the operation $x\boldsymbol{S}y$ gives another amplifier obtained connecting, x and y in series. Since the input of y is the output of x, the output value of y will be $\mathbf{A_y}$ times greater than its input value, namely the output value of x, which is $\mathbf{A_x}$ times greater than the input value of x Thus, the compound amplification, namely the output value of y divided by the input value of x will be the product of $\mathbf{A_y}$ and $\mathbf{A_x}$. But $\mathbf{A_y} * \mathbf{A_x} \in \mathbb{R}$. Hence, $x\boldsymbol{S}y \in \boldsymbol{CA}$. (In practice that means that whatever amplification (**A** > 1) or attenuation (**A** < 1) is needed, it is attainable with at most two components.)

- *Associativity*. Connecting in series is obviously *physically* associative; hence, $(x\boldsymbol{S}y)\boldsymbol{S}z = x\boldsymbol{S}(y\boldsymbol{S}z)$ (To prove the fact that such a group is abelian, multiplicative and so on, is outside *PrsADT*. Anyhow, the proofs are trivial, similar to this one.)

- *Identity*. Carrying on this plainness approach, it is obvious that the identity element, $\mathbf{1}_{CA}$, is a pair of wires or – to give it a minimal electrical embodiment – a resistor or any kind of passive circuit letting the input value unchanged (**A** = 1). Of course, for any amplifier x the output value will remain unchanged (and equal to the input value), i.e., the equality chain $\forall\, x \in \boldsymbol{CA}{:}\mathbf{1}_{CA}\, \boldsymbol{S} x = x\boldsymbol{S}\mathbf{1}_{CA} = x$ holds.

- *Invertibility*. Likewise, for each amplifier x, having an amplification factor of $\mathbf{A_x}$, there is another amplifier y (its *inverse element*, x^{-1}), having an amplification factor of $\mathbf{A_y} = 1/\mathbf{A_x}$. Hence, the equality chain $\forall\, x, y \in \boldsymbol{CA} :\ x\boldsymbol{S}x^{-1} = x^{-1}\boldsymbol{S}\, x = \mathbf{1}_{CA}$ holds.

The need for the unusual theorem above has to be justified, because it is not employed immediately neither in the nested mechanisms (4.1.1) nor in the appliance (Chapter 6):

- As shown above (4.1.1, 4.2.1, 4.2.3, 4.3.2) and reinforced below (in the rest of this chapter and in the next one), CybMd cannot rely on the unacceptable reductionist approach, considering that physical feedback loops could be modelled as linear just assembling a sufficiently great number of such simple "$\boldsymbol{CA}$ pairs".

- *A fortiori*, for living systems there is no feedback loop reducible to linear combinations of $\boldsymbol{CA}$ pairs (even nonlinearity is function of time).

- *A fortiori*, even macrochronic modelling of homeostasis in oversimplified living systems is too chaoplex for such linear combinations.

- Hence, to legitimate the use of **A** and **β**, in both nested mechanisms and toolkit appliance, the theorem above is necessary, since it proves that $\boldsymbol{CA}$ pairs are able to model oversimplified living systems, since, as group, they allow finding homomorphic projections in real-world feedback loops (albeit in very simple biologic systems).

- On the other hand, from *PrsCSIT* it proves that CybMd can apply tractable mathematics – provided it is not old-fashioned.

5.2.2. *Reaction*: (Negative) Feedback Is Vital for Stability in Living Systems

While experiencing amplification *is* natural – at least as regards the main fields governing our sensations (*electromagnetic* and *acoustic*, respectively) – *reaction* had to be realised cognitively. Even the section title sounds confusing, without explanation: "Choosing GST as *Lingua Franca* [...] was much more than a linguistic bridge [...] because it is able to express in a transdisciplinarily palatable jargon the vital relationship between cognition and cybernetic, automatic, or intentional systems" [2]. However, not even GST terminology suffices always. (This is a key reason for including a memetic approach, 5.4.4.)

After clarifying "*feedback*" as both *process* and *mechanism*, for improving communication with ecologists, [6] goes on in transdisciplinary speech: ""*feedback*"" was adopted for *cybernetic* systems from electronics (as forerunner of *automatic* systems), but for *intentional* systems the much older (and mentally more nearby) term "*reaction*" is kept for three reasons" [6]. The reasons (2.3.2) are further qualified for living systems: "a) [...] is a most conspicuous *sign of life*; b) it evokes *resistance* to environment *hostility* (a living

system *counteracts*); c) [...] evokes also *persistence*, *continuance* through *stability* (here the undertone is to react to any specific time model applied, not necessarily *antagonism*)" [6].

In addition, after explaining that "*negative* feedback tends to keep parameter values, is corrective, conservative [whereas] *positive* feedback tends to modify parameter values, is evolutive, innovative" [6], the same glossary entry contrasts the fundamental *positive* role of *negative* feedback in *living* systems (it "promotes symmetry, stationariness, stability, reversibility") with the huge dangers linked to *positive* feedback in *all kind of* systems (it "promotes chain reactions, system increase (perhaps catastrophic, leading to system annihilation), instability, irreversibility") [6].

Consequently, negative feedback is vital for stability in *all kind of* systems, at any level of organization from simple automata to robots, or from cells to benthic communities. So, why is "negative" weakened by putting it in parentheses? Because ecologic systems need positive feedback too as trigger of ontogenesis and generator of biologic growth. Therefore, at least to compensate irreversible losses – even at subcellular level –, both feedback species are vital. (Nevertheless, positive feedback is outside the scope of ADT.)

In short: a) negative feedback networks can be seen as an amplifier with $\mathbf{A} \leq 1$ (technologically, for the *ARC* mechanism it can be thought of as a potentiometer); b) for CybMd it suffices only "within an outer doll".

Even so it is a challenge to *coordinate* it with an equally oversimplified amplifier.

5.2.3. *Coordination*: Stability (Homeostasis) of Oversimplified Living Systems

Considering only Figure 5.1, this section about *coordinating* two circuits connected in series is useless. Its role is highlighted by Figure 5.2 that "shows the block diagram for feedback in a cybernetic system reduced to essentials" [14]. The new element in Figure 5.2 is time itself: **Δt** (the interval between two successive discrete time moments, 5.1.2). Here, in line with the microchronic view "the input value "*before* feeding back" would be seen in the $\boldsymbol{n}^{\textbf{-th}}$ moment, while the value "*after* feeding back" would be seen in the $(\boldsymbol{n+1})^{\textbf{-th}}$ moment.). More generally, for any process where time cannot be neglected the feedback loop should reflect the two successive time moments: $\mathbf{V_{ipr}}$, (*pr*, from *preceding* or *previous*, was the input value *before* being amplified) and $\mathbf{V_{ic}}$ (*c*, from *current,* is the value that enters the amplifier *now*). Referring to discrete time moments, $\mathbf{V_{ic} = V_i}$ at $\mathbf{t_n}$ and $\mathbf{V_{ipr} = V_i}$ at $\mathbf{t_{n-1}}$" [14].

$$\mathbf{V_{ic} = V_{iin} + V_\beta = V_{iin} + \beta * V_o = V_{iin} + \beta * A * V_{ipr}}.$$

Hence, the basic *microchronic feedback* formula is: $\mathbf{V_{iin} = V_{ic} - \beta * A * V_{ipr}}$

Thus, *because* **Δt** cannot be overseen anymore, the *microchronic* amplification is:

$$\mathbf{A = V_o / V_{iin} = V_o /(V_{ic} - \beta * A * V_{ipr}) > 0}$$

Of course, *macrochronic*, $\mathbf{V_{ic} = V_{ipr} = V}$i; thus:

$\mathbf{A = V_o / V_{iin} = V_o / (Vi*(1 - \beta*A)) = A / (1 - \beta*A)}$ (regaining the Barkhausen formula, 5.1.1).

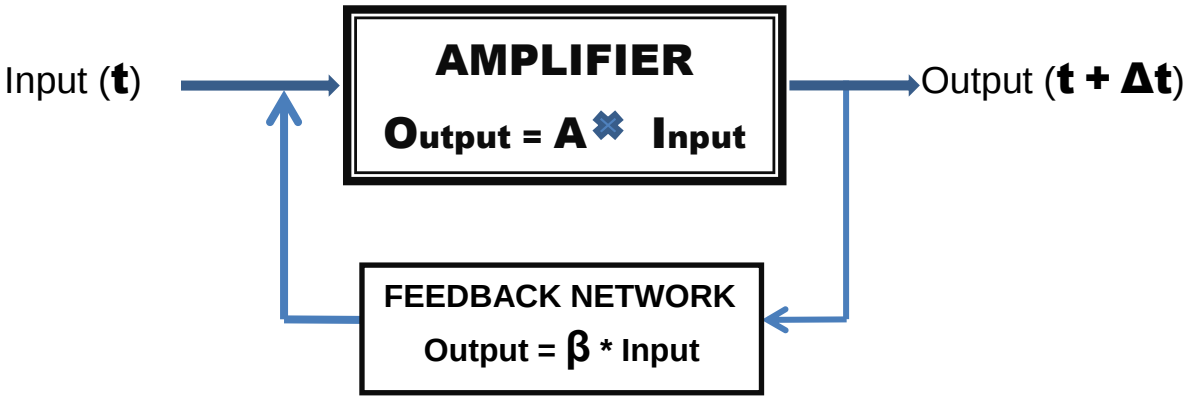


Figure 5.2. Wiener: β < 0 to enable homeostasis (stability) in cybernetic systems (simplified from [14])

"The role of an engineering approach is to elucidate the mechanisms (in particular feedback interactions) responsible for such observed phenomena" [24]. Unfortunately, for modelling complex living systems "elucidating" is not enough. Indeed, apparently, infinite speed should be rather easier to conceive mentally than to meet in engineering practice. Although harder to assimilate cognitively, in electronics – due to the great speed of signal propagation – feedback mechanisms can mostly afford to sidestep an explicit temporal dimension. Yet such circumvention is impossible when modelling processes in living systems. Homeostasis is no exception.

5.3. *ARCH*: Hysteresis as Tool to Counter Anthropogenic Disturbance

As outermost doll, *ARCH* must prove that it conveys the functionality of all the mechanisms it contains and their potential to model homeostasis as well as to answer "What-if questions. Thus, it is scrutinised as *transdisciplinary ARCH from Claude Bernard to system biology* (5.3.1). Likewise, as *engineering ARCH* its becoming is examined from the necessity of *intractable equations* in conventional homeostasis models to the sufficiency of *simple FOR loops* in its current implementation (5.3.2). Finally, its key add-on to CybMd is outlined: the way nature is *adapting* ecosystems to exogenic perturbation is illustrated via a most worrying threat: *unpredictable anthropogenic disturbance* (5.3.3).

5.3.1. Transdisciplinary *ARCH*: From Claude Bernard to System Biology

Claude Bernard discovered the ancestor concept of homeostasis: "La fixité du milieu intérieur est la condition d'une vie libre et indépendante [...] This independence […] derives from the fact that in the living being, the tissues are […] protected by a veritable internal environment" [en.wikipedia.org/wiki/Claude_Bernard].

After a century, Wiener [26] explained the "*how*" of the "*what*" intuited by Bernard. However, the conceptual debate about this kind of topic is not even now settled, as shown by a recent comprehensive study carried out by ESF (*E*uropean *S*cience *F*oundation)

having the relevant title "*FORWARD LOOK Systems Biology*: *a Grand Challenge for Europe*": "an important task of Systems Biology is to identify functional units (subsystems) that realise such '*dynamic motifs*', including for example 'oscillations', 'amplification', '*hysteresis'*, '*homeostasis*'. Stability and bifurcation analysis are important tools for this task. [...] Many of the *existing systems-theoretic techniques* are not well suited for *short time series*, *uncertainty in data* and for *systems involving many variables*: there is a need for *basic research to develop new methodologies*. Systems Biology is not the application of existing engineering tools to cell biology but a merger of both fields; both fields should co-evolve" [24][1].

Nevertheless, reductionists disagree: "mathematical modelling approaches have been applied to study the properties of biological systems [...] for analysing the properties of large metabolic networks and predicting the phenotypic behaviour of microorganisms [...]. Moreover, the stochastic kinetics modelling framework considering the stochastic nature of biochemical reactions has been used to predict the concentration of molecular components in the cell [...] The recent effort to build a whole-cell model has made the development of integrative modelling approaches necessary for the analysis of metabolism." [41]. Yet they are dissatisfied about the results: "In the absence of modelbased approaches, process optimisation in the bioindustry relies on extensive, and in certain cases unnecessary, experimentation". [...] Due to the lack of mechanistic information (and in many cases absence of proper modelling practice), mathematical models of biological processes are usually limited both in terms of range of validity and predictive capability. [...] Moreover, mathematical models of biological systems, generally, lack transferability to other, even similar, processes without a complete re-estimation of the model parameters. Bioprocess models usually focus on the significant process variables and their interconnectivity around specific operating conditions. [T]he primary "tools" utilised to describe the observed macroscopic behaviour are usually limited to nutrient and metabolite concentrations, a remnant from a period when those were the only readily measurable quantities analytically." [41].

Thus, after a century and a half, another quote from Bernard is still of worrying modernity: "Although the application of mathematics to every aspect of science is its ultimate goal, biology is still too complex and poorly understood. Therefore, for now the goal of medical science should be to discover all the new facts possible. Qualitative analysis must always precede quantitative analysis" [en.wikipedia.org/wiki/ Claude_Bernard].

5.3.2. Engineering *ARCH*: From Intractable Equations to Simple *FOR* Loops

As entailed by the Matryoshka approach (4.1.1), for both (transdisciplinary) *research* and (service-oriented) *engineering* reasons, *ARCH* is neither an essentially different mechanism from, nor an application of *ARC*. It is rather based on *ARC* but finalised as mechanism for PoC applications to be used in ecologic research where atemporal

[1] My italics RF.

modelling is totally unsuitable. Moreover, the syntagm “oversimplified living systems” suggests that without developing *ARC* into *ARCH*, not even the simplest predator-prey communities could be modelled acceptably. This development is outlined below.

The following assessments stem from an authority in investigating liver homeostasis: “For many chemicals, this leakage is in a steady state, or homeostasis, which is really a dynamic steady state, i.e., fluctuating regularly in time” [23]. Findings reported in [23] “suggest that dynamic models of liver tests might improve the ability to detect when the liver is transitioning away from homeostasis in the presence of variation that is currently considered clinically insignificant. A candidate for the SDE for liver homeostasis is [...]“ [23].

$$\mathrm{d}\mathbf{X}_t = -t^{-1/2}\boldsymbol{\zeta}(\mathbf{X}_t - \boldsymbol{\mu})\,\mathrm{d}t + t^{-1/4}\boldsymbol{\sigma}\,\mathrm{d}\mathbf{B}_t$$

Figure 5.3. Stochastic Differential Equation for homeostasis (snapshot from [23])

Despite its high complexity, the equation in Figure 5.3 was only partially satisfying. Positive aspects are rather general:

- *System complexity*. “There is nothing here that makes the approach restricted to liver biology. Similar homeostasis models can be developed for other combinations of clinical measurements“ [23].

- *Scope*. ”Once we have a good quantitative definition of homeostasis, the specificity problem can be attacked using pathodynamic (nonequilibrium) models of disease. Such phenotypic models may then map to specific genotypes or genotype-environment combinations and may provide a good transitional link between systems biology and clinical medicine“ [23]. (Comments in 5.4.4.)

On the other hand, the same equation is self-assessed as still unsatisfying as regards:

- *Adequacy*. “Unfortunately, this equation is not physically intuitive because the homeostatic force appears to be relative to the amount of time between observations. [...]. However, it will require further study to see if there is a formulation that leads to a solution that is only a function of *t*“ [23].

- *Solution quality*. “Better homeostasis models [...] may lead to predicting which patients are susceptible to idiosyncratic reactions, even though the actual rare event may not be amenable to such modelling“ [23].

A similar case is illustrated by the “numeric simulations” in Figure 5.4: “By developing some new analysis technique, we are able to show that the two boundary equilibria are both saddle points and the interior equilibrium is globally asymptotically stable”. The problem (a variant of applying the Lotka-Volterra model, discussed in 6.4.2) is of certain interest to ecologists, but they’re not interested in “globally asymptotically stable solutions” and in visually incomprehensible curves even less.

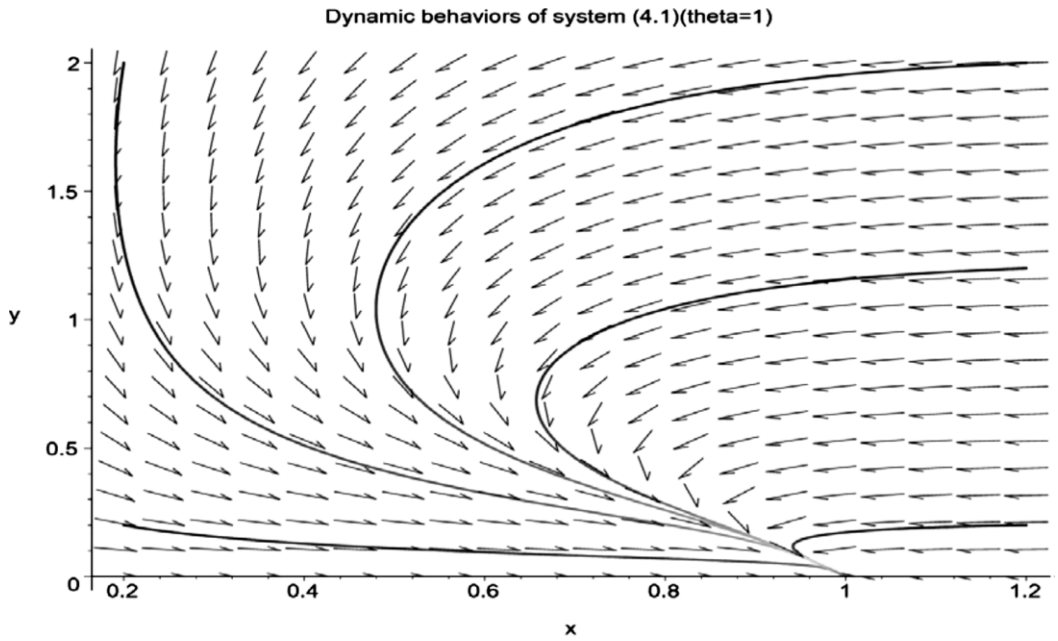


Figure 5.4. Numeric simulations with θ=1 (initial conditions (x(0),y(0))=(1.2,1.2),(0.2,2),(1.2,2),(1.2,0.2),(0.2,0.2) (snapshot from [50])

In short, from *PrsADT* both cases are considered as blatant examples of "intractable mathematics" with affordable "in house" research resources (3.3.1). Hence, the boundedly rational approach applied in *ARCH*, replacing such equations by simple *FOR* loops (5.3.3, 5.4.3, 6.2.3) is compulsory.

5.3.3. Adapting to Perturbation: Anthropogenic Disturbance Is Unpredictable

From a biologic perspective homeostasis is a key way to ensure *preservation*. The *homeostatic state* results from an *adaptation process* triggered by a *perturbation* (4.2.3). Now, the "What for" question – vital for modelling – can be answered (from *PrsADT*): the process leading to a homeostatic state should be a fitting balance between efficiency and resource consumption, since to counteract successfully perturbations negative feedback must be intense, requiring valuable additional resources. Nature always achieve *homeostasis* via *hysteretic delay*, lagging in response to perturbation (the most acceptable meaning of *hysteresis* is "*lag in response*", 5.4.4). Whether this is because it solves the same (engineering) trade-off (saving resources *genotypically*, during phylogenesis, when homeostasis emerges or *phenotypically*, delaying the reaction until it becomes inexorable) or just because adaptation cannot be achieved suddenly seems to be yet unsettled (5.4.3, 5.4.4). (Despite being outside the scope of *ADT*, the issue is paramount for devising suitable WISC. 7.3.3, 7.3.4.)

In CybMd the possibility to "*TIMEOUT*" the response (5.1.2) increases not only the effectiveness of WISC (7.3.1, 7.3.4) but adds also significant functionality to the mechanisms aimed at modelling homeostasis (5.1.2, 5.1.3). For ecologic research that is vital, considering uncontrolled anthropogenic disturbance.

As shown in Figure 5.5, beside the intended input, *perturbations* also enter the system (to suggest their upsetting effect, Chiller font was used; the subliminal message linked to this font will emerge in 5.4.4). Since they are processed similarly to the desired input, negative feedback will reduce their effect "**1 + |β|*A**" times – that means, with a sufficiently great **|β|** (almost) as much as needed. The price? More energy spent for amplification. In electronics and robotics there is also more risk regarding system stability. Paradoxically, in ecologic research such risk is irrelevant, since concern about stability is mostly the trigger for research. (Hence, the aim of saving resources fades away in face of that of preserving ecologic systems.)

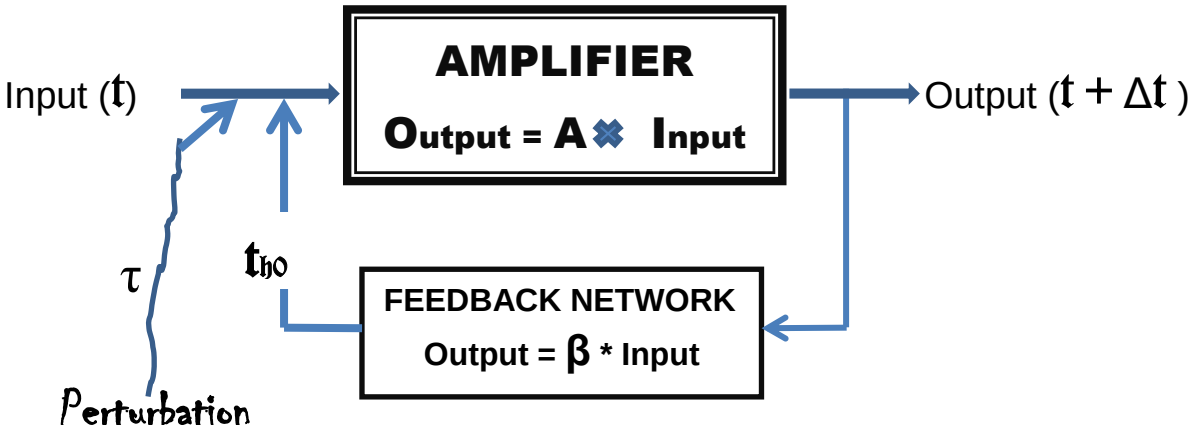


Figure 5.5. Perturbation as trigger of homeostasis with hysteretic delay in cybernetic systems (adapted from [14])

The two time species in the figure (τ for perturbation and t for the system) are fundamental from a biologic perspective, giving reason for the next sub-chapter.

5.4. Wienerian Time: A Newcomer in Microchronic Cybernetic Modelling

The need for a coherent conceptual framework for the temporal dimension of living systems goes far beyond the requirements of biologic modelling and is blatantly illustrated by the surprisingly widespread *confusion about Bergsonian time as temporal dimension in biology* (5.4.1). Next are given the *reasons to use Wienerian time in ecologic models* as placeholder for such a framework (5.4.2). The way *WiTi* is applied in *correlating distinct time dimensions* is abridged in this new light, facing the challenges of modelling homeostasis with hysteretic delay (5.4.3). Finally, the subjective experience acquired is described, approaching the issue of *time as chaoplex feature of living systems* from a *memetic engineering* stance (5.4.4).

5.4.1. Confusion about Bergsonian Time as Temporal Dimension in Biology

"*Bergsonian time*" (in the meaning given by Wiener, [25]) is so variously (mis)understood that a generally graspable model was unlikely to develop. Even when replacing the term by the more commonly used "*biologic time*" the concept has no universally accepted connotation. Confusion penetrated valuable scholarly papers[2] as illustrated by examples taken from different fields (including authorities, as the first example shows):

- *ASPET* (American Society for Pharmacology and Experimental Therapeutics): "The Dedrick approach can describe the plasma concentration-time curves in humans from the relevant profiles in animal species by *transforming the chronological time into the biologic time* based on the allometric theory" [66] (at an annual meeting of experimental biology).

- "The most commonly used *markers of internal biologic time in humans are the circadian rhythms of body temperature* and the circadian rhythms of hormones cortisol and melatonin**"** [35].

- "Testing and reasoning are two main closely related diagnostic activities. Diagnostic testing is realized in *Newtonian (short) time* while diagnostic reasoning in *(long) Bergsonian time*. *Both sort of time are used to analyze different kinds of problems*" [29].

- "*Slowing down or even „stopping" the cosmic and biologic time* represents" [38].

- "The neoclassical school uses a *Newtonian time* concept in which *time is homogeneous, mathematically discontinuous and causal inertia*, while most Austrian economists adopt Henri *Bergson's concept of time*, where *time is subjective and continuous, meaning that events are linked with each other*" [67].

- "[S]trategy studies have developed essentially around two *monotemporal approaches* (i.e., *process and content*), emphasising how they are respectively related to *two polarised monodimensional conceptions of time (i.e., Newtonian time and Bergsonian time*)" [52].

To show that such misunderstanding stems from the difficulty to discriminate between *macrochronic/static* and *microchronic/dynamic* just an example about pseudo-synonyms

First [TheFree]dictionary definitions of two well-known "*synonyms*":

Contemporaneous: Originating, existing, or happening during the same period of time […] *Contemporary* is used more often of persons, *contemporaneous* of events and facts [...] *Simultaneous* more narrowly specifies occurrence of events at the same time [...]

Synchronous: [...] Moving or operating at the same rate. [...] occurring at the same time; [...] recurring exactly together and at the same rate.

Despite the syncretic definitions and the rather fuzzy synonymity, a vague tendency towards semantic specialization is discernable: "*contemporaneous*" has a static undertone, suggesting *coexistence in time* (of beings or objects) whereas "*synchronous*" has a dynamic undertone calling to mind *processes* that are (more or less) *related in time*.

In short, all concepts about time are so logically incoherent that without systematising them, CybMd is hard to advance. What is more, the confusion originates perhaps also from software designers (not even robots need BeTi, circular time suffices) and from mathematicians (transferring *atemporality* from *numbers* to *bits* and from *bits* to *models*).

[2] Italics in the quotes are mine, RF.

5.4.2. Wienerian Time. (Sufficient?) Reasons to Use it in Ecologic Models

Paraphrasing both Poincaré and [14], the reasons to use WiTi in modelling homeostasis with hysteretic delay were *"PrsADT convenient"*. The main reason – that motivated also the implementation of WiTi in *ARCH* before [14] was accepted for publishing – was the easiness to apply it for *handling different species of time* (5.4.3). Whether or not the reasons to use WiTi in *ADT* are also sufficient to apply it more largely, to its primary purpose, as tool for (non-algorithmic) CybMd, is outside the competence and the scope of *ADT* (that explains the "*?*" in the section title).

Since *"PrsADT convenient"* means here – as well as in 5.4.3 and 6.4 – *pragmatic, boundedly rational* and "*Just in Time*", below is a fuzzy X-ray image of [14] taken from *PrsADT* stance when WiTi was yet unborn.

"Choosing this title for the first chapter of [25], Wiener highlights the weight of these two contrasting kinds of time focusing on the opposition *reversibility-irreversibility*. Irreversible time was first called "human time" by Bergson [...] and later "living matter time" by Vernadsky [...]. Recently, this time was even tighter linked to living-system complexity: "internal (or subjective) time (as well as subjective space) of a complex system is determined by the content of its memory [...]. They are produced by information processes occurring in a complex system" [18]" [14].

Skipping Vernadsky and Bergson, modelling homeostasis showed that "biologic time" has *irreversibility* as defining feature and is "linked to *living-system complexity*" [18]. For implementing *ARCH* (and validating thus all three nested mechanisms), the useful features of WiTi (as only accessible mathematic extension of NeTi) are expressed – in decreasing order of reflection in correlating different species of time – by two *P*(remises) and three *C*(riteria) from [14] (the key syntagmata/words are *italicised*):

- *P2*. "the extended time must be *physically compatible with* usual *Newtonian time*".

- *P4*. "To ease transdisciplinary research, GST must be kept as *Lingua Franca* (it [...] proved mandatory for cybernetic modelling of living systems)". (Now, the premise requires qualifying: GST is not anymore sufficient, because modelling living systems requires cybernetics, 3.2.1, 3.2.2, 4.3.2, 5.2.2.)

- *C2*. "the new time species should be not just *mathematically tractable* but also "*mathematically convenient*"".

- *C4*. "The [...] application domain should be relevant to transdisciplinary research, *living systems, chaoplex environments*".

- *C5*. ""*Irreversibility* should be [...] related to time that model *physical irreversible processes*". (Still partially implemented, 5.4.3.)

5.4.3. Wienerian Time in Modelling: Correlating Distinct Time Dimensions

Returning to the relations between *perturbation* and *system* (5.3.3), Figure 5.5 conveys that there are *two species of time* but postpones explanation until time-related issues are set up. Thus, before showing how WiTi eased correlating the two time species, it has to be proved matter-of-factly that τ (perturbation time) and t (system time) are not just

representations of two *distinct (Newtonian) time functions*, but *dissimilar temporal dimensions*. ("Matter-of-factly" means here "substantiated by the case of *homeostasis with hysteretic delay*", beyond theoretic assumptions for CybMd in 4.2.2, 4.2.3, 4.3.2.)

Indeed, without *exogenic perturbation* (entering the system) there may be *stability* (5.1.2, 5.2) – or even *thermodynamic equilibrium* (4.3.2, [23]) but there is *no homeostasis*. Certainly, *homeostasis* could be approximated by *equilibrium* but only *macrochronically*. This perspective is acceptable even in living systems *iff* processes are of explicit chemical nature and fast enough, as the successful model (4.3.2) of liver homeostasis shows [23]: "current methods of detecting *abnormalities* are crude and do not use clinical measurements efficiently. A *dynamic model* of homeostasis whose variation can be considered largely as biological randomness provides a standard for measuring changes [...] The homeostasis model should be the simplest dynamic case for describing the behaviour of clinical measurements. If models for this case are *unknown* or *unknowable*, *it is not likely that specific dynamic disease models can be constructed*. [S]uch a model exists for liver tests. It may be possible now to generalise it to the *various liver lesions*" [23]. (Some fragments are *italicised* here to impair redundancy, since they are commented upon in 5.4.4.)

Thus, in biology as a whole, current models are able to handle perturbation only if a macrochronic view is acceptable. However, even for liver lesions equations (as in Figure 5.3) solve only "the homeostasis part of the problem but not the patterns of deviation from homeostasis (pathodynamics)" [23]. (In a self-citation, [DC Trost - Pharmaceutical Sciences Encyclopedia: Drug …, 2010 - Wiley Online Library] the term is defined "Pathodynamics is a term used by the author to describe a quantitative approach to disease that includes how the biological system changes over time".)

Based on this very relevant instance of (liver functionality) perturbation, the problem can be stated for any perturbation that cannot be regarded macrochronically (i.e., for all imaginable kinds of perturbation occurring in ecologic research):

Perturbation cannot be modelled as system feature not because it is chaoplex – and sometimes even partly unknown – but because, as *trigger* of homeostasis it is the decisive factor in system ontogenesis. In short: no perturbation, no system to model microchronically (with or without hysteretic delay). Hence, if the system in Figure 5.5 is modelled evolving in the time dimension **t**, the perturbation must be modelled evolving in another time dimension $\mathfrak{T}$ *necessarily compatible* with the system time **t** because in $\mathfrak{T}$ are two moments that correspond to essential instants in **t**. (For homeostasis, **t** is a species of biologic time, but not inevitably an instance of Wienerian time, $\mathfrak{w}$. Though, since in 6.3.2 $\mathbf{Im}\,(\mathfrak{w}) = \mathfrak{t}$ was implemented, the notations of [14] are kept).

"there must be temporal accessibility relations between the Kripke worlds carrying out homeostasis with hysteretic delay. The time dimensions require temporal correlation in (at least) two instants:

- $\tau_0 > t_0$ (entailed by the *intrinsic exogenic* nature of any perturbation);

- $\tau_h = t_{h0}$ (τ_h i is the moment in Newtonian time when the system begins to counteract the perturbation, after the hysteretic delay; t_{h0} is the moment in system time when the β network is connected to the amplifier input; since there can be *only one* "big bang" moment $t_0 = t_{h0}$, when the system starts to react – delayed or not)." [14].

The challenging problem of linking the two time dimensions in the practical instance of the research-toolkit appliance architecture, where t *must be* discrete (4.1.1, 5.1.1, 5.1.2, 5.2.3) and τ *must be* irreversible (to mirror credibly unpredictable anthropogenic disturbance) is settled in 6.3.2.

5.4.4. Time as Chaoplex Feature of Living Systems. A Memetic Approach

Despite keeping on with the restraint to avoid any elaboration on memetic engineering (3.2.1, [5], [7]), there are at least eleven reasons to approach from a memetic stance *the status of time in (dealing with) living systems.* The (not strict) order is of decreased detail:

- A) The very validation of non-algorithmic software could be affected by conceptual confusion regarding the process nature of any service, in "*Service-Oriented x*" (7.1.1).

- B) *"BR + JIT"*. *Simplicity* (as *output* of bounded rationality) is ineffective if it is not accessible "Just in (*end-user subjective, irreversible*) Time" (3.2.2).

- C) *GST* is operational as *Lingua Franca* for investigating living systems only considering time-related memes specific to the disciplines involved (3.2.2).

- D) The above reason is *a fortiori* valid to support CybMd as transdisciplinary endeavour (3.3.3, 4.2.1, 4.2.3, 5.4.1, 5.4.2).

- E) The reason is even more *a fortiori* relevant when employing non-algorithmic software in CybMd (situation hardly to avoid in non-trivial cases).

- F) There are already two instances where discontent regarding a chaoplex model (Figure 5.3) have memetic explanation: F1) The misleading equivalence "pathodynamics" = "nonequilibrium" and the illusion (here and now, 7.2.1) about a "good quantitative definition of homeostasis" cannot link (*macrochronic*) clinical medicine to (*microchronic*) system biology ([23], 5.3.2). F2) Variation of a (*diachronic*) dynamic model can*NOT* be considered biological (*diachronic*) randomness; thus the pessimistic sentence "If models for this case are *unknown* or *unknowable*, *it is not likely that specific dynamic disease models can be constructed*" is reasonable ([23], 5.4.3).

- G) *"Plan A"* failed firstly because of the memetic-originated confusion about "predictive" models (6.2.1, 6.2.2).

- H) *"Plan B1"* failed firstly because the service provider was unable to persuade the end user "Just-in-Time" to bridge the memetic gap regarding the nature of homeostasis: microchronic *process* or macrochronic (equilibrium) *state* (6.2.1).

- I)”G + H” as chaoplexity-related design-space dimensions should be mirrored in the interface escaping the memetic dilemma: *clarity* (to reduce *cognitive* complexity) or *richness* (to manage *structural* complexity) via the interface (6.2.1, 6.2.4.).

- J) In fact, memetic engineering was applied in implementing the interface (6.4.3).

The core idea is that all reasons stem from the time memeplex, more precisely focused on the memes related to the generalised confusion described in 5.4.1.

In short, the only boundedly rational acceptable approach is to consider time as (obviously *chaoplex*) *feature* of *living systems*, having *irreversibility* as distinctive attribute.

The best conclusion stems from Wiener himself: "[T]he individuality of the body is that of a flame rather than that of a stone, of a form rather than as a bit of substance" [26]. Memetically, the “flame” suggests not only the *pattern* but the *process* of “burning in time” – that is to say *time* itself.

SIXTH CHAPTER: RESEARCH TOOLKIT APPLIANCE

Tool for Exploring Homeostasis in Benthic Communities

As user-oriented output of *ADT* the research toolkit should be carried out merging – as seamless as possible – all three perspectives (Figures 4.1, 4.2, 4.3) in its *rationale*, *approach*, and – consequently – in the (appliance-oriented) *state of the art* (6.1). Almost as corollary, the toolkit *architecture* is based on the *interface as versatile, multifunctional proofing tool* (6.2). As a result, the appliance *structure* underlines the *interface implementation* process carried out at two levels: A) *Implementing the design space dimensions bottom-up*, as required by nesting the mechanisms as Russian dolls (6.3). B) As research toolkit aimed at *WISC* for *benthic communities in Transylvanian lotic systems*, the *interface* must be engineered with more care than required by usual anthropocentric design (6.4).

6.1. Rationale, Approach, and State of the Art

The subchapter answers to three questions: *Why*? (*rationale*, 6.1.1), *How* to attack it? (*approach*, 6.1.2). *Where from*? (a double *State of the Art*, entailed by the approach). The first confronts the problem, focusing on *ecolinguistics* that is practically *blocked by conflicting memes* (6.1.3) while the second backs the solution, *"What-if" scenarios in managing* chaoplex *living systems* (6.1.4).

6.1.1. Rationale: Coalescing Six Objectives From Three Distinct Perspectives

For the sake of conciseness, the (interconnected) targets are concentrated into six objectives, two for each perspective (ordered in line with Figures 4.1, 4.2, 4.3: *PrsADT*, *PrsCSIT*, *PrsU*.

- *O1*. Designing a PoC (3.2.4, 4.1.4, 7.1.2) application for validating the outermost (4.1.1, 5.1.3, 5.2.2) mechanism, namely *ARCH* (4.1.1. 5.3.2, 5.4.2).

- *O2*. Developing this application as a service-oriented appliance (4.1.1, 7.1.1), having all its functionality simple accessible through the interface (3.2.4, 4.1.1), to ease the quantitative testing (3.1.3, 3.2.3) required for validation (3.1.1).

- *O3*. Choosing an application domain sufficiently challenging (3.2.4) to require models (4.3.1, 5.1.2) able to mirror all concepts, approaches (3.2.2), and mechanisms (3.2.3) of [2] to be highlighted, mainly the synergistic effect of the thesis pillars *"BR + JIT"* (3.2.2).

- *O4*. Choosing a process to be modelled manifesting both cognitive and structural complexity, able to reflect pertinently the *ADT* original contributions mainly as regards microchronic modelling (3.3.3, 4.3.2, 5.3.3, 5.4.3) of living systems as distinct subfield of CybMd (4.2.3, 5.1.1, 5.1.2, 5.2.2).

- *O5*. Carrying out a research toolkit for investigating homeostasis with hysteretic delay (4.1.1, 4.2.3, 5.2.3, 5.3.2, 5.3.3, 5.4.3) in benthic communities (3.2.4, 4.1.1, 5.2.1).

- *O6*. Using as much as possible the significant information (4.3.3) about biodiversity in Transylvanian lotic systems (3.2.4, 4.2.3).

The last objective is very sensitive because: “What confuses this question from the very beginning, is the multiplicity of possible definitions of “diversity” and “stability.” There are probably no better instances of formalization indeterminacy in any scientific context. For instance, a reasonable first attempt to define diversity would be to equate the diversity of a community to the number of species in it, that is, its species “richness.” The trouble is that there is ample reason to doubt that richness captures all that is relevant about diversity, whether or not we are interested in only its relationship to stability. [...] Stability turns out to be even more difficult to define. [...] At one extreme, stability can be defined to require that a community be truly in equilibrium: it does not change in either its composition (the abundances of every component) or in the interactions among these components. At the practical level, this definition faces the problem of vacuous scope: almost no natural community satisfies such a strict requirement of equilibrium.” [21].

The non-ortogonality of the objectives above simplifies the application architectural design space (6.2.4). The price: a requiring approach to verify that all start-vector components are *included in the* condensed *objectives* and *mirrored by the interface* (6.3, 6.4).

6.1.2. Approach: Integrating All Proofs in a Flexible, Fault-Tolerant Interface

As non-negotiable matter, the *premises* (4.1.2) call for special attention, after being compelled to give up the initial plans because of lacking temporal data. Thus, the first premise requiring that “temporal evolution should be reflected through repeated measurements” seems impossible to respect *ad litteram*, since – partially excepting initial scenario input – all successive input is obtained either by running the model or by educated guess. However, when analysing where this verbalisation comes from (“a basic feature of biologic systems is that they evolve in time” and “the evolution of biologic systems must be modelled as process”) it becomes clear that the premise is obeyed in both letter and spirit through *microchronic* modelling. The other two premises are followed straightforwardly implementing the innermost mechanism, *DOMINO* (5.1.2, 5.1.3, 6.3.2).

The *criteria* (4.1.3) have been obeyed as follows:

- A) Illustrating chaoplexity was more than obeyed, since: a) *homeostasis* was modelled including *hysteretic delay*; b) employing an original multifaceted mechanism (*ARCH*); c) the “intensely dynamic (uncertain) environment” was illustrated by its most relevant occurrence: unforeseen, intense anthropogenic disturbance (4.1.1, 5.3.3, 5.4.3).

- B) Was obeyed “as much as possible”. The only simulation is at low level implementation: because of Java syntax, “*return*” from a (virtual) exception handler was simulated through “*return -4*” (6.3.1, 6.3.2).

- C) Because of very strict user-deadlines, the Scandinavian method was practically skipped over, being replaced by (late) consultation.

- D) Was more than obeyed, since the model is both simple and highly flexible (5.1.1, 5.1.2, 5.2.3, 5.4.3).

- E) The circumstantial criterion was impossible to follow after both *PlanA* and *PlanB1* failed (4.3.1, 4.3.3).

The desideratum of PoC application is obeyed by the very existence of the appliance described below.

In short, the approach must be focused on "*E*" because: a) it is paramount from *PrsU*; b) it is the only criterion neglected so far; c) an emergency solution (part of *Plan B1*, 1.1.3) was available (as outlined in 4.3.3) d) the causes of plan failures were identified (both plan failed because of memetic related causes, 5.4.4); e) some major causes where already investigated from memetic stance in transdisciplinary research context in post-thesis work [4], [5]. Hence, the only both possible and effectual solution must start from adapting the solution in 4.3.3 to the extreme case of lacking *any temporal information*. The only two possibilities to create plausible temporal data without repeated measurements are by *microchronic modelling* or by *educated guess*. Practically, that means *WISC*.

6.1.3. Ecolinguistics Blocked by Conflicting Memes. State of the Art

The "State of the Art" below is neither independent of nor redundant with the various "State of the Art" in *ADT*. Thus, considering "*ecolinguistics*" as "*linguistic ecology*" and language as main "*memetic carrier*", taking advantage of applying in [5] memetics to ecology, via ecolinguistics, the following is an updated, very abridged summary of [5] focused on the main ambiguous – or even clashing – deeply rooted memes (except the time memeplex, already amply investigated in 5.3.1, 5.3.2, 5.4.1):

- "Protecting the environment" presents a threefold linguistic trap, imposing to opt for "*protecting*" or "*preserving*" and "*growth*" or "*development*". [D]espite their synonymy, for preserving the environment intense negative feedback would suffice but for protecting it, some positive feedback is necessary. Obviously, in "sustainable growth", the two types of feedback need a delicate balance and replacing "growth" by "development" does not change things – except increasing confusion.

- "[B]iocenosis" means a system where the relations are controlled by negative feedback able to ensure homeostasis; "ecosystem" means a system where stability is imperilled mainly because relations are altered in an uncontrolled manner by outside factors.

- [E]cologists have to identify the generators of positive feedback and to impose measures to reduce it drastically and urgent. (No lengthy quantitative, "predictive" modelling is necessary since it is irrelevant if the model output asserts that the system will blow up in a century or in two.)

- No need to refer to memetics to explain that homeostasis is good whereas avalanche is bad. [...] The nuances of "sustainable" (*per se* or in syntagm with "growth" or "development") and "ecological" (in its modern meaning of "natural" or "balanced") become clear".

- Humans are natural too. Hence, children should be taught that protecting the environment (i.e., respecting all species) does not mean to ignore the risk of being

"included as individual in the trophic chain" (e.g., by sharks when taking imprudently a bath in (*their*?) habitat, or having a crocodile as pet). Balance is unavoidable.

- Since natural processes – mainly after arbitrary human intervention – are unpredictable, ecologists have to shift from a "descriptive logic" towards a process-oriented "logic of action" [24]. That is a key motivation for using WISC, as described below.

6.1.4. "What-If" Scenarios in Managing Living Systems. State of the Art

As expected, WISC are used in situations of high complexity. To increase relevance the two 2013 instances described below are from the most chaoplex domain (military applications) and from large-scale ecology (applying landscape science to natural resource management):

- a) "One of the greatest challenges for a complex system practitioner is that the outcome will be highly context or history dependent. A challenge for military applications is that commanders may find it difficult to rely upon systems that lack a quantifiable measure of effectiveness [...] Complexity, in its purest sense, is also challenging to use because it does not always indicate what people might need to do differently in specific contexts [...] These last two concerns can be partially addressed by testing distinct scenarios numerous times, and comparing their outcomes. [...] Various methods can be used to study complex systems. Simulation using computer models is by far the preferred tool. Simulations allow a series of thought experiments to test various 'what if' scenarios" [30].

- b) "Models have been used to explore the behavior of systems, to make ex-ante assessments of policies, inputs to the planning process, and scenario creation [...] [D]ifferent types of change can be modeled and "what if" scenarios explored. The model's incorporation of "bottom up" and non-linear information enables it to *focus on questions difficult to answer with more standard methods*" [58][3].

Both examples refer to critical resource *management*, requiring "BR + "JIT". Thus, *Bounded rationality*: [30] regards the "*commander*" as "*complex system practitioner*" (and explains that Canada is not part of US because in 1775 crucial decisions made by a revolutionary american general where "ill-fated"). Likewise, both exploit WISC as alternative method. *"Just in Time"*: [58] is explicit about "*questions difficult to answer with more standard methods*".

In short, WISC are a valuable research tool *per se*, not just an emergency exit for running models without the necessary input data. As regards scenarios the appliance is aimed at, examples are hinted at in 6.4.3 and 6.4.4.

[3] My italics, RF.

6.2. Architecture: Interface as Versatile, Multifunctional Proofing Tool

The appliance architectonics is shaped by its *first* two *design*-(sub-)*space dimensions* (as research *toolkit* aimed at WISC): *predictability* and *simplicity* (6.2.1). The first two require (once more) "*State of the Art*", namely *predictability* to shed light again on *ecologic stability: what do (not) predict current models* (6.2.2) and *simplicity*, expressed through *a simple (discrete) time model for many-sided scenarios* (6.2.3). Other dimensions vital for "WISC-based-research" (e.g., *cogency*, *ecolinguistics*) are arrived at *mixing richness with clarity* in *a multifunctional, user-friendly interface* (6.2.4).

6.2.1. First Toolkit Design-Space Dimensions: *Predictability* and *Simplicity*

For the sake of brevity, design-space dimensions that are either already taken care of in the Matryoshka-kind mechanism design or common features of PoC applications, are grouped – rather loosely – in an all-embracing $\mathbf{S_{ADT}}$ (sub-)space. Therefore, the only dimensions to be considered carefully are the four defining features for a toolkit aiming at developing WISC. Thus, the appliance design space becomes a (vaguely defined) subset of the Cartesian product:

$$\mathbf{S_{appliance}} = \mathbf{S_{ADT}} \times \mathbf{S_{WISC}}$$

where $\mathbf{S_{WISC}}$ is shrunk to its main dimensions (from *PrsADT*),

$$\mathbf{S_{WISC}} = \{Predictability, Simplicity, Cogency, Ecolinguistics\}$$

Since the last two dimensions, are in fact both consequence, blend, and looked-for mirroring of all architectural features of a research toolkit, they are shaped as design-space in their own rights together with defending $\mathbf{S_{WISC}}$ (6.2.4). Thus, only the first two dimensions should be justified here (the first highlights the *cognitive* chaoplexity of the research subfield, the second its *structural* chaoplexity):

- *Predictability*. As *sine qua non* precondition for (any kind of) *preservation*, *predictability* is both the only scientific backbone of and the most widespread practical approach to *ecology*. The problem is crucial because evolution predictability is fundamental for end-users (researchers or environmental supervisors alike). Moreover, for the targeted appliance end-users, practically all they have here and now is *experimental ecology*: a huge amount of data, professionally collected and meticulously preprocessed in laboratory is available for further research waiting for fitting models to exploit its vast research potential. Unfortunately, this was impossible because of the yet unbridged memetic gap regarding *the status of time in* (dealing with) *living systems* (5.2, 5.4). The fundamental conceptual disparity is the difference between "*predictive*" (in "predictive models") and "*predictable*" (in "predictable evolution"). In the syntagm "predictive models" the connotations are confusing: "*pre*" means "in advance" but "in advance *of knowing for sure*" (because of *uncertainty*). It does not mean *necessarily* "in advance *of happening*" (because of *future contingency*).

- *Simplicity*. "Human-induced environmental changes differ from most natural changes in which they happen at a faster rate [...]. The first response of populations is usually

phenotypically plastic alterations of morphology, physiology and behaviour. This plasticity can be favourable and move the population closer to an adaptive peak in the altered environment […], or be maladaptive and move the population further from the peak and increase the risk of extinction. […]. Results show that a primary reaction is plastic alterations of behaviour, with some adjustments being adaptive while others are not. […] Human disturbances can be dramatic and resolution of the limit of flexibility and the possibility of genetic adaptation should be important targets of future research" [15]. The keywords of [15] ("contemporary adaptation; environmental deterioration; evolution; mate choice; natural selection; phenotypic plasticity") are even more telling: all refer to *processes* that occur in *irreversible* (Bergsonian) *time*. Hence, to attain simplicity able to model "phenotypically plastic alterations" (all three words epitomise irreversibility, as it is obvious when replacing them by their antonyms: "genotypically elastic states"), a discrete time model is necessary but not at all sufficient: it must fit many-sided WISC that mirror system response to *microchronically irreversible* "human-induced environmental changes" [15].

6.2.2. *Predictability*. System Stability: What Do (Not) Predict Predictive Models

"Predictive modelling is the process by which a model is created or chosen to try to best predict the probability of an outcome" [wikipedia.org/wiki/Predictive_modeling]. The application areas mentioned in this entry illustrate the meaning of "probability of an outcome". Besides the reasons above, this "State of the Art" about the limits of predictive models is crucial as persuasion exercise for at least four reasons:

- A) For researchers in ecology or evolutionary biology, predictive models are extrapolative in *space* but not in *time* because living systems are open and (macroscopically) nondeterministic *par excellence* (i.e., even statistical determinism becomes irrelevant, [14]).

- B) To defend assumption "I)" for CybMd: "predictive models predict *synchronically* (*biodiversity* seen as spatial distribution" [14] of *living systems*) "but cannot predict *diachronically* (*stability* seen as *evolution*" [14] of *living systems*).

- C) Likewise for assumption "J)": "to *predict evolution* a statistically relevant amount of *temporal information* is required" [14]. That means repeating frequently measurements carried out in a sufficiently great number of locations or the – not "Just-in-Time" – solution given as example in 4.2.3).

- D) Key corollary: lacking any temporal data (about the *real* system evolution) the only way to get an idea about a *plausible* evolution is to run credible (i.e., microchronic) CybMd (e.g., to get a virtual system response to a relevant scenario). In other words, to get in due time an answer to a "What-if" question.

From a both transdisciplinary and pragmatic perspective, the limits are best illustrated through examples of applicative research domains where predictive models are currently used, focusing first on living systems:

- "In their seminal test of island biogeography *predictions*, Kinkel et al. [...] found evidence for species turnover and equilibrium in the phyllosphere, but not for a

species-area relationship. [...] Another exception is the derivation of log-normal species-abundance curves for a pseudomonad community [...], which is not in agreement with the zero-sum multinomial curve *predicted* by Hubbell's [...] neutral theory of biodiversity. Community assembly rules aim to *predict* spatial species distributions" [64].

- "Acid mine drainage (AMD) is a global problem that may have serious human health and environmental implications. Laboratory and field tests are commonly used for predicting AMD, [...]. Furthermore, these tests are often conducted at small-scale over a short period of time. Subsequently, extrapolation of these results into large-scale setting of mine sites introduce huge uncertainties for decision-makers. This study presents machine learning techniques to develop models to predict AMD quality using historical monitoring data of a mine site" [65].

- "But given that there is no guarantee that the future will resemble the past, why is this inductive gamble so effective, and why is inductive reasoning so prevalent? We suggest the predictive power of the genome is actually only part of the story. In no small part it is also because organisms are active agents who do work on their environments, transforming its state in reliable, predictable and often homeostatic ways [49].

- From a deeper theoretical point of view, one of the "philosophically intriguing [...] problems [is] the *uniqueness* problem: ecological systems are supposed to be unique both because they are contingent historical entities, and structurally because of their complexity. By itself being a historical entity does not present any problem for scientific study [...]. However, ecological systems are supposed to have histories that result in highly contingent structures - part of their complexity - that make them special. [...] There is an important philosophical lesson here: especially when a new discipline is being formed, the structure of the phenomena – how they are distinguished and classified – are in part determined by the models used to represent them. [...] Consequently, classification is not theoretically innocent [...] The most general and uncontroversial theoretical result to date is that progressively larger populations are required for safety in the face of demographic, environmental, and random catastrophic stochasticity. Moreover, because of the structural uncertainly of these models, apparently slight differences in assumptions and techniques routinely lead to widely divergent predictions" [21].

6.2.3. *Simplicity*. A Simple (Discrete) Time Model for Many-Sided Scenarios

"Body size is associated with fundamental biological processes such as metabolism, movement, and the rate of reproduction and evolution. Although allometric principles should also influence the range of potential behavioral responses for a given organism, evidence for such large-scale and cross-taxon relationships is lacking. [...] [B]ody size affects the likelihood of attack and the costs of predator avoidance. [...] We found a weak but significant relationship between two metrics of prey size (mean species-level prey mass and mean species level predator: prey size ratio) and two of the five prey response variables: risk-induced changes in prey habitat use and prey fecundity were significantly correlated with prey body size and the predator: prey ratio. Risk-induced reductions in prey activity were positively correlated with prey mass" [55].

The research above refers *only* to a single: a) allometric component (body size); b) anti-predator behaviour component (fear-based risk management); c) predator-prey species pair; d) reduced and well-known habitat. The result of [55] was a "weak relationship between two metrics of prey size and two of the five prey response variables". It was cited not to add another instance to the scores of examples proofing domain chaoplexity given in the (sub)chapters above, but to show that the distinctive simple time model allows getting similar – albeit just plausible – results, due to blending *irreversibility* (required by "fundamental biological processes such as metabolism, movement, and the rate of reproduction and evolution") with easy *discretisation*. This substantially simplified approach is possible by investigating stability via sampling – in fact or simulated through WISC – according to the Nyquist–Shannon–Kotelnikov theorem, without becoming aware neither of the biologic process *per se*, nor of the mathematics involved to model it.

The next example [28] is even more significant because, replacing the relatively rarely used *allometry* by the universally known "*temperature* effects on predator-prey interactions caused by global warming" it addresses a crucial environmental preservation issue:

"Temperature effects on predator–prey interactions are fundamental to better understand the effects of global warming. Previous studies never considered local adaptation of both predators and prey at different latitudes, and ignored the novel population combinations of the same predator–prey species system that may arise because of northward dispersal. […] We advocate the novel common garden experimental approach using predators and prey obtained from natural temperature gradients spanning the predicted temperature increase in the northern populations as a powerful approach to gain mechanistic insights into how community modules will be affected by global warming. It can be used as a space-for-time substitution to inform how predator–prey interaction may gradually evolve to long-term warming" [28].

As regards other dimensions vital for "WISC-based-research" (for instance, *cogency*) they are arrived at *mixing richness with clarity* in *a multifunctional interface* (below).

6.2.4. Mixing Richness with Clarity: a Multifunctional, User-Friendly Interface

While "Richness" (from *PrsU*) corresponds entirely to "Multifunctional" (from *PrsADT*), "Clarity" (from *PrsU*) means not just "User-Friendly" (from *PrsADT*) – as in most customary applications – but, because the appliance is a workhorse research tool, it entails also other connotations of the polysemantic concept of "Clarity". Just a few motivations:

- a) "Simulations allow a series of thought experiments to test various 'what if' scenarios" [30]. However, thought experiments call for all connotations of "Clarity".

- b) Main (other) connotations of "Clarity" required by a research toolkit (e.g.: *expressiveness*, *persuasiveness*, *eloquence*) are strongly related to language.

- c) Ecologists are aware of the language problem. They gave even two meanings to the new subfield of ecology, that emerged from this concern: *ecolinguistics*.

- d) The many "State of the Art" sections above enhanced the assertion that cognitive chaoplexity is – at least in great part – of memetic nature.

- e) Language is a major, but indiscriminate "*memetic carrier*" (6.1.3). If a "What-if" question is distorted by a dubious meme, the answer is irrelevant – no matter if given by a person or by a scenario.

- f) Moreover, "To increase the preparedness for the unexpected it is useful to work with "what if" scenarios covering the range given [from climate models]. Results should be communicated as scenarios and not predictions" [40].

Therefore ecolinguistics was easily qualified as crucial design-space dimension of a toolkit aimed at WISC. (The idea gained weight after lacking the opportunity to apply the Scandinavian method.)

On the other hand, the real problem was not selecting suitable dimensions for $\mathbf{S_{WISC}}$ but its very existence because:

- a) It seems strange to disconnect a design-(sub)space from $\mathbf{S_{appliance}}$, bearing in mind that *anthropocentric design* was a precondition from all perspectives (4.3.1).

- b) Corollary: *anthropocentrism* is a *sine qua non* dimension of $\mathbf{S_{ADT}}$ together with other dimensions reflecting key concepts/approaches of [2], like: *bounded rationality*, *"Just-in-Time"*, *uncertainty* that should be validated (3.2.1, 3.2.2).

- c) As required by objective *O2* (6.1.1), *anthropocentrism* too should be validated by *quantitative testing* (3.1.3, 3.2.3), albeit indirectly via a key aspect (*simplicity*, 7.2.4).

- d) Moreover, most central anthropocentric design features (e.g., semantic validation of analog input) remain unnoticed because they are expressed through sliders that hide significant parts of their (multi)functionality.

In spite of the above, $\mathbf{S_{WISC}}$ was unavoidable because WISC involves a design, where architecture, structure, implementation and validation are interrelated in a way that needs cautious blending (6.3.3, 6.4.3, 7.2.4, 7.3).

6.3. Structure: Implementing Dolls Bottom-Up. Testing Only the Outer

After searching with no success for *an (integrated?) development environment* (6.3.1), the main problem was still to *emulate dynamic propagation of exceptions* (6.3.2). Next, the *workhorse nature* of the *toolkit* reveals itself in a self-reflexive manner: the interface is in fact a series of very short-lived "prototype design" loops consisting of *adding functionality*, *in six short-lived steps* (6.3.3).

6.3.1. Choosing an (Integrated?) Development Environment

The title is rather standard than actual. In fact, there were five steps:

- *Failing to find* an IDE. The last Google-Scholar-2013-search-update (May,3) gave for *"exception handling"* + *"fault tolerance"* about 83 results, all of them referring to "exception handling" in line with the prevalent paradigm, namely confusing – in both Java (semantic) spirit and (syntactic) letter –"*exception* (handling)" with *"error* (recovery)*"*. Likewise, for *"research programming languages":* about 7 results, none referring either to "*exception*

handling or to "*fault tolerance*". (What is more, the only reference to a research programming *language* meant a new *platform* (Aeolus) built still on Java.)

- *Substituting simulation through emulation*. To avoid "returning to square 1" as regards inacceptable risks when trying to validate simulated features (2.3.3, 3.1.3, 3.2.1, 3.2.2, 4.1.3, 4.3.1, 6.1.1), exceptions were emulated (6.3.2).

- *Setting up a "monocrystal approach"*. The need for such a challenging approach was heightened by lacking expected communication with end-users. It consists of an extreme variant of "Successive Prototyping", where the appliance (in this case the toolkit itself) expands continually like a silicon monocrystal. That means implementing the mechanisms bottom-up, testing always the outermost, and looping in a "cut and try" manner (programming, integrating, implementing, testing, revising, restarting, 6.3.3).

- *Assessing consequences*. Here *PrsADT* and *PrsU* are equally important. Thus, the parameters to be exhaustively tested (6.4.4) are chosen according to their relevance from both perspectives **A**, **β**, **DIU**, **Δt** (6.3.3, 6.4.3).

- *Choosing IDE pieces*. In this context, the selection was very easy, albeit not very pleasing: the most pragmatic mix was [38] (to enable emulating exception propagation low-level simulating only handler exits), [19] (to have a minimal API support for multithreading) as well as [42], [43], [44], and [45] (to enable suitable communication through and with the bit by bit developing interface).

6.3.2. Emulating Dynamic Propagation of Exceptions

"For cybernetic modelling of living systems exceptions are crucial, since *reactivity* is *a* (some authors consider "is *the most*") defining characteristic of life" [6].

To defend the way exceptions are handled in the research toolkit, four facets have to be elucidated for each of their three cardinal features (2.3.2), namely how could it be: a) supported by Windows32 API functions; b) emulated in Java; c) relevant for PoC validation (*PrsADT*); d) worthwhile for WISC (*PrsU*).

- *Unexpectedness.* a) The exception raised by "*Set Event*" is received through "*Wait*" and balanced via its two variants: with "*TIMEOUT*" (to keep asynchronicity at polling loop level) and without (to favour synchronicity). b) Java syntax implies no restriction. c) Vital for three reasons: c1) First and *sine qua non* component of non-algorithmic software (as a whole); c2) implementing simplicity through discrete time (**Δt**). c3) minimal handling of uncertainty due to future contingents (**DIU**) d) "For ecology exceptions are threefold vital, as way to ensure "Just in Time" reaction to: a) unforeseen anthropogenic disturbances (essential for preserving ecologic systems); b) exogenous stimuli (from environment); c) endogenous stimuli (from system/body/organ endoceptors, essential for homeostasis)." [6].

- *Consequence.* ""Consequence" is used here instead of its more common synonym "*importance*" to highlight its role in risk management. [...] The syntagm "*dynamic propagation*" – now perceived as pleonastic – was coined (probably in the early 70ies) when exception handling was introduced in real-time programming, to suggest both the

key architectonic role of "*propagation*" (the superior programming unit is better suited for strategic decision making, since it is more context-aware) as well as the *caveat* that the exception should be conveyed from callee to caller (i.e., *dynamically*, not in line with the *static* block structure)" [6]. a) Propagation is straightforward using "*RETURN - n*". b) Java syntax is *inacceptable* since it forbids exiting from an exception handler using "*RETURN - n*" (as allowed for any other programming unit). c) Vital for validation because of the Matryoshka architectonics: *ARCH* could not model perturbation or hysteresis in (neither in BeTi, nor in any other time) if exceptions raised by *DOMINO* would not be propagated outwards, via **DIU**. d) Likewise vital for normative scenarios on countering anthropogenic disturbance.

- *Rareness.* Is skipped over because it is not essential from any perspective.

6.3.3. Toolkit as Workhorse: Adding Functionality in Six Short-Lived Steps

Following the conclusion of 6.2.4 and the *"*monocrystal approach*"* (6.3.1), the appliance was built beginning – and ending – with the interface in ten steps, each step integrating programming (code for both the mechanism component and the interface elements necessary to test it), debugging, testing, and commenting (from all perspectives). Here *PrsADT* keeps the connotations used in describing the mechanisms to be validated (Chapter 5), *PrsCSIT* is focused on *non-algorithmic CybMd*, and *PrsU* is split in two sub-perspectives in order of increased stakeholder interest: *PrsUb/e* (*biology/ecology*, regarding the entire research undertaking, at strategic echelon) and *PrsUw* (focused on WISC for field research, at tactic echelon). According to the approach set up in 6.1.2, as substitute for applying the Scandinavian method, the last four steps are outlined in 6.4.3, from *PrsUw*. The numeric values used in the first six steps have the interpretations (from the first two *PrsU* sub-perspectives, *PrsADT* and *PrsCSIT*) below:

- **A**. *PrsUb/e*: growth; reproduction rate. *PrsADT*: amplification. *PrsCSIT*: real number.
- **β**. *PrsUb/e*: growth reduction rate. *PrsADT*: transfer factor. *PrsCSIT*: real number.
- $\mathbf{A_{fb}}$. *PrsUb/e*: growth rate. *PrsADT*: feedback amplification. *PrsCSIT*: real number.
- **Δt**. *PrsUb/e*: time span, lifespan. *PrsADT*: time granule. *PrsCSIT*: step in *FOR* loop.
- **DIU**. *PrsUb/e*: hysteretic delay. *PrsADT*: delay time. *PrsCSIT*: *WAIT* with *TIMEOUT*.

In short, the appliance models *stability of linear systems in discrete time* (from *PrsADT* and *PrsCSIT*) or *homeostasis* (*with hysteretic delay*) *of oversimplified living systems* (from *PrsU*). The six implementation steps are listed below (from *PrsADT*, details in 6.4.4, AV2):

- a) *DOMINO*. Implemented as innermost mechanism (5.1.3)
- b) *Macrochronic stability* (according to the Barkhausen relation, 5.1.1).
- c) *ARC*. Implemented as middle mechanism (5.2.3) using discrete time (5.1.2).
- d) *Exception propagation from DOMINO to ARC* (emulated in Java, 6.3.2).
- e) *ARCH*. Implemented as outermost mechanism (5.3.2, 5.3.3).
- f) *Exception propagation from DOMINO to ARCH* (emulated in Java, 6.3.2).

6.4. Interface Shaping: Benthic Species in Transylvanian Lotic Systems

The only foundation of WISC is *the Lotka-Volterra model*; however, its *strong conditions, hard to meet in real habitats* (6.4.1) require another *State of the Art* regarding the complicated *variants of applying the model* (6.4.2). Next, *the interface* can be *shaped for predator-prey model scenarios* (*in four steps*, 6.4.3). Finally, the *software infrastructure for quantitative testing of* **A**, **β**, **DIU**, and **Δt** as most significant parameters for validating non-algorithmic software necessary for CybMd is described in detail (6.4.4).

6.4.1. Lotka-Volterra Model: Strong Conditions, Hard to Meet in Real Habitats

"The Lotka-Volterra equations, also known as the predator-prey equations, are a pair of first-order, non-linear, differential equations frequently used to describe the dynamics of biological systems in which two species interact, one a predator and one its prey. [...] The [...] system of equations is an example of a Kolmogorov model [...] which is a more general framework that can model the dynamics of ecological systems with predator-prey interactions, competition, disease, and mutualism. [...] The Lotka-Volterra model makes a number of assumptions about the environment and evolution of the predator and prey populations: 1. The prey population finds ample food at all times. 2. The food supply of the predator population depends entirely on the prey populations. 3. The rate of change of population is proportional to its size. 4. During the process, the environment does not change in favour of one species and the genetic adaptation is sufficiently slow" [wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations].

Though, besides the four "assumptions[4]" above, there are also other oversimplifying conditions, hard to meet in any benthic setting – or even in any usual habitat altogether: "The prey are assumed to have an unlimited food supply, and to reproduce exponentially unless subject to predation; [...]. The rate of predation upon the prey is assumed to be proportional to the rate at which the predators and the prey meet" [wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations].

With such unnatural restrictions, it is not surprising that the "equations have periodic solutions and do not have a simple expression in terms of the usual trigonometric functions. However, a linearization of the equations yields a solution similar to simple harmonic motion [...] with the population of predators following that of prey by 90" [wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations].

What is more, in practice (as the State of the Art below brings to light), all kind of additional parameters complicate the model without considering real settings, in real habitats.

[4] Labelling as "*assumptions*" such difficult to meet "*(pre)conditions*" reveals a serious memetic problem: mathematicians "*assume* their own *assumptions*" i.e., they "take something for granted" [thefreedictionary.com/assumptions] what is extremely improbable in real world settings.

6.4.2. Variants of Applying the Lotka-Volterra Model. State of the Art

In [51] the authors "proposed a patchy predator-prey model with one patch as refuge and the other as open habitat, and incorporated prey refuge in the considered model explicitly. [They] applied an analytical approach to study the dynamic consequences of the simplest forms of refuge used by prey and the migration efficiency. The results have shown that the refuge used by prey and the migration efficiency play an important role in the dynamic consequences of the interacting populations and the equilibrium density of two interacting populations" [51]. Thus, despite being a "patchy model", variant details do not refer to other species, environment changes ("patchy ignoring assumption 4"), anthropogenic disturbances and so on, but to "migration", "global stability" (the last two concepts are keywords!), or "**φ(x)**" as "functional response of the predator population" to "**x**" that "represents a quantity of prey population that occupied a refuge". Instead of showing that such a habitat is – at least – plausible, the paper highlights the fact that "**φ(x)**" satisfies the following assumption: $\varphi(0)=0,\varphi'(x)>0(x>0)$" [51].

The same matter is handled in the same way also in [50], where the results given as example of intractable mathematics in Figure 5.4, "implicate that in the Lotka-Volterra predator-prey system, the stabilizing effect of the mutual interference of predator species dominate the destabilizing effects of prey refuge" [50].

Likewise, in [35] the authors "derive from first principles the functional response of the predator and the reproduction rate of the prey in the case that the prey form groups as a defence against the predator and the latter captures only single prey".

For the sake of shortness, variant examples with parameters representing "effort of the prey and predator", "death rate of the predator", "capture coefficient of the prey", or "consumption rate of prey by a predator" are skipped over.

In short, the variants above, published in April-May 2013 in prestigious journals like *Applied Mathematics and Computation*, *Mathematical Biosciences*, or *Journal of mathematical biology*, show that: a) paradigm shifting towards CybMd is very slow; b) all three sub-perspectives of *PrsU* are exceedingly disregarded, favouring irrelevant details like "boundary equilibria" or "saddle points" (no field experiment is proposed); c) the Lotka-Volterra *equations*, overshadow the *model* – at least in the domain of ecologic modelling, where, instead of reducing *complexity*, they added *complicatedness* (6.4.3); d) corollary: the model *per se* is useful in real-world situations/habitats only for WISC.

6.4.3. Shaping the Interface: Four Steps for Predator-Prey Model Scenarios

"While much has been written about an alleged new science of complexity in recent years, no compelling operational distinction has yet been presented to distinguish complexity from complicatedness. In general, complex systems are supposed to exhibit "emergent" properties, that is those that, in some sense or other, resist reductionist explanation. [...] Beyond that, no stance will be taken on the complexity versus complicatedness issue" [21]. Based on the authority of *The Stanford Encyclopedia of Philosophy* (quoted in a similar "State of the Art" context, in 4.2.2), the research toolkit uses the (atypical) simplicity of the Lotka-Volterra model just for plausible scenarios, *assumed as such*.

Thus, the numeric values used in the four steps mentioned in 6.3.3 have the following (possible) interpretations from *PrsUw*: **A**: Prey population growth as if unimpeded by any hindrance. **β**: Prey population growth reduction rate. $\mathbf{A_{fb}}$: Prey population growth rate in a stable state. **Δt**: Prey population lifespan. **DIU**: In *normative* scenarios evaluates outcome in a given time span. (Details in 7.3.1.)

In short, the appliance interface must ease setting up WISC (7.3.3, 7.3.4).

In this context, the four implementation steps are (details in 6.4.4 and 7.3.2):

- g) *Sliders.* Are multifunctional and general, i.e., usable for various kinds of input, languages and psychophysical laws governing cognition ([2], 4.3.1, 6.2.4)

- h) *Menus and buttons*. Besides their functionality, they play also the role of "*Field-researcher guide for devising WISC*" (4.1.1) trying to utter the topics in a (quasi)ecologic language, where "*language*" has four connotations (7.3.2). (*Language* as system of signs – in the meaning used in semiotics – is applied in "i)" below).

- i) *Analog, approximate, uncertain input.* After testing numerically the functionality added through the six steps (6.3.3), most sliders were redesigned to accept also a boundedly rational – albeit primeval – form of "Computing with words" (enabling thus the simplest form of expressing degrees of uncertainty via word-based data input ([2], 4.3.1).

- j) *Logarithmic input.* Iff "i)" gets immediate student acceptance (November 2013, 8.2.1), the uncertainty slider will be redesigned for logarithmic input (to test whether a variant of the *Weber-Fechner* law is applicable to cognitive matters). Otherwise, this step remains for long range future work (8.2.2).

6.4.4. Software Infrastructure for Quantitative Testing of A, β, DIU, and Δt

The testing infrastructure illustrates best interface implementation because it is:

- Essential for carrying out quantitative testing to validate the concepts, approaches, and mechanisms of [2] – the apex of *ADT* as a whole (7.2).

- Essential for using the toolkit appliance as footing for service-oriented validation of non-algorithmic CybMd (7.3).

- The central point of the "*growing monocrystal*" (as *ad hoc* variant of successive prototyping, 6.3.3) as well as the first both *archetypal* and *serviceable* interface component.

- The only interface aspect not subject to change, a month before finishing *ADT* caused by skipping over the Scandinavian method for developing software, 6.1.2, 6.2.4, 6.3.3).

- A convenient prolegomena for defending PoC validation in SOE (7.1).

As entailed by the five reasons above, for testing the main numeric values expressing the appliance functionality from *PrsADT* (6.3.3), the following infrastructure was built:

- *Sliders*. The defining values for (macrochronic) stability of a basic cybernetic system (Barkhausen feedback loop), namely amplification factor **A**, feedback network transfer factor **β** (**|β| <1**), and feedback amplification factor $\mathbf{A_{fb}}$ were tested via sliders that conceal their functionality (6.2.4). For instance, semantic validation (automatic at analog input) was

explicitly adapted for numeric input setting up limits, far beyond the true-to-life values that will be set up by the end-user for credible WISC (e.g., **A < 2** means inexorable extinction rate), to ease testing “pathologic cases” necessary to PoC validation. On the other hand, allowing **β < 0.01** would be time wasting and possible confusing about the system propensity to attain a homeostatic state in due time.

- *API functions and instructions*. For the sake of reducing redundancy with 7.2, and *AV1*, here are listed five *API* functions necessary for minimal multithreading [19] (6.3.1) and one Java instruction [39], necessary to emulate dynamic exception propagation: *Sleep*, *Wait*, *Set Event*, *Reset Event*, *Create Thread*, *Return*.

SEVENTH CHAPTER: VALIDATION

Validating Process-Oriented Modelling Mechanisms

The validation framework had to follow the current trend in *service-oriented engineering* - in the original meaning of "*service*" –, focusing on *PoC validation* (7.1). In this context, the *concepts, approaches,* and *mechanisms* featured in [2] are systematically *tested quantitatively* in line with the industrial paradigm (7.2). Ending thus the process of validating the thesis ideas, the *concepts* grouped under the label "*Non-algorithmic cybernetic modelling of living systems*" (3.3.4) are submitted to *service-oriented validation* (7.3).

7.1. "Proof-of-Concept" Validation in Service-Oriented Engineering

The intrinsic *process* nature of any *service* entailed two (on-going) shifts in software engineering: a) Spawning various *Service-Oriented "x"*, where *x*, ranges from application to computing, to engineering, *restricting* (unduly) *meanings* and *broadening* (already existing) *confusions* (7.1.1). b) *Revisiting validation* based on *PoC* applications (the significance of this shift requires a (final) *State of the Art* (7.1.2). Only on this groundwork can the key problem of *assessing service quality in process-oriented software engineering* be dealt with (7.1.3). This shapes the framework for the very process of validating [2] as specified below and carried out in the whole chapter.

7.1.1. Service-Oriented "*x*": Restricting Meanings and Broadening Confusions

After giving the 23 connotations of *service* (15 of them as noun) [thefreedictionary.com/ service] adds a usage note: "Aside from specialized senses [...], the verb service is used principally in the sense "to repair or maintain": service the washing machine. In the sense "to supply goods or services to," serve is the correct choice".

Though, a Google Scholar search (July 13, 2013) for "Service-Oriented" gives about 260,000 results, while "User-Oriented" gives only about 55,800 results. The probably reason: while "*user-oriented*" keeps its traditional meaning, "*service-oriented*" refers not to "*service*" in the meaning above but – after limiting excessively its semantics – to "*Web service*" ("Web services are message-oriented systems based on document exchanges. But the development of web services on both client and server sides is increasingly based on object-oriented implementations" [62].) Thus, the crucial question "*Who* is the service for?" gets a rather unexpected (and odd) answer, "The software engineer", instead of the habitual one, "The application end user". In other words, the conventional "service *provider*" is more interested in helping *him/herself* (with a more convenient application *structure*) than in helping the *user* (usually, his/her employer!) who pays for the service (hoping to get a more appropriate application *architecture*).

To justify the explanation above just one example referring to Service-Oriented "*x*" (where *x* stays mainly for *architecture* but occasionally also for *software engineering*

or - very telling – for *programming*): “Service-orientation is a promising paradigm to decompose inward-oriented organisational processes into outward-oriented ITG Service Components […] In service-oriented software engineering, a component is evolving into a service with standard interfaces for communications and the ability to dynamically locate necessary services at runtime [...] The SOA integrates them into a distributed computing system by means of service-oriented programming” [53].

7.1.2. Validation Revisited: “Proof-of-Concept” Applications. State of the Art

“Proof-of-Concept” is understood as specified in 3.2.4. PoC models became popular even in most advanced exploratory research domains. For instance, [36] presents a “model based bionic muscle with hyperbolic force-velocity relation” in a chaoplex biologic domain (Bionics and Biomechanics): ”This proof of concept can be seen as a well-founded starting point for the development of Hill-type artificial muscles” [36].

Therefore, this “State of the Art” refers to the novel subfield of *PoC validation* – swiftly developing and replacing conventional (prototype-based) validation practice. The following three examples are from 2013 papers in cutting-edge engineering research:

- In “Ceramic matrix composite technology for aircraft turbine engine applications” [37]: “Progress in CMC component fabrication, evaluation, and testing is presented in which the goal is to advance from the proof of concept validation [...] to a system/subsystem or prototype demonstration in a relevant environment” [37].

- In “Intelligent Fiber Optic Systems” [63]: “Finally, a proof-of-concept FBG-based high strain measurement system is developed. […] These strain data are used to validate our fixture design”. [63].

- In “Magnetostrictive aluminum composite with electrically tunable stiffness” [61]: “Shifts in modal frequencies as a function of applied magnetic field for a cantilevered, proof-of-concept composite were measured. […] The optimized composite shows a threefold increase in modal frequency shifts compared to the proof-of-concept [61].

As a result of this rapidly rising acceptance, appeared an institutional accelerator: “The proof of concept center accelerates the commercialization of innovations out of the university and into the marketplace. It does this by providing seed funding to novel, early stage research that most often would not be funded by any other conventional source” [17].

The main domain where acceleration is needed appears to be *validation in SOE*.

7.1.3. Assessing Service Quality in Process-Oriented Software Engineering

Concentrating the above to its essence, the problem is to set up a consistent quality valuation framework based on the requirements of PoC validation (7.1.2), removing the hurdles put to user-centred design by distorting the meaning of *service* (7.1.1),

After abridging the inferences from both 7.1.1 and 7.1.2, such a framework would be:

- *Service-oriented means user-oriented*. “*Services* are surely *processes*. Thus, *service-oriented* engineering is *based on processes*. [...] *Software Engineering* is regarded as adult research subdomain of *Computer Science* (not just as innovative subdomain of

IT) [and] *Service-Oriented* refers to *Engineering* as a whole, key feature of *post-industrial* engineering (not to *Software Engineering*, that *was always* service-oriented)" [14]. Since: a) this stance is fully endorsed by the examples and explanations in 7.1.1; b) it proved to be useless to fight syntagmata spawned within old paradigms, a boundedly rational (hence, simplified) way out – already followed in the section title – is proposed: in the rest of this chapter "*process*-oriented" should be read as "*service*-oriented".

- *Proving concepts is necessary and sufficient*. The assertion refers to *research in engineering* and is confined below to *transdisciplinary modelling*. While necessity is self-explaining, sufficiency needs defending. Based on the relevant examples in 7.1.2, the reason is threefold pragmatic, considering the very possibility to: a) continue engineering research in medium-sized East-European universities; b) approach exploratory transdisciplinary research; c) model chaoplex systems.

Hence, the validation framework of [2] is founded on two pillars: a) all thesis and *ADT* contributions should be "Proof-of-Concept" validated; b) process-oriented software (inherently *non-algorithmic*) should be comprehensively tested using product-oriented (inherently *quantitative*), conventional, "generally accepted" validation methods (7.2).

7.2. Quantitative Testing of Magnitudes Needed to Prove Thesis Ideas

To clear up testing, the *standoff* about *product-leaning quantity vs. service-leaning quality* is *solved* pragmatically (7.2.1), explaining the testing approach that follows: **A** *AND* **β** *ensure* together *simplified (linear) cybernetic systems stability* (7.2.2). On the contrary, the two temporal dimensions – despite aiming both at *fighting chaoplexity*, making the most of *anthropocentrism* – are unlike enough to be tested separately: **Δt** illustrates *discrete time replacing films by snapshots* (7.2.3), while the three *features of the (second) time* dimension (**D**elay, **I**rreversibility, **U**ncertainty) *are even more human(centred)* (7.2.4).

7.2.1. Solving Standoff: Product-Leaning *Quantity* vs. Service-Leaning *Quality*

To impair misunderstanding, there is neither a logical nor a philosophical dilemma, but a pragmatic explanation that leads to grouping or separating magnitudes to be tested. Thus:

- All assertions about holism/reductionism, chaoplexity/mechanistic model, and so on, are expressed at *present tense* (that holds for quotations from authorities too (e.g., *Stanford Encyclopedia of Philosophy*: "in some sense or other, resist reductionist explanation" [21], 6.4.3). In short, it is "*ignoramus*", not "*ignorabimus*" (just as in 2.1.1).

- Moreover, all assertions about cognitive chaoplexity (on the whole, 3.3.4, 4.2, 6.2.1) or structural chaoplexity (in biology, 4.2.1) are based on evidence quoted from very recent scholarly papers. In short, it is about "here and now".

- That holds, *a fortiori*, for [2] and *ADT* where the contributions to be proved are "macroscopically holistic". (Moreover, it is the only possible approach for a thesis having as pillars "bounded rationality" and "Just in Time", 3.2.2.) In short, chaoplexity – qualitative *par excellence* – cannot impede quantitative testing.

- Other main ideas to be proved – as new ways to valorise *anthropocentrism* (6.2.4), or even *transdisciplinarity* (mainly, when the research domain chosen to illustrate it involves CybMd of living systems, 3.2.4) – are hardly explainable when reducing abstraction level.

- In spite of the above, in software engineering qualitative (*holistic*) models can be defended by quantitative (*reductionist*) testing. Thus, *testing* distinct magnitudes is compatible with *interpreting* their synergistic effect, meeting the terms of both [1] and *O2* (3.1.3, 6.1.1).

- As regards the contributions of *ADT* as self-contained research, the problem is even simpler: the numeric values used in the implementation steps *do have* interpretations from *PrsU*: from *PrsUb*/*e* they are already given (6.3.3) and from *PrsUw* those numeric values *must have* interpretations to be able to validate a service (7.3.1).

7.2.2. **A** *AND* **β** Ensure Simplified (Linear) Cybernetic Systems Stability

The *AND* in the title – suggesting the Boolean operator – highlights an immediate example of testing *separately* and interpreting *as one* (7.2.1), since stability (even for macrochronically assessed linear systems, 5.1.1, 5.1.2) is unthinkable without feedback (5.2.2) and depends on the value of **1 - β*A** (5.1.1), where **β*A** is the product of two complex numbers. Hence, it is impossible to consider stability as function of only either **A** or **β**. However:

- Any feedback where **β** is not a *purely* negative number with **|β| < 1** is outside the scope of testing from both *PrsADT* and *PrsU* (in line with the current research targets).

- Both **A** and **β** can (and should) be varied for fine-tuning CybMd.

- Even more for modelling homeostasis.

- Even more for using the appliance to devise WISC (central for descriptive scenarios, *sine qua non* for normative ones).

Therefore testing separately is mandatory but not with the classical method of employing parameters: for software engineers it is ineffectual and for ecologists it is even confusing (see Figure 5.4). To be suitable, a relevant value is looked for to keep the “parameter” constant while modifying the “variable”. Before getting from the end user more telling values, the following were used for the first tests (AV2):

- For *testing* **A**, **β** was set at the value of the golden ratio (**0.619**, estimated as ratio of consecutive Fibonacci numbers: **13/21**) because there is (anecdotal) evidence that the reproduction of rabbits in Australia could be related to the Fibonacci series.

- For *testing* **β**, **A** was set at the value of **1000** because stability is endangered by high amplification rates.

7.2.3. **Δt**. Discrete Time Fights Chaoplexity Replacing Films by Snapshots

The role of **Δt** restated in the title, albeit crucial from *PrsU* (above all in scenarios), cannot justify testing a magnitude that from *PrsADT* has (apparently) only technologic role. (Indeed, it is no point to test neither Kotelnikov sampling, nor the *SLEEP* function.) Moreover, it takes only two values during the validation process: for *SLEEP* **10** message

(the minimal value, **1** ms is not used in view of fault tolerance) but for the first tests a user-friendly value of **5** sec was used, labelled below **Δtid** (from "**Δt** for *i*nterface *d*evelopment").

Though, as the exercise below shows, allowing **Δt** to take negative values, there are reasons to go beyond the (apparently, exclusively) procedural role of **Δt** (since the reasons emerge clearer after describing how **Δt** can be used to "predict the system past", the main reasons from *PrsADT* are presented at the end of 7.2.3 and those from *PrsU* in 7.3.1).

Exercise: investigating past evolution of linear cybernetic systems starting from a stable present state. *Notations*: $\mathbf{A_{hs}}$: amplification when the *h*omeostatic *s*tate is deemed arrived at); $\mathbf{A_{-n}}$: amplification at the $\boldsymbol{n^{-th}}$ iteration, (i.e., before $\boldsymbol{n}$*$\boldsymbol{\Delta tid}$ or $\boldsymbol{\Delta tid}$ seconds ago).

Starting from the Barkhausen relation above and replacing $\mathbf{A_{fb}}$ by $\mathbf{A_{hs}}$ and **A** by $\mathbf{A_{-1}}$:

$\mathbf{A_{hs} = A_{-1}/(1 + \beta * A_{-1})}$; where from, $\mathbf{A_{-1} = A_{hs}/(1 - \beta * A_{hs})}$; likewise, for the $\boldsymbol{n^{-th}}$ iteration:

$\mathbf{A_{-n} = A_{-(n-1)}/(1 - \beta * A_{-(n-1)})}$. To correlate loop length with the usual (forward) loop (7.2.2), the exit condition is $\mathbf{A_{-n} > 100 * A_{hs}}$. Since it is suitable to examine visually the past system states, the (module of the) sleep duration in the exercise, $-\boldsymbol{\Delta_{tex}}$, is kept the same as for the first tests: *SLEEP* $(-\boldsymbol{\Delta_{tex}}) =$ *SLEEP* $(\boldsymbol{\Delta i_t})$.

Reasons (from *PrsADT*). Using discrete moments going backwards in time shows that:

- The difference between the two temporal dimensions represented by $\boldsymbol{\Delta t}$ and ***DIU***, respectively, is by far not reduced to *time granularity*.

- $\boldsymbol{\Delta t}$ belongs undisputable to *Newtonian* time since it is *reversible*.

- The perceptible reluctance of specialists in robotics and automatic control to accept that they use (albeit implicitly, as in reverse time series) *reversible time* is groundless.

- If $\mathbf{Im(\mathfrak{w}) = t}$ (5.4.3), the exercise puts an end to "the arrow of time" taboo in software engineering and endorses the claims that "mathematically, $\mathbf{t = -t}$ has no major consequence [...]. However, from a physical stance, changing the sign of $\mathbf{t}$ means inverting "the arrow of time" (in the meaning given by Eddington)" [14].

- Corollary: it defends implementing WiTi in the appliance, despite the concept has not yet proved its appropriateness for CybMd.

7.2.4. D,I,U. The (Second) Time Features Are Even More Human(Centred)

Studying competitive bicycle pelotons [22] has identified three hysteresis types:

1. "peloton decelerates rapidly […] followed by a proportionately longer acceleration [(] predictably and periodically, primarily as riders enter and exit corners […] known as the accordion effect […] resembles vehicle traffic hysteresis" [22].

2. "peloton speed accelerates rapidly […] followed by a proportionately longer duration of low flow. [It] is […] unpredictably and aperiodically […] characterized by a delay in the reintegration [and] is essentially the inverse process of vehicle traffic hysteresis" [22].

3. "peloton transitions […] to a period when […] power output remains roughly constant, such as when […] riders proceed up sufficiently steep hills and weaker cyclists lose positions in the peloton" [22].

[22] ends: "[1.] results more from […] adjustments due to […] constraints […] externally determined […]. [2.] is driven largely by limitations in cyclists' competitive fitness and their simultaneously opposing objectives […]. [3.] results from intrinsic differences in physiological fitness […]. [P]eloton hysteresis is a selforganizing dynamical process within competitive systems […]. We may predict these kinds of hysteresis to be observable in rapidly moving herds, flocks, and sperm aggregates, among other biological collectives" [22].

In short, even in simple "competitive systems ", acting in known environments, hysteresis is manifold as both *causes* it stems from ("externally determined", "opposing objectives", "intrinsic differences in physiological fitness") and *effects* (from "accordion effect" to system disintegration). Hence, appropriate testing should be focused on the most significant feature of hysteretic delay, *BeTi*, highlighting its major aspects: *unpredictability* and *irreversibility*.

Thus, the leitmotif of counteracting anthropogenic disturbance (4.1.1, 4.1.2, 5.3.3, 5.4.3, 6.1.2, 6.3.3, 6.4.3) comes to life representing pollution by the Heaviside function. In the current implementation, because of the Boolean nature of the "event" set by *Set Event* (6.4.4), it is defined as function of τ (the perturbation time, 5.4.3, AV2).

7.3. Service-Oriented Validation of Cybernetic Modelling Concepts

Ending above the validation of the contributions in [2], the contributions of *ADT* as self-contained PhD research should be proved too. The *architecture to validate* is represented by S_{WISC}, the *design space for WISC* (7.3.1) and is usable via the *interface* construed *as mirror for functionality*, enabling the *user-validation* of **A**, **β**, **Δt**, *and* **DIU** (7.3.2). This is illustrated by two kind of *scenarios: descriptive*, to interpret *facts* (7.3.3), and, much more significative, *normative*, to *prepare actions* (7.3.4).

7.3.1. *Architecture to Validate*: S_{WISC}, Design Space for "What-if" Scenarios

Validating S_{WISC} becomes easier, because it takes advantage of: a) validating above all (old and new) mechanisms ([2], 5.2, 5.3) applied in non-algorithmic CybMd; b) the swiftly rising acceptance of PoC validation in the last years (3.2.4, 7.1.2); c) the undeniable improvement provided by the appliance, when the end-user compares CybMd (as reflected in the toolkit) with the models available before (the predictive models in the recent past); d) quantitative testing (7.2, *AV2*) the four magnitudes (**A**, **β**, **Δt**, and **DIU**) necessary and sufficient for WISC, the toolkit (6.2, 6.4) is designed for. (Their interpretation from *PrsUw*, abridged in 6.4.3, is detailed below.)

- **A**. Prey population growth as if unimpeded by any hindrance.

- **β**. Prey population growth reduction rate due to any factor impeding the current generation to reproduce itself at the same rate as the previous one. In scenarios closer to the – rather utopical – Lotka-Volterra model, the only hindrance is predation. In more

plausible scenarios, the homeostatic state of a benthic community is deemed as arrived at, considering the same environment factors, except predation. In realist scenarios, starting from a homeostatic state in a (partially known) ecologic system, a (new) impairment is usually a (more efficient) functional response of the predator to increased effects of (unexpected) anthropogenic disturbance or (predictable) natural habitat perturbation. Such a scenario yields a new homeostatic state (favouring the predator).

- $\mathbf{A_{fb}}$. Prey population growth rate in a stable (homeostatic) state. If the impairment is major $\mathbf{A_{fb}}$ decreases swiftly from the ideal value of **A** approaching the level of zero growth.

- **Δt**. Normally it is the prey population lifespan. On the other hand, it gives great flexibility to scenarios, allowing "to take samples" of the ecologic system state when desired. Moreover, allowing **Δt** to take negative values, as suggested above (7.2.3), such samples could predict the system past (via short, boundedly rational, plausible reverse time series). Besides, there are other reasons too, to give semantic value to **Δt**: a) credible scenarios are simple, but simplicity entails discrete time to answer key "What-if" questions (a scenario engaging – even as suggestion – some complicatedness as in Figures 5.3 or 5.4 cannot get user acceptance); b) discrete time is necessary in any memetic engineering exercise involving ecolinguistics (7.3.4); c) it is the only chance of the research toolkit to use the peculiar simplicity of the Lotka-Volterra model in – at least – plausible scenarios.

- **DIU**. Likewise, besides expressing hysteretic delay in (open-ended) *descriptive* scenarios, it is very valuable in *normative* scenarios, to evaluate how the looked-for outcome of an environment preservation action can be reached in a given time span (7.3.4).

7.3.2. *Interface, Mirror for Functionality*: User-Validating Look and Language

The appliance interface, as research toolkit, has to reflect all the functionality necessary and sufficient to set up WISC for *preservation of ecological* (sub)*systems*. Specifically, scenarios aiming at investigating ecologic stability in (oversimplified) *benthic communities in Transylvanian lotic systems* (3.2.4, 6.1.1, 6.4.1).

In this context, the sliders, menus and buttons presented Figure 7.1, used for quantitative testing **A**, **β**, **Δt**, and **DIU** (6.4.4) should be completed for playing the role of ""*Field-researcher guide for devising WISC*" (4.1.1, 6.4.3). In this context, cognitive chaoplexity can be substantially reduced taking advantage of the four connotations of "*language*":

- *Natural language*. It is to expect that Romanian is favoured for dialog (to ease interaction, above all in field experiments) and English for outputting results (to ease scientific collaboration). That implies "Google-like" versatility in choosing language.

- *CybMd language.* Choosing GST as *Lingua Franca* proved to be essential for transdisciplinary collaboration (4.3.2, 5.4.2, 5.4.4). To strengthen the link in practice, all terms used to label interface components should be clear for any intended user category. (Example: acronyms should be avoided as much as possible, despite the need to save space on laptop screens.)

- *Language* as "*memetic carrier*". The "(quasi)" prefixing "ecologic language" (4.1.1, 6.4.3) suggests that it is vital to avoid distorting "What-if" questions/answers by dubious memes induced in transdisciplinary language from misunderstood Computer Science language (not just deceiving syntagms like "predictive modelling" but also confusing ones like "service-oriented"), as well as to exploit ecolinguistics (6.1.3, 6.2.1. 6.2.4).

- *Language* as system of signs (as understood in semiotics). "Based on the concept of "Computer-Aided Semiosis" [...] next to (atemporal) text or images, multimodal interfaces of modern computers could assist humans in understanding complex messages (textual or sign-based, atemporal or temporal)" [6]. Since WISC (despite not being human-to-human communication) are extremely interaction-oriented, several guidelines of [57] about "non-verbal communication (the silent language)", regarding kinesics, haptics, proxemics, or chronemics could be applied in animated images because "language reflects cultural values" [57]. In a larger sense, it is a first step from "Computing with *words*" towards "Computing with *images* or *sounds*". (Example: warnings about pollution could be transmitted using personalised, emotional animations; text could use "telling" fonts as Chiller, *iff* the end-user deems that the message is better conveyed.)

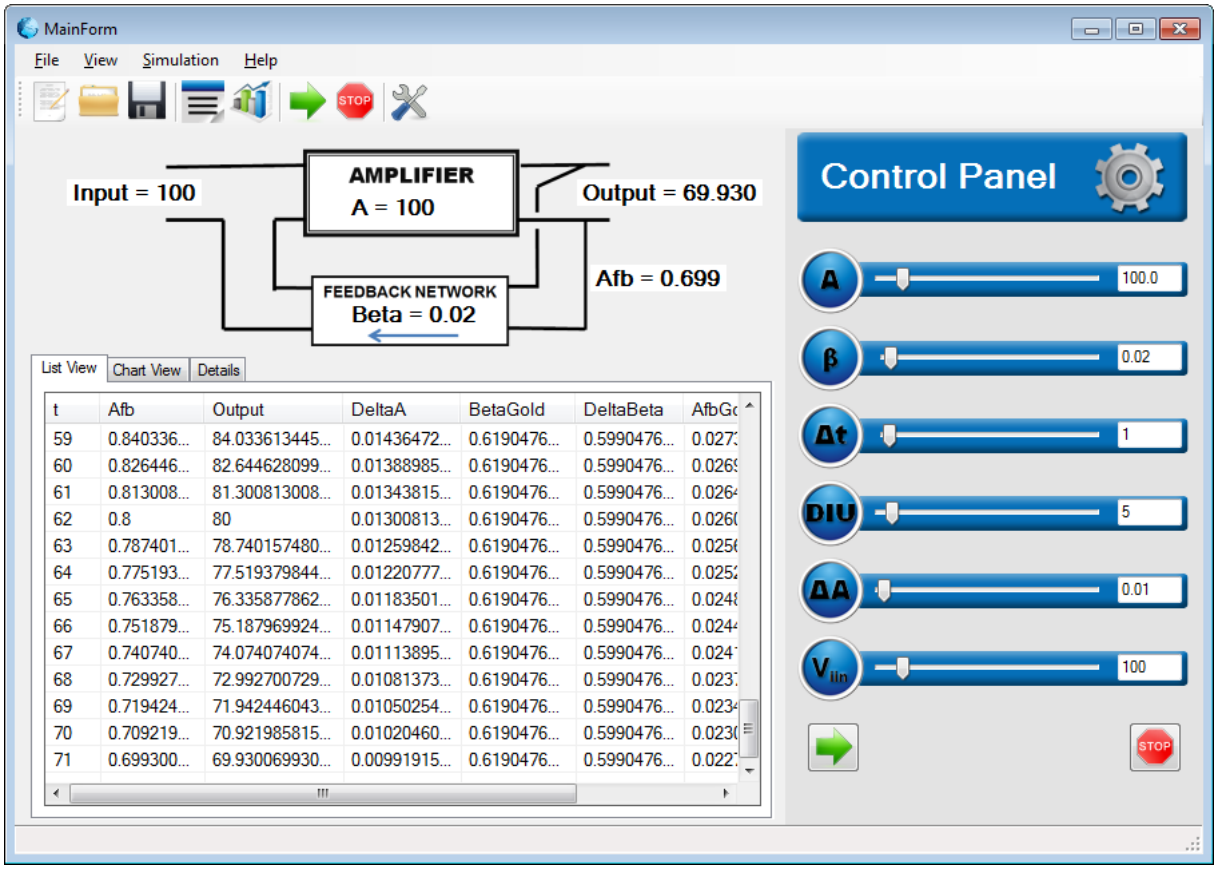


Figure 7.1. Toolkit interface accepted by the end user at July 19th 2013.

While the examples above can be implemented through options available in menus, since they refer rather to personal taste, other choices, depending on the enterprise culture of the (extended) research team, should be "parameterised" only by users having

administrator access rights. (Examples: when an ecologic emergency occurs, what are the urgency degrees from e-mail, to Skype, to SMS, to mobile, to hotline, 7.3.3.)

7.3.3. *Descriptive* Scenarios for Field Ecology Research: Interpreting *Facts*

Because ecolinguistics is a crucial S_{WISC} dimension (6.1.3, 6.2.4, 6.4.3), applying it in scenarios is at hand. Though, descriptive scenarios need clearing up:

- In ecology, describing systems is just a mean to preserve them; thus, action-oriented normative scenarios are understandably preferred over descriptive ones.

- Applying scenarios instead of models to describe a system suggests markedly low model acceptance within the ecologic researcher community (as expected in the case of the strange simplicity of Lotka-Volterra models, 6.4.1, 6.4.2, 7.3.1).

- Even when temporal information is lacking it seems better to sample benthos using real benthometers instead of “virtual benthometers”.

However, there are at least two resource-saving reasons to use descriptive scenarios:

- a) Severe lack of time to get temporal information about system evolution.

- b) Easing ecology student laboratory work as required in field ecology research.

Therefore, below are listed some interface features illustrating its suitability for both kind of scenarios, followed by a simple example of an educational scenario.

After launching the toolkit from wherever within a Windows platform computing environment, and granting access rights (at present are only two user categories: *administrator* and *researcher*, following the guidelines given in the examples, 7.3.1), the appliance takes control of the screen until the user pushes the "STOP" button (not just to be user-friendly but to reinstate sane application development practices, avoiding uncontrolled computer-human interaction). According to the rights, the user chooses the type of scenario and other menu options (7.3.2) and inputs the desired start values. At this stage of interface development, there are only two scenario choices: *descriptive* (with input values for **A**, **β**, and **Δt**) and *normative* (with input values for **A**, **β**, **Δt**, and **DIU**). The input values are taken from the data base for *benthic communities in Transylvanian lotic systems* (now a fragment in *Microsoft Excel* is accessible from the appliance).

Example: What could be the growth of the prey population species ***X*** in a stable (homeostatic) state ($\mathbf{A_{fb}}$) in interference with the predator species ***Y*** if the unimpeded prey population growth is **A = 700**, its generation lifespan is **Δt = 6 hours**, and **three fifths** of each generation escapes predation from ***Y***.(**β = 0.4**)?

7.3.4. *Normative* Scenarios for Preserving Benthic Species: Preparing *Actions*

A fundamental role of ecolinguistics is to convey believable warnings about the risks of neglecting environment preservation measures. Normative scenarios are a means at hand to spread such messages – and are favoured compared to descriptive ones – because they:

- “[T]ake values and interests into account, [...] describe a desirable future or set a specific goal and explore possible ways to reach that goal [and] are useful for finding ways of reaching specified goals or testing alternative policy interventions to see how effective

and efficient they are" [The *Environmental Terminology and Discovery Service*: glossary. eea.europa.eu/terminology/concept_html?term=normative%20scenario].

- Are more credible – *iff* assumed as such (6.4.3) – since the subliminal message is "we are people like you, that don't pretend to know all the facts". (The users empathise easier with those – people or even software – that put "What-if" questions.)

- Show (very) reduced cognitive chaoplexity since non-actions are caused rather by lack of political will than by limited technological possibilities (simplicity can be valorised).

- Ease monitoring species behaviour *in situ* (an environment preservation action is inherently based on controlling states and their variations caused by perturbations).

On the other hand, the CybMd incorporated in the toolkit – despite being intended initially only at *describing* (hysteretic) *delay* –, *warns* against *delaying* (actions). Thus, it is possible to use **DIU** to get a first feeling about (plausible processes towards) homeostasis of a benthic predator-prey species pair when the existing homeostatic state was disrupted by anthropogenic disturbance. In other words, it is possible to evaluate how the looked-for (or be afraid of) outcome of an environment preservation action (or inaction) can succeed (or fall through) in the time span described by **DIU** (7.3.1). Since ecolinguistics is central, a novel memetic engineering exercise is proposed, outlining a scenario aimed at alerting about benthic species unbalance. Yet "unbalance" and "homeostasis" are fundamentally contradictory terms. That requires explaining, because the assertions about homeostasis as "*adaptation process* triggered by a *perturbation"* (in the system environment, 4.2.3, 5.3.3) holds but needs illustration to verify whether it is likely or even possible.

Example: How could be affected the system homeostasis in the example above (7.3.3) by a disturbance described by a Heaviside function with ***ad* = 100**, occurred two days ago if the pollution effects are not cleaned up in a week (**DIU = 2 + 7 days**)?

EIGHTH CHAPTER: ASSESSMENT

Conclusions and Future Work

Because of the *ADT* role, context, and content, both *Conclusions* (8.1) and *Future work* (8.2) are outlined against distinct dimensions (according to the various "*PrsX*").

8.1. Conclusions

Since *fulfilling requirements*, i.e., *quantitatively validating the contributions* of [2] (8.1.1) is the *raison d'être* of *ADT*, the main conclusions are from *PrsADT*, following the "validation thread". Next, the transdisciplinary research is evaluated: *novel concepts for modelling homeostasis via diverse kinds of time* (8.1.2) and *building mechanisms and refining methods for illustrating concepts* (8.1.3). Ultimately, the two undertakings are assessed *integrated* under the title of "*Non-Algorithmic Cybernetic Modelling of Living Systems*" (8.1.4).

8.1.1. Fulfilling Requirements: Quantitatively Validating Thesis Contributions

The "validation thread" to follow, starts with selecting *concepts and approaches to be emphasised in relevant modelling* (3.2.2) and *mechanisms to be demonstrated via PoC software* (3.2.3). Then, its direction is established *choosing a research domain fitting with the new perspective* (3.2.4). After passing through the two transdisciplinary research chapters, the thread is abided by the *rationale* for (6.1.1) and *approach* to (6.1.2) the *interface as versatile, multifunctional proofing tool* (6.2). On this foundation the *toolkit functionality* (6.3.3) is reflected by *the interface*, as validation infrastructure (6.4.4). Finally, the *PoC validation* framework for *service-oriented engineering* is set up (7.1) and the validation process itself is carried out via *quantitative testing of magnitudes needed to prove* the contributions in [2] (7.2).

Accordingly, the conclusions about complying with [1], refer the *sine qua non* category of contributions, updating the conclusions of [2] based on the failed application. For the sake of both, clearness and brevity, the references to section 9.1.2 of [2], "*Achievements and Failures*", that needed amending are put into curly brackets ("{}").

- *Bounded rationality* as both *strategy* and *mechanism*. BR was investigated related to {decision making, behavioural economics}, approximation, and uncertainty setting up its main role in the post-industrial era: fighting cognitive chaoplexity. Instead of "behavioural economics" is "biology", while "decision making" is not anymore at a general level, being applied in normative scenarios. The {application domain "Non-deterministic e-Teaching"} was replaced by "Non-algorithmic cybernetic modelling" and {BR as "psychological stabiliser"} with the two much more relevant "biological stabiliser" and "ecological stabiliser".

- *"Just in Time"*: as key requirement for any service providing, hence, *a fortiori* for (non-trivial) modelling as transdisciplinary service. {For decision making in chaoplex and

risky situations, even "*Rationale 4: precision is harmful*" (from a JIT stance)}. Besides its "first level endorsement" there is a "second level" – albeit rather subliminal – involved in the ecolinguistic discourse backed by (normative) scenario results.

- *"BR + JIT"*. The synergistic effect is obvious when applied to CybMd, as for instance, when testing **A** and **β** *separately* but interpreting the results *as one* (7.2.2), or when assessing time (discrete-time, BeTi), even at (quantitatively tested) parameter level (**Δt**, **DIU**).

- *GST as*: a) *Lingua Franca* (living systems cannot be investigated without it). b) scientific infrastructure for Cybernetics (crucial processes – for instance, reaching/maintaining stability of ecologic systems – cannot be modelled without it). {BR was expressed in terms of GST, including the key aspect of BR as feedback},

In this regard, the enrichment was significant and twofold: a) Instead of using the *Lingua Franca* for a general metascience (as GST itself) it was used for the science that became the most important successor of GST, namely cybernetics. b) Moreover, this language was applied effectively in software engineering: for instance, the key concept of (negative) feedback expressed by **β** (feedback-network transfer factor) was essential in modelling homeostasis in all stages from giving (adaptable, transdisciplinary) semantic value to modelling, to setting up design space, to mirroring architecture in the interface, to quantitative testing, to validating, to proposing it as key input for WISC.

- *Transdisciplinarity*. As it was shown (2.3.1, 3.1.1, 3.2.2) servicing other domains is crucial to avoid the syndrome of "solution in search of a problem". {Choosing GST as *Lingua Franca* for psychologically-oriented addressees was much more than a linguistic bridge}. As transdisciplinary research axis of *ADT*, it is assessed in 8.1.2 and 8.1.3.

- *User-centeredness*. Anthropocentric interfaces are vital. Likewise, detailed in 8.1.3.

8.1.2. Novel Concepts for Modelling Homeostasis via Diverse Species of Time

From *PrsADT* it is a welcomed but not needed advance in transdisciplinary research, obtained through serendipity and useful as validation framework. On the contrary, from *PrsCSIT* as one (transdisciplinary research) bridge pillar (Figure 3.1) and *PrsUb*/*e* as the other pillar, it is the very research substance (Chapters Four and Five). (From *PrsUw* it is the unnoticed technological toolkit background.) For clarity, the research results are grouped in three categories: *concepts*, *approaches*, *mechanisms*. For concision, in each category are only three entities, outlined very abridged (below, concepts are dealt with; the other two categories are shown in the next section): Key word/expression (*KW*), *core idea* (*CI*), *practicability* (*PCT*) – mainly for software engineering (*PCTS*) or university education (*PCTU*), in addition to CSIT curricula (AI, modelling and simulation, software engineering, etc.).

- *Concepts*. They augment the field of CybMd via diverse *kinds of time*:

C1. *Microchronic modelling stability of linear systems*. *KW*: microchronic stability. *CI*: Extending the (*atemporal* or – at most – *synchronic*) Barkhausen relation for *stability* (of amplifiers with feedback loop) to domains where a temporal dimension is unavoidable (i.e., where *processes* CANNOT be modelled as *events*). *PCTS*: modelling living systems

(biology, ecology, medicine – in a large range from clinical measurements to pathodynamics –, as well as in pharmaceutical sciences, automation, robotics, or any other chaoplex, service-oriented application. *PCTU*: besides the domains cited before, electric engineering, electronic engineering, any domain of GST (above all cybernetics or synergetics).

C2. *Discrete-time modelling of living systems*. *KW*: discrete time; time granule. *CI*: a) *Discrete time* is – *now* and (not only) *here* – the only way to avoid intractable mathematics (mainly partially suitable differential equations) in modelling. b) The *time granule* (b1) has semantic value, (b2) is variable within a very large range (e.g., prey population lifespan), (b3) appropriate for WISC (simple *movies* are a sequence of *pictures*), (b4) simulating (future or past) discrete events (b5) that occur in BeTi, and (b6) is Kripke-compatible with *irreversible* time (the time granule **Δt** belongs undisputable to NeTi since it is *reversible*). *PCTS*: Idem. *PCTU*: Idem plus logic and metamathematics (mainly in line with the Carnap/Tarski model-theoretic approach). (The "Kotelnikov-sampling" potential of the time granule was not investigated; depending on user-interest it could be explored in the future. 8.2.2.)

C3. *Correlating distinct temporal dimensions*. *KW*: homeostasis; hysteretic delay; BeTi. *CI*: Perturbation (anthropogenically caused), as trigger of (ecologic) system homeostasis acts in *irreversible* time. $\boldsymbol{\tau}$ (perturbation time) and $\mathbf{t}$ (system time) are not just representations of two (distinct, Newtonian) *time functions*, but *dissimilar temporal dimensions*. Their correlation (mainly when *hysteretic delay* matters) requires non-algorithmic CybMd. *PCTS*: modelling homeostasis with hysteretic delay (mainly after major disruption of ecosystem homeostasis, in normative scenarios aimed at counteracting anthropogenic disturbances). *PCTU*: biology, ecology.

8.1.3. Refining Methods and Building Mechanisms for Illustrating Concepts

The approaches and mechanisms set up for confirming the practicability and validity of the new research results are separated from the concepts they should illustrate, not because they are considered marginal, but given that they are only partly original. Thus, "*building*" means that the innermost mechanism was only modified (with significantly enriched semantics) before being nested in the Matryoshka doll framework together with the two novel mechanisms. As regards "*refining*", it acknowledges that the only *ADT* contribution is adjusting conventional approach to both "BR + JIT" viewpoint and current circumstances.

- *Approaches*. (In parentheses is indicated the standard approach):

A1. *Monocrystal approach*. (*Successive prototyping*.) From both *PrsADT* and *PrsUb/e*, the "proof of concept" approach entails a series of very short-lived "interface design" variants, looping in a "cut and try" manner (established in 6.3.1, carried out in 6.3.3) where the toolkit expands continually like a silicon monocrystal.

A2. *Separating complexity from complicatedness*. (*Occam's Razor*.) Considering that: a) the only model relevant for homeostasis of predator-prey species pair – albeit its radical simplifications in complexity and extent – is the Lotka-Volterra model (6.4.1); b) in ecologic

modelling the Lotka-Volterra *equations* overshadow the *model* adding *complicatedness* (to conditions, hard to meet in real habitats, 6.4.2); c) all sub-perspectives of *PrsU* are disregarded (focusing on irrelevant mathematic details, 6.4.3, Figure 5.4), the model was exploited in the only practicable boundedly rational way: for setting up WISC in discrete time, in real-world situations/habitats (7.3.1), based on plausible input data (gathered from recent field experiments or expressing skilled-researcher educated guess, 7.3.3, 7.3.4).

A3. *Interface as Factotum*. (*Anthropocentrism; Scandinavian method; SOE*.) To keep the user role as protagonist in SOE, the application functionality is concentrated in the interface (6.3.3), designed not only for ecologic research (adaptable to diverse scenario types, habitats, species, predator-prey interaction modes, 6.4.3) but even as software infrastructure for quantitative testing and, thus, as (technologic) validation framework (6.4.4).

- *Mechanisms*. They are here only reminded for the sake of completeness because: a) their crucial technologic role was repeatedly emphasised above; b) from a transdisciplinary stance they are less important; c) their second-degree role is revealed in 8.1.4.

M1. *ARC* (*A*mplification, *R*eaction, *C*oordination) for modelling stability (homeostasis).

M2. *ARCH* (*ARC* with *H*ysteresis), for modelling homeostasis in case of disturbance.

M3. Analog slider for boundedly rational data input (including uncertain, fuzzy data).

8.1.4. Integration: "*Non-Algorithmic Cybernetic Modelling of Living Systems*"

Returning now to the "validation thread", that means reinterpreting the conclusions about *evaluating transdisciplinary research* (8.1.2, 8.1.3) from *PrsADT*, it becomes clear that the Cartesian cut between the research contributions of [2] and *ADT* research results is not necessary. Here it will be shown that such a categorical separation has to be revisited:

Indeed, stated recurrently as leitmotif of the "validation thread" and set up definitively through the hiatus between *product*-oriented validation (7.2) and *service*-oriented validation (7.3), the fracture is neither necessary, nor defendable. That is illustrated by the software engineering mechanisms that embody the essence of the research undertaking as a whole, since their key role cannot be distributed between "validating [2]" and "enabling CybMd". (For instance, the innermost mechanism nested in the Matryoshka structure is developed neither in ADT, nor in [2] but is a prerequisite for both validating and modelling.)

Referring to time spans, the (categorical) opposition *product-service* appears to be:

- *Useful in short range*. The contributions of [2] are systematically *tested quantitatively* in line with the industrial paradigm (7.2) to comply with [1] in *letter*. On the other hand, the contributions of *ADT* as self-contained PhD research (grouped under the label "*Non-algorithmic cybernetic modelling of living systems*) are submitted to *service-oriented validation* (7.3) to comply with [1] in *spirit* too (only a relevant application can "emphasise clearly the thesis contributions").

- *Unclear in middle range*. [2] is titled "*Bounded Rationality in Agent Orientation - "Just-In-Time" Visual Pattern Recognition*". In fact, "*Bounded Rationality*" was not ratified in "*Agent Orientation*" and "*Just-In-Time*" was not ratified in "*Visual Pattern Recognition*".

Both – and above all their synergistic blend, “BR + JIT” – were validated in an *ad hoc* sub-branch of CybMd applied to biology and ecology.

- *Unnatural in long range*. In the post-industrial era, in both engineering and transdisciplinary research, both are fundamental: BR to fight (mainly, cognitive) chaoplexity and JIT for any service providing. Hence, *a fortiori “BR + JIT”* is vital for (nontrivial) modelling as transdisciplinary service. (*A fortiori* for CybMd. *A fortiori* for *non-algorithmic* CybMd.)

In short, it can be inferred that, *de facto*, [2] should be titled: “*Bounded Rationality and “Just-In-Time” in Cybernetic Modelling of Living systems*” (The *de jure* aspects are totally outside both the scope and the competence of *ADT*.)

8.2. Future Work

The targets are ranked on time horizons. *Short and middle range* aims are labelled *intentions* (8.2.1) while *long range* aims are considered just as *ideas* to think about (8.2.2).

8.2.1. Short and Middle Range Intentions

The toolkit practicability can be substantiated only considering user priorities: the immediate need is to gather field data – above all minimal temporal information for “What-if” scenario input. As a result, the *short range intentions* refer to university education, having as horizon November 2013. The syllabi topics result from the chain of implications: evolution prediction → field (temporal) data → research personnel (partially) skilled in working with field evidence → students → ecolinguistic discourse → using (first) and developing (next) WISC → toolkit (interface) practice → laboratory work → basic knowledge about “Ecolinguistics-Based (Normative) Scenarios” (and/or similar topics).

Based on this educational and evidential framework, the *middle range intentions* – regarding software engineering development – can be carried out having as horizon, 2014:

The first major task is *reshaping the interface*, shifting from the “quantitative-testing-look” to its cardinal role of “*Field-researcher guide for devising WISC*” (4.1.1, 6.4.3). In this context, the sliders, menus and buttons used for quantitative testing should be completed/updated/relabelled for playing their key role (7.3.2). The first steps should regard:

- *Sliders.* They will be redesigned for various kinds of input languages (6.4.3) but the psychophysical laws governing cognition will be established empirically (“negociating” with students by “cut-and-try” methods, rather than based on intractable mathematics). First will be redesigned the slider for approximate, uncertain input (whether the variation follows a kind of *Weber-Fechner* law is not of immediate interest, 8.2.2).

- *Menus and buttons*. The (quasi)ecologic language (6.4.3) will be continously renewed, in the rhythm of assimilating/demolishing the main ecologic memes ([6], 6.2.4).

- *Romanian version of the interface*. Will follow the changes above.

- *Nonnumeric communication*. This first step towards semiotic-oriented interfaces should be extended from sliders to the interface as a whole. The next step in this direction will be nontextual communication (through suggestive icons proposed by students).

For the sake of continuity with the intentions above, nonverbal communication (7.3.2) will be the first idea proposed below for further transdisciplinary research.

8.2.2. Long Range Ideas

The time horizon is set between two years ahead and "if ever" not just because of future contingency, but above all, considering the problems posed by assessing service quality in post-industrial engineering research (7.1.3). The main question before engaging in any transdisciplinary research should be: "Does the matter really matter to the user?" ("user-pulled" research, 2.2.2, 3.1.1). Accordingly, below only the subset of "open questions" in [2] that was approached in *ADT* and could raise user interest (mainly from the sub-perspectives of biologic or ecologic transdisciplinary research) will be considered:

- *Semiotic-oriented interfaces*. The trend towards more meaningful, while concise user-application interaction, unavoidable in the context of WISC, should be followed, beyond the last middle-range intention (replacing text by icons), since this communication – albeit non*textual* is still *verbal* (the metaphors are *word-based*; e.g., "recycle bin", "save", most emoticons). The "psycholinguistic expressiveness" is enhanced by multimodal messages, based on suggestive ecolinguistic discourse. (For instance, a flood alert having a waterfall-like sound associated with short film sequences about a recently flooded town, has a greater "ecolinguistic impact" than a text even written with Chiller font.)

- *Memetic engineering*. For transdisciplinary research as a whole – and particularly for ecolinguistics in the context of preserving ecologic systems – the task is crucial and threefold: dismantling *vicious* memes, detaching *conflicting* memes, clarifying *fuzzy* memes:

- *Vicious memes*. Regarding the "State of the Art" about the limits of predictive models (6.2.2) as memetic engineering exercise, the path is set up for annihilating other malicious memes too. Thus, for instance, the recent memetic distortion of "service-oriented" is hard to defeat because it stems from inside software engineering (in 7.1.1 the meme was just exposed, not at all overthrown).

- *Conflicting memes*. Ecology harbours more conflicting memes than exposed when approaching the toolkit architecture (6.1.3). Transdisciplinary research can help only after basic clarifications will come from ecologists themselves.

- *Fuzzy memes*. A most difficult situation is when the meme to fight is elusive (a relevant example is the multiple confusion about "*biologic time*", labelled or not as Bergsonian, 5.4.1). Hopefully, the whole time memeplex will be carefully investigated.

References

Documents and Papers Shaping the Addendum

[1] MINISTERUL EDUCAŢIEI, CERCETĂRII, TINERETULUI ŞI SPORTULUI DIRECŢIA GENERALĂ ÎNVĂŢĂMÂNT SUPERIOR. *Ordinul MINISTRULUI EDUCAŢIEI, CERCETĂRII, TINERETULUI ŞI SPORTULUI, Nr.5743 din 12.09.2012 privind atribuirea titlului de DOCTOR*. [Appendix 3]

[2] Fabian, R.D. *Bounded Rationality in Agent Orientation - "Just-in-Time" Visual Pattern Recognition*. (Doctoral dissertation, PhD Thesis in Computer Science and Information Technology, Sibiu, 2011, Copyright: LBUS, Ralf D. Fabian, http://bcu. ulbsibiu. ro/digitale/doctorate/Ralf_Fabian_Phd_Thesis.pdf), 2011.

[3] "Proof-of-Concept" User-Validation Assessment by Angela Bănăduc [Appendix 4]

[4] Brumar, C.I., R.D. Fabian, M.-J. Manolescu, V. Chiş. Memetic Engineering for Permanent Education in Line with Sustainable Growth. *Int. J. of Computers, Communications & Control*, **7**, 5, 807-815, http://www.journal.univagora.ro/download/pdf/632.pdf, 2012.

[5] Oprean C., C.I. Brumar, R.D. Fabian, B.E. Bărbat, Memetics in Continuing Education. Rationale, Approach, Examples, *3re International conference on applied social science* (ICASS 2013) [proceedings of], Taipei, Taiwan, 2013, VOL 2, pp. 86-92, ISBN 978-1-61275-052-1.

[6] Brumar, C.I., R.D. Fabian, B.E. Bărbat. *CSITAO Carnap-like Glossary*. (Not updated since September 2011 and interrupted after February 2013.) [http://bcu.ulbsibiu.ro/digitale/doctorate/glossary_csitao.pdf].

[7] Brumar, C.I. *Memetic engineering in cybernetic modelling of chaoplex systems. Application: Ecolinguistic discourse in normative scenarios for preserving benthic communities*. (In preparation.)

[8] Anand, M., *et al*. Ecological systems as complex systems: challenges for an emerging science. *Diversity* **2**, 3, 395-410, 2010.

[9] Curtean-Bănăduc A., Olosutean H., The Trichoptera larvae communities' functional feeding structue analysis – indicator of the matter cycling process in the lotic ecosystems. A Carpathian river basin case study, *Proceedings of the 11th International Conference on Environment, Ecosystems and Development* (EED 2013), pp. 219 – 224, 2013.

[10] Curtean–Bănăduc A., Bănăduc D., Sîrbu I., Predictive modelling of biodiversity - a case study of a second order Carpathian River, *Danube News*, No. 19, Volume 11, pp. 11 – 13, 2009.

[11] Sîrbu I., Curtean-Bănăduc A., Modelling approaches of correlated lotic communities ecological synthetic measures and environmental factors, *Acta oecologica*, Vol XII, Nr. 1-2, Editura Universităţii "Lucian Blaga" din Sibiu, ISSN 1221-5015, pp. 5-15, 2005.

[12] Bărbat, B.E. DOMINO: Trivalent Logic Semantics in Bivalent Syntax Clothes. *Int. J. of Computers, Communications & Control*, **2**, 4, 303-313, 2007.

[13] Bărbat, B.E. NEWTON, HUSSERL, WIENER: A Temporal Golden Braid. (Invited paper at *ICCCC 2010*). *Abstracts of ICCCC Papers, International Conference on Computers, Communications & Control*, 12, Agora University Editing House, Oradea, 2010.

[14] Bărbat, B.E. Complex Time for Chaoplex Systems: From Newton to Wiener. *Computer Sc. J. of Moldova,* **21**, 2(62), 2013. (To appear.)

[15] Candolin, U. Population responses to anthropogenic disturbance: lessons from three-spined sticklebacks Gasterosteus aculeatus in eutrophic habitats. *Journal of Fish Biology*, **75**, 8, 2108-2121, 2009.

[16] Godin, B., J.P. Lane. (2013). Pushes and Pulls: Hi (S) tory of the Demand Pull Model of Innovation. *Science, Technology & Human Values*. Science Technology Human Values Project on the Intellectual History of Innovation Working Paper No. 13, http://www.csiic.ca/PDF/Demand-pull.pdf, 2013.

[17] Gulbranson, C.A., D.B Audretsch. *Proof of Concept Centers: Accelerating the Commercialization of University Innovation*. http://www.kauffman.org/uploadedFiles /POC _Centers_01242008.pdf, 2008.

[18] Luksha, P.O. *Memory as producer of subjective time and space in complex systems*. [http://www.mdpi.org/fis2005/], 2005.

[19] Microsoft Developer Network (MSDN). *Windows API Reference*. 3/25/2010 http://msdn.microsoft.com/en-us/library/aa383749(v=vs.85).aspx, 2010.

[20] Øhrstrøm, P., P. Hasle. "Future Contingents". *The Stanford Encyclopedia of Philosophy* (Summer 2011 Edition), Edward N. Zalta (ed.), http://plato.stanford.edu/archives/sum2011/entries/future-contingents/, 2011.

[21] Sarkar, S. "Ecology", *The Stanford Encyclopedia of Philosophy* (Spring 2009 Edition), Edward N. Zalta (ed.), plato.stanford.edu/archives/spr2009/entries/ecology/, 2009.

[22] Trenchard, H. Hysteresis in competitive bicycle pelotons. *Complex Adaptive Systems - Resilience, Robustness and Evolvability. 2010 AAAI Fall Symposium FS* , 130-137, 2010.

[23] Trost, D.C., *et al*. A model for liver homeostasis using modified mean-reverting Ornstein–Uhlenbeck process. *Computational and Mathematical Methods in Medicine*, **11**, 1, 27–47, 2010.

[24] Bruhn, T. *et.al*. *FORWARD LOOK Systems Biology: a Grand Challenge for Europe*. European Science Foundation, 2007.

[25] Wiener, N. Cybernetics. or Control and Communication in the Animal and Machine. MIT Press, Cambridge, 1948.

[26] Wiener, N. *The Human Use of Human Beings: Cybernetics and Society*. Boston: Houghton Mifflin, 1950; 2nd ed., 1954; New York: Da Capo Press, 1988.

References (published after defending the thesis, December 2nd, 2011)

[27] Barkhuizen, W.F., J.H.C. Pretorius, L. Pretorius. An integrated systems approach to risk management within a technology-driven industry, using the design structure matrix and fuzzy logic. *South African Journal of Industrial Engineering*, **23**, 2, 202-214, 2012.

[28] Block, M., *et al*. Local genetic adaptation generates latitude-specific effects of warming on predator–prey interactions. *Global Change Biology*, **19**, 3, 689-696, 2013.

[29] Borowczyk, H., Lindstedt, P., Manerowski, J. NEWTONIAN AND BERGSONIAN TIME IN TECHNICAL DIAGNOSTICS. *Journal of KONES Powertrain and Transport*, **19**, 2, 61-66, 2012.

[30] Blouin, S. Is Your World Complex? An Overview of Complexity Science and its Potential for Military Applications. *Canadian Military Journal*,**13**, 2, 26-36, 2013.

[31] Donohue, I., *et al*. On the dimensionality of ecological stability. *Ecology Letters*,**16**, 4 (April), 421-429, Wiley Online Library, 2013.

[32] Esmaeilzadeh, H., *et al*. Power challenges may end the multicore era. *Communications of the ACM* **56**, 2, 93-102, 2013.

[33] Farrar, C.R., Todd, M. D. Potential Collaborative Research topics with Korea's Agency for Defense Development. *No. LA-UR-12-24296*. Los Alamos National Laboratory (LANL). University of California Jacobs School of Engineering (UCSD) Engineering Institute, 2012.

[34] Flanagan, D. *Java in a Nutshell*. Fifth Edition, O'Reilly Media, Inc., Sebastopol, CA, 2013.

[35] Geritz, S.A.H., M. Gyllenberg. Group defence and the predator's functional response. *Journal of mathematical biology*, **66**, 4-5, 705-717, 2013.

[36] Haeufle, D.F.B., *et al*. Proof-of-concept: model based bionic muscle with hyperbolic force-velocity relation. Applied Bionics and Biomechanics, **9**, 3, 267-274, 2012.

[37] Halbig, M.C., *et al*. Evaluation of Ceramic Matrix Composite Technology for Aircraft Turbine Engine Applications. *NASA technical reports*. (The NASA server was unavailable for public access in March 2013 “while the agency conducts a review of the site's content to ensure that it does not contain technical information that is subject to U.S. export control laws and regulations”), 2013.

[38] Jinaru, A. The practice of euthanasia – a criminal or compassionate act? *Proc. of the Int. Conf*. *Knowledge and Action within the Knowledge Based Society*, **3**, (Social and Political Philosophy, Ethics, Psychology and Educational Sciences), 149-167, Baia Mare, 2010.

[39] Gosling J., *et al*. *The Java® Language Specification*. Java SE 7 Edition, 2013-02-28, http://docs.oracle.com/javase/specs/jls/se7/html/index.html, 2013 .

[40] Koboltschnig, G.R. Climate Research and Adaptation Strategies - Examples from the European Alps. In *Dating Torrential Processes On Fans and Cones: Methods and Their Application for Hazard and Risk Assessment*. (M. Schneuwly-Bollschweiler, M. Stoffel, F. Rudolf-Miklau, Eds.) Dordrecht ; New York: Springer, 2013.

[41] Koutinas, M., *et al*.Bioprocess systems engineering: transferring traditional process engineering principles to industrial biotechnology. Computational and Structural Biotechnology Journal **3**, 4 [http://dx.doi.org/10.5936/csbj.201210022], 2013.

[42] Microsoft Developer Network (MSDN). *NET Development*. http://msdn.microsoft. com/en-us/library/ff361664.aspx (retrieved May 2013).

[43] Microsoft Developer Network (MSDN). *.NET Framework 4.5.* http://msdn.microsoft.com/en-us/library/w0x726c2(v=VS.110).aspx, 2013.

[44] Microsoft Developer Network (MSDN).*Visual Studio 2012*, http://msdn.microsoft.com/en-US/vstudio, 2012.

[45] Microsoft Developer Network (MSDN). *.NET Windows Forms References*. http://msdn.microsoft.com/en-us/library/ms229608.aspx, 2013.

[46] Morozov, A., J.C. Poggiale. From spatially explicit ecological models to mean-field dynamics: The state of the art and perspectives. *Ecological Complexity*, **10**, 1-11, June 2012.

[47] Chesson, P. Scale transition theory: Its aims, motivations and predictions. *Ecological Complexity,* **10**, 52–68, June 2012.

[48] Romanczuk , P., L. Schimansky-Geier. Mean-field theory of collective motion due to velocity alignment. *Ecological Complexity*, **10**, 83-92, June 2012.

[49] Laland, K.N., *et al*. More on how and why: cause and effect in biology revisited. *Biology & Philosophy*, 1-27, Springer Science+Business Media B.V., 2012.

[50] Ma, Z., *et al*. Dynamic behaviors of a Lotka-Volterra predator-prey model incorporating a prey refuge and predator mutual interference. *Applied Mathematics and Computation*, **219**, 15, 7945-7953, 2013.

[51] Ma, Z., *et al*. The effect of prey refuge in a patchy predator-prey system. *Mathematical Biosciences*, **243**, 1, 126-130, 2013.

[52] Mocciaro Li Destri, A., G.B. Dagnino. Learning to synthesise contradictions: an Austrian approach to bridging time concepts in the strategic theory of the firm. *Int. J. of Strategic Change Management*, **4**, 2, 99-128, Inderscience Publishers, 2012.

[53] Muller, C. Linkage Mechanisms for component-based Services and ITGovernance. *Journal of Systems Integration*, **4**, 1, 3-12, 2013.

[54] Patterson, T. 9.2 Ecosystem Services. FINAL DRAFT 1/9/2013 1[http://gis.fs. fed.us/psw/publications/reports/psw_sciencesynthesis2013/psw_sciencesynthesis] 2013_ 9_2.pdf], 2013.

[55] Preisser, E.L., J.L. Orrock. The allometry of fear: interspecific relationships between body size and response to predation risk. *Ecosphere*, *3*(9), art77, http://dx.doi.org/10.1890/ES12-00084.1, 2012.

[56] Raczynski, S., Universidad Panamericana. *Object properties in discrete event simulation: Attributes, functions and non-typical behavior*. [http://www.researchgate.net/publication /235794387_Supplementary_text_for_PSM_and_Bluesss.../file/32bfe51393a1e169b9.pdf], Mexico City, Mexico, 2013.

[57] Reis, J.L. *Multicultural Competency: Verbal and Non Verbal Communication*. facweb.northseattle.edu, VerbalandNonverbalComm.ppt, 2012.

[58] Robinson, G.M., D.A. Carson. Applying landscape science to natural resource management. *Ecology and Society* **18**, 1, 32.http://dx.doi.org/10.5751/ES-05639-180132, 2013.

[59] de Rigo, D., *et al*. Toward Open Science at the European Scale: Geospatial Semantic Array Programming for Integrated Environmental Modelling. *Geophysical Research*, Abstracts 15, EGU General Assembly, 2013.

[60] Scharl, A. *et al*. Media Watch on Climate Change – Visual Analytics for Aggregating and Managing Environmental Knowledge from Online Sources. *46th Hawaii International Conference on System Sciences* , 955-964, Maui, Hawaii [http://eprints.weblyzard.com/63/] Official URL: http://www.ecoresearch.net/climate, 2013.

[61] Scheidler, J., M.J. Dapino,. Magnetostrictive aluminum composite with electrically tunable stiffness. *Proceedings of SPIE* Industrial and Commercial Applications of Smart Structures Technologies, **8690**, Abstracts, 74, 2013.

[62] Seth, A., Agarwal, H., Singla, A.R. Testing and Evaluation of Service Oriented Systems. *arXiv preprint arXiv:1302.1912*, 2013.

[63] Wang, G. *et al*.High-strain measurement using fiber Bragg grating sensors. *Proceedings of SPIE* Industrial and Commercial Applications of Smart Structures Technologies, **8690**, Abstracts, 74, 2013.

[64] Meyer, K.M., Leveau, J.H. Microbiology of the phyllosphere: a playground for testing ecological concepts. *Oecologia*, **168**, 3, 621-629, 2012.

[65] Betrie, G.D. *et al*. Predicting copper concentrations in acid mine drainage: a comparative analysis of five machine learning techniques. *Environmental Monitoring* , Springer, 2012.

[66] Sayama, H., Komura, H., Kogayu, M. Application of Hybrid Approach Based on Empirical and Physiological Concept for Predicting Pharmacokinetics in Humans - Usefulness of Exponent on Prospective Evaluation of Predictability. *Drug Metabolism and Disposition*, **41**, 2, 498-507, [http://dmd.aspetjournals.org/content/41/2/498.short], 2013.

[67] Yu, F-L.T. Two perspectives of time in economics: the neoclassical school (Newtonian) versus the Austrian school (Bergsonian). *Int. J. of Pluralism and Economics Education*, **3**, 1, 91-103, Inderscience Publishers, 2012.

[68] Wu, J., *et al*. A nanotechnology enhancement to Moore's law. *Applied Computational Intelligence and Soft Computing*, Hindawi Publishing Corporation, **2013**, Article ID 426962, 13 pages http://dx.doi.org/10.1155/2013/426962, 2013.

APPENDICES: Code, Test Results, Scenarios, and (a Few) Comments

APPENDIX 1: AV1. Code of threads relevant to validation

Framework for describing API functions necessary for minimal multithreading **[19]:**
a) API function syntax. b) Relevant parameters for the current version (*July 19-th, 2013* (commenting semantics and/or values). c) Idem for possible changes in November 2013, after testing the toolkit in laboratory work.

- ***Sleep***. a)

```
VOID WINAPI Sleep(
  _In_  DWORD dwMilliseconds
);
```

b) `dwMilliseconds`. Referred to as "**Δt**". Semantics as step in *FOR* loop: time granule (in discrete microchronic view). Values used: 10 ms, 5 sec (reasons in 7.2.3)

c) Semantics: species life span; time granule for "Kotelnikov-sampling". For fine-tuning discrete time testing it has no value limits but a set of magnitude units to select from: {*seconds*, *minutes*, *hours*, *weeks*, *months*}.

- ***Wait***. a)

```
DWORD WINAPI WaitForSingleObject(
  _In_  HANDLE hHandle,
  _In_  DWORD dwMilliseconds
);
```

b) Second `dwMilliseconds`. Referred to as "**DIU**". Semantics:

b1) "***D***elay" facet: usual time-out interval (used mainly for hysteretic delay),

b2) "***I****rreversibility*" facet: conveyed together with *Set Event* (see below),

b3) "***U****ncertainty*" facet: asynchronous time-out interval, applying the exception exit of the trivalent *IF* in (mainly) Kleene connotation ("**u**" interpreted as "temporary lack of knowledge") meaning for the user: "*UNCERTAIN in the time span given*" (it did not happened yet)

c) Possible future facet, "*Creativity*" (not outlined yet): applying the third *IF* value in (mainly) Łukasiewicz connotation ("**i**" interpreted as "unknowable" or "problematical") meaning for the user: "*VAGUE in the time span given*".

- ***Set Event***. a)

```
BOOL WINAPI SetEvent(
  _In_  HANDLE hEvent
);
```

b) `hEvent`. Referred to as "event". Semantics: "something happened in the (real world) environment", expressing thus a simple form of irreversibility (any "*Undo*" is excluded). The event has *Boolean* nature (despite its syntactic expression as *integer* event handle) and must be set by a thread *not* synchronised with the waiting thread.

c) For expressing the "**I**" facet of **DIU**, (i.e., time span in irreversible, Bergsonian, time) `SetEvent` will be exploited also for perturbation functions other than the Heaviside function, expressing pollution in the current implementation (AV2).

For exclusively technological reasons, the next two API functions are necessary too:

- ***Reset Event***. a)

```
BOOL WINAPI ResetEvent(
  _In_  HANDLE hEvent
);
```

Necessary for system (re)initialising (in testing) and for system rebirth ("big bang" in "What-if" scenarios).

- ***Create Thread***. a)

```
HANDLE WINAPI CreateThread(
  _In_opt_   LPSECURITY_ATTRIBUTES lpThreadAttributes,
  _In_       SIZE_T dwStackSize,
  _In_       LPTHREAD_START_ROUTINE lpStartAddress,
  _In_opt_   LPVOID lpParameter,
  _In_       DWORD dwCreationFlags,
  _Out_opt_  LPDWORD lpThreadId
);
```

Necessary to: a) implement minimal multithreading; b) correlate distinct time dimensions without affecting (through inappropriate synchronization) model operation; c) allow developing flexible "What-if" scenarios.

"Return" as: A) service-oriented mechanism; B) dynamic exception propagator

Java instruction syntax [39]:

```
ReturnStatement:
    return Expression_opt ;
```

- A) Return (-1).

Role: Message to the user: "service failed".

- B) Return (-4).

Role: Message to the caller: "exception occurred".

Necessary to exit the thread that emulates dynamic exception propagation: it exits a simulated exception handler, since Java syntax forbids exiting from an exception handler by "`return Expression`" (as allowed for any other programming entity).

(Because after debugging *DOMINO* not all exit codes proved to be necessary, "*-2*" and "*-3*" are yet unused and kept for future development in November 2013.)

```
HANDLE hIterationThread; // thread for Iterations
HANDLE hDecisionThread; // thred for decisions
HANDLE hTesterThread; // thread for testing

DWORD WINAPI IterationThreadProc(LPVOID lpParam);
DWORD WINAPI DecisionThreadProc(LPVOID lpParam);
DWORD WINAPI TesterThreadProc(LPVOID lpParam);
BOOL checkHomeostaticState(double dVal);
INT domino();

// macrochronic version
INT domino(){

	INT RetVal;
	// return value propagating the possible exception in DOMONO, means:
	// RetVal ==  0 - the event waited for occured before timeout duration elapsed time;
	// no exception
	// RetVal == -1 - homeostatic state not attained but still good chances (system
	// stabilizabil in due time),
	// RetVal == -2 - homeostatic state not attained microchronicaly (in current
	// interration) but still good chances (system stabilizabil in due time)
	// RetVal == -3 - homeostatic state not attained microchronicaly (forlast itteration)
	// but still good chances (system stabilizabil in due time)
	// RetVal == -4 - the event did not occur in due time; exception raised
	// (simulated/propagated via return value)

	INT dwWaitResult;
	// dwWaitResult == WAIT_TIMEOUT means: since timeout elapsed (simulated) exception is
	// rased
	// dwWaitResult == WAIT_OBJECT_0 means: event happend before timeout duration elapsed
	// (here less then dwTimeout/1000 s)

	ResetEvent(hEventObject);
	// hEventObject denotes the manner WAIT terminates

	dwWaitResult = WaitForSingleObject(hEventObject, dwTimeout);
	//dwTimeout - Wait duration (ms)

	if(dwWaitResult == WAIT_OBJECT_0)
	{
		// bivalent IF (favorable wait termination)
		if(homeostaticState){

			RetVal = 0;
			// SUCCESS, system stable
		}
		else{
			// inform caller
			// homeostatic state not attained but still good chances
			RetVal = -1;
		}
	}
	else{
		// event not occured in due time
		RetVal = -4;
	}

	return RetVal;
	// return t1-t6 == t0, hence retunr -1, -2, -3 have same meaning
}
```

```
DWORD WINAPI TesterThreadProc(LPVOID lpParam){
	Sleep(2000);
	SetEvent(hEventObject);
	return 1;
}

DWORD WINAPI DecisionThreadProc(LPVOID lpParam){
	DWORD dResult;
	BOOL bStopped = FALSE;
	dwTimeout*=1000; //ms to sec

	do{
		dResult = domino();
		switch(dResult){
		case 0:
			bStopped = TRUE;
			//MessageBox::Show("0");
			break;
		case -1:
			bStopped = FALSE;
			//MessageBox::Show("-1");
			SetEvent(hContinueEvent);
			break;
		case -4:
			bStopped = TRUE;
			MessageBox::Show("Big problem! (Simulated exception)");

		}
	}

	while(!bStopped);

	ProjectGUI::MainForm::autoRef->BeginInvoke(ProjectGUI::MainForm::autoRef-
>updateDecisionMsgDelegate,"Decision terminated: "+dResult);
	return dResult;
}
```

APPENDIX 2: AV2. Quantitative testing results

Possible semantics of the tested magnitudes:

- **A**. *PrsUb/e*: (uncontrolled) growth; gross reproduction rate. *PrsADT*: amplification factor. *PrsCSIT*: real number associated to a cybernetic amplifier ***CA*** (according to the theorem in 5.2.1).

- **β**. *PrsUb/e*: growth reduction rate; species adaptation rate to habitat perturbation, aiming to achieve a homeostatic state (not necessarily the previous one). *PrsADT*: transfer factor of the feedback network. *PrsCSIT*: real number (**|β| <1**) associated to a cybernetic amplifier ***CA*** (according to the theorem in 5.2.1).

- $\mathbf{A_{fb}}$. *PrsUb/e*: (sustainable) growth rate; reproduction rate in stable (homeostatic) state. Because of ecologic chaoplexity, it is difficult to define it, depending on various, unknown and dynamic factors as: fertility of (a particular) prey species after interference with (a particular) predator species, mortality (caused by all causes, including predation, lack of food, anthropogenic pollution, and so on), biologic evolution/decline, immigration/emigration, mutation, as well as other hard to assess rates (6.4.3). *PrsADT*: feedback amplification factor. *PrsCSIT*: similar to **A**, but in a specific, dynamic context.

- **Δt**. *PrsUb/e*: minimum time span for observable biologic changes; usually, generation lifespan. *PrsADT*: interval between two successive discrete time moments (microchronic view); **Δt --> 0** (macrochronic view). *PrsCSIT*: modelling time granularity (e.g., in deterministic Lotka-Volterra-like models it tends towards time derivatives, 5.3.2, 6.4.1).

- **DIU**. *PrsUb/e*: hysteretic delay; time required to start adapting to perturbation (usually, in case of counteracting anthropogenic disturbance). *PrsADT*: time span (in perturbation, irreversible, Bergsonian, time) between perturbation occurrence and feedback start. *PrsCSIT*: *WAIT* with non-algorithmic TIMEOUT: *WAIT* is terminated irreversibly in both cases (either system reaction or exception).

- *Heaviside function*. ***H*** (**τ**) = **0** for $\tau < \tau_0$ (the moment in **τ** when perturbation occurs)

H (**τ**) **= *ad*** **f**or $\tau > \tau_0$ (***ad***, from *a*nthropogenic *d*isturbance, is a constant representing pollution intensity; thus, hysteretic delay can be tested as function of perturbation intensity, useful to parameterise realist WISC, 7.3.4).

For testing purposes, ***H*** (**τ**) could remain undefined for $\tau = 0$. However, since for future work it is convenient to define ***H*** (**τ**) as sigmoid function limit, *H* (**τ**) = 1/2 for $\tau = \tau_0$.

For all tests the homeostatic state was considered arrived at, when the feedback amplification factor, calculated according to the Barkhausen relation (for negative feedback), $\mathbf{A_{fb} = A/(1 + \beta * A)}$, varies with less than **1%** from one iteration to the next one.

Benthic quantitative data from several sampling stations together with biotope characteristics were used in “What-if” scenarious to analyse the influence of chemical oxygen demand (COD-Mn) the density of filtrators (FDs). [9][10][11]

Sampling station id: S4; Input Viin: 116.78; A: 23; Beta: 0.2; DeltaA: 0.001

t	Afb	Output(Vo)	DeltaA	AfbGoldR	DeltaAfbGoldR	OutputGoldR	DeltaAfbAfbG
0	0	2685.94	23	0	23	2685.94	0
1	4.107142857	479.6321429	18.89285714	1.509375	21.490625	176.2648125	2.597767857
2	2.254901961	263.327451	1.852240896	0.780290792	0.729084208	91.12235864	1.474611169
3	1.554054054	181.4824324	0.700847907	0.526143791	0.254147001	61.4430719	1.027910263
4	1.18556701	138.4505155	0.368487044	0.396877568	0.129266223	46.34736237	0.788689443
5	0.958333333	111.9141667	0.227233677	0.318601583	0.078275985	37.20629288	0.63973175
6	0.804195804	93.91398601	0.154137529	0.266115702	0.052485881	31.07699174	0.538080102
7	0.692771084	80.90180723	0.11142472	0.228476821	0.037638881	26.68152318	0.464294263
8	0.608465608	71.05661376	0.084305476	0.200165769	0.028311052	23.37535847	0.40829984
9	0.54245283	63.34764151	0.066012778	0.178097345	0.022068424	20.79820796	0.364355485
10	0.489361702	57.14765957	0.053091128	0.160411823	0.017685522	18.73289273	0.328949879
11	0.445736434	52.05310078	0.043625268	0.14592145	0.014490373	17.04070695	0.299814984
12	0.409252669	47.79252669	0.036483765	0.133832086	0.012089364	15.62891106	0.275420583
13	0.378289474	44.17664474	0.030963195	0.123592631	0.010239456	14.43314739	0.254696843
14	0.351681957	41.06941896	0.026607516	0.114808652	0.008783978	13.40735441	0.236873305
15	0.328571429	38.37057143	0.023110529	0.107190413	0.007618239	12.5176964	0.221381016
16	0.308310992	36.00455764	0.020260437	0.100520291	0.006670121	11.73875963	0.207790701
17	0.29040404	33.91338384	0.017906952	0.094631661	0.00588863	11.05108542	0.195772379
18	0.274463007	32.05178998	0.015941033	0.089394781	0.005236881	10.43952249	0.185068226
19	0.260180995	30.38393665	0.014282012	0.08470712	0.00468766	9.89209751	0.175473875
20	0.247311828	28.88107527	0.012869168	0.080486586	0.004220535	9.399223463	0.166825242
21	0.235655738	27.51987705	0.01165609	0.076666667	0.003819919	8.953133333	0.158989071
22	0.225048924	26.28121331	0.010606814	0.073192908	0.003473759	8.547467798	0.151856016
23	0.215355805	25.14925094	0.009693118	0.070020296	0.003172612	8.176970136	0.14533551
24	0.206463196	24.11077199	0.00889261	0.067111296	0.002908999	7.83725719	0.139351899
25	0.198275862	23.15465517	0.008187334	0.064434365	0.002676931	7.524645144	0.133841497
26	0.190713101	22.27147595	0.007562761	0.061962797	0.002471568	7.236015394	0.128750304
27	0.18370607	21.45319489	0.007007031	0.059673832	0.002288964	6.968710156	0.124032238
28	0.177195686	20.69291217	0.006510385	0.057547957	0.002125876	6.720450375	0.119647729
29	0.171130952	19.98467262	0.006064733	0.055568339	0.001979618	6.489270594	0.115562614
30	0.165467626	19.32330935	0.005663326	0.053720387	0.001847952	6.2734668	0.111747239
31	0.160167131	18.70431755	0.005300495	0.051991389	0.001728998	6.07155436	0.108175742
32	0.155195682	18.12375169	0.004971449	0.050370216	0.001621173	5.88223381	0.104825466
33	0.15052356	17.57814136	0.004672121	0.048847087	0.001523128	5.704362864	0.101676473
34	0.146124524	17.06442186	0.004399037	0.04741337	0.001433717	5.536933346	0.098711154
35	0.141975309	16.57987654	0.004149215	0.046061415	0.001351955	5.379052069	0.095913893
36	0.138055222	16.12208884	0.003920087	0.044784423	0.001276992	5.229924896	0.093270799
37	0.134345794	15.68890187	0.003709428	0.043576326	0.001208097	5.088843378	0.090769468
38	0.130830489	15.27838453	0.003515305	0.042431696	0.00114463	4.955173504	0.088398793
39	0.127494457	14.88880266	0.003336032	0.04134566	0.001086036	4.828346174	0.086148797

40	0.124324324	14.51859459	0.003170132	0.04031383	0.00103183	4.707849094	0.084010494
41	0.121308017	14.16635021	0.003016307	0.039332248	0.000981583	4.59321987	0.081975769
42	0.118434604	13.830793	0.002873413	0.038397329	0.000934919	4.484040067	0.080037275
43	0.115694165	13.51076459	0.002740439	0.037505824	0.000891505	4.379930113	0.078188341
44	0.113077679	13.20521141	0.002616486	0.036654777	0.000851047	4.280544889	0.076422902
45	0.110576923	12.91317308	0.002500756	0.035841496	0.000813281	4.185569902	0.074735427
46	0.108184384	12.63377234	0.002392539	0.035063521	0.000777975	4.094717967	0.073120863
47	0.105893186	12.36620626	0.002291198	0.034318602	0.000744919	4.007726304	0.071574584
48	0.103697024	12.1097385	0.002196162	0.033604675	0.000713926	3.924353997	0.070092349
49	0.101590106	11.86369258	0.002106918	0.032919847	0.000684828	3.844379771	0.068670259
50	0.0995671	11.62744589	0.002023006	0.032262374	0.000657473	3.767600027	0.067304726
51	0.09762309	11.40042445	0.00194401	0.031630648	0.000631726	3.693827112	0.065992442
52	0.095753539	11.18209825	0.001869551	0.031023187	0.000607461	3.62288779	0.064730352
53	0.093954248	10.97197712	0.00179929	0.030438619	0.000584568	3.554621881	0.06351563
54	0.092221331	10.76960706	0.001732917	0.029875673	0.000562946	3.488881054	0.062345659
55	0.090551181	10.57456693	0.00167015	0.029333171	0.000542501	3.425527754	0.06121801
56	0.088940449	10.38646558	0.001610733	0.028810021	0.000523151	3.364434238	0.060130428
57	0.087386018	10.20493921	0.00155443	0.028305204	0.000504817	3.305481716	0.059080814
58	0.085884989	10.02964899	0.001501029	0.027817773	0.000487431	3.248559581	0.058067215
59	0.084434655	9.860279001	0.001450334	0.027346846	0.000470927	3.193564715	0.057087809
60	0.083032491	9.696534296	0.001402164	0.026891598	0.000455248	3.140400869	0.056140893
61	0.081676136	9.538139205	0.001356355	0.02645126	0.000440339	3.088978094	0.055224877
62	0.080363382	9.384835779	0.001312754	0.026025109	0.00042615	3.039212242	0.054338273
63	0.07909216	9.236382393	0.001271223	0.025612472	0.000412637	2.991024499	0.053479687
64	0.077860528	9.092552471	0.001231631	0.025212716	0.000399756	2.944340972	0.052647812
65	0.076666667	8.953133333	0.001193861	0.024825247	0.000387469	2.899092311	0.05184142
66	0.075508864	8.817925148	0.001157803	0.024449506	0.00037574	2.855213364	0.051059358
67	0.074385511	8.686739974	0.001123353	0.024084971	0.000364536	2.812642864	0.05030054
68	0.073295092	8.559400892	0.001090419	0.023731145	0.000353825	2.771323146	0.049563947
69	0.072236181	8.435741206	0.001058912	0.023387565	0.00034358	2.731199884	0.048848616
70	0.07120743	8.315603715	0.001028751	0.023053792	0.000333773	2.692221851	0.048153638
71	0.07020757	8.198840049	0.00099986	0.022729412	0.00032438	2.654340706	0.047478158

Sampling station id: S4; Input Viin: 116.78; A: 23; Beta: 0.03; DeltaA: 0.01;

t	Afb	Output(Vo)	DeltaA	AfbGoldR	DeltaAfbGoldR	OutputGoldR	DeltaAfbAfbG
0	0	2685.94	23	0	23	2685.94	0
1	8.812260536	1029.095785	14.18773946	1.509375	21.490625	176.2648125	7.302885536
2	5.450236967	636.478673	3.36202357	0.780290792	0.729084208	91.12235864	4.669946175
3	3.945111492	460.7101201	1.505125475	0.526143791	0.254147001	61.4430719	3.418967701
4	3.091397849	361.0134409	0.853713643	0.396877568	0.129266223	46.34736237	2.694520282
5	2.541436464	296.7889503	0.549961385	0.318601583	0.078275985	37.20629288	2.222834881
6	2.157598499	251.9643527	0.383837965	0.266115702	0.052485881	31.07699174	1.891482797
7	1.874490628	218.9030155	0.283107872	0.228476821	0.037638881	26.68152318	1.646013806
8	1.657060519	193.5115274	0.217430109	0.200165769	0.028311052	23.37535847	1.45689475
9	1.484828922	173.3983215	0.172231597	0.178097345	0.022068424	20.79820796	1.306731577
10	1.34502924	157.0725146	0.139799682	0.160411823	0.017685522	18.73289273	1.184617416

11	1.22928915	143.556387	0.11574009	0.14592145	0.014490373	17.04070695	1.0833677
12	1.131889764	132.1820866	0.097399386	0.133832086	0.012089364	15.62891106	0.998057677
13	1.04879161	122.4778842	0.083098154	0.123592631	0.010239456	14.43314739	0.925198979
14	0.977060323	114.1011045	0.071731287	0.114808652	0.008783978	13.40735441	0.862251671
15	0.914512922	106.7968191	0.0625474	0.107190413	0.007618239	12.5176964	0.80732251
16	0.859491779	100.3714499	0.055021144	0.100520291	0.006670121	11.73875963	0.758971487
17	0.810715545	94.6753613	0.048776234	0.094631661	0.00588863	11.05108542	0.716083883
18	0.767178119	89.59106071	0.043537426	0.089394781	0.005236881	10.43952249	0.677783338
19	0.728078506	85.02500791	0.039099613	0.08470712	0.00468766	9.89209751	0.643371386
20	0.692771084	80.90180723	0.035307422	0.080486586	0.004220535	9.399223463	0.612284499
21	0.660729675	77.16001149	0.032041409	0.076666667	0.003819919	8.953133333	0.584063009
22	0.631521142	73.74903899	0.029208533	0.073192908	0.003473759	8.547467798	0.558328234
23	0.604785696	70.62687352	0.026735447	0.070020296	0.003172612	8.176970136	0.5347654
24	0.580221998	67.75832492	0.024563698	0.067111296	0.002908999	7.83725719	0.513110702
25	0.557575758	65.11369697	0.02264624	0.064434365	0.002676931	7.524645144	0.493141393
26	0.536630891	62.66775548	0.020944866	0.061962797	0.002471568	7.236015394	0.474668095
27	0.517202609	60.39892062	0.019428283	0.059673832	0.002288964	6.968710156	0.457528776
28	0.499131944	58.28862847	0.018070664	0.057547957	0.002125876	6.720450375	0.441583988
29	0.482281401	56.32082198	0.016850544	0.055568339	0.001979618	6.489270594	0.426713062
30	0.46653144	54.48154158	0.015749961	0.053720387	0.001847952	6.2734668	0.412811053
31	0.451777647	52.7585936	0.014753793	0.051991389	0.001728998	6.07155436	0.399786258
32	0.437928408	51.14127951	0.013849239	0.050370216	0.001621173	5.88223381	0.387558192
33	0.424903011	49.62017366	0.013025397	0.048847087	0.001523128	5.704362864	0.376055924
34	0.412630068	48.18693936	0.012272943	0.04741337	0.001433717	5.536933346	0.365216698
35	0.401046207	46.83417611	0.011583861	0.046061415	0.001351955	5.379052069	0.354984792
36	0.39009498	45.55529172	0.010951228	0.044784423	0.001276992	5.229924896	0.345310557
37	0.379725937	44.34439491	0.010369043	0.043576326	0.001208097	5.088843378	0.336149611
38	0.369893857	43.19620457	0.00983208	0.042431696	0.00114463	4.955173504	0.32746216

Sampling station id: S7; Input Viin: 202.97; A: 23; Beta: 0.03; DeltaA: 0.001

t	Afb	Output(Vo)	DeltaA	AfbGoldR	DeltaAfbGoldR	OutputGoldR	DeltaAfbAfbG
0	0	4668.31	23	0	23	4668.31	0
1	13.60946746	2762.313609	9.390532544	1.509375	21.490625	306.3578438	12.10009246
2	9.663865546	1961.47479	3.945601909	0.780290792	0.729084208	158.375622	8.883574755
3	7.491856678	1520.62215	2.172008869	0.526143791	0.254147001	106.7914052	6.965712887
4	6.117021277	1241.571809	1.374835401	0.396877568	0.129266223	80.55423993	5.720143709
5	5.168539326	1049.058427	0.948481951	0.318601583	0.078275985	64.66656332	4.849937743
6	4.474708171	908.2315175	0.693831155	0.266115702	0.052485881	54.01350413	4.208592469
7	3.945111492	800.7392796	0.529596679	0.228476821	0.037638881	46.3739404	3.716634671
8	3.527607362	715.9984663	0.41750413	0.200165769	0.028311052	40.62764608	3.327441593
9	3.19001387	647.4771151	0.337593492	0.178097345	0.022068424	36.14841814	3.011916524
10	2.911392405	590.9253165	0.278621465	0.160411823	0.017685522	32.55878778	2.750980582
11	2.677532014	543.4586729	0.233860391	0.14592145	0.014490373	29.61767674	2.531610564
12	2.478448276	503.0506466	0.199083738	0.133832086	0.012089364	27.16389859	2.344616189
13	2.306920762	468.2357071	0.171527514	0.123592631	0.010239456	25.08559621	2.183328132
14	2.157598499	437.9277674	0.149322263	0.114808652	0.008783978	23.30271215	2.042789847

15	2.026431718	411.3048458	0.131166781	0.107190413	0.007618239	21.75643808	1.919241305
16	1.910299003	387.7333887	0.116132715	0.100520291	0.006670121	20.40260354	1.809778712
17	1.806755695	366.7172035	0.103543308	0.094631661	0.00588863	19.20738832	1.712124034
18	1.713859911	347.8621461	0.092895785	0.089394781	0.005236881	18.14445863	1.62446513
19	1.63004961	330.8511694	0.0838103	0.08470712	0.00468766	17.19300421	1.54534249
20	1.554054054	315.4263514	0.075995556	0.080486586	0.004220535	16.33636227	1.473567468
21	1.484828922	301.3757263	0.069225132	0.076666667	0.003819919	15.56103333	1.408162255
22	1.421508035	288.5234858	0.063320887	0.073192908	0.003473759	14.85596454	1.348315127
23	1.363366924	276.7225845	0.058141111	0.070020296	0.003172612	14.21201943	1.293346628
24	1.309794989	265.8490888	0.053571935	0.067111296	0.002908999	13.62157982	1.242683692
25	1.260273973	255.7978082	0.049521016	0.064434365	0.002676931	13.07824306	1.195839608
26	1.21436114	246.4788807	0.045912832	0.061962797	0.002471568	12.57658884	1.152398344
27	1.171676006	237.815079	0.042685134	0.059673832	0.002288964	12.11199778	1.112002174
28	1.131889764	229.7396654	0.039786242	0.057547957	0.002125876	11.68050876	1.074341807
29	1.094716802	222.1946692	0.037172962	0.055568339	0.001979618	11.27870571	1.039148463
30	1.059907834	215.1294931	0.034808967	0.053720387	0.001847952	10.90362696	1.006187447
31	1.027244305	208.4997767	0.032663529	0.051991389	0.001728998	10.55269214	0.975252917
32	0.996533795	202.2664645	0.03071051	0.050370216	0.001621173	10.22364272	0.94616358
33	0.967606226	196.3950358	0.028927569	0.048847087	0.001523128	9.914493325	0.918759139
34	0.940310711	190.8548651	0.027295515	0.04741337	0.001433717	9.623491705	0.892897341
35	0.914512922	185.6186879	0.025797789	0.046061415	0.001351955	9.349085447	0.868451507
36	0.890092879	180.6621517	0.024420043	0.044784423	0.001276992	9.089894298	0.845308456
37	0.866943083	175.9634376	0.023149796	0.043576326	0.001208097	8.844686936	0.823366757
38	0.844966936	171.502939	0.021976147	0.042431696	0.00114463	8.612361416	0.80253524
39	0.824077392	167.2629882	0.020889544	0.04134566	0.001086036	8.391928608	0.782731732
40	0.804195804	163.2276224	0.019881587	0.04031383	0.00103183	8.182498122	0.763881974
41	0.785250939	159.3823831	0.018944865	0.039332248	0.000981583	7.983266287	0.745918691
42	0.767178119	155.7141428	0.01807282	0.038397329	0.000934919	7.793505843	0.72878079
43	0.749918487	152.2109553	0.017259632	0.037505824	0.000891505	7.612557074	0.712412663
44	0.733418367	148.861926	0.01650012	0.036654777	0.000851047	7.439820141	0.69676359
45	0.717628705	145.6570983	0.015789662	0.035841496	0.000813281	7.274748442	0.681787209
46	0.702504582	142.5873549	0.015124124	0.035063521	0.000777975	7.116842831	0.667441061
47	0.688004786	139.6443314	0.014499795	0.034318602	0.000744919	6.965646582	0.653686184
48	0.674091442	136.82034	0.013913344	0.033604675	0.000713926	6.820740973	0.640486767
49	0.660729675	134.1083022	0.013361767	0.032919847	0.000684828	6.681741412	0.627809828
50	0.647887324	131.5016901	0.012842351	0.032262374	0.000657473	6.548294035	0.61562495
51	0.635534678	128.9944736	0.012352646	0.031630648	0.000631726	6.420072692	0.60390403
52	0.623644252	126.5810738	0.011890426	0.031023187	0.000607461	6.296776286	0.592621065
53	0.612190578	124.2563215	0.011453674	0.030438619	0.000584568	6.178126418	0.581751959
54	0.601150026	122.0154208	0.011040551	0.029875673	0.000562946	6.063865281	0.571274353
55	0.590500642	119.8539153	0.010649384	0.029333171	0.000542501	5.953753796	0.56116747
56	0.580221998	117.7676589	0.010278644	0.028810021	0.000523151	5.847569937	0.551411977
57	0.570295066	115.7527895	0.009926932	0.028305204	0.000504817	5.745107243	0.541989862
58	0.560702097	113.8057045	0.009592969	0.027817773	0.000487431	5.646173472	0.532884323
59	0.551426516	111.92304	0.00927558	0.027346846	0.000470927	5.550589401	0.52407967
60	0.54245283	110.1016509	0.008973686	0.026891598	0.000455248	5.45818774	0.515561232

61	0.533766535	108.3385936	0.008686295	0.02645126	0.000440339	5.368812158	0.507315276
62	0.525354043	106.6311101	0.008412492	0.026025109	0.00042615	5.282316396	0.499328934
63	0.517202609	104.9766134	0.008151434	0.025612472	0.000412637	5.198563474	0.491590136
64	0.509300266	103.3726749	0.007902343	0.025212716	0.000399756	5.117424962	0.48408755
65	0.501635769	101.817012	0.007664497	0.024825247	0.000387469	5.038780325	0.476810522
66	0.494198539	100.3074774	0.00743723	0.024449506	0.00037574	4.962516325	0.469749032
67	0.486978615	98.84204954	0.007219924	0.024084971	0.000364536	4.888526479	0.462893645
68	0.479966611	97.41882304	0.007012004	0.023731145	0.000353825	4.816710559	0.456235466
69	0.473153672	96.03600082	0.006812939	0.023387565	0.00034358	4.746974143	0.449766107
70	0.46653144	94.69188641	0.006622232	0.023053792	0.000333773	4.679228199	0.443477648
71	0.460092018	93.38487698	0.006439422	0.022729412	0.00032438	4.613388706	0.437362607
72	0.45382794	92.11345699	0.006264078	0.022414033	0.000315379	4.549376305	0.431413907
73	0.447732139	90.87619233	0.006095801	0.022107287	0.000306746	4.487115983	0.425624853
74	0.441797925	89.67172493	0.005934214	0.021808823	0.000298464	4.426536777	0.419989103
75	0.436018957	88.49876777	0.005778968	0.021518311	0.000290512	4.367571505	0.414500647
76	0.430389222	87.3561003	0.005629736	0.021235436	0.000282874	4.310156518	0.409153785
77	0.424903011	86.2425642	0.00548621	0.020959903	0.000275534	4.25423147	0.403943108
78	0.419554907	85.15705947	0.005348104	0.020691428	0.000268475	4.199739108	0.398863479
79	0.414339759	84.0985408	0.005215148	0.020429744	0.000261684	4.146625074	0.393910015
80	0.409252669	83.06601423	0.00508709	0.020174596	0.000255148	4.094837726	0.389078073
81	0.404288979	82.05853401	0.00496369	0.019925743	0.000248853	4.04432797	0.384363236
82	0.399444251	81.07519972	0.004844727	0.019682954	0.000242789	3.995049106	0.379761298
83	0.394714261	80.1151536	0.00472999	0.01944601	0.000236944	3.946956679	0.375268251
84	0.39009498	79.17757802	0.004619282	0.019214703	0.000231307	3.900008354	0.370880276
85	0.385582565	78.26169321	0.004512415	0.018988835	0.000225869	3.854163784	0.36659373
86	0.381173351	77.36675505	0.004409214	0.018768214	0.00022062	3.809384496	0.362405137
87	0.376863837	76.49205309	0.004309514	0.018552662	0.000215553	3.765633787	0.358311176
88	0.37265068	75.63690862	0.004213157	0.018342004	0.000210658	3.722876619	0.354308676
89	0.368530684	74.80067297	0.004119996	0.018136077	0.000205927	3.681079528	0.350394607
90	0.364500792	73.98272583	0.004029892	0.017934722	0.000201355	3.640210538	0.34656607
91	0.360558081	73.18247374	0.003942711	0.017737789	0.000196933	3.600239075	0.342820292
92	0.356699752	72.39934864	0.003858329	0.017545134	0.000192655	3.561135893	0.339154618
93	0.352923124	71.63280651	0.003776628	0.017356619	0.000188515	3.522873006	0.335566505
94	0.34922563	70.88232615	0.003697494	0.017172112	0.000184507	3.485423614	0.332053518
95	0.345604808	70.14740796	0.003620822	0.016991487	0.000180626	3.448762049	0.328613322
96	0.342058299	69.42757287	0.00354651	0.016814621	0.000176865	3.412863708	0.325243677
97	0.338583836	68.72236125	0.003474462	0.0166414	0.000173221	3.377705003	0.321942436
98	0.335179248	68.03133197	0.003404588	0.016471712	0.000169689	3.343263309	0.318707536
99	0.331842447	67.35406146	0.003336801	0.016305449	0.000166263	3.309516913	0.315536998
100	0.328571429	66.69014286	0.003271018	0.016142509	0.00016294	3.276444972	0.31242892
101	0.325364267	66.03918517	0.003207162	0.015982793	0.000159716	3.244027465	0.309381474
102	0.322219109	65.40081255	0.003145158	0.015826207	0.000156586	3.212245159	0.306392902
103	0.319134175	64.77466352	0.003084934	0.015672659	0.000153548	3.181079564	0.303461516
104	0.316107752	64.16039032	0.003026424	0.015522062	0.000150597	3.150512903	0.30058569
105	0.313138189	63.55765827	0.002969562	0.015374332	0.00014773	3.120528075	0.297763858
106	0.310223901	62.96614513	0.002914289	0.015229387	0.000144945	3.091108624	0.294994514

107	0.307363357	62.38554056	0.002860544	0.015087149	0.000142237	3.062238708	0.292276208
108	0.304555085	61.81554555	0.002808272	0.014947544	0.000139605	3.033903073	0.28960754
109	0.301797664	61.25587193	0.00275742	0.014810499	0.000137045	3.006087023	0.286987165
110	0.299089727	60.70624187	0.002707937	0.014675944	0.000134555	2.978776397	0.284413783
111	0.296429952	60.16638742	0.002659775	0.014543812	0.000132132	2.951957543	0.28188614
112	0.293817067	59.63605008	0.002612885	0.014414038	0.000129774	2.925617297	0.279403029
113	0.291249842	59.11498037	0.002567225	0.014286559	0.000127479	2.89974296	0.276963282
114	0.28872709	58.60293748	0.002522752	0.014161316	0.000125244	2.87432228	0.274565774
115	0.286247666	58.09968886	0.002479424	0.014038249	0.000123067	2.849343428	0.272209417
116	0.283810464	57.60500987	0.002437202	0.013917303	0.000120946	2.824794986	0.269893161
117	0.281414413	57.11868347	0.002396051	0.013798423	0.00011888	2.800665924	0.26761599
118	0.279058481	56.64049988	0.002355932	0.013681557	0.000116866	2.776945585	0.265376924
119	0.276741668	56.17025629	0.002316813	0.013566654	0.000114903	2.753623673	0.263175014
120	0.274463007	55.70775656	0.002278661	0.013453664	0.000112989	2.730690231	0.261009343
121	0.272221565	55.25281098	0.002241442	0.013342541	0.000111123	2.708135635	0.258879023
122	0.270016436	54.80523597	0.002205129	0.013233239	0.000109302	2.685950574	0.256783197
123	0.267846745	54.36485385	0.002169691	0.013125713	0.000107526	2.664126039	0.254721032
124	0.265711645	53.93149261	0.0021351	0.013019921	0.000105793	2.642653314	0.252691724
125	0.263610315	53.50498567	0.00210133	0.01291582	0.000104101	2.62152396	0.250694495
126	0.26154196	53.08517171	0.002068355	0.01281337	0.000102449	2.600729805	0.24872859
127	0.259505811	52.67189439	0.00203615	0.012712534	0.000100837	2.580262936	0.246793277
128	0.25750112	52.26500224	0.002004691	0.012613271	9.93E-05	2.560115687	0.244887848
129	0.255527164	51.86434841	0.001973956	0.012515547	9.77E-05	2.540280628	0.243011616
130	0.253583241	51.46979052	0.001943922	0.012419326	9.62E-05	2.520750559	0.241163916
131	0.251668673	51.0811905	0.001914569	0.012324573	9.48E-05	2.5015185	0.2393441
132	0.249782798	50.69841442	0.001885875	0.012231254	9.33E-05	2.48257768	0.237551543
133	0.247924976	50.32133233	0.001857822	0.012139338	9.19E-05	2.463921534	0.235785637
134	0.246094586	49.9498181	0.00183039	0.012048794	9.05E-05	2.445543692	0.234045792
135	0.244291025	49.58374934	0.001803561	0.01195959	8.92E-05	2.427437974	0.232331435
136	0.242513707	49.22300717	0.001777318	0.011871697	8.79E-05	2.409598378	0.23064201
137	0.240762064	48.86747619	0.001751643	0.011785087	8.66E-05	2.392019081	0.228976977
138	0.239035544	48.51704427	0.001726521	0.011699731	8.54E-05	2.374694426	0.227335812
139	0.237333609	48.17160252	0.001701935	0.011615603	8.41E-05	2.357618922	0.225718006
140	0.235655738	47.83104508	0.001677871	0.011532676	8.29E-05	2.34078723	0.224123062
141	0.234001424	47.4952691	0.001654313	0.011450925	8.18E-05	2.324194168	0.2225505
142	0.232370176	47.16417458	0.001631249	0.011370324	8.06E-05	2.307834695	0.220999852
143	0.230761513	46.83766429	0.001608663	0.01129085	7.95E-05	2.291703913	0.219470663
144	0.22917497	46.51564368	0.001586543	0.01121248	7.84E-05	2.275797061	0.21796249
145	0.227610094	46.19802078	0.001564876	0.01113519	7.73E-05	2.260109508	0.216474904
146	0.226066444	45.88470611	0.00154365	0.011058958	7.62E-05	2.244636749	0.215007486
147	0.224543591	45.57561261	0.001522853	0.010983763	7.52E-05	2.229374403	0.213559828
148	0.223041117	45.27065555	0.001502474	0.010909584	7.42E-05	2.214318207	0.212131533
149	0.221558617	44.96975243	0.0014825	0.0108364	7.32E-05	2.199464013	0.210722217
150	0.220095694	44.67282297	0.001462923	0.010764191	7.22E-05	2.184807782	0.209331503
151	0.218651963	44.37978895	0.001443731	0.010692938	7.13E-05	2.170345583	0.207959025
152	0.217227049	44.09057423	0.001424914	0.010622622	7.03E-05	2.156073589	0.206604427

153	0.215820587	43.80510463	0.001406462	0.010553225	6.94E-05	2.14198807	0.205267362
154	0.214432221	43.52330785	0.001388367	0.010484729	6.85E-05	2.128085397	0.203947492
155	0.213061603	43.24511348	0.001370618	0.010417116	6.76E-05	2.114362033	0.202644487
156	0.211708395	42.97045287	0.001353208	0.01035037	6.67E-05	2.100814529	0.201358025
157	0.210372267	42.69925912	0.001336127	0.010284473	6.59E-05	2.087439528	0.200087794
158	0.209052899	42.43146701	0.001319368	0.010219411	6.51E-05	2.074233756	0.198833489
159	0.207749977	42.16701292	0.001302922	0.010155166	6.42E-05	2.06119402	0.197594812
160	0.206463196	41.90583483	0.001286782	0.010091724	6.34E-05	2.04831721	0.196371472
161	0.205192256	41.64787225	0.001270939	0.01002907	6.27E-05	2.035600291	0.195163186
162	0.203936868	41.39306615	0.001255388	0.009967189	6.19E-05	2.023040302	0.193969679
163	0.202696748	41.14135895	0.00124012	0.009906067	6.11E-05	2.010634357	0.192790681
164	0.201471619	40.89269446	0.001225129	0.00984569	6.04E-05	1.99837964	0.191625929
165	0.20026121	40.64701785	0.001210409	0.009786044	5.96E-05	1.986273401	0.190475166
166	0.199065259	40.40427558	0.001195951	0.009727117	5.89E-05	1.974312959	0.189338142
167	0.197883507	40.16441538	0.001181752	0.009668895	5.82E-05	1.962495696	0.188214611
168	0.196715703	39.92738625	0.001167804	0.009611366	5.75E-05	1.950819056	0.187104337
169	0.195561602	39.69313834	0.001154101	0.009554518	5.68E-05	1.939280543	0.186007084
170	0.194420964	39.46162299	0.001140638	0.009498338	5.62E-05	1.927877721	0.184922625
171	0.193293554	39.23279267	0.00112741	0.009442815	5.55E-05	1.916608211	0.183850739
172	0.192179144	39.00660094	0.00111441	0.009387938	5.49E-05	1.905469688	0.182791207
173	0.191077511	38.78300241	0.001101633	0.009333694	5.42E-05	1.894459883	0.181743817
174	0.189988435	38.56195275	0.001089076	0.009280074	5.36E-05	1.883576575	0.180708362
175	0.188911704	38.34340862	0.001076731	0.009227066	5.30E-05	1.872817598	0.179684638
176	0.187847109	38.12732767	0.001064596	0.00917466	5.24E-05	1.862180834	0.178672448
177	0.186794445	37.91366848	0.001052664	0.009122847	5.18E-05	1.851664211	0.177671598
178	0.185753513	37.70239057	0.001040932	0.009071615	5.12E-05	1.841265706	0.176681898
179	0.184724119	37.49345434	0.001029395	0.009020956	5.07E-05	1.83098334	0.175703163
180	0.18370607	37.28682109	0.001018048	0.008970859	5.01E-05	1.820815178	0.174735212
181	0.182699182	37.08245294	0.001006888	0.008921315	4.95E-05	1.810759328	0.173777867
182	0.181703271	36.88031285	0.000995911	0.008872316	4.90E-05	1.800813939	0.172830955

Sampling station id: S40; Input Viin: 189.73; A: 23.6; Beta: 0.08; DeltaA: 0.001

t	Afb	Output(Vo)	DeltaA	AfbGoldR	DeltaAfbGoldR	OutputGoldR	DeltaAfbAfbG
0	0	4477.628	23.6	0	23.6	4477.628	0
1	8.171745152	1550.425208	15.42825485	1.511897498	22.0881025	286.8523124	6.659847654
2	4.941373534	937.5268007	3.230371618	0.780964387	0.730933111	148.1723731	4.160409147
3	3.541416567	671.9129652	1.399956968	0.526449968	0.254514419	99.88335245	3.014966598
4	2.7595884	523.5767072	0.781828166	0.397051755	0.129398214	75.33262939	2.362536646
5	2.260536398	428.8915709	0.499052002	0.318713826	0.078337928	60.46957428	1.941822572
6	1.914341337	363.2079818	0.346195062	0.266194006	0.052519821	50.50498872	1.648147331
7	1.660101294	314.9710186	0.254240042	0.228534538	0.037659467	43.35985797	1.431566756
8	1.465474416	278.044461	0.194626878	0.200210067	0.028324471	37.98585602	1.265264349
9	1.311694086	248.867719	0.15378033	0.178132413	0.022077654	33.79706276	1.133561673
10	1.187122736	225.2327968	0.12457135	0.160440272	0.017692141	30.44033279	1.026682464
11	1.084160235	205.6977214	0.102962501	0.145944991	0.014495281	27.69014312	0.938215244
12	0.997632736	189.280859	0.086527499	0.133851888	0.012093103	25.39571868	0.863780848

13	0.923896023	175.2907924	0.073736713	0.123609518	0.01024237	23.45243378	0.800286505
14	0.860309128	163.2264509	0.063586895	0.114823224	0.008786294	21.78541031	0.745485904
15	0.804911323	152.7158254	0.055397805	0.107203115	0.007620109	20.33964698	0.697708208
16	0.756216355	143.476929	0.048694969	0.100531462	0.006671653	19.07383423	0.655684893
17	0.713077109	135.2921199	0.043139246	0.094641561	0.0058899	17.95634343	0.618435548
18	0.6745941	127.9907386	0.038483009	0.089403615	0.005237946	16.96254789	0.585190485
19	0.640052072	121.4370796	0.034542028	0.084715052	0.004688563	16.07298691	0.55533702
20	0.608875129	115.5218782	0.031176943	0.080493747	0.004221306	15.27207861	0.528381382
21	0.580594371	110.15617	0.028280758	0.076673164	0.003820583	14.54719948	0.503921207
22	0.554824149	105.2667858	0.025770222	0.07319883	0.003474334	13.88801406	0.481625319
23	0.531244372	100.7929948	0.023579777	0.070025716	0.003173115	13.28597903	0.461218657
24	0.509587148	96.6839696	0.021657224	0.067116275	0.00290944	12.73397091	0.442470873
25	0.489626556	92.89684647	0.019960592	0.064438955	0.002677321	12.22600286	0.425187601
26	0.471170739	89.39522441	0.018455817	0.061967041	0.002471914	11.75700668	0.409203699
27	0.454055718	86.14799138	0.017115021	0.059677769	0.002289272	11.3226631	0.394377949
28	0.438140502	83.12839745	0.015915216	0.057551618	0.002126151	10.91926841	0.380588884
29	0.4233032	80.31331611	0.014837302	0.055571752	0.001979865	10.54362853	0.367731448
30	0.40943789	77.68265094	0.01386531	0.053723577	0.001848175	10.19297431	0.355714313
31	0.39645209	75.21885499	0.012985801	0.051994377	0.001729201	9.864893095	0.344457713
32	0.384264687	72.90653901	0.012187403	0.050373021	0.001621356	9.557273189	0.333891666
33	0.372804246	70.73214963	0.011460441	0.048849725	0.001523296	9.268258324	0.323954521
34	0.362007608	68.68370352	0.010796638	0.047415855	0.00143387	8.996210176	0.314591753
35	0.351818724	66.75056649	0.010188884	0.046063761	0.001352094	8.739677293	0.305754963
36	0.342187681	64.92326876	0.009631043	0.04478664	0.001277121	8.497369192	0.297401041
37	0.333069888	63.19334989	0.009117793	0.043578425	0.001208215	8.268134639	0.289491463
38	0.324425382	61.55322776	0.008644506	0.042433687	0.001144739	8.05094337	0.281991696
39	0.316218244	59.99608747	0.008207138	0.04134755	0.001086137	7.844870601	0.274870695
40	0.3084161	58.51578672	0.007802144	0.040315627	0.001031923	7.649083869	0.268100474
41	0.300989695	57.10677482	0.007426405	0.039333958	0.000981669	7.462831791	0.261655737
42	0.293912524	55.76402311	0.007077171	0.038398959	0.000934999	7.285434429	0.255513565
43	0.287160518	54.48296505	0.006752006	0.037507379	0.00089158	7.116274994	0.249653139
44	0.280711771	53.25944429	0.006448747	0.036656262	0.000851116	6.954792681	0.244055508
45	0.274546301	52.08966961	0.00616547	0.035842916	0.000813346	6.800476459	0.238703385
46	0.268645843	50.97017576	0.005900458	0.03506488	0.000778036	6.65285967	0.233580963
47	0.26299367	49.89778907	0.005652172	0.034319904	0.000744976	6.511515311	0.228673767
48	0.257574435	48.86959749	0.005419236	0.033605924	0.00071398	6.376051914	0.223968511
49	0.252374027	47.88292412	0.005200408	0.032921045	0.000684878	6.246109923	0.219452982
50	0.247379455	46.93530398	0.004994572	0.032263525	0.000657521	6.121358505	0.21511593
51	0.242578735	46.02446345	0.00480072	0.031631754	0.00063177	6.001492743	0.210946981
52	0.237960797	45.14830201	0.004617938	0.031024251	0.000607503	5.886231142	0.206936546
53	0.233515396	44.30487612	0.004445401	0.030439643	0.000584608	5.775313425	0.203075753
54	0.229233041	43.4923848	0.004282356	0.029876659	0.000562983	5.668498571	0.199356381
55	0.225104922	42.70915681	0.004128119	0.029334123	0.000542537	5.565563066	0.195770799
56	0.221122854	41.95363916	0.003982067	0.028810938	0.000523184	5.466299341	0.192311916
57	0.217279222	41.22438683	0.003843632	0.02830609	0.000504849	5.370514376	0.188973133
58	0.21356693	40.52005357	0.003712293	0.027818629	0.000487461	5.278028447	0.185748301

59	0.209979358	39.83938359	0.003587572	0.027347673	0.000470956	5.188674002	0.182631685
60	0.206510326	39.18120406	0.003469032	0.026892398	0.000455275	5.102294644	0.179617928
61	0.203154053	38.54441843	0.003356273	0.026452033	0.000440365	5.018744222	0.17670202
62	0.19990513	37.92800027	0.003248923	0.026025858	0.000426175	4.937886003	0.173879272
63	0.196758487	37.33098779	0.003146642	0.025613197	0.000412661	4.859591925	0.17114529
64	0.19370937	36.75247882	0.003049117	0.025213419	0.000399779	4.783741924	0.168495952
65	0.190753314	36.19162625	0.002956056	0.024825928	0.000387491	4.710223313	0.165927386
66	0.187886122	35.64763391	0.002867192	0.024450167	0.000375761	4.638930231	0.163435955
67	0.185103846	35.11975278	0.002782276	0.024085612	0.000364555	4.569763129	0.161018235
68	0.18240277	34.60727756	0.002701076	0.023731768	0.000353844	4.502628308	0.158671002
69	0.179779389	34.10954354	0.002623381	0.02338817	0.000343598	4.437437495	0.156391219
70	0.1772304	33.6259237	0.00254899	0.02305438	0.00033379	4.374107457	0.15417602
71	0.174752681	33.15582608	0.002477719	0.022729983	0.000324397	4.312559646	0.152022698
72	0.172343284	32.69869136	0.002409396	0.022414588	0.000315394	4.252719872	0.149928696
73	0.169999424	32.25399066	0.002343861	0.022107827	0.000306762	4.194518008	0.147891597
74	0.16771846	31.82122349	0.002280963	0.021809349	0.000298478	4.137887714	0.145909112
75	0.165497896	31.39991585	0.002220564	0.021518822	0.000290526	4.082766185	0.143979074
76	0.163335363	30.98961852	0.002162533	0.021235935	0.000282888	4.029093916	0.142099429
77	0.161228617	30.58990545	0.002106747	0.020960388	0.000275546	3.976814495	0.140268228
78	0.159175525	30.20037231	0.002053092	0.020691901	0.000268487	3.925874396	0.138483624
79	0.157174064	29.82063509	0.002001461	0.020430205	0.000261696	3.876222803	0.136743859
80	0.15522231	29.45032886	0.001951754	0.020175046	0.000255159	3.827811439	0.135047264
81	0.153318435	29.0891066	0.001903875	0.019926181	0.000248864	3.780594408	0.133392253
82	0.151460697	28.73663809	0.001857737	0.019683382	0.0002428	3.734528052	0.131777315
83	0.149647441	28.39260894	0.001813256	0.019446428	0.000236954	3.689570813	0.130201013
84	0.147877087	28.05671964	0.001770354	0.019215112	0.000231317	3.645683113	0.128661975
85	0.14614813	27.72868467	0.001728957	0.018989233	0.000225878	3.602827235	0.127158896
86	0.144459135	27.40823172	0.001688995	0.018768604	0.000220629	3.560967212	0.125690531
87	0.142808733	27.09510093	0.001650402	0.018553042	0.000215561	3.520068732	0.124255691
88	0.141195616	26.78904418	0.001613117	0.018342376	0.000210666	3.48009904	0.12285324
89	0.139618534	26.48982441	0.001577082	0.01813644	0.000205936	3.441026853	0.121482093
90	0.138076293	26.19721507	0.001542241	0.017935078	0.000201363	3.402822278	0.120141215
91	0.136567751	25.91099949	0.001508542	0.017738137	0.000196941	3.365456732	0.118829614
92	0.135091817	25.63097037	0.001475935	0.017545474	0.000192662	3.328902877	0.117546342
93	0.133647443	25.35692928	0.001444374	0.017356952	0.000188522	3.293134548	0.11629049
94	0.132233628	25.08868618	0.001413815	0.017172438	0.000184514	3.258126694	0.11506119
95	0.130849412	24.82605899	0.001384215	0.016991806	0.000180632	3.223855316	0.113857606
96	0.129493876	24.56887318	0.001355536	0.016814934	0.000176872	3.190297417	0.112678943
97	0.128166138	24.31696138	0.001327738	0.016641706	0.000173228	3.157430945	0.111524432
98	0.126865351	24.07016299	0.001300787	0.016472012	0.000169695	3.125234749	0.110393339
99	0.125590702	23.8283239	0.001274649	0.016305743	0.000166269	3.093688533	0.109284959
100	0.124341412	23.5912961	0.00124929	0.016142797	0.000162946	3.062772809	0.108198615
101	0.123116731	23.35893744	0.001224681	0.015983075	0.000159721	3.032468863	0.107133656
102	0.12191594	23.13111129	0.001200791	0.015826483	0.000156592	3.002758713	0.106089456
103	0.120738346	22.90768633	0.001177594	0.01567293	0.000153553	2.973625077	0.105065415
104	0.119583283	22.68853622	0.001155063	0.015522328	0.000150602	2.945051334	0.104060954

105	0.11845011	22.47353945	0.001133172	0.015374593	0.000147735	2.917021498	0.103075518
106	0.117338212	22.26257905	0.001111898	0.015229643	0.00014495	2.889520186	0.102108569
107	0.116246995	22.05554242	0.001091217	0.015087401	0.000142242	2.862532589	0.101159594
108	0.115175887	21.85232109	0.001071108	0.014947791	0.00013961	2.836044445	0.100228096
109	0.114124337	21.65281055	0.00105155	0.014810742	0.00013705	2.810042018	0.099313596
110	0.113091815	21.4569101	0.001032522	0.014676182	0.000134559	2.784512067	0.098415633
111	0.112077809	21.26452262	0.001014007	0.014544046	0.000132136	2.759441833	0.097533763
112	0.111081824	21.07555447	0.000995985	0.014414268	0.000129778	2.734819007	0.096667556

Sampling station id: S19; Input Viin: 146.56; A: 23.2; Beta: 0.002; DeltaA: 0.01

t	Afb	Output(Vo)	DeltaA	AfbGoldR	DeltaAfbGoldR	OutputGoldR	DeltaAfbAfbG
0	0	3400.192	23.2	0	23.2	3400.192	0
1	22.17125382	3249.41896	1.028746177	1.510229386	21.68977061	221.3392188	20.66102444
2	21.22986823	3111.449488	0.941385594	0.780519064	0.729710322	114.3928741	20.44934916
3	20.36516854	2984.719101	0.864699689	0.52624757	0.254271495	77.12684381	19.83892097
4	19.56815115	2867.908232	0.797017392	0.396936614	0.129310956	58.17503015	19.17121453
5	18.83116883	2759.896104	0.736982316	0.318639634	0.07829698	46.69982472	18.5125292
6	18.14768461	2659.724656	0.683484225	0.266142248	0.052497385	39.00580793	17.88154236
7	17.51207729	2566.570048	0.635607311	0.228496389	0.03764586	33.48843073	17.28358091
8	16.91948658	2479.719953	0.592590714	0.200180787	0.028315601	29.33849618	16.71930579
9	16.36568849	2398.555305	0.553798094	0.178109234	0.022071553	26.10368941	16.18757925
10	15.84699454	2322.535519	0.518693952	0.160421469	0.017687766	23.51137043	15.68657307
11	15.36016949	2251.186441	0.486825044	0.145929431	0.014492037	21.38741748	15.21424006
12	14.90236382	2184.090442	0.457805668	0.1338388	0.012090631	19.61541454	14.76852502
13	14.47105788	2120.878244	0.431305939	0.123598356	0.010240444	18.11457507	14.34745953
14	14.06401552	2061.222114	0.407042365	0.114813593	0.008784763	16.82708017	13.94920193
15	13.67924528	2004.830189	0.384770236	0.107194719	0.007618873	15.71045809	13.57205056
16	13.31496786	1951.44169	0.364277423	0.100524079	0.006670641	14.73280898	13.21444378
17	12.96958855	1900.822898	0.345379309	0.094635018	0.005889061	13.86970825	12.87495353
18	12.64167393	1852.763731	0.327914619	0.089397776	0.005237242	13.10213806	12.55227616
19	12.32993197	1807.07483	0.311741959	0.08470981	0.004687966	12.41506972	12.24522216
20	12.03319502	1763.585062	0.296736952	0.080489014	0.004220796	11.79646985	11.95270601
21	11.75040519	1722.139384	0.282789834	0.07666887	0.003820144	11.23658956	11.67373632
22	11.48060174	1682.596991	0.269803445	0.073194916	0.003473954	10.72744689	11.40740683
23	11.22291022	1644.829721	0.257691525	0.070022133	0.003172783	10.26244388	11.15288808
24	10.97653293	1608.720666	0.246377287	0.067112985	0.002909149	9.836079015	10.90941995
25	10.74074074	1574.162963	0.235792189	0.064435921	0.002677063	9.443728607	10.67630482
26	10.51486584	1541.058738	0.225874896	0.061964236	0.002471685	9.081478391	10.45290161
27	10.29829545	1509.318182	0.21657039	0.059675167	0.002289069	8.745992504	10.23862029
28	10.09046625	1478.858733	0.207829205	0.057549198	0.002125969	8.434410451	10.03291705
29	9.890859482	1449.604366	0.199606768	0.055569496	0.001979702	8.144265347	9.835289985
30	9.698996656	1421.48495	0.191862826	0.053721469	0.001848027	7.873418458	9.645275187
31	9.514435696	1394.435696	0.18456096	0.051992402	0.001729067	7.620006403	9.462443294
32	9.336767547	1368.396652	0.177668149	0.050371167	0.001621235	7.382398213	9.28639638
33	9.165613148	1343.312263	0.171154399	0.048847982	0.001523185	7.1591602	9.116765166
34	9.000620732	1319.130975	0.164992415	0.047414213	0.001433769	6.949026997	8.95320652

35	8.841463415	1295.804878	0.159157318	0.04606221	0.001352002	6.750877565	8.795401204
36	8.687837028	1273.289395	0.153626386	0.044785175	0.001277036	6.563715184	8.643051854
37	8.539458186	1251.542992	0.148378842	0.043577038	0.001208137	6.386650686	8.495881148
38	8.396062536	1230.526925	0.14339565	0.042432371	0.001144667	6.218888328	8.353630165
39	8.257403189	1210.205011	0.138659347	0.041346301	0.001086071	6.059713835	8.216056888
40	8.1232493	1190.543417	0.134153889	0.040314439	0.001031861	5.908484237	8.08293486
41	7.993384785	1171.510474	0.129864515	0.039332827	0.000981612	5.764619185	7.954051958
42	7.867607162	1153.076506	0.125777623	0.038397881	0.000934946	5.627593512	7.829209281
43	7.745726496	1135.213675	0.121880667	0.037506351	0.00089153	5.496930823	7.708220145
44	7.62756444	1117.895844	0.118162056	0.036655281	0.00085107	5.372197963	7.590909159
45	7.512953368	1101.098446	0.114611072	0.035841977	0.000813303	5.253000221	7.47711139
46	7.401735579	1084.798367	0.111217788	0.035063982	0.000777996	5.138977157	7.366671598
47	7.293762575	1068.973843	0.107973004	0.034319043	0.000744939	5.02979896	7.259443532
48	7.188894398	1053.604363	0.104868178	0.033605099	0.000713944	4.925163266	7.155289299
49	7.086999022	1038.670577	0.101895375	0.032920254	0.000684845	4.824792356	7.054078769
50	6.987951807	1024.154217	0.099047215	0.032262764	0.000657489	4.7284307	6.955689043
51	6.891634981	1010.038023	0.096316826	0.031631023	0.000631741	4.63584278	6.860003958
52	6.797937178	996.3056728	0.093697803	0.031023548	0.000607475	4.546811172	6.76691363
53	6.706753006	982.9417206	0.091184171	0.030438966	0.000584582	4.461134839	6.676314041
54	6.617982656	969.9315381	0.08877035	0.029876007	0.000562959	4.378627617	6.588106649
55	6.531531532	957.2612613	0.086451125	0.029333494	0.000542513	4.299116864	6.502198038
56	6.447309916	944.9177412	0.084221616	0.028810332	0.000523162	4.222442255	6.418499584
57	6.36523266	932.8884987	0.082077255	0.028305504	0.000504828	4.148454701	6.336927156
58	6.285218899	921.1616818	0.080013761	0.027818063	0.000487441	4.077015382	6.257400836
59	6.207191781	909.7260274	0.078027118	0.027347127	0.000470937	4.007994881	6.179844654
60	6.131078224	898.5708245	0.076113557	0.02689187	0.000455257	3.941272396	6.104186355
61	6.056808688	887.6858814	0.074269536	0.026451522	0.000440348	3.87673504	6.030357167
62	5.984316962	877.061494	0.072491726	0.026025363	0.000426159	3.814277198	5.958291599
63	5.913539967	866.6884176	0.070776995	0.025612718	0.000412645	3.753799956	5.887927249
64	5.844417574	856.5578396	0.069122394	0.025212954	0.000399764	3.695210574	5.819204619
65	5.77689243	846.6613546	0.067525143	0.024825478	0.000387477	3.638422013	5.752066953
66	5.710909807	836.9909413	0.065982623	0.024449731	0.000375747	3.583352504	5.686460076
67	5.646417445	827.5389408	0.064492362	0.024085188	0.000364543	3.529925154	5.622332257
68	5.583365422	818.2980362	0.063052024	0.023731356	0.000353832	3.47806759	5.559634065
69	5.521706017	809.2612338	0.061659405	0.02338777	0.000343586	3.427711628	5.498318246
70	5.461393597	800.4218456	0.06031242	0.023053991	0.000333779	3.378792978	5.438339606
71	5.402384501	791.7734724	0.059009096	0.022729605	0.000324386	3.331250968	5.379654895
72	5.344636933	783.3099889	0.057747567	0.022414221	0.000315384	3.285028294	5.322222712
73	5.288110868	775.0255288	0.056526065	0.02210747	0.000306752	3.240070787	5.266003398
74	5.232767954	766.9144713	0.055342914	0.021809001	0.000298469	3.196327207	5.210958953
75	5.178571429	758.9714286	0.054196525	0.021518484	0.000290517	3.153749039	5.157052944
76	5.125486037	751.1912337	0.053085391	0.021235605	0.000282879	3.112290325	5.104250432
77	5.073477957	743.5689293	0.052008081	0.020960067	0.000275538	3.071907487	5.052517889
78	5.022514721	736.0997575	0.050963235	0.020691588	0.000268479	3.032559183	5.001823133
79	4.972565158	728.7791495	0.049949563	0.0204299	0.000261688	2.994206161	4.952135258
80	4.923599321	721.6027165	0.048965837	0.020174748	0.000255152	2.956811131	4.903424572

81	4.875588433	714.5662408	0.048010888	0.019925891	0.000248857	2.920338642	4.855662542
82	4.828504829	707.6656677	0.047083605	0.019683099	0.000242793	2.884754971	4.80882173
83	4.7823219	700.8970976	0.046182929	0.019446152	0.000236947	2.85002802	4.762875748
84	4.737014048	694.2567788	0.045307852	0.019214842	0.00023131	2.816127216	4.717799206
85	4.692556634	687.7411003	0.044457413	0.01898897	0.000225872	2.783023424	4.673567664
86	4.648925938	681.3465854	0.043630697	0.018768347	0.000220623	2.750688866	4.630157591
87	4.606099111	675.0698856	0.042826827	0.018552791	0.000215556	2.719097037	4.58754632
88	4.564054139	668.9077746	0.042044971	0.01834213	0.00021066	2.688222636	4.545712009
89	4.522769807	662.8571429	0.041284333	0.0181362	0.00020593	2.658041499	4.504633606
90	4.482225657	656.9149923	0.04054415	0.017934843	0.000201358	2.628530536	4.464290814
91	4.442401961	651.0784314	0.039823696	0.017737907	0.000196935	2.599667669	4.424664054
92	4.403279684	645.3446705	0.039122277	0.01754525	0.000192658	2.571431782	4.385734435
93	4.364840458	639.7110175	0.038439227	0.017356732	0.000188517	2.543802663	4.347483725
94	4.327066547	634.1748732	0.03777391	0.017172223	0.000184509	2.516760964	4.309894325
95	4.289940828	628.7337278	0.037125719	0.016991595	0.000180628	2.490288146	4.272949234
96	4.253446759	623.3851569	0.03649407	0.016814727	0.000176867	2.464366445	4.236632031
97	4.217568354	618.1268179	0.035878405	0.016641504	0.000173223	2.438978829	4.20092685
98	4.182290164	612.9564465	0.035278189	0.016471813	0.000169691	2.41410896	4.165818351
99	4.147597254	607.8718535	0.03469291	0.016305548	0.000166265	2.389741159	4.131291706
100	4.113475177	602.870922	0.034122077	0.016142606	0.000162942	2.365860376	4.097332571
101	4.079909961	597.9516038	0.033565217	0.015982889	0.000159718	2.342452153	4.063927072
102	4.046888083	593.1119174	0.033021878	0.015826301	0.000156588	2.319502602	4.031061782
103	4.014396456	588.3499446	0.032491626	0.015672751	0.00015355	2.296998372	3.998723705
104	3.982422411	583.6638286	0.031974045	0.015522152	0.000150599	2.274926627	3.966900259
105	3.950953678	579.0517711	0.031468733	0.01537442	0.000147732	2.253275017	3.935579258
106	3.919978373	574.5120303	0.030975306	0.015229474	0.000144946	2.232031659	3.904748899
107	3.889484979	570.0429185	0.030493394	0.015087235	0.000142239	2.211185116	3.874397744
108	3.859462337	565.6428001	0.030022642	0.014947628	0.000139607	2.190724371	3.844514709
109	3.82989963	561.3100898	0.029562707	0.014810581	0.000137047	2.170638813	3.815089049
110	3.80078637	557.0432503	0.029113261	0.014676025	0.000134556	2.150918216	3.786110345
111	3.772112383	552.8407908	0.028673987	0.014543891	0.000132134	2.131552722	3.757568492
112	3.743867803	548.7012652	0.02824458	0.014414116	0.000129775	2.112532825	3.729453687
113	3.716043055	544.6232701	0.027824748	0.014286636	0.00012748	2.093849357	3.701756419
114	3.688628848	540.6054439	0.027414207	0.014161391	0.000125245	2.075493469	3.674467457
115	3.661616162	536.6464646	0.027012686	0.014038323	0.000123068	2.05745662	3.647577839
116	3.63499624	532.7450489	0.026619922	0.013917376	0.000120947	2.039730565	3.621078864
117	3.608760577	528.8999502	0.026235662	0.013798494	0.000118881	2.022307339	3.594962083
118	3.582900914	525.109958	0.025859663	0.013681627	0.000116867	2.005179248	3.569219287
119	3.557409225	521.373896	0.02549169	0.013566723	0.000114904	1.988338856	3.543842502
120	3.53227771	517.6906212	0.025131515	0.013453732	0.00011299	1.971778974	3.518823978
121	3.507498791	514.0590227	0.02477892	0.013342608	0.000111124	1.955492652	3.494156182
122	3.483065097	510.4780207	0.024433693	0.013233305	0.000109303	1.939473167	3.469831792
123	3.458969466	506.9465649	0.024095632	0.013125778	0.000107527	1.923714013	3.445843688
124	3.435204928	503.4636342	0.023764538	0.013019984	0.000105794	1.908208897	3.422184943
125	3.411764706	500.0282353	0.023440222	0.012915882	0.000104102	1.892951725	3.398848823
126	3.388642206	496.6394017	0.0231225	0.012813432	0.00010245	1.877936596	3.375828774

127	3.365831012	493.2961931	0.022811194	0.012712594	0.000100838	1.863157796	3.353118418
128	3.343324879	489.9976943	0.022506133	0.012613331	9.93E-05	1.848609789	3.330711548
129	3.321117728	486.7430142	0.022207151	0.012515606	9.77E-05	1.834287212	3.308602122
130	3.299203641	483.5312856	0.021914087	0.012419384	9.62E-05	1.820184863	3.286784257
131	3.277576854	480.3616637	0.021626787	0.01232463	9.48E-05	1.806297704	3.265252224
132	3.256231754	477.2333258	0.0213451	0.01223131	9.33E-05	1.792620844	3.244000444
133	3.235162874	474.1454708	0.02106888	0.012139394	9.19E-05	1.779149545	3.22302348
134	3.214364886	471.0973177	0.020797988	0.012048848	9.05E-05	1.765879205	3.202316038
135	3.193832599	468.0881057	0.020532287	0.011959644	8.92E-05	1.752805361	3.181872956
136	3.173560954	465.1170935	0.020271645	0.01187175	8.79E-05	1.739923682	3.161689204
137	3.15354502	462.1835581	0.020015935	0.011785139	8.66E-05	1.72722996	3.141759881
138	3.133779987	459.2867949	0.019765033	0.011699782	8.54E-05	1.714720113	3.122080205
139	3.114261168	456.4261168	0.019518819	0.011615653	8.41E-05	1.702390173	3.102645515
140	3.094983991	453.6008538	0.019277177	0.011532726	8.29E-05	1.690236288	3.083451266
141	3.075943997	450.8103521	0.019039995	0.011450974	8.18E-05	1.678254714	3.064493023
142	3.057136833	448.0539743	0.018807163	0.011370373	8.06E-05	1.666441811	3.045766461
143	3.038558256	445.3310981	0.018578577	0.011290898	7.95E-05	1.654794043	3.027267358
144	3.020204124	442.6411164	0.018354132	0.011212527	7.84E-05	1.643307972	3.008991597
145	3.002070393	439.9834369	0.018133731	0.011135236	7.73E-05	1.631980253	2.990935157
146	2.984153118	437.357481	0.017917275	0.011059004	7.62E-05	1.620807634	2.973094114
147	2.966448445	434.7626841	0.017704673	0.010983808	7.52E-05	1.609786952	2.955464637
148	2.948952613	432.198495	0.017495832	0.010909628	7.42E-05	1.598915128	2.938042985
149	2.931661949	429.6643753	0.017290664	0.010836444	7.32E-05	1.588189166	2.920825506
150	2.914572864	427.159799	0.017089085	0.010764234	7.22E-05	1.577606151	2.90380863
151	2.897681855	424.6842526	0.01689101	0.010692981	7.13E-05	1.567163244	2.886988874
152	2.880985496	422.2372343	0.016696359	0.010622664	7.03E-05	1.55685768	2.870362831
153	2.864480443	419.8182537	0.016505053	0.010553267	6.94E-05	1.546686768	2.853927176
154	2.848163426	417.4268317	0.016317017	0.01048477	6.85E-05	1.536647886	2.837678656
155	2.83203125	415.0625	0.016132176	0.010417157	6.76E-05	1.526738481	2.821614093
156	2.816080792	412.7248009	0.015950458	0.01035041	6.67E-05	1.516956062	2.805730383
157	2.800309	410.413287	0.015771793	0.010284513	6.59E-05	1.507298204	2.790024487
158	2.784712886	408.1275206	0.015596113	0.01021945	6.51E-05	1.497762545	2.774493437
159	2.769289534	405.8670741	0.015423353	0.010155205	6.42E-05	1.488346778	2.759134329
160	2.754036087	403.631529	0.015253447	0.010091762	6.34E-05	1.479048657	2.743944325
161	2.738949754	401.420476	0.015086333	0.010029107	6.27E-05	1.46986599	2.728920647
162	2.724027804	399.2335149	0.014921951	0.009967226	6.19E-05	1.460796642	2.714060578
163	2.709267564	397.0702541	0.01476024	0.009906103	6.11E-05	1.451838526	2.69936146
164	2.694666419	394.9303104	0.014601145	0.009845726	6.04E-05	1.442989609	2.684820693
165	2.680221811	392.8133087	0.014444607	0.00978608	5.96E-05	1.434247906	2.670435731
166	2.665931237	390.7188821	0.014290574	0.009727153	5.89E-05	1.425611481	2.656204085
167	2.651792246	388.6466715	0.014138992	0.00966893	5.82E-05	1.417078443	2.642123315
168	2.637802438	386.5963253	0.013989808	0.009611401	5.75E-05	1.408646947	2.628191037
169	2.623959464	384.5674991	0.013842973	0.009554552	5.68E-05	1.40031519	2.614404912
170	2.610261026	382.559856	0.013698438	0.009498372	5.62E-05	1.392081415	2.600762654
171	2.596704871	380.5730659	0.013556155	0.009442849	5.55E-05	1.383943901	2.587262022
172	2.583288794	378.6068056	0.013416077	0.009387971	5.49E-05	1.375900972	2.573900823

173	2.570010635	376.6607586	0.013278159	0.009333727	5.42E-05	1.367950986	2.560676908
174	2.556868277	374.7346147	0.013142357	0.009280106	5.36E-05	1.360092344	2.547588171
175	2.543859649	372.8280702	0.013008628	0.009227098	5.30E-05	1.352323479	2.534632551
176	2.530982719	370.9408274	0.01287693	0.009174692	5.24E-05	1.344642861	2.521808027
177	2.518235498	369.0725947	0.012747221	0.009122878	5.18E-05	1.337048996	2.50911262
178	2.505616036	367.2230862	0.012619462	0.009071646	5.12E-05	1.329540422	2.49654439
179	2.493122421	365.392022	0.012493615	0.009020986	5.07E-05	1.32211571	2.484101435
180	2.48075278	363.5791275	0.012369641	0.008970889	5.01E-05	1.314773463	2.471781891
181	2.468505277	361.7841335	0.012247503	0.008921345	4.95E-05	1.307512314	2.459583933
182	2.456378113	360.0067762	0.012127165	0.008872345	4.90E-05	1.300330928	2.447505768
183	2.444369521	358.246797	0.012008592	0.008823881	4.85E-05	1.293227997	2.43554564
184	2.432477772	356.5039423	0.011891749	0.008775943	4.79E-05	1.286202243	2.423701829
185	2.420701169	354.7779633	0.011776604	0.008728524	4.74E-05	1.279252414	2.411972645
186	2.409038046	353.068616	0.011663122	0.008681614	4.69E-05	1.272377287	2.400356433
187	2.397486772	351.3756614	0.011551274	0.008635205	4.64E-05	1.265575663	2.388851567
188	2.386045746	349.6988646	0.011441026	0.00858929	4.59E-05	1.258846369	2.377456456
189	2.374713397	348.0379954	0.01133235	0.008543861	4.54E-05	1.25218826	2.366169536
190	2.363488183	346.392828	0.011225214	0.00849891	4.50E-05	1.245600209	2.354989273
191	2.352368592	344.7631408	0.011119591	0.008454429	4.45E-05	1.239081119	2.343914163
192	2.341353141	343.1487163	0.011015451	0.008410412	4.40E-05	1.232629911	2.332942729
193	2.330440373	341.549341	0.010912768	0.00836685	4.36E-05	1.226245531	2.322073523
194	2.319628859	339.9648056	0.010811513	0.008323737	4.31E-05	1.219926945	2.311305122
195	2.308917197	338.3949045	0.010711662	0.008281067	4.27E-05	1.213673143	2.300636131
196	2.29830401	336.8394357	0.010613187	0.008238831	4.22E-05	1.207483132	2.290065179
197	2.287787946	335.2982013	0.010516064	0.008197025	4.18E-05	1.201355941	2.279590921
198	2.277367677	333.7710068	0.010420269	0.00815564	4.14E-05	1.19529062	2.269212037
199	2.267041901	332.257661	0.010325776	0.008114671	4.10E-05	1.189286235	2.25892723
200	2.256809339	330.7579767	0.010232563	0.008074112	4.06E-05	1.183341874	2.248735226
201	2.246668733	329.2717694	0.010140606	0.008033956	4.02E-05	1.177456639	2.238634776
202	2.236618849	327.7988586	0.010049883	0.007994198	3.98E-05	1.171629655	2.228624651
203	2.226658477	326.3390663	0.009960373	0.007954831	3.94E-05	1.165860059	2.218703645

Sampling station id: S19; Input Viin: 146.56; A: 23.2; Beta: 0.03; DeltaA: 0.01

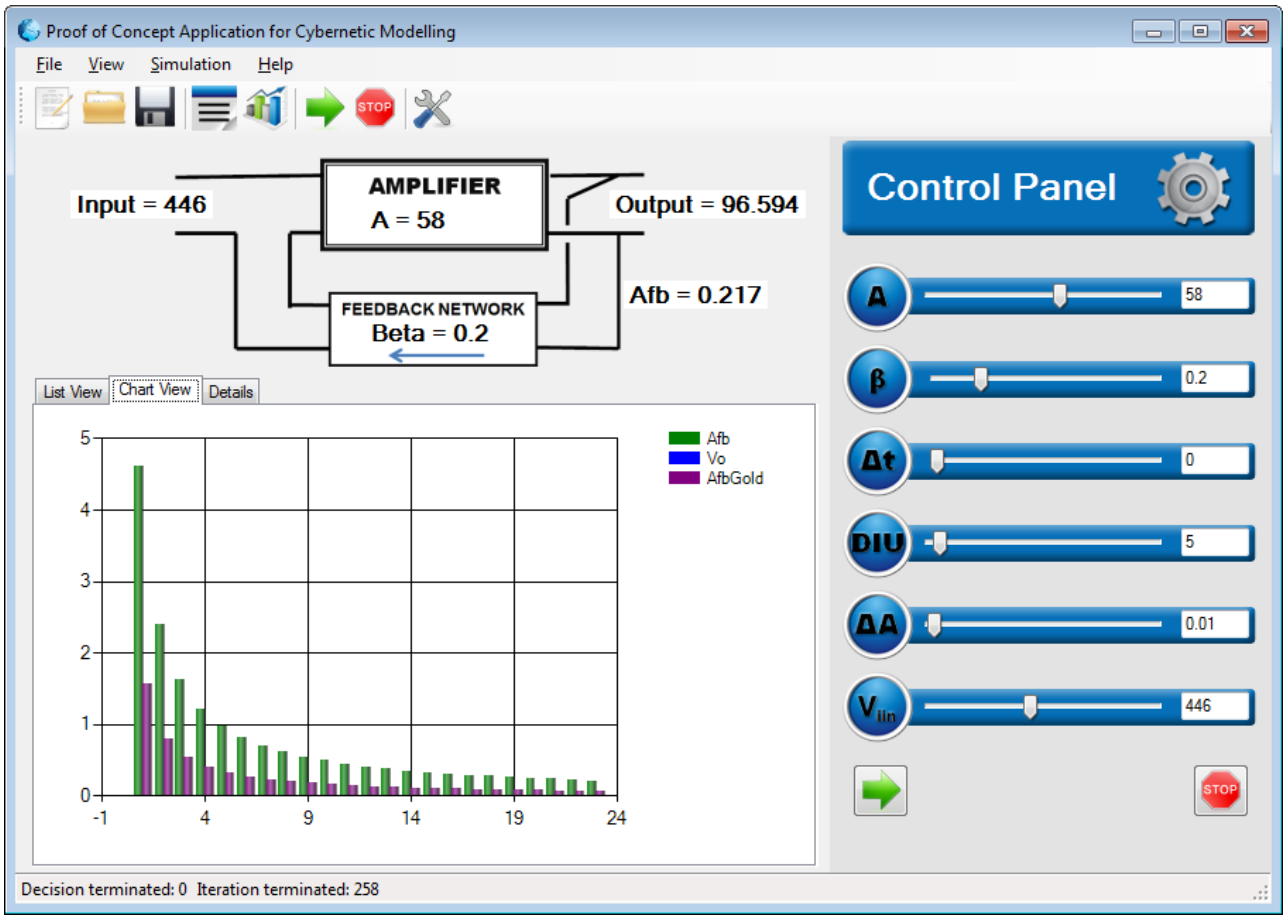
t	Afb	Output(Vo)	DeltaA	AfbGoldR	DeltaAfbGoldR	OutputGoldR	DeltaAfbAfbG
0	0	3400.192	23.2	0	23.2	3400.192	0
1	13.67924528	2004.830189	9.520754717	1.510229386	21.68977061	221.3392188	12.1690159
2	9.698996656	1421.48495	3.980248628	0.780519064	0.729710322	114.3928741	8.918477591
3	7.512953368	1101.098446	2.186043288	0.52624757	0.254271495	77.12684381	6.986705798
4	6.131078224	898.5708245	1.381875144	0.396936614	0.129310956	58.17503015	5.73414161
5	5.178571429	758.9714286	0.952506796	0.318639634	0.07829698	46.69982472	4.859931795
6	4.482225657	656.9149923	0.696345772	0.266142248	0.052497385	39.00580793	4.216083408
7	3.950953678	579.0517711	0.531271978	0.228496389	0.03764586	33.48843073	3.72245729
8	3.53227771	517.6906212	0.418675968	0.200180787	0.028315601	29.33849618	3.332096923
9	3.193832599	468.0881057	0.338445111	0.178109234	0.022071553	26.10368941	3.015723365
10	2.914572864	427.159799	0.279259735	0.160421469	0.017687766	23.51137043	2.754151396
11	2.680221811	392.8133087	0.234351053	0.145929431	0.014492037	21.38741748	2.53429238

12	2.48075278	363.5791275	0.199469031	0.1338388	0.012090631	19.61541454	2.34691398
13	2.308917197	338.3949045	0.171835583	0.123598356	0.010240444	18.11457507	2.185318841
14	2.159344751	316.4735666	0.149572447	0.114813593	0.008784763	16.82708017	2.044531158
15	2.027972028	297.2195804	0.131372723	0.107194719	0.007618873	15.71045809	1.920777309
16	1.911667765	280.1740277	0.116304263	0.100524079	0.006670641	14.73280898	1.811143687
17	1.80798005	264.9775561	0.103687715	0.094635018	0.005889061	13.86970825	1.713345032
18	1.714961561	251.3447664	0.093018489	0.089397776	0.005237242	13.10213806	1.625563785
19	1.631046119	239.0461192	0.083915442	0.08470981	0.004687966	12.41506972	1.546336309
20	1.554959786	227.8949062	0.076086334	0.080489014	0.004220796	11.79646985	1.474470772
21	1.485655738	217.7377049	0.069304048	0.07666887	0.003820144	11.23658956	1.408986868
22	1.422265817	208.4472781	0.063389921	0.073194916	0.003473954	10.72744689	1.349070901
23	1.36406397	199.9172154	0.058201847	0.070022133	0.003172783	10.26244388	1.294041836
24	1.310438319	192.05784	0.053625651	0.067112985	0.002909149	9.836079015	1.243325334
25	1.260869565	184.7930435	0.049568754	0.064435921	0.002677063	9.443728607	1.196433644
26	1.214914118	178.0578132	0.045955447	0.061964236	0.002471685	9.081478391	1.152949882
27	1.172190784	171.7962813	0.042723334	0.059675167	0.002289069	8.745992504	1.112515617
28	1.132370168	165.9601718	0.039820616	0.057549198	0.002125969	8.434410451	1.07482097
29	1.095166163	160.5075529	0.037204005	0.055569496	0.001979702	8.144265347	1.039596667
30	1.060329068	155.4018282	0.034837096	0.053721469	0.001848027	7.873418458	1.006607599
31	1.027639972	150.6109142	0.032689096	0.051992402	0.001729067	7.620006403	0.97564757
32	0.996906153	146.1065658	0.030733818	0.050371167	0.001621235	7.382398213	0.946534986
33	0.967957276	141.8638184	0.028948877	0.048847982	0.001523185	7.1591602	0.919109295
34	0.940642232	137.8605255	0.027315045	0.047414213	0.001433769	6.949026997	0.893228019
35	0.914826498	134.0769716	0.025815733	0.04606221	0.001352002	6.750877565	0.868764288
36	0.890389929	130.4955481	0.024436569	0.044785175	0.001277036	6.563715184	0.845604755
37	0.86722488	127.1004785	0.023165049	0.043577038	0.001208137	6.386650686	0.823647842
38	0.845234625	123.8775867	0.021990255	0.042432371	0.001144667	6.218888328	0.802802254
39	0.824332007	120.8140989	0.020902619	0.041346301	0.001086071	6.059713835	0.782985706
40	0.80443828	117.8984743	0.019893727	0.040314439	0.001031861	5.908484237	0.764123841
41	0.785482124	115.12026	0.018956157	0.039332827	0.000981612	5.764619185	0.746149296
42	0.767398783	112.4699656	0.018083341	0.038397881	0.000934946	5.627593512	0.729000901
43	0.750129333	109.938955	0.01726945	0.037506351	0.00089153	5.496930823	0.712622982
44	0.733620035	107.5193524	0.016509297	0.036655281	0.00085107	5.372197963	0.696964755
45	0.717821782	105.2039604	0.015798253	0.035841977	0.000813303	5.253000221	0.681979805
46	0.702689605	102.9861885	0.015132177	0.035063982	0.000777996	5.138977157	0.667625623
47	0.68818225	100.8599905	0.014507355	0.034319043	0.000744939	5.02979896	0.653863207
48	0.6742618	98.81980935	0.01392045	0.033605099	0.000713944	4.925163266	0.640656701
49	0.660893345	96.86052871	0.013368454	0.032920254	0.000684845	4.824792356	0.627973092
50	0.648044693	94.97743017	0.012848653	0.032262764	0.000657489	4.7284307	0.615781929
51	0.635686103	93.1661552	0.01235859	0.031631023	0.000631741	4.63584278	0.604055079
52	0.623790062	91.42267154	0.01189604	0.031023548	0.000607475	4.546811172	0.592766515
53	0.612331081	89.74324324	0.011458981	0.030438966	0.000584582	4.461134839	0.581892115
54	0.601285507	88.1244039	0.011045574	0.029876007	0.000562959	4.378627617	0.5714095
55	0.590631365	86.56293279	0.010654142	0.029333494	0.000542513	4.299116864	0.561297871
56	0.580348209	85.0558335	0.010283156	0.028810332	0.000523162	4.222442255	0.551537877
57	0.570416994	83.60031471	0.009931214	0.028305504	0.000504828	4.148454701	0.54211149

Sampling station id: S19; Input Viin: 146.56; A: 232; Beta: 0.03; DeltaA: 0.01

t	Afb	Output(Vo)	DeltaA	AfbGoldR	DeltaAfbGoldR	OutputGoldR	DeltaAfbAfbG
0	0	34001.92	232	0	232	34001.92	0
1	29.14572864	4271.59799	202.8542714	1.604214686	230.3957853	235.1137043	27.54151396
2	15.54959786	2278.949062	13.59613079	0.804890137	0.799324548	117.9646985	14.74470772
3	10.60329068	1554.018282	4.946307179	0.537214687	0.26767545	78.73418458	10.06607599
4	8.044382802	1178.984743	2.558907875	0.403144394	0.134070294	59.08484237	7.641238408
5	6.480446927	949.7743017	1.563935874	0.322627641	0.080516753	47.284307	6.157819287
6	5.425631431	795.1805426	1.054815496	0.268918695	0.053708945	39.41272396	5.156712736
7	4.66613033	683.8680611	0.759501101	0.230539914	0.038378781	33.78792978	4.435590416
8	4.093154552	599.8927311	0.572975778	0.201747484	0.02879243	29.56811131	3.891407068
9	3.645505971	534.2853551	0.447648581	0.179348426	0.022399058	26.28530536	3.466157545
10	3.28611898	481.6135977	0.359386991	0.161426063	0.017922364	23.65860376	3.124692917
11	2.991232594	438.395049	0.294886386	0.146760249	0.014665813	21.50918216	2.844472345
12	2.744912447	402.2943682	0.246320147	0.134537321	0.012222929	19.71778974	2.610375126
13	2.536073459	371.6869261	0.208838988	0.124193836	0.010343485	18.20184863	2.411879622
14	2.356765542	345.4075579	0.179307916	0.115327258	0.008866579	16.90236288	2.241438285
15	2.20113852	322.5988615	0.155627023	0.107642341	0.007684917	15.77606151	2.093496179
16	2.064791741	302.6158775	0.136346779	0.100917621	0.00672472	14.79048657	1.96387412
17	1.944351324	284.9641301	0.120440417	0.094983721	0.0059339	13.92081415	1.849367603
18	1.837187203	269.2581565	0.107164121	0.089708888	0.005274833	13.14773463	1.747478315
19	1.741218853	255.1930351	0.09596835	0.084989097	0.004719791	12.45600209	1.656229756
20	1.654778887	242.5243937	0.086439966	0.080741121	0.004247976	11.83341874	1.574037766
21	1.576515357	231.0540908	0.07826353	0.07689758	0.003843541	11.27010938	1.499617777
22	1.505320529	220.6197768	0.071194828	0.073403342	0.003494239	10.75799376	1.431917188
23	1.440278123	211.0871617	0.065042407	0.070212858	0.003190484	10.29039646	1.370065265
24	1.380623661	202.3442038	0.059654462	0.067288171	0.002924687	9.861754299	1.31333549
25	1.325714286	194.2966857	0.054909375	0.064597393	0.002690777	9.467393962	1.261116892
26	1.275005496	186.8648055	0.05070879	0.062113543	0.00248385	9.103360914	1.212891952
27	1.22803303	179.9805209	0.046972466	0.059813635	0.002299908	8.766286325	1.168219395
28	1.184398611	173.5854605	0.043634418	0.057677965	0.00213567	8.453282506	1.126720647
29	1.143758627	167.6292644	0.040639984	0.055689547	0.001988418	8.161859976	1.088069081
30	1.105815062	162.0682555	0.037943566	0.053833659	0.001855887	7.889861107	1.051981403
31	1.070308175	156.8643661	0.035506887	0.05209748	0.00173618	7.635406611	1.018210695
32	1.037010549	151.984266	0.033297626	0.050469788	0.001627692	7.396852061	0.986540761
33	1.005722213	147.3986475	0.031288336	0.048940723	0.001529065	7.172752313	0.95678149
34	0.976266622	143.0816361	0.029455591	0.047501584	0.001439138	6.961832204	0.928765037
35	0.948487326	139.0103025	0.027779296	0.046144666	0.001356918	6.762962275	0.90234266
36	0.92224519	135.164255	0.026242136	0.044863118	0.001281548	6.57513854	0.877382072
37	0.897416061	131.5252978	0.024829129	0.043650829	0.001212289	6.397465528	0.853765231
38	0.873888805	128.0771433	0.023527255	0.042502334	0.001148496	6.229142015	0.831386472
39	0.851563647	124.8051681	0.022325158	0.041412725	0.001089609	6.069448935	0.810150922
40	0.830350752	121.6962062	0.021212895	0.040377587	0.001035138	5.91773912	0.789973165
41	0.810169018	118.7383713	0.020181734	0.039392935	0.000984652	5.773428528	0.770776083
42	0.790945043	115.9209055	0.019223975	0.038455163	0.000937772	5.635988729	0.75248988

43	0.772612229	113.2340482	0.018332814	0.037561002	0.000894161	5.504940444	0.735051227
44	0.755110012	110.6689233	0.017502217	0.036707478	0.000853524	5.379847956	0.718402534
45	0.738383195	108.2174411	0.016726816	0.035891882	0.000815596	5.260314275	0.702491313
46	0.722381368	105.8722132	0.016001828	0.035111742	0.00078014	5.145976924	0.687269625
47	0.707058393	103.6264781	0.015322974	0.034364794	0.000746948	5.036504271	0.672693599
48	0.692371971	101.4740361	0.014686422	0.033648965	0.000715829	4.931592317	0.658723006
49	0.678283242	99.40919191	0.014088729	0.032962349	0.000686616	4.830961875	0.645320893
50	0.664756447	97.42670487	0.013526795	0.032303194	0.000659155	4.734356091	0.632453253
51	0.651758625	95.52174402	0.012997822	0.031669884	0.000633309	4.641538252	0.62008874
52	0.639259341	93.689849	0.012499284	0.03106093	0.000608955	4.552289851	0.608198411
53	0.627230453	91.92689521	0.012028888	0.030474951	0.000585978	4.466408872	0.596755502
54	0.615645897	90.22906273	0.011584556	0.029910673	0.000564278	4.38370826	0.585735224
55	0.604481501	88.59280875	0.011164397	0.029366912	0.000543762	4.304014563	0.575114589
56	0.593714812	87.01484287	0.010766689	0.028842568	0.000524344	4.227166715	0.564872245
57	0.583324952	85.492105	0.01038986	0.028336619	0.000505948	4.153014953	0.554988333
58	0.573292478	84.02174558	0.010032474	0.027848116	0.000488504	4.081419842	0.545444362
59	0.563599261	82.60110776	0.009693217	0.027376169	0.000471946	4.012251398	0.536223092



APPENDIX 3: AS3. Copy of decision to postpone thesis validation

MINISTERUL EDUCAȚIEI, CERCETĂRII, TINERETULUI ȘI SPORTULUI
DIRECȚIA GENERALĂ ÎNVĂȚĂMÂNT SUPERIOR

ORDINUL MINISTRULUI EDUCAȚIEI, CERCETĂRII, TINERETULUI ȘI SPORTULUI

Nr. 5743 din 12.09.2012

privind atribuirea titlului de DOCTOR

LISTA
persoanelor cărora li se atribuie titlul de DOCTOR
în domeniul specificat și, după caz, cu distincția menționată,
urmare a validării tezei de doctorat de către CNATDCU,
întrunit în data de 12.09.2012

Anexa nr. 35
Instituția Universitatea "Lucian Blaga" din Sibiu

Nr. crt.	Numele și Prenumele	Domeniul	Distincția
1	Aioanei V. Genoveva	Drept	-
2	Alexandrescu C. Adrian	Teologie	-
3	Barb A. I. Elena Alina (Pitic)	Calculatoare și tehnologia informației	-
4	Bogdan L. Silvia	Filologie	-
5	Bojescu V. Elena Rodica (Giurgiu)	Filologie	-
6	Botaș S. Luminița (Zamfirescu)	Finanțe	-
7	Brabete M. Alina Camelia (Cătană)	Medicină	-
8	Brîndușă M. Emilia Elena (Bodea)	Filologie	-
9	Ciocan G. Mihaela	Medicină	-
10	Citrea E. C. Ianina Mihaela (Roberts)	Management	-
11	Drugaș G. M. Șerban George Paul	Teologie	-
12	Dumitrescu I. Elena Carmen (Berbescu)	Medicină	-
13	Dumitru D. Gheorghe Simion	Medicină	-
14	Fărcaș I. Raluca Maria (Frîncu)	Istorie	-
15	Ganea R. Vasile Romulus	Teologie	-
16	Ghiță - Nică I. Florentina (Petrică)	Filologie	-
17	Hila R. Maria (Hulber)	Filologie	-
18	Horșia V. Dragoș Ovidiu	Medicină	-
19	Iaru D. Cristina Anca (Danciu)	Inginerie industrială	-
20	Ionac V. Ioana	Finanțe	-
21	Ionescu G. Mădălina Ioana	Drept	MAGNA CUM LAUDE
22	Jilăveanu M. Simona Elena (Radu)	Medicină	-
23	Lostun V. Doru Viorel	Istorie	-
24	Mihăiescu A. Mirela Elena (Mitrea)	Medicină	-
25	Mihuț A. Oana Elena (Picu)	Medicină	-
26	Moga C. Doru Florian Cornel	Medicină	-
27	Muntean I. Elena Cristina (Rezi)	Medicină	-
28	Murariu A. A. Mihaela Andreea	Filologie	-
29	Mutu I. Cătălin Cosmin	Medicină	-
30	Nichita F. Iohana Raluca (Popescu)	Istorie	-
31	Olariu N. Maria Elena (Scridon)	Medicină	-
32	Osnaga I. Bianca Medina	Filologie	-
33	Pașa S. Alexandru	Inginerie și management	-
34	Popa M. Livia Mirela	Medicină	-
35	Răhăian N. Liliana Cristina (Cuibus)	Medicină	-
36	Rusu G. Gabriel George	Medicină	-
37	Sabău D. Andreea Maria (Smarandache)	Medicină	-
38	Săceleanu M. Mircea Vicențiu	Medicină	-
39	Stoia I. Dan Ioan	Medicină	-
40	Stoica A. Augustin	Inginerie mecanică	-

ORDINUL MINISTRULUI EDUCAȚIEI, CERCETĂRII, TINERETULUI ȘI SPORTULUI

Nr. 5743 din 12.09.2012

privind atribuirea titlului de DOCTOR

LISTA
persoanelor cărora li se atribuie titlul de DOCTOR
în domeniul specificat și, după caz, cu distincția menționată,
urmare a validării tezei de doctorat de către CNATDCU,
întrunit în data de 12.09.2012

Anexa nr. 35
Instituția **Universitatea "Lucian Blaga" din Sibiu**

Nr. crt.	Numele si Prenumele	Domeniul	Distincția
41	Stroia I. Marius Daniel	Filologie	-
42	Surdu I. Alexandru	Filologie	
43	Șerban R. Liiana Carmen (Prodan)	Medicină	
44	Tătulea Gh. George Robert	Teologie	MAGNA CUM LAUDE
45	Tecoantă I. Ovidiu Ioan	Medicină	-
46	Tomița Ș. Maria (Neagoi)	Finante	-
47	Tudor N. Nicolae	Inginerie și management	-
48	Tudose Ș. Silviu	Teologie	MAGNA CUM LAUDE
49	Ureche G. Maria Corina (Beca)	Medicină	-
50	Virvorea D. Daniela (Popa)	Management	-
51	Vlad Stoica G. Ana (Tirnovean)	Management	-
52	Zaharia E. Adrian	Drept	-
53	Zerbes G. Mihai Victor	Inginerie industrială	-

DIRECTOR GENERAL,
Prof. univ. dr. Ioan Ștefan GROZA

ANEXĂ

Adresa Nr. 57898 din 17.09.2012

LISTA persoanelor pentru care CNATDCU, reunit în data de 12.09.2012,
a invalidat argumentat teza de doctorat sau, după caz,
a propus amânarea luării deciziei privind validarea tezei de doctorat

Instituția **Universitatea "Lucian Blaga" din Sibiu**

Nr. crt.	Numele si Prenumele	Domeniul	Rezoluția	Argumentarea Rezoluției
1	Bantaș P. Mihaela Zoe (Hărătău)	Medicină	Se amână luarea deciziei privind validarea tezei de doctorat	Lucrarea va fi analizată în cadrul următoarei întruniri.
2	Băcilă I. Ionuț Ciprian	Medicină	Se amână luarea deciziei privind validarea tezei de doctorat	Lucrarea va fi analizată în cadrul următoarei întruniri.
3	Boicean G. Adrian Gheorghe	Medicină	Se invalidează teza de doctorat	Fără publicații suficiente.
4	Bolca A. Ciprian Nicolae	Medicină	Se invalidează teza de doctorat	Publicații insuficiente
5	Drăgoiu C. L. Alexandra Paula (Dumitrescu)	Management	Nediscutat	
6	Fabian A. Ralf Detlef	Calculatoare și tehnologia informației	Se amână luarea deciziei privind validarea tezei de doctorat	Teza este insuficient susținută din punct de vedere al validării contribuțiilor. Recomandăm extinderea tezei cu un capitol substanțial de validare, bazat pe metodologii științifice și inginerești general acceptate/ utilizate, prin care contribuțiile tezei să fie clar reliefate. Comisia se va pronunța asupra acordării titlului după completarea tezei cu observațiile de mai sus.
7	Filote G. Veronica Isabela (Crăciun)	Inginerie industrială	Se amână luarea deciziei privind validarea tezei de doctorat	Se solicită avizul comisiei Ingineria resurselor vegetale și animale.
8	Hojda G. Axenia (Rădulescu)	Inginerie industrială	Se amână luarea deciziei privind validarea tezei de doctorat	Se solicită avizul comisiei Ingineria resurselor vegetale și animale.
9	Mândrean L. Liliana Nicoleta	Inginerie industrială	Se amână luarea deciziei privind validarea tezei de doctorat	Se solicită avizul comisiei Ingineria resurselor vegetale și animale.
10	Mihalca G. George Liviu	Inginerie industrială	Se amână luarea deciziei privind validarea tezei de doctorat	Se solicită avizul comisiei Ingineria resurselor vegetale și animale.

APPENDIX 4: AS4. “Proof-of-Concept” User-Validation Assessment

“Proof-of-Concept” User-Validation Assessment

The evaluation of the self-adjustment and support capacity of ecosystems has become a hotspot in the research world in the fields of environment protection, sustainable management of natural resources and biodiversity. Although progress has been attained regarding the know how/knowledge on the structure of ecological systems, the analysis of ecological processes and the modelling still require a considerable scientifical effort.

Creating a modelling system which allows the simulation and prediction (with an acceptable degree of error) of the dynamics of ecological systems, represents an essential step in the elaboration of management plans for a sustainable use of ecosystem resources and services, at the same time being an important tool for sizing the socio-economical development.

The research of Mr. Ralf D. Fabian takes part in this context, summing up with the elaboration of the "Tool for Exploring Homeostasis in Benthic Communities”/ Proof of Concept Application Toolkit and is based on the *interface* construed *as mirror for functionality*, enabling the *user-validation* of A, β, Δt, *and* DIU.

The application uses two kinds of *scenarios: descriptive*, to interpret *facts*, and, much more significative, *normative*, to *prepare actions*.

The assessment below:

1. Refers to the "Tool for Exploring Homeostasis in Benthic Communities” as defined in Chapter 6 and submitted to “Proof-of-Concept” user validation according to Chapter 7 of “Validating Process-Oriented Modelling Mechanisms” (Addendum to the PhD Thesis “Bounded Rationality in Agent Orientation – “Just-In-Time” Visual Pattern Recognition”).

2. Follows the request of Ralf D. Fabian, author of the Addendum, as designer of the toolkit for ecologic research, focused on setting up “What-if” scenarios aimed at *preserving of* (oversimplified) *ecological* (sub) *systems*.

3. Was performed by member(s) of the Department of Environmental Sciences of the Faculty of Sciences of the “Lucian Blaga” University of Sibiu, involved in the project in the field of biodiversity – stability – self-regulation – productivity assessment in the lotic ecosystems cases and in the field of sustainable management of the ecosystems resources and services.

4. Was based on the following background:

At first (October 2012, "*Plan A*") it was intended to design only one application ("*Preservation of Ecological Systems Modelled Using GST Applied to Benthic Communities in Transylvanian Lotic Systems*") for both validating and continuing the thesis. Likewise, it was necessary to speed up and reduce the scope of the "Proof-of-Concept" (PoC) application for to (new title): "*Homeostasis in Benthic Communities Modelled Via GST*".

It has been necessary to set up a "Plan B" – but from the joint post-thesis work, only and partially could be used in ADT. Plan B suggests a new approach to CybMd ("*Non-algorithmic cybernetic modelling of living systems*").

5. Is confined to the claims regarding the toolkit interface as expressed in Figure 7.1, at July 19^{-th} 2013, as well as to the refinements, the designer accepted to carry out in November 2013 based on suggestions proposed until October 31^{-st} after extended testing the toolkit in laboratory work in line with the short and middle range intentions in Chapter. 8.

To conclude:

- The toolkit architectonics as expressed in Chapter 6 is reflected in the interface.
- The toolkit functionality accessible through the interface is sufficient to set up "What-if" scenarios aiming at investigating ecologic stability in (oversimplified) benthic communities in Transylvanian lotic systems.
- I therefore consider that the results obtained by Mr Ralf D. Fabian are in accordance with the previously set objectives, the toolkit is functional, easy to use, having an appealing interface, being adequate for cybernetic modelling of ecological systems.
- Transdisciplinary approach represents an essential prerequisite for attaining progress in the know-how of applied sciences, which leads/determines us to promote the development of cybernetic modelling applications on ecological systems using What-if scenarios.

Assoc. prof. PhD Angela Bănăduc

APPENDIX 5: AC5. Integral code of the toolkit

```
#pragma once
#include <windows.h>
#include <math.h>
#include "ProjectInfoDlg.h"
#include "AboutDlg.h"

#pragma comment (lib, "user32.lib")

using namespace System;
using namespace System::Windows::Forms;


// global variables
double Viin; // value for amplification input
double A; // amplification value
double Afb; // amplification value with feedback
double beta; // feedback ratio
int t; // snapshot id
int delta_t; // minimum time span for observable changes
int diu; // delay
double delta_A; // desired amplification gain

//-------------------------------------------------------------
HANDLE hEventObject;  // handle of the Event waited for
HANDLE hCloseEvent; // handle of termination
HANDLE hContinueEvent; // handle of iterations
DWORD dwTimeout=5000; // Wait duration (ms), see domino()
DWORD dwCloseWaitMillies=2000; // Wait duration (ms) for before closing handle
BOOL homeostaticState=FALSE; //

HANDLE hIterationThread; // thread for Iterations
HANDLE hDecisionThread; // thred for decisions
HANDLE hTesterThread; // thread for testing

DWORD WINAPI IterationThreadProc(LPVOID lpParam);
DWORD WINAPI DecisionThreadProc(LPVOID lpParam);
DWORD WINAPI TesterThreadProc(LPVOID lpParam);
BOOL checkHomeostaticState(double dVal);
INT domino();

// macrochronic version
INT domino(){

	INT RetVal;
	// return value propagating the possible exception in DOMONO, means:
	// RetVal ==  0 - the event waited for occured before timeout duration elapsed time;
no exception
	// RetVal == -1 - homeostatic state not attained but still good chances (system
stabilizabil in due time),
	// RetVal == -2 - homeostatic state not attained microchronicaly (in current
interration) but still good chances (system stabilizabil in due time)
	// RetVal == -3 - homeostatic state not attained microchronicaly (forlast itteration)
but still good chances (system stabilizabil in due time)
	// RetVal == -4 - the event did not occur in due time; exception raised
(simulated/propagated via return value)

	INT dwWaitResult;
	// dwWaitResult == WAIT_TIMEOUT means: since timeout elapsed (simulated) exception is
rased
	// dwWaitResult == WAIT_OBJECT_0 means: event happend before timeout duration elapsed
(here less then dwTimeout/1000 s)
```

```
	ResetEvent(hEventObject);
	// hEventObject denotes the manner WAIT terminates

	dwWaitResult = WaitForSingleObject(hEventObject, dwTimeout);
	//dwTimeout - Wait duration (ms)

	if(dwWaitResult == WAIT_OBJECT_0)
	{
		// bivalent IF (favorable wait termination)
		if(homeostaticState){

			RetVal = 0;
			// SUCCESS, system stable
		}
		else{
			// inform caller
			// homeostatic state not attained but still good chances
			RetVal = -1;
		}
	}
	else{
		// event not occured in due time
		RetVal = -4;
	}

	return RetVal;
	// return t1-t6 == t0, hence retunr -1, -2, -3 have same meaning
}
DWORD WINAPI TesterThreadProc(LPVOID lpParam){

	Sleep(200);
	SetEvent(hEventObject);
	//MessageBox::Show(L"Init set event");
	return 1;
}

BOOL checkHomeostaticState(double dVal){
	if(dVal<=delta_A)
		return true;
	else
		return false;
}
DWORD WINAPI DecisionThreadProc(LPVOID lpParam){
	DWORD dResult;
	BOOL bStopped = FALSE;
	dwTimeout*=1000; //ms to sec

	do{
		dResult = domino();
		switch(dResult){
		case 0:
			bStopped = TRUE;
			//MessageBox::Show("0");
			break;
		case -1:
			bStopped = FALSE;
			//MessageBox::Show("-1");
			SetEvent(hContinueEvent);
			break;
		case -4:
			bStopped = TRUE;
```

```
                        MessageBox::Show("Big problem! (Simulated exception)");

                }
        }

        while(!bStopped);

        ProjectGUI::MainForm::autoRef->BeginInvoke(ProjectGUI::MainForm::autoRef-
>updateDecisionMsgDelegate,"Decision terminated: "+dResult);
        //MessageBox::Show("Decision terminated"+dResult);
        return dResult;
}
DWORD WINAPI IterationThreadProc(LPVOID lpParam){

        int nResult;
        int nIterationResult;
        int i=0;
        // i means the ith iteration (repetitions)
        int period = delta_t*1000; //ms to sec

        double Vo = 0.0;
        double VoGold = 0.0;

        double betaGold = 13.0/21.0; //silver
        double deltaBeta = 0.0;

        t=0;
        Afb = 0.0;

        double Apr = 0.0;
        double AprGold = 0.0;

        double Amp = 0.0;
        double AmpGold = 0.0;
        double AfbGold = 0.0;
        double deltaAfb = 0.0;
        double deltaAfbGold = 0.0;
        double deltaAfbAfbG = 0.0;
        int count = 0;

        do{
                nResult = WaitForSingleObject (hContinueEvent, 1000);
                if (nResult == WAIT_TIMEOUT)
                        break; // no need to continue

                Apr = Amp;
                AprGold = AmpGold;

                if( i == 0 ){
                        Amp = A; AmpGold = A;
                        Afb = 0; AfbGold = 0;
                        Vo = Viin * Amp;
                        VoGold = Viin * AmpGold;
                }
                else{
                        Amp = Amp/(1+beta*Amp);
                        AmpGold = AmpGold/(1+betaGold*AmpGold);
                        Afb = Amp;
                        AfbGold = AmpGold;
                        Vo = Viin * Amp;
                        VoGold = Viin * AmpGold;
                }
```

```
		// Barkhausen
		// macrochronic
		// discrete time

		t=i;
		deltaAfb = abs(Apr-Amp);
		deltaAfbGold = abs(AprGold-AmpGold);

		deltaAfbAfbG = abs(Afb-AfbGold);
		deltaBeta = abs(beta-betaGold);

		homeostaticState = checkHomeostaticState(deltaAfb);

		ProjectGUI::MainForm::autoRef->BeginInvoke(ProjectGUI::MainForm::autoRef-
>addListaDataDelegate ,
			t, Afb, Vo, deltaAfb, betaGold, deltaBeta, AfbGold, deltaAfbGold,
VoGold, deltaAfbAfbG);

		Sleep(period);

		i++;
		ResetEvent(hContinueEvent);
		SetEvent(hEventObject);

	}
	while(nResult == WAIT_OBJECT_0);
	//while(deltaAfb>delta_A);

	ProjectGUI::MainForm::autoRef->BeginInvoke(ProjectGUI::MainForm::autoRef-
>updateIterationMsgDelegate,"Iteration terminated: " + nResult);
	//MessageBox::Show("Iteration terminated");
	return nResult;
}
namespace ProjectGUI {

	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Data;
	using namespace System::Drawing;
	using namespace System::Data;
	using namespace Microsoft::Office::Interop;


	/// <summary>
	/// Summary for MainForm
	/// </summary>
	public ref class MainForm : public System::Windows::Forms::Form
	{
	public:
		MainForm(void)
		{
			InitializeComponent();
			//other constructor code here
			dlgPrInfo = gcnew ProjectInfoDlg();
			prInfo = gcnew ProjectInformation();

			dlgAbout = gcnew AboutDlg();
			autoRef = this;
			addListaDataDelegate = gcnew AddListData(this,
&MainForm::AddListDataMethod);
			updateDecisionMsgDelegate = gcnew UpdateDecisionMsg(this,
&MainForm::UpdateDecisionMsgMethod);
```

```
                    updateIterationMsgDelegate = gcnew UpdateIterationMsg(this,
&MainForm::UpdateIterationMsgMethod);

            }

            delegate void AddListData(int nT, double dAfb, double dVo, double dDeltaA,
double dBetaGold, double dDeltaBeta,double dAfbGold, double dDeltaAfbGold, double dVoGold,
double dDeltaAfbAfbG);
            AddListData^ addListaDataDelegate;
            delegate void UpdateDecisionMsg(String^ msg);
            UpdateDecisionMsg^ updateDecisionMsgDelegate;
            delegate void UpdateIterationMsg(String^ msg);
            UpdateIterationMsg^ updateIterationMsgDelegate;

        private: System::Windows::Forms::ColumnHeader^  columnHeader7;
        public: static MainForm^ autoRef;
        private: System::Windows::Forms::ColumnHeader^  columnHeader8;
        private: System::Windows::Forms::ToolStripSeparator^  toolStripSeparator3;
        private: System::Windows::Forms::ToolStripButton^  tsbListView;
        private: System::Windows::Forms::ToolStripButton^  tsbChartView;
        private: System::Windows::Forms::ToolStripSeparator^  toolStripSeparator4;
        private: System::Windows::Forms::ToolStripButton^  tsbStart;
        private: System::Windows::Forms::ToolStripButton^  tsbStop;
        private: System::Windows::Forms::ToolStripSeparator^  toolStripSeparator5;
        private: System::Windows::Forms::ToolStripButton^  tsbSettings;

        private: System::Windows::Forms::Panel^  panel3;
        private: System::Windows::Forms::Panel^  panel4;
        private: System::Windows::Forms::Panel^  panel5;
        private: System::Windows::Forms::Panel^  panel8;
        private: System::Windows::Forms::Panel^  panel7;
        private: System::Windows::Forms::Panel^  panel6;
        private: System::Windows::Forms::ToolStripMenuItem^  startToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  stopToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  terminateToolStripMenuItem;
        private: System::Windows::Forms::ColumnHeader^  columnHeader10;
        private: System::Windows::Forms::ToolStripStatusLabel^  tsslbDecisionMsg;
        private: System::Windows::Forms::ToolStripStatusLabel^  tsslbIterationMsg;

        private: System::Windows::Forms::ColumnHeader^  columnHeader9;


        protected:
            /// <summary>
            /// Clean up any resources being used.
            /// </summary>
            ~MainForm()
            {
                if (components)
                {
                    delete components;
                }
            }
        private: System::Windows::Forms::StatusStrip^  statusStrip1;
        protected:
        private: System::Windows::Forms::MenuStrip^  menuStrip1;
        private: System::Windows::Forms::ToolStripMenuItem^  fileToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  openToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  saveAsToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  exitToolStripMenuItem;
        private: System::Windows::Forms::ToolStripMenuItem^  simulationToolStripMenuItem;
```

```
private: System::Windows::Forms::ToolStrip^  toolStrip1;
private: System::Windows::Forms::ToolStripButton^  tsbOpen;

private: System::Windows::Forms::ToolStripSeparator^  toolStripSeparator1;
private: System::Windows::Forms::SplitContainer^  splitContainer1;

private: System::Windows::Forms::TabControl^  tabControl;
private: System::Windows::Forms::TabPage^  tabList;
private: System::Windows::Forms::TabPage^  tabChart;
private: System::Windows::Forms::OpenFileDialog^  openFileDialog;

private: System::ComponentModel::IContainer^  components;
private: System::Windows::Forms::ToolStripMenuItem^  newToolStripMenuItem;

private:
	/// <summary>
	/// Required designer variable.
	/// </summary>
	ProjectInfoDlg ^dlgPrInfo;
	AboutDlg^ dlgAbout;
private: System::Windows::Forms::ToolStripMenuItem^  viewToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^  helpToolStripMenuItem;
private: System::Windows::Forms::ToolStripButton^  tsbNew;
private: System::Windows::Forms::ToolStripButton^  tsbSaveAs;


private: System::Windows::Forms::ToolStripMenuItem^  dataChartToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^  dataListToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^  detailsToolStripMenuItem;
private: System::Windows::Forms::ToolStripSeparator^  toolStripSeparator2;
private: System::Windows::Forms::ToolStripMenuItem^  settingsToolStripMenuItem;
private: System::Windows::Forms::ToolStripMenuItem^  aboutToolStripMenuItem;
private: System::Windows::Forms::TabPage^  tabDetails;
private: System::Windows::Forms::DataVisualization::Charting::Chart^  chartView;

private: System::Windows::Forms::GroupBox^  groupBox1;
private: System::Windows::Forms::TextBox^  tbDeltaTDescription;
private: System::Windows::Forms::Label^  label9;
private: System::Windows::Forms::TextBox^  tbBetaDescription;
private: System::Windows::Forms::Label^  label8;
private: System::Windows::Forms::TextBox^  tbAmplificationDescription;
private: System::Windows::Forms::Label^  label7;
private: System::Windows::Forms::Label^  label6;
private: System::Windows::Forms::TextBox^  tbOutputDescription;
private: System::Windows::Forms::Label^  label5;
private: System::Windows::Forms::TextBox^  tbInputDescription;
private: System::Windows::Forms::TextBox^  tbPrDescription;
private: System::Windows::Forms::Label^  label2;
private: System::Windows::Forms::Label^  label4;
private: System::Windows::Forms::Label^  label3;
private: System::Windows::Forms::Label^  label1;
private: System::Windows::Forms::TextBox^  tbPrName;
private: System::Windows::Forms::ComboBox^  cbPrType;
private: System::Windows::Forms::DateTimePicker^  dtPrDate;
private: System::Windows::Forms::ListView^  listView;

private: System::Windows::Forms::ColumnHeader^  columnHeader1;
private: System::Windows::Forms::ColumnHeader^  columnHeader2;
private: System::Windows::Forms::ColumnHeader^  columnHeader3;
private: System::Windows::Forms::SaveFileDialog^  saveFileDialog;
private: System::Windows::Forms::Panel^  panel1;
private: System::Windows::Forms::Panel^  panel2;
```

```
        private: System::Windows::Forms::Label^  label10;
        private: System::Windows::Forms::Label^  lbOutput;
        private: System::Windows::Forms::Label^  lbAfb;
        private: System::Windows::Forms::Label^  lbInput;
        private: System::Windows::Forms::Label^  lbBetaFeedback;
        private: System::Windows::Forms::Label^  lbAmplifier;
        private: System::Windows::Forms::ColumnHeader^  columnHeader4;
        private: System::Windows::Forms::ColumnHeader^  columnHeader5;
        private: System::Windows::Forms::ColumnHeader^  columnHeader6;

        private: System::Windows::Forms::Button^  btStop;
        private: System::Windows::Forms::Button^  btStart;

        private: Dotnetrix::Controls::TrackBar^  trbAmplification;
        private: Dotnetrix::Controls::TrackBar^  trbBeta;
        private: Dotnetrix::Controls::TrackBar^  trbDeltaT;
        private: Dotnetrix::Controls::TrackBar^  trbVii;
        private: Dotnetrix::Controls::TrackBar^  trbDIU;
        private: Dotnetrix::Controls::TrackBar^  trbDeltaA;

        private: System::Windows::Forms::TextBox^  tbDIUVal;
        private: System::Windows::Forms::TextBox^  tbViiVal;
        private: System::Windows::Forms::TextBox^  tbDeltaAVal;
        private: System::Windows::Forms::TextBox^  tbDeltaTVal;
        private: System::Windows::Forms::TextBox^  tbBetaVal;
        private: System::Windows::Forms::TextBox^  tbAVal;
        private: ProjectInformation ^prInfo;


#pragma region Windows Form Designer generated code
                        /// <summary>
                        /// Required method for Designer support - do not modify
                        /// the contents of this method with the code editor.
                        /// </summary>
                        void InitializeComponent(void)
                        {
                                System::ComponentModel::ComponentResourceManager^  resources =
(gcnew System::ComponentModel::ComponentResourceManager(MainForm::typeid));
                                System::Windows::Forms::DataVisualization::Charting::ChartArea^
chartArea1 = (gcnew System::Windows::Forms::DataVisualization::Charting::ChartArea());
                                System::Windows::Forms::DataVisualization::Charting::Legend^
legend1 = (gcnew System::Windows::Forms::DataVisualization::Charting::Legend());
                                System::Windows::Forms::DataVisualization::Charting::Series^
series1 = (gcnew System::Windows::Forms::DataVisualization::Charting::Series());
                                System::Windows::Forms::DataVisualization::Charting::Series^
series2 = (gcnew System::Windows::Forms::DataVisualization::Charting::Series());
                                System::Windows::Forms::DataVisualization::Charting::Series^
series3 = (gcnew System::Windows::Forms::DataVisualization::Charting::Series());
                                this->statusStrip1 = (gcnew
System::Windows::Forms::StatusStrip());
                                this->tsslbDecisionMsg = (gcnew
System::Windows::Forms::ToolStripStatusLabel());
                                this->tsslbIterationMsg = (gcnew
System::Windows::Forms::ToolStripStatusLabel());
                                this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());
                                this->fileToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                                this->newToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                                this->openToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
```

```
                            this->saveAsToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->toolStripSeparator1 = (gcnew
System::Windows::Forms::ToolStripSeparator());
                            this->exitToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->viewToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->dataListToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->dataChartToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->detailsToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->toolStripSeparator2 = (gcnew
System::Windows::Forms::ToolStripSeparator());
                            this->settingsToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->simulationToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->startToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->stopToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->terminateToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->helpToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->aboutToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
                            this->toolStrip1 = (gcnew System::Windows::Forms::ToolStrip());
                            this->tsbNew = (gcnew
System::Windows::Forms::ToolStripButton());
                            this->tsbOpen = (gcnew
System::Windows::Forms::ToolStripButton());
                            this->tsbSaveAs = (gcnew
System::Windows::Forms::ToolStripButton());
                            this->toolStripSeparator3 = (gcnew
System::Windows::Forms::ToolStripSeparator());
                            this->tsbListView = (gcnew
System::Windows::Forms::ToolStripButton());
                            this->tsbChartView = (gcnew
System::Windows::Forms::ToolStripButton());
                            this->toolStripSeparator4 = (gcnew
System::Windows::Forms::ToolStripSeparator());
                            this->tsbStart = (gcnew
System::Windows::Forms::ToolStripButton());
                            this->tsbStop = (gcnew
System::Windows::Forms::ToolStripButton());
                            this->toolStripSeparator5 = (gcnew
System::Windows::Forms::ToolStripSeparator());
                            this->tsbSettings = (gcnew
System::Windows::Forms::ToolStripButton());
                            this->splitContainer1 = (gcnew
System::Windows::Forms::SplitContainer());
                            this->panel1 = (gcnew System::Windows::Forms::Panel());
                            this->lbOutput = (gcnew System::Windows::Forms::Label());
                            this->lbAfb = (gcnew System::Windows::Forms::Label());
                            this->lbInput = (gcnew System::Windows::Forms::Label());
                            this->lbBetaFeedback = (gcnew System::Windows::Forms::Label());
                            this->lbAmplifier = (gcnew System::Windows::Forms::Label());
```

```
                            this->tabControl = (gcnew
System::Windows::Forms::TabControl());
                            this->tabList = (gcnew System::Windows::Forms::TabPage());
                            this->listView = (gcnew System::Windows::Forms::ListView());
                            this->columnHeader1 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->columnHeader2 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->columnHeader3 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->columnHeader4 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->columnHeader5 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->columnHeader6 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->columnHeader7 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->columnHeader8 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->columnHeader9 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->columnHeader10 = (gcnew
System::Windows::Forms::ColumnHeader());
                            this->tabChart = (gcnew System::Windows::Forms::TabPage());
                            this->chartView = (gcnew
System::Windows::Forms::DataVisualization::Charting::Chart());
                            this->tabDetails = (gcnew System::Windows::Forms::TabPage());
                            this->label4 = (gcnew System::Windows::Forms::Label());
                            this->label3 = (gcnew System::Windows::Forms::Label());
                            this->label1 = (gcnew System::Windows::Forms::Label());
                            this->tbPrName = (gcnew System::Windows::Forms::TextBox());
                            this->cbPrType = (gcnew System::Windows::Forms::ComboBox());
                            this->dtPrDate = (gcnew
System::Windows::Forms::DateTimePicker());
                            this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
                            this->tbDeltaTDescription = (gcnew
System::Windows::Forms::TextBox());
                            this->label9 = (gcnew System::Windows::Forms::Label());
                            this->tbBetaDescription = (gcnew
System::Windows::Forms::TextBox());
                            this->label8 = (gcnew System::Windows::Forms::Label());
                            this->tbAmplificationDescription = (gcnew
System::Windows::Forms::TextBox());
                            this->label7 = (gcnew System::Windows::Forms::Label());
                            this->label6 = (gcnew System::Windows::Forms::Label());
                            this->tbOutputDescription = (gcnew
System::Windows::Forms::TextBox());
                            this->label5 = (gcnew System::Windows::Forms::Label());
                            this->tbInputDescription = (gcnew
System::Windows::Forms::TextBox());
                            this->tbPrDescription = (gcnew
System::Windows::Forms::TextBox());
                            this->label2 = (gcnew System::Windows::Forms::Label());
                            this->panel8 = (gcnew System::Windows::Forms::Panel());
                            this->trbVii = (gcnew Dotnetrix::Controls::TrackBar());
                            this->tbViiVal = (gcnew System::Windows::Forms::TextBox());
                            this->panel7 = (gcnew System::Windows::Forms::Panel());
                            this->trbDIU = (gcnew Dotnetrix::Controls::TrackBar());
                            this->tbDIUVal = (gcnew System::Windows::Forms::TextBox());
                            this->panel6 = (gcnew System::Windows::Forms::Panel());
                            this->trbDeltaA = (gcnew Dotnetrix::Controls::TrackBar());
```

```
                            this->tbDeltaAVal = (gcnew System::Windows::Forms::TextBox());
                            this->panel5 = (gcnew System::Windows::Forms::Panel());
                            this->trbDeltaT = (gcnew Dotnetrix::Controls::TrackBar());
                            this->tbDeltaTVal = (gcnew System::Windows::Forms::TextBox());
                            this->panel4 = (gcnew System::Windows::Forms::Panel());
                            this->trbBeta = (gcnew Dotnetrix::Controls::TrackBar());
                            this->tbBetaVal = (gcnew System::Windows::Forms::TextBox());
                            this->panel3 = (gcnew System::Windows::Forms::Panel());
                            this->trbAmplification = (gcnew
Dotnetrix::Controls::TrackBar());
                            this->tbAVal = (gcnew System::Windows::Forms::TextBox());
                            this->btStop = (gcnew System::Windows::Forms::Button());
                            this->btStart = (gcnew System::Windows::Forms::Button());
                            this->panel2 = (gcnew System::Windows::Forms::Panel());
                            this->label10 = (gcnew System::Windows::Forms::Label());
                            this->openFileDialog = (gcnew
System::Windows::Forms::OpenFileDialog());
                            this->saveFileDialog = (gcnew
System::Windows::Forms::SaveFileDialog());
                            this->statusStrip1->SuspendLayout();
                            this->menuStrip1->SuspendLayout();
                            this->toolStrip1->SuspendLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->splitContainer1))->BeginInit();
                            this->splitContainer1->Panel1->SuspendLayout();
                            this->splitContainer1->Panel2->SuspendLayout();
                            this->splitContainer1->SuspendLayout();
                            this->panel1->SuspendLayout();
                            this->tabControl->SuspendLayout();
                            this->tabList->SuspendLayout();
                            this->tabChart->SuspendLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->chartView))->BeginInit();
                            this->tabDetails->SuspendLayout();
                            this->groupBox1->SuspendLayout();
                            this->panel8->SuspendLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbVii))->BeginInit();
                            this->panel7->SuspendLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbDIU))->BeginInit();
                            this->panel6->SuspendLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbDeltaA))->BeginInit();
                            this->panel5->SuspendLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbDeltaT))->BeginInit();
                            this->panel4->SuspendLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbBeta))->BeginInit();
                            this->panel3->SuspendLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbAmplification))->BeginInit();
                            this->panel2->SuspendLayout();
                            this->SuspendLayout();
                            //
                            // statusStrip1
                            //
                            this->statusStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^  >(2) {this->tsslbDecisionMsg,
                                    this->tsslbIterationMsg});
                            this->statusStrip1->Location = System::Drawing::Point(0, 584);
```

```
                            this->statusStrip1->Name = L"statusStrip1";
                            this->statusStrip1->Size = System::Drawing::Size(890, 22);
                            this->statusStrip1->TabIndex = 0;
                            this->statusStrip1->Text = L"statusStrip1";
                            //
                            // tsslbDecisionMsg
                            //
                            this->tsslbDecisionMsg->Name = L"tsslbDecisionMsg";
                            this->tsslbDecisionMsg->Size = System::Drawing::Size(0, 17);
                            //
                            // tsslbIterationMsg
                            //
                            this->tsslbIterationMsg->Name = L"tsslbIterationMsg";
                            this->tsslbIterationMsg->Size = System::Drawing::Size(0, 17);
                            //
                            // menuStrip1
                            //
                            this->menuStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^  >(4) {this->fileToolStripMenuItem,
                                    this->viewToolStripMenuItem, this-
>simulationToolStripMenuItem, this->helpToolStripMenuItem});
                            this->menuStrip1->Location = System::Drawing::Point(0, 0);
                            this->menuStrip1->Name = L"menuStrip1";
                            this->menuStrip1->Size = System::Drawing::Size(890, 24);
                            this->menuStrip1->TabIndex = 1;
                            this->menuStrip1->Text = L"menuStrip1";
                            //
                            // fileToolStripMenuItem
                            //
                            this->fileToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array< System::Windows::Forms::ToolStripItem^  >(5) {this->newToolStripMenuItem,
                                    this->openToolStripMenuItem, this-
>saveAsToolStripMenuItem, this->toolStripSeparator1, this->exitToolStripMenuItem});
                            this->fileToolStripMenuItem->Name = L"fileToolStripMenuItem";
                            this->fileToolStripMenuItem->Size = System::Drawing::Size(37,
20);
                            this->fileToolStripMenuItem->Text = L"&File";
                            //
                            // newToolStripMenuItem
                            //
                            this->newToolStripMenuItem->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"newToolStripMenuItem.Image")));
                            this->newToolStripMenuItem->Name = L"newToolStripMenuItem";
                            this->newToolStripMenuItem->ShortcutKeys =
static_cast<System::Windows::Forms::Keys>((System::Windows::Forms::Keys::Control |
System::Windows::Forms::Keys::N));
                            this->newToolStripMenuItem->Size = System::Drawing::Size(161,
22);
                            this->newToolStripMenuItem->Text = L"&New...";
                            this->newToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::newToolStripMenuItem_Click);
                            //
                            // openToolStripMenuItem
                            //
                            this->openToolStripMenuItem->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"openToolStripMenuItem.Image")));
                            this->openToolStripMenuItem->Name = L"openToolStripMenuItem";
                            this->openToolStripMenuItem->ShortcutKeys =
static_cast<System::Windows::Forms::Keys>((System::Windows::Forms::Keys::Control |
System::Windows::Forms::Keys::O));
```

```
                            this->openToolStripMenuItem->Size = System::Drawing::Size(161,
22);
                            this->openToolStripMenuItem->Text = L"&Open...";
                            this->openToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::openToolStripMenuItem_Click);
                            //
                            // saveAsToolStripMenuItem
                            //
                            this->saveAsToolStripMenuItem->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"saveAsToolStripMenuItem.Image")));
                            this->saveAsToolStripMenuItem->Name =
L"saveAsToolStripMenuItem";
                            this->saveAsToolStripMenuItem->ShortcutKeys =
static_cast<System::Windows::Forms::Keys>((System::Windows::Forms::Keys::Control |
System::Windows::Forms::Keys::S));
                            this->saveAsToolStripMenuItem->Size =
System::Drawing::Size(161, 22);
                            this->saveAsToolStripMenuItem->Text = L"&Save as...";
                            this->saveAsToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::saveAsToolStripMenuItem_Click);
                            //
                            // toolStripSeparator1
                            //
                            this->toolStripSeparator1->Name = L"toolStripSeparator1";
                            this->toolStripSeparator1->Size = System::Drawing::Size(158,
6);
                            //
                            // exitToolStripMenuItem
                            //
                            this->exitToolStripMenuItem->Name = L"exitToolStripMenuItem";
                            this->exitToolStripMenuItem->ShortcutKeys =
static_cast<System::Windows::Forms::Keys>((System::Windows::Forms::Keys::Alt |
System::Windows::Forms::Keys::X));
                            this->exitToolStripMenuItem->Size = System::Drawing::Size(161,
22);
                            this->exitToolStripMenuItem->Text = L"E&xit";
                            this->exitToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::exitToolStripMenuItem_Click);
                            //
                            // viewToolStripMenuItem
                            //
                            this->viewToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array< System::Windows::Forms::ToolStripItem^  >(5) {this->dataListToolStripMenuItem,
                                    this->dataChartToolStripMenuItem, this-
>detailsToolStripMenuItem, this->toolStripSeparator2, this->settingsToolStripMenuItem});
                            this->viewToolStripMenuItem->Name = L"viewToolStripMenuItem";
                            this->viewToolStripMenuItem->Size = System::Drawing::Size(44,
20);
                            this->viewToolStripMenuItem->Text = L"&View";
                            //
                            // dataListToolStripMenuItem
                            //
                            this->dataListToolStripMenuItem->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"dataListToolStripMenuItem.Image")));
                            this->dataListToolStripMenuItem->Name =
L"dataListToolStripMenuItem";
                            this->dataListToolStripMenuItem->Size =
System::Drawing::Size(130, 22);
                            this->dataListToolStripMenuItem->Text = L"Data List";
```

```
                            this->dataListToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::dataListToolStripMenuItem_Click);
                            //
                            // dataChartToolStripMenuItem
                            //
                            this->dataChartToolStripMenuItem->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"dataChartToolStripMenuItem.Image")));
                            this->dataChartToolStripMenuItem->Name =
L"dataChartToolStripMenuItem";
                            this->dataChartToolStripMenuItem->Size =
System::Drawing::Size(130, 22);
                            this->dataChartToolStripMenuItem->Text = L"Data Chart";
                            this->dataChartToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::dataChartToolStripMenuItem_Click);
                            //
                            // detailsToolStripMenuItem
                            //
                            this->detailsToolStripMenuItem->Name =
L"detailsToolStripMenuItem";
                            this->detailsToolStripMenuItem->Size =
System::Drawing::Size(130, 22);
                            this->detailsToolStripMenuItem->Text = L"Details";
                            this->detailsToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::detailsToolStripMenuItem_Click);
                            //
                            // toolStripSeparator2
                            //
                            this->toolStripSeparator2->Name = L"toolStripSeparator2";
                            this->toolStripSeparator2->Size = System::Drawing::Size(127,
6);
                            //
                            // settingsToolStripMenuItem
                            //
                            this->settingsToolStripMenuItem->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"settingsToolStripMenuItem.Image")));
                            this->settingsToolStripMenuItem->Name =
L"settingsToolStripMenuItem";
                            this->settingsToolStripMenuItem->Size =
System::Drawing::Size(130, 22);
                            this->settingsToolStripMenuItem->Text = L"Settings";
                            //
                            // simulationToolStripMenuItem
                            //
                            this->simulationToolStripMenuItem->DropDownItems-
>AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^  >(3) {this-
>startToolStripMenuItem,
                                    this->stopToolStripMenuItem, this-
>terminateToolStripMenuItem});
                            this->simulationToolStripMenuItem->Name =
L"simulationToolStripMenuItem";
                            this->simulationToolStripMenuItem->Size =
System::Drawing::Size(76, 20);
                            this->simulationToolStripMenuItem->Text = L"&Simulation";
                            //
                            // startToolStripMenuItem
                            //
                            this->startToolStripMenuItem->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"startToolStripMenuItem.Image")));
                            this->startToolStripMenuItem->Name = L"startToolStripMenuItem";
```

```
                            this->startToolStripMenuItem->Size = System::Drawing::Size(128,
22);
                            this->startToolStripMenuItem->Text = L"Start";
                            this->startToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::startToolStripMenuItem_Click);
                            //
                            // stopToolStripMenuItem
                            //
                            this->stopToolStripMenuItem->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"stopToolStripMenuItem.Image")));
                            this->stopToolStripMenuItem->Name = L"stopToolStripMenuItem";
                            this->stopToolStripMenuItem->Size = System::Drawing::Size(128,
22);
                            this->stopToolStripMenuItem->Text = L"Stop";
                            this->stopToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::stopToolStripMenuItem_Click);
                            //
                            // terminateToolStripMenuItem
                            //
                            this->terminateToolStripMenuItem->Name =
L"terminateToolStripMenuItem";
                            this->terminateToolStripMenuItem->Size =
System::Drawing::Size(128, 22);
                            this->terminateToolStripMenuItem->Text = L"Terminate";
                            //
                            // helpToolStripMenuItem
                            //
                            this->helpToolStripMenuItem->DropDownItems->AddRange(gcnew
cli::array< System::Windows::Forms::ToolStripItem^  >(1) {this->aboutToolStripMenuItem});
                            this->helpToolStripMenuItem->Name = L"helpToolStripMenuItem";
                            this->helpToolStripMenuItem->Size = System::Drawing::Size(44,
20);
                            this->helpToolStripMenuItem->Text = L"&Help";
                            //
                            // aboutToolStripMenuItem
                            //
                            this->aboutToolStripMenuItem->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"aboutToolStripMenuItem.Image")));
                            this->aboutToolStripMenuItem->Name = L"aboutToolStripMenuItem";
                            this->aboutToolStripMenuItem->Size = System::Drawing::Size(107,
22);
                            this->aboutToolStripMenuItem->Text = L"&About";
                            this->aboutToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MainForm::aboutToolStripMenuItem_Click);
                            //
                            // toolStrip1
                            //
                            this->toolStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^  >(11) {this->tsbNew, this->tsbOpen,
                                    this->tsbSaveAs, this->toolStripSeparator3, this-
>tsbListView, this->tsbChartView, this->toolStripSeparator4, this->tsbStart,
                                    this->tsbStop, this->toolStripSeparator5, this-
>tsbSettings});
                            this->toolStrip1->Location = System::Drawing::Point(0, 24);
                            this->toolStrip1->Name = L"toolStrip1";
                            this->toolStrip1->Size = System::Drawing::Size(890, 39);
                            this->toolStrip1->TabIndex = 2;
                            this->toolStrip1->Text = L"toolStrip1";
                            //
                            // tsbNew
```

```
                            //
                            this->tsbNew->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;
                            this->tsbNew->Image = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"tsbNew.Image")));
                            this->tsbNew->ImageScaling =
System::Windows::Forms::ToolStripItemImageScaling::None;
                            this->tsbNew->ImageTransparentColor =
System::Drawing::Color::Magenta;
                            this->tsbNew->Name = L"tsbNew";
                            this->tsbNew->Size = System::Drawing::Size(36, 36);
                            this->tsbNew->Text = L"New";
                            this->tsbNew->Click += gcnew System::EventHandler(this,
&MainForm::tsbNew_Click);
                            //
                            // tsbOpen
                            //
                            this->tsbOpen->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;
                            this->tsbOpen->Image = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"tsbOpen.Image")));
                            this->tsbOpen->ImageScaling =
System::Windows::Forms::ToolStripItemImageScaling::None;
                            this->tsbOpen->ImageTransparentColor =
System::Drawing::Color::Magenta;
                            this->tsbOpen->Name = L"tsbOpen";
                            this->tsbOpen->Size = System::Drawing::Size(36, 36);
                            this->tsbOpen->Text = L"Open";
                            this->tsbOpen->Click += gcnew System::EventHandler(this,
&MainForm::tsbOpen_Click);
                            //
                            // tsbSaveAs
                            //
                            this->tsbSaveAs->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;
                            this->tsbSaveAs->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources->GetObject(L"tsbSaveAs.Image")));
                            this->tsbSaveAs->ImageScaling =
System::Windows::Forms::ToolStripItemImageScaling::None;
                            this->tsbSaveAs->ImageTransparentColor =
System::Drawing::Color::Magenta;
                            this->tsbSaveAs->Name = L"tsbSaveAs";
                            this->tsbSaveAs->Size = System::Drawing::Size(36, 36);
                            this->tsbSaveAs->Text = L"Save as";
                            //
                            // toolStripSeparator3
                            //
                            this->toolStripSeparator3->Name = L"toolStripSeparator3";
                            this->toolStripSeparator3->Size = System::Drawing::Size(6, 39);
                            //
                            // tsbListView
                            //
                            this->tsbListView->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;
                            this->tsbListView->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources->GetObject(L"tsbListView.Image")));
                            this->tsbListView->ImageScaling =
System::Windows::Forms::ToolStripItemImageScaling::None;
                            this->tsbListView->ImageTransparentColor =
System::Drawing::Color::Magenta;
                            this->tsbListView->Name = L"tsbListView";
                            this->tsbListView->Size = System::Drawing::Size(36, 36);
```

```
                            this->tsbListView->Text = L"ListView";
                            //
                            // tsbChartView
                            //
                            this->tsbChartView->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;
                            this->tsbChartView->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources->GetObject(L"tsbChartView.Image")));
                            this->tsbChartView->ImageScaling =
System::Windows::Forms::ToolStripItemImageScaling::None;
                            this->tsbChartView->ImageTransparentColor =
System::Drawing::Color::Magenta;
                            this->tsbChartView->Name = L"tsbChartView";
                            this->tsbChartView->Size = System::Drawing::Size(36, 36);
                            this->tsbChartView->Text = L"ChartView";
                            //
                            // toolStripSeparator4
                            //
                            this->toolStripSeparator4->Name = L"toolStripSeparator4";
                            this->toolStripSeparator4->Size = System::Drawing::Size(6, 39);
                            //
                            // tsbStart
                            //
                            this->tsbStart->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;
                            this->tsbStart->Image = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"tsbStart.Image")));
                            this->tsbStart->ImageScaling =
System::Windows::Forms::ToolStripItemImageScaling::None;
                            this->tsbStart->ImageTransparentColor =
System::Drawing::Color::Magenta;
                            this->tsbStart->Name = L"tsbStart";
                            this->tsbStart->Size = System::Drawing::Size(36, 36);
                            this->tsbStart->Text = L"Start";
                            this->tsbStart->Click += gcnew System::EventHandler(this,
&MainForm::tsbStart_Click);
                            //
                            // tsbStop
                            //
                            this->tsbStop->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;
                            this->tsbStop->Image = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"tsbStop.Image")));
                            this->tsbStop->ImageScaling =
System::Windows::Forms::ToolStripItemImageScaling::None;
                            this->tsbStop->ImageTransparentColor =
System::Drawing::Color::Magenta;
                            this->tsbStop->Name = L"tsbStop";
                            this->tsbStop->Size = System::Drawing::Size(36, 36);
                            this->tsbStop->Text = L"Stop";
                            this->tsbStop->Click += gcnew System::EventHandler(this,
&MainForm::tsbStop_Click);
                            //
                            // toolStripSeparator5
                            //
                            this->toolStripSeparator5->Name = L"toolStripSeparator5";
                            this->toolStripSeparator5->Size = System::Drawing::Size(6, 39);
                            //
                            // tsbSettings
                            //
                            this->tsbSettings->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;
```

```
                            this->tsbSettings->Image =
(cli::safe_cast<System::Drawing::Image^  >(resources->GetObject(L"tsbSettings.Image")));
                            this->tsbSettings->ImageScaling =
System::Windows::Forms::ToolStripItemImageScaling::None;
                            this->tsbSettings->ImageTransparentColor =
System::Drawing::Color::Magenta;
                            this->tsbSettings->Name = L"tsbSettings";
                            this->tsbSettings->Size = System::Drawing::Size(36, 36);
                            this->tsbSettings->Text = L"Settings";
                            //
                            // splitContainer1
                            //
                            this->splitContainer1->Dock =
System::Windows::Forms::DockStyle::Fill;
                            this->splitContainer1->Location = System::Drawing::Point(0,
63);
                            this->splitContainer1->Name = L"splitContainer1";
                            //
                            // splitContainer1.Panel1
                            //
                            this->splitContainer1->Panel1->BackColor =
System::Drawing::SystemColors::Control;
                            this->splitContainer1->Panel1->Controls->Add(this->panel1);
                            this->splitContainer1->Panel1->Controls->Add(this->tabControl);
                            //
                            // splitContainer1.Panel2
                            //
                            this->splitContainer1->Panel2->BackColor =
System::Drawing::SystemColors::ControlLight;
                            this->splitContainer1->Panel2->Controls->Add(this->panel8);
                            this->splitContainer1->Panel2->Controls->Add(this->panel7);
                            this->splitContainer1->Panel2->Controls->Add(this->panel6);
                            this->splitContainer1->Panel2->Controls->Add(this->panel5);
                            this->splitContainer1->Panel2->Controls->Add(this->panel4);
                            this->splitContainer1->Panel2->Controls->Add(this->panel3);
                            this->splitContainer1->Panel2->Controls->Add(this->btStop);
                            this->splitContainer1->Panel2->Controls->Add(this->btStart);
                            this->splitContainer1->Panel2->Controls->Add(this->panel2);
                            this->splitContainer1->Size = System::Drawing::Size(890, 521);
                            this->splitContainer1->SplitterDistance = 573;
                            this->splitContainer1->TabIndex = 3;
                            //
                            // panel1
                            //
                            this->panel1->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"panel1.BackgroundImage")));
                            this->panel1->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Zoom;
                            this->panel1->Controls->Add(this->lbOutput);
                            this->panel1->Controls->Add(this->lbAfb);
                            this->panel1->Controls->Add(this->lbInput);
                            this->panel1->Controls->Add(this->lbBetaFeedback);
                            this->panel1->Controls->Add(this->lbAmplifier);
                            this->panel1->Location = System::Drawing::Point(12, 12);
                            this->panel1->Name = L"panel1";
                            this->panel1->Size = System::Drawing::Size(553, 156);
                            this->panel1->TabIndex = 2;
                            //
                            // lbOutput
                            //
                            this->lbOutput->AutoSize = true;
```

```
                            this->lbOutput->BackColor = System::Drawing::Color::White;
                            this->lbOutput->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                                    static_cast<System::Byte>(0)));
                            this->lbOutput->Location = System::Drawing::Point(409, 26);
                            this->lbOutput->Name = L"lbOutput";
                            this->lbOutput->Size = System::Drawing::Size(84, 20);
                            this->lbOutput->TabIndex = 6;
                            this->lbOutput->Text = L"Output = ";
                            //
                            // lbAfb
                            //
                            this->lbAfb->AutoSize = true;
                            this->lbAfb->BackColor = System::Drawing::Color::White;
                            this->lbAfb->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                                    static_cast<System::Byte>(0)));
                            this->lbAfb->Location = System::Drawing::Point(419, 90);
                            this->lbAfb->Name = L"lbAfb";
                            this->lbAfb->Size = System::Drawing::Size(57, 20);
                            this->lbAfb->TabIndex = 5;
                            this->lbAfb->Text = L"Afb = ";
                            //
                            // lbInput
                            //
                            this->lbInput->AutoSize = true;
                            this->lbInput->BackColor = System::Drawing::Color::White;
                            this->lbInput->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                                    static_cast<System::Byte>(0)));
                            this->lbInput->Location = System::Drawing::Point(27, 26);
                            this->lbInput->Name = L"lbInput";
                            this->lbInput->Size = System::Drawing::Size(71, 20);
                            this->lbInput->TabIndex = 4;
                            this->lbInput->Text = L"Input = ";
                            //
                            // lbBetaFeedback
                            //
                            this->lbBetaFeedback->AutoSize = true;
                            this->lbBetaFeedback->BackColor =
System::Drawing::Color::White;
                            this->lbBetaFeedback->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 12, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
                                    static_cast<System::Byte>(0)));
                            this->lbBetaFeedback->Location = System::Drawing::Point(236,
117);
                            this->lbBetaFeedback->Name = L"lbBetaFeedback";
                            this->lbBetaFeedback->Size = System::Drawing::Size(67, 20);
                            this->lbBetaFeedback->TabIndex = 3;
                            this->lbBetaFeedback->Text = L"Beta = ";
                            //
                            // lbAmplifier
                            //
                            this->lbAmplifier->AutoSize = true;
                            this->lbAmplifier->BackColor = System::Drawing::Color::White;
                            this->lbAmplifier->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 12, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
                                    static_cast<System::Byte>(0)));
                            this->lbAmplifier->Location = System::Drawing::Point(236, 37);
                            this->lbAmplifier->Name = L"lbAmplifier";
```

```
                this->lbAmplifier->Size = System::Drawing::Size(41, 20);
                this->lbAmplifier->TabIndex = 2;
                this->lbAmplifier->Text = L"A = ";
                //
                // tabControl
                //
                this->tabControl->Controls->Add(this->tabList);
                this->tabControl->Controls->Add(this->tabChart);
                this->tabControl->Controls->Add(this->tabDetails);
                this->tabControl->Location = System::Drawing::Point(12, 169);
                this->tabControl->Name = L"tabControl";
                this->tabControl->SelectedIndex = 0;
                this->tabControl->Size = System::Drawing::Size(557, 348);
                this->tabControl->TabIndex = 0;
                //
                // tabList
                //
                this->tabList->Controls->Add(this->listView);
                this->tabList->Location = System::Drawing::Point(4, 22);
                this->tabList->Name = L"tabList";
                this->tabList->Padding = System::Windows::Forms::Padding(3);
                this->tabList->Size = System::Drawing::Size(549, 322);
                this->tabList->TabIndex = 0;
                this->tabList->Text = L"List View";
                this->tabList->UseVisualStyleBackColor = true;
                //
                // listView
                //
                this->listView->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
                this->listView->Columns->AddRange(gcnew cli::array<
System::Windows::Forms::ColumnHeader^  >(10) {this->columnHeader1, this->columnHeader2,
                    this->columnHeader3, this->columnHeader4, this-
>columnHeader5, this->columnHeader6, this->columnHeader7, this->columnHeader8,
                    this->columnHeader9, this->columnHeader10});
                this->listView->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 10, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
                    static_cast<System::Byte>(0)));
                this->listView->GridLines = true;
                this->listView->HeaderStyle =
System::Windows::Forms::ColumnHeaderStyle::Nonclickable;
                this->listView->Location = System::Drawing::Point(6, 6);
                this->listView->Name = L"listView";
                this->listView->Size = System::Drawing::Size(537, 310);
                this->listView->TabIndex = 0;
                this->listView->UseCompatibleStateImageBehavior = false;
                this->listView->View = System::Windows::Forms::View::Details;
                //
                // columnHeader1
                //
                this->columnHeader1->Text = L"t";
                this->columnHeader1->Width = 33;
                //
                // columnHeader2
                //
                this->columnHeader2->Text = L"Afb";
                this->columnHeader2->Width = 73;
                //
                // columnHeader3
                //
                this->columnHeader3->Text = L"Output";
                this->columnHeader3->Width = 85;
```

```
                                //
                                // columnHeader4
                                //
                                this->columnHeader4->Text = L"DeltaA";
                                this->columnHeader4->Width = 81;
                                //
                                // columnHeader5
                                //
                                this->columnHeader5->Text = L"BetaGoldR";
                                this->columnHeader5->Width = 81;
                                //
                                // columnHeader6
                                //
                                this->columnHeader6->Text = L"DeltaBeta";
                                this->columnHeader6->Width = 78;
                                //
                                // columnHeader7
                                //
                                this->columnHeader7->Text = L"AfbGoldR";
                                this->columnHeader7->Width = 73;
                                //
                                // columnHeader8
                                //
                                this->columnHeader8->Text = L"DeltaAfbGoldR";
                                //
                                // columnHeader9
                                //
                                this->columnHeader9->Text = L"OutputGoldR";
                                //
                                // columnHeader10
                                //
                                this->columnHeader10->Text = L"DeltaAfbAfbG";
                                //
                                // tabChart
                                //
                                this->tabChart->Controls->Add(this->chartView);
                                this->tabChart->Location = System::Drawing::Point(4, 22);
                                this->tabChart->Name = L"tabChart";
                                this->tabChart->Padding = System::Windows::Forms::Padding(3);
                                this->tabChart->Size = System::Drawing::Size(549, 322);
                                this->tabChart->TabIndex = 1;
                                this->tabChart->Text = L"Chart View";
                                this->tabChart->UseVisualStyleBackColor = true;
                                //
                                // chartView
                                //
                                this->chartView->BorderlineDashStyle =
System::Windows::Forms::DataVisualization::Charting::ChartDashStyle::Dot;
                                chartArea1->Name = L"ChartArea1";
                                this->chartView->ChartAreas->Add(chartArea1);
                                legend1->Name = L"Legend1";
                                this->chartView->Legends->Add(legend1);
                                this->chartView->Location = System::Drawing::Point(0, 3);
                                this->chartView->Name = L"chartView";
                                this->chartView->Palette =
System::Windows::Forms::DataVisualization::Charting::ChartColorPalette::Bright;
                                series1->ChartArea = L"ChartArea1";
                                series1->CustomProperties = L"DrawingStyle=Cylinder";
                                series1->Legend = L"Legend1";
                                series1->Name = L"Afb";
                                series2->ChartArea = L"ChartArea1";
                                series2->CustomProperties = L"DrawingStyle=Cylinder";
```

```
series2->Legend = L"Legend1";
series2->Name = L"Vo";
series3->ChartArea = L"ChartArea1";
series3->CustomProperties = L"DrawingStyle=Cylinder";
series3->Legend = L"Legend1";
series3->Name = L"AfbGold";
this->chartView->Series->Add(series1);
this->chartView->Series->Add(series2);
this->chartView->Series->Add(series3);
this->chartView->Size = System::Drawing::Size(546, 313);
this->chartView->TabIndex = 0;
this->chartView->Text = L"chart1";
//
// tabDetails
//
this->tabDetails->Controls->Add(this->label4);
this->tabDetails->Controls->Add(this->label3);
this->tabDetails->Controls->Add(this->label1);
this->tabDetails->Controls->Add(this->tbPrName);
this->tabDetails->Controls->Add(this->cbPrType);
this->tabDetails->Controls->Add(this->dtPrDate);
this->tabDetails->Controls->Add(this->groupBox1);
this->tabDetails->Location = System::Drawing::Point(4, 22);
this->tabDetails->Name = L"tabDetails";
this->tabDetails->Padding = System::Windows::Forms::Padding(3);
this->tabDetails->Size = System::Drawing::Size(549, 322);
this->tabDetails->TabIndex = 2;
this->tabDetails->Text = L"Details";
this->tabDetails->UseVisualStyleBackColor = true;
//
// label4
//
this->label4->AutoSize = true;
this->label4->Location = System::Drawing::Point(13, 287);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(66, 13);
this->label4->TabIndex = 20;
this->label4->Text = L"Project type:";
//
// label3
//
this->label3->AutoSize = true;
this->label3->Location = System::Drawing::Point(229, 287);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(68, 13);
this->label3->TabIndex = 19;
this->label3->Text = L"Current date:";
//
// label1
//
this->label1->AutoSize = true;
this->label1->Location = System::Drawing::Point(13, 255);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(72, 13);
this->label1->TabIndex = 18;
this->label1->Text = L"Project name:";
//
// tbPrName
//
this->tbPrName->Location = System::Drawing::Point(104, 252);
this->tbPrName->Name = L"tbPrName";
this->tbPrName->Size = System::Drawing::Size(381, 20);
```

```
                            this->tbPrName->TabIndex = 17;
                            //
                            // cbPrType
                            //
                            this->cbPrType->FormattingEnabled = true;
                            this->cbPrType->Items->AddRange(gcnew cli::array<
System::Object^  >(2) {L"Enumerative", L"Reconstructive"});
                            this->cbPrType->Location = System::Drawing::Point(104, 284);
                            this->cbPrType->Name = L"cbPrType";
                            this->cbPrType->Size = System::Drawing::Size(115, 21);
                            this->cbPrType->TabIndex = 16;
                            //
                            // dtPrDate
                            //
                            this->dtPrDate->Location = System::Drawing::Point(303, 284);
                            this->dtPrDate->Name = L"dtPrDate";
                            this->dtPrDate->Size = System::Drawing::Size(182, 20);
                            this->dtPrDate->TabIndex = 15;
                            //
                            // groupBox1
                            //
                            this->groupBox1->Controls->Add(this->tbDeltaTDescription);
                            this->groupBox1->Controls->Add(this->label9);
                            this->groupBox1->Controls->Add(this->tbBetaDescription);
                            this->groupBox1->Controls->Add(this->label8);
                            this->groupBox1->Controls->Add(this-
>tbAmplificationDescription);
                            this->groupBox1->Controls->Add(this->label7);
                            this->groupBox1->Controls->Add(this->label6);
                            this->groupBox1->Controls->Add(this->tbOutputDescription);
                            this->groupBox1->Controls->Add(this->label5);
                            this->groupBox1->Controls->Add(this->tbInputDescription);
                            this->groupBox1->Controls->Add(this->tbPrDescription);
                            this->groupBox1->Controls->Add(this->label2);
                            this->groupBox1->Location = System::Drawing::Point(6, 6);
                            this->groupBox1->Name = L"groupBox1";
                            this->groupBox1->Size = System::Drawing::Size(488, 228);
                            this->groupBox1->TabIndex = 14;
                            this->groupBox1->TabStop = false;
                            this->groupBox1->Text = L"Descriptions";
                            //
                            // tbDeltaTDescription
                            //
                            this->tbDeltaTDescription->Location =
System::Drawing::Point(98, 195);
                            this->tbDeltaTDescription->Name = L"tbDeltaTDescription";
                            this->tbDeltaTDescription->Size = System::Drawing::Size(381,
20);
                            this->tbDeltaTDescription->TabIndex = 18;
                            //
                            // label9
                            //
                            this->label9->AutoSize = true;
                            this->label9->Location = System::Drawing::Point(7, 193);
                            this->label9->Name = L"label9";
                            this->label9->Size = System::Drawing::Size(41, 13);
                            this->label9->TabIndex = 17;
                            this->label9->Text = L"Delta t:";
                            //
                            // tbBetaDescription
                            //
```

```
                            this->tbBetaDescription->Location = System::Drawing::Point(98,
169);
                            this->tbBetaDescription->Name = L"tbBetaDescription";
                            this->tbBetaDescription->Size = System::Drawing::Size(381, 20);
                            this->tbBetaDescription->TabIndex = 16;
                            //
                            // label8
                            //
                            this->label8->AutoSize = true;
                            this->label8->Location = System::Drawing::Point(7, 169);
                            this->label8->Name = L"label8";
                            this->label8->Size = System::Drawing::Size(32, 13);
                            this->label8->TabIndex = 15;
                            this->label8->Text = L"Beta:";
                            //
                            // tbAmplificationDescription
                            //
                            this->tbAmplificationDescription->Location =
System::Drawing::Point(98, 143);
                            this->tbAmplificationDescription->Name =
L"tbAmplificationDescription";
                            this->tbAmplificationDescription->Size =
System::Drawing::Size(381, 20);
                            this->tbAmplificationDescription->TabIndex = 14;
                            //
                            // label7
                            //
                            this->label7->AutoSize = true;
                            this->label7->Location = System::Drawing::Point(7, 143);
                            this->label7->Name = L"label7";
                            this->label7->Size = System::Drawing::Size(69, 13);
                            this->label7->TabIndex = 13;
                            this->label7->Text = L"Amplification:";
                            //
                            // label6
                            //
                            this->label6->AutoSize = true;
                            this->label6->Location = System::Drawing::Point(7, 118);
                            this->label6->Name = L"label6";
                            this->label6->Size = System::Drawing::Size(42, 13);
                            this->label6->TabIndex = 12;
                            this->label6->Text = L"Output:";
                            //
                            // tbOutputDescription
                            //
                            this->tbOutputDescription->Location =
System::Drawing::Point(98, 115);
                            this->tbOutputDescription->Name = L"tbOutputDescription";
                            this->tbOutputDescription->Size = System::Drawing::Size(381,
20);
                            this->tbOutputDescription->TabIndex = 11;
                            //
                            // label5
                            //
                            this->label5->AutoSize = true;
                            this->label5->Location = System::Drawing::Point(7, 93);
                            this->label5->Name = L"label5";
                            this->label5->Size = System::Drawing::Size(34, 13);
                            this->label5->TabIndex = 10;
                            this->label5->Text = L"Input:";
                            //
                            // tbInputDescription
```

```
                            //
                            this->tbInputDescription->Location = System::Drawing::Point(98,
90);
                            this->tbInputDescription->Name = L"tbInputDescription";
                            this->tbInputDescription->Size = System::Drawing::Size(381,
20);
                            this->tbInputDescription->TabIndex = 9;
                            //
                            // tbPrDescription
                            //
                            this->tbPrDescription->Location = System::Drawing::Point(10,
35);
                            this->tbPrDescription->Multiline = true;
                            this->tbPrDescription->Name = L"tbPrDescription";
                            this->tbPrDescription->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
                            this->tbPrDescription->Size = System::Drawing::Size(469, 50);
                            this->tbPrDescription->TabIndex = 6;
                            //
                            // label2
                            //
                            this->label2->AutoSize = true;
                            this->label2->Location = System::Drawing::Point(7, 19);
                            this->label2->Name = L"label2";
                            this->label2->Size = System::Drawing::Size(52, 13);
                            this->label2->TabIndex = 5;
                            this->label2->Text = L"Scenario:";
                            //
                            // panel8
                            //
                            this->panel8->BackColor = System::Drawing::Color::Transparent;
                            this->panel8->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"panel8.BackgroundImage")));
                            this->panel8->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
                            this->panel8->Controls->Add(this->trbVii);
                            this->panel8->Controls->Add(this->tbViiVal);
                            this->panel8->Location = System::Drawing::Point(9, 379);
                            this->panel8->Name = L"panel8";
                            this->panel8->Size = System::Drawing::Size(301, 52);
                            this->panel8->TabIndex = 17;
                            //
                            // trbVii
                            //
                            this->trbVii->Location = System::Drawing::Point(50, 14);
                            this->trbVii->Maximum = 1000;
                            this->trbVii->Name = L"trbVii";
                            this->trbVii->Size = System::Drawing::Size(184, 45);
                            this->trbVii->TabIndex = 7;
                            this->trbVii->TickStyle =
System::Windows::Forms::TickStyle::None;
                            this->trbVii->Value = 100;
                            this->trbVii->Scroll += gcnew System::EventHandler(this,
&MainForm::trbVii_Scroll);
                            //
                            // tbViiVal
                            //
                            this->tbViiVal->Location = System::Drawing::Point(240, 14);
                            this->tbViiVal->Name = L"tbViiVal";
                            this->tbViiVal->Size = System::Drawing::Size(48, 20);
                            this->tbViiVal->TabIndex = 11;
```

```
                            this->tbViiVal->Text = L"100";
                            //
                            // panel7
                            //
                            this->panel7->BackColor = System::Drawing::Color::Transparent;
                            this->panel7->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"panel7.BackgroundImage")));
                            this->panel7->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
                            this->panel7->Controls->Add(this->trbDIU);
                            this->panel7->Controls->Add(this->tbDIUVal);
                            this->panel7->Location = System::Drawing::Point(9, 263);
                            this->panel7->Name = L"panel7";
                            this->panel7->Size = System::Drawing::Size(301, 52);
                            this->panel7->TabIndex = 16;
                            //
                            // trbDIU
                            //
                            this->trbDIU->BackColor = System::Drawing::Color::Transparent;
                            this->trbDIU->Location = System::Drawing::Point(50, 14);
                            this->trbDIU->Maximum = 100;
                            this->trbDIU->Minimum = 1;
                            this->trbDIU->Name = L"trbDIU";
                            this->trbDIU->Size = System::Drawing::Size(184, 45);
                            this->trbDIU->TabIndex = 10;
                            this->trbDIU->TickFrequency = 10;
                            this->trbDIU->TickStyle =
System::Windows::Forms::TickStyle::None;
                            this->trbDIU->Value = 5;
                            this->trbDIU->Scroll += gcnew System::EventHandler(this,
&MainForm::trbDIU_Scroll);
                            //
                            // tbDIUVal
                            //
                            this->tbDIUVal->Location = System::Drawing::Point(240, 14);
                            this->tbDIUVal->Name = L"tbDIUVal";
                            this->tbDIUVal->Size = System::Drawing::Size(48, 20);
                            this->tbDIUVal->TabIndex = 11;
                            this->tbDIUVal->Text = L"5";
                            //
                            // panel6
                            //
                            this->panel6->BackColor = System::Drawing::Color::Transparent;
                            this->panel6->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"panel6.BackgroundImage")));
                            this->panel6->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
                            this->panel6->Controls->Add(this->trbDeltaA);
                            this->panel6->Controls->Add(this->tbDeltaAVal);
                            this->panel6->Location = System::Drawing::Point(9, 321);
                            this->panel6->Name = L"panel6";
                            this->panel6->Size = System::Drawing::Size(301, 52);
                            this->panel6->TabIndex = 15;
                            //
                            // trbDeltaA
                            //
                            this->trbDeltaA->Location = System::Drawing::Point(50, 14);
                            this->trbDeltaA->Maximum = 100;
                            this->trbDeltaA->Name = L"trbDeltaA";
                            this->trbDeltaA->Size = System::Drawing::Size(184, 45);
```

```
                            this->trbDeltaA->TabIndex = 9;
                            this->trbDeltaA->TickStyle =
System::Windows::Forms::TickStyle::None;
                            this->trbDeltaA->Value = 10;
                            this->trbDeltaA->Scroll += gcnew System::EventHandler(this,
&MainForm::trbDeltaA_Scroll);
                            //
                            // tbDeltaAVal
                            //
                            this->tbDeltaAVal->Location = System::Drawing::Point(240, 14);
                            this->tbDeltaAVal->Name = L"tbDeltaAVal";
                            this->tbDeltaAVal->Size = System::Drawing::Size(48, 20);
                            this->tbDeltaAVal->TabIndex = 11;
                            this->tbDeltaAVal->Text = L"0.1";
                            //
                            // panel5
                            //
                            this->panel5->BackColor = System::Drawing::Color::Transparent;
                            this->panel5->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"panel5.BackgroundImage")));
                            this->panel5->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
                            this->panel5->Controls->Add(this->trbDeltaT);
                            this->panel5->Controls->Add(this->tbDeltaTVal);
                            this->panel5->Location = System::Drawing::Point(9, 205);
                            this->panel5->Name = L"panel5";
                            this->panel5->Size = System::Drawing::Size(301, 52);
                            this->panel5->TabIndex = 14;
                            //
                            // trbDeltaT
                            //
                            this->trbDeltaT->Location = System::Drawing::Point(54, 14);
                            this->trbDeltaT->Maximum = 100;
                            this->trbDeltaT->Name = L"trbDeltaT";
                            this->trbDeltaT->Size = System::Drawing::Size(180, 45);
                            this->trbDeltaT->TabIndex = 8;
                            this->trbDeltaT->TickStyle =
System::Windows::Forms::TickStyle::None;
                            this->trbDeltaT->Value = 1;
                            this->trbDeltaT->Scroll += gcnew System::EventHandler(this,
&MainForm::trbDeltaT_Scroll);
                            //
                            // tbDeltaTVal
                            //
                            this->tbDeltaTVal->Location = System::Drawing::Point(240, 14);
                            this->tbDeltaTVal->Name = L"tbDeltaTVal";
                            this->tbDeltaTVal->Size = System::Drawing::Size(48, 20);
                            this->tbDeltaTVal->TabIndex = 11;
                            this->tbDeltaTVal->Text = L"1";
                            //
                            // panel4
                            //
                            this->panel4->BackColor = System::Drawing::Color::Transparent;
                            this->panel4->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"panel4.BackgroundImage")));
                            this->panel4->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
                            this->panel4->Controls->Add(this->trbBeta);
                            this->panel4->Controls->Add(this->tbBetaVal);
                            this->panel4->Location = System::Drawing::Point(9, 147);
```

```
                            this->panel4->Name = L"panel4";
                            this->panel4->Size = System::Drawing::Size(301, 52);
                            this->panel4->TabIndex = 13;
                            //
                            // trbBeta
                            //
                            this->trbBeta->Location = System::Drawing::Point(54, 14);
                            this->trbBeta->Maximum = 100;
                            this->trbBeta->Name = L"trbBeta";
                            this->trbBeta->Size = System::Drawing::Size(180, 45);
                            this->trbBeta->TabIndex = 6;
                            this->trbBeta->TickStyle =
System::Windows::Forms::TickStyle::None;
                            this->trbBeta->Value = 20;
                            this->trbBeta->Scroll += gcnew System::EventHandler(this,
&MainForm::trbBeta_Scroll);
                            //
                            // tbBetaVal
                            //
                            this->tbBetaVal->Location = System::Drawing::Point(240, 14);
                            this->tbBetaVal->Name = L"tbBetaVal";
                            this->tbBetaVal->Size = System::Drawing::Size(48, 20);
                            this->tbBetaVal->TabIndex = 11;
                            this->tbBetaVal->Text = L"0.2";
                            //
                            // panel3
                            //
                            this->panel3->BackColor = System::Drawing::Color::Transparent;
                            this->panel3->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"panel3.BackgroundImage")));
                            this->panel3->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
                            this->panel3->Controls->Add(this->trbAmplification);
                            this->panel3->Controls->Add(this->tbAVal);
                            this->panel3->Location = System::Drawing::Point(9, 89);
                            this->panel3->Name = L"panel3";
                            this->panel3->Size = System::Drawing::Size(301, 52);
                            this->panel3->TabIndex = 12;
                            //
                            // trbAmplification
                            //
                            this->trbAmplification->BackColor =
System::Drawing::Color::Transparent;
                            this->trbAmplification->Location = System::Drawing::Point(50,
13);
                            this->trbAmplification->Maximum = 1000;
                            this->trbAmplification->Name = L"trbAmplification";
                            this->trbAmplification->Size = System::Drawing::Size(184, 45);
                            this->trbAmplification->TabIndex = 5;
                            this->trbAmplification->TickFrequency = 100;
                            this->trbAmplification->TickStyle =
System::Windows::Forms::TickStyle::None;
                            this->trbAmplification->Value = 100;
                            this->trbAmplification->Scroll += gcnew
System::EventHandler(this, &MainForm::trbAmplification_Scroll);
                            //
                            // tbAVal
                            //
                            this->tbAVal->Location = System::Drawing::Point(240, 13);
                            this->tbAVal->Name = L"tbAVal";
                            this->tbAVal->Size = System::Drawing::Size(48, 20);
```

```
                            this->tbAVal->TabIndex = 11;
                            this->tbAVal->Text = L"10.0";
                            //
                            // btStop
                            //
                            this->btStop->Image = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"btStop.Image")));
                            this->btStop->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
                            this->btStop->Location = System::Drawing::Point(257, 444);
                            this->btStop->Name = L"btStop";
                            this->btStop->Size = System::Drawing::Size(40, 38);
                            this->btStop->TabIndex = 2;
                            this->btStop->TextAlign =
System::Drawing::ContentAlignment::BottomCenter;
                            this->btStop->UseVisualStyleBackColor = true;
                            this->btStop->Click += gcnew System::EventHandler(this,
&MainForm::btStop_Click);
                            //
                            // btStart
                            //
                            this->btStart->Image = (cli::safe_cast<System::Drawing::Image^
>(resources->GetObject(L"btStart.Image")));
                            this->btStart->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
                            this->btStart->Location = System::Drawing::Point(16, 444);
                            this->btStart->Name = L"btStart";
                            this->btStart->Size = System::Drawing::Size(40, 38);
                            this->btStart->TabIndex = 2;
                            this->btStart->TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
                            this->btStart->UseVisualStyleBackColor = true;
                            this->btStart->Click += gcnew System::EventHandler(this,
&MainForm::btStart_Click);
                            //
                            // panel2
                            //
                            this->panel2->BackColor = System::Drawing::Color::Transparent;
                            this->panel2->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"panel2.BackgroundImage")));
                            this->panel2->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
                            this->panel2->Controls->Add(this->label10);
                            this->panel2->Location = System::Drawing::Point(9, 3);
                            this->panel2->Name = L"panel2";
                            this->panel2->Size = System::Drawing::Size(301, 69);
                            this->panel2->TabIndex = 0;
                            //
                            // label10
                            //
                            this->label10->AutoSize = true;
                            this->label10->BackColor = System::Drawing::Color::Transparent;
                            this->label10->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 20, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                                  static_cast<System::Byte>(0)));
                            this->label10->ForeColor = System::Drawing::Color::White;
                            this->label10->Location = System::Drawing::Point(11, 16);
                            this->label10->Name = L"label10";
                            this->label10->Size = System::Drawing::Size(191, 31);
                            this->label10->TabIndex = 0;
                            this->label10->Text = L"Control Panel";
```

```
                            //
                            // saveFileDialog
                            //
                            this->saveFileDialog->FileName = L"output.xls";
                            //
                            // MainForm
                            //
                            this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                            this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
                            this->ClientSize = System::Drawing::Size(890, 606);
                            this->Controls->Add(this->splitContainer1);
                            this->Controls->Add(this->toolStrip1);
                            this->Controls->Add(this->statusStrip1);
                            this->Controls->Add(this->menuStrip1);
                            this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedSingle;
                            this->Icon = (cli::safe_cast<System::Drawing::Icon^
>(resources->GetObject(L"$this.Icon")));
                            this->MainMenuStrip = this->menuStrip1;
                            this->Name = L"MainForm";
                            this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
                            this->Text = L"Proof of Concept Application for Cybernetic
Modelling";
                            this->statusStrip1->ResumeLayout(false);
                            this->statusStrip1->PerformLayout();
                            this->menuStrip1->ResumeLayout(false);
                            this->menuStrip1->PerformLayout();
                            this->toolStrip1->ResumeLayout(false);
                            this->toolStrip1->PerformLayout();
                            this->splitContainer1->Panel1->ResumeLayout(false);
                            this->splitContainer1->Panel2->ResumeLayout(false);
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->splitContainer1))->EndInit();
                            this->splitContainer1->ResumeLayout(false);
                            this->panel1->ResumeLayout(false);
                            this->panel1->PerformLayout();
                            this->tabControl->ResumeLayout(false);
                            this->tabList->ResumeLayout(false);
                            this->tabChart->ResumeLayout(false);
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->chartView))->EndInit();
                            this->tabDetails->ResumeLayout(false);
                            this->tabDetails->PerformLayout();
                            this->groupBox1->ResumeLayout(false);
                            this->groupBox1->PerformLayout();
                            this->panel8->ResumeLayout(false);
                            this->panel8->PerformLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbVii))->EndInit();
                            this->panel7->ResumeLayout(false);
                            this->panel7->PerformLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbDIU))->EndInit();
                            this->panel6->ResumeLayout(false);
                            this->panel6->PerformLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbDeltaA))->EndInit();
                            this->panel5->ResumeLayout(false);
                            this->panel5->PerformLayout();
```

```
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbDeltaT))->EndInit();
                            this->panel4->ResumeLayout(false);
                            this->panel4->PerformLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbBeta))->EndInit();
                            this->panel3->ResumeLayout(false);
                            this->panel3->PerformLayout();
                            (cli::safe_cast<System::ComponentModel::ISupportInitialize^
>(this->trbAmplification))->EndInit();
                            this->panel2->ResumeLayout(false);
                            this->panel2->PerformLayout();
                            this->ResumeLayout(false);
                            this->PerformLayout();

                     }
#pragma endregion


       private: void UpdateDecisionMsgMethod(String^ msg){
                            tsslbDecisionMsg->Text=msg;
                     }
       private: void UpdateIterationMsgMethod(String^ msg){
                            tsslbIterationMsg->Text = msg;
                     }
       private: void AddListDataMethod(int nT, double dAfb, double dVo, double dDeltaA,
double dBetaGold, double dDeltaBeta,
                            double dAfbGold, double dDeltaAfbGold, double dVoGold, double
dDeltaAfbAfbG){

                                   // populate ListView
                                   ListViewItem ^ lviTemp = gcnew
ListViewItem(nT.ToString());

                                   lviTemp->SubItems->Add(dAfb.ToString());
                                   lviTemp->SubItems->Add(dVo.ToString());
                                   lviTemp->SubItems->Add(dDeltaA.ToString());
                                   lviTemp->SubItems->Add(dBetaGold.ToString());
                                   lviTemp->SubItems->Add(dDeltaBeta.ToString());

                                   lviTemp->SubItems->Add(dAfbGold.ToString());
                                   lviTemp->SubItems->Add(dDeltaAfbGold.ToString());
                                   lviTemp->SubItems->Add(dVoGold.ToString());
                                   lviTemp->SubItems->Add(dDeltaAfbAfbG.ToString());

                                   listView->Items->Add(lviTemp);

                                   lviTemp->EnsureVisible();

                                   // update Chart
                                   chartView->Series["Afb"]->Points->AddXY(nT,dAfb);
                                   //chartView->Series["Vo"]->Points->AddXY(nT,dVo);
                                   chartView->Series["AfbGold"]->Points-
>AddXY(nT,dAfbGold);

                                   // update labels
                                   lbAmplifier->Text = "A = " + A;
                                   lbOutput->Text = String::Format("Output =
{0:F3}\r\n",dVo);
                                   lbAfb->Text = String::Format("Afb = {0:F3}\r\n",dAfb);
                                   lbInput->Text = "Input = " + Viin;
                                   lbBetaFeedback->Text = "Beta = " + beta;
```

```
                        }


        private: System::Void newToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
                                onNew(sender,e);
                        }

        private: System::Void aboutToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
                                dlgAbout->ShowDialog();
                        }
        private: System::Void dataChartToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
                                tabControl->SelectTab(1);
                        }
        private: System::Void dataListToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
                                tabControl->SelectTab(0);
                        }
        private: System::Void detailsToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
                                tabControl->SelectTab(2);
                        }
        private: System::Void saveAsToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
                                onSaveAs(sender, e);
                        }
        private: System::Void tsbNew_Click(System::Object^  sender, System::EventArgs^  e) {
                                onNew(sender,e);
                        }

        private: void onSaveAs(System::Object^  sender, System::EventArgs^  e) {
                                //this->saveFileDialog->FileName = "*.xls";
                                //if (this->saveFileDialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){

                                Excel::Application ^exApp = gcnew Excel::ApplicationClass();

                                // Add a workbook (comes with three Worksheets)
                                Excel::Workbook^   exWb  = exApp->Workbooks-
>Add(Type::Missing);

                                // Delete the last two worksheets
                                //safe_cast<Excel::Worksheet^>(exApp->ActiveWorkbook-
>Sheets[3])->Delete();
                                //safe_cast<Excel::Worksheet^>(exApp->ActiveWorkbook-
>Sheets[2])->Delete();

                                // make worksheet active
                                safe_cast<Excel::Worksheet^>(exApp->ActiveWorkbook->Sheets-
>Item[1])->Select(Type::Missing);

                                // Create a variable for the active Worksheet's tracking handle
                                // (first Worksheet is the default active one)
                                Excel::Worksheet^  exWs  = safe_cast<Excel::Worksheet^>(exApp-
>ActiveSheet);

                                // Rename the active worksheet
                                // info sheet
                                exWs->Name = "Info";
                                int row=1;
```

```
                            int col=1;
                            // Put Column titles
                            exWs->Cells[row, col]     = "Name";
                            exWs->Cells[row, col+1]   = "Value";
                            exWs->Cells[row, col+2]   = "Description";

                            // set column width

                            exWs->Range["A1:B1", Type::Missing]->EntireColumn->ColumnWidth
= 12;
                            exWs->Range["C1", Type::Missing]->EntireColumn->ColumnWidth =
255;


                            row++;
                            // put column data
                            exWs->Cells[row, col]     = "Project";
                            exWs->Cells[row, col+1]   = "";
                            exWs->Cells[row, col+2]   = prInfo->ProjectName;
                            row++;
                            exWs->Cells[row, col]     = "Scenario";
                            exWs->Cells[row, col+1]   = "";
                            exWs->Cells[row, col+2]   = prInfo->ProjectDescription;
                            row++;
                            exWs->Cells[row, col]     = "Date";
                            exWs->Cells[row, col+1]   = "";
                            exWs->Cells[row, col+2]   = prInfo->ProjectDate;
                            row++;
                            exWs->Cells[row, col]     = "Viin";
                            exWs->Cells[row, col+1]   = Viin;
                            exWs->Cells[row, col+2]   = prInfo->InputDescription;
                            row++;
                            exWs->Cells[row, col]     = "Amplification";
                            exWs->Cells[row, col+1]   = A;
                            exWs->Cells[row, col+2]   = prInfo->AmplificationDescription;
                            row++;
                            exWs->Cells[row, col]     = "Beta";
                            exWs->Cells[row, col+1]   = beta;
                            exWs->Cells[row, col+2]   = prInfo->BetaDescription;
                            row++;
                            exWs->Cells[row, col]     = "Delta t";
                            exWs->Cells[row, col+1]   = delta_t;
                            exWs->Cells[row, col+2]   = prInfo->DeltaTDescription;
                            row++;
                            exWs->Cells[row, col]     = "Vo";
                            exWs->Cells[row, col+1]   = "";
                            exWs->Cells[row, col+2]   = prInfo->OutputDescription;

                            // data sheet
                            safe_cast<Excel::Worksheet^>(exApp->ActiveWorkbook->Sheets-
>Item[2])->Select(Type::Missing);

                            exWs  = safe_cast<Excel::Worksheet^>(exApp->ActiveSheet);

                            exWs->Name = "Data";


                            row=1;
                            col=1;
                            // Put Column titles
                            exWs->Cells[row, col]     = "t";
                            exWs->Cells[row, col+1]   = "Afb";
                            exWs->Cells[row, col+2]   = "Output(Vo)";
```

```
                                exWs->Cells[row, col+3]   = "DeltaA";
                                exWs->Cells[row, col+4]   = "BetaGoldRatio";
                                exWs->Cells[row, col+5]   = "DeltaBeta";
                                exWs->Cells[row, col+6]   = "AfbGoldR";
                                exWs->Cells[row, col+7]   = "DeltaAfbGoldR";
                                exWs->Cells[row, col+8]   = "OutputGoldR";
                                exWs->Cells[row, col+9]   = "DeltaAfbAfbG";

                                row++;
                                // add data

                                for(int i=0; i<listView->Items->Count; ++i){
                                        exWs->Cells[row, col]     = listView->Items[i]-
>SubItems[0]->Text; //"t";
                                        exWs->Cells[row, col+1]   = listView->Items[i]-
>SubItems[1]->Text; //"Afb";
                                        exWs->Cells[row, col+2]   = listView->Items[i]-
>SubItems[2]->Text; //"Output(Vo)";
                                        exWs->Cells[row, col+3]   = listView->Items[i]-
>SubItems[3]->Text; //"DeltaA";
                                        exWs->Cells[row, col+4]   = listView->Items[i]-
>SubItems[4]->Text; //"BetaGold";
                                        exWs->Cells[row, col+5]   = listView->Items[i]-
>SubItems[5]->Text; //"DeltaBeta";

                                        exWs->Cells[row, col+6]   = listView->Items[i]-
>SubItems[6]->Text; //"AfbGold";
                                        exWs->Cells[row, col+7]   = listView->Items[i]-
>SubItems[7]->Text; //"DeltaAfbGold";
                                        exWs->Cells[row, col+8]   = listView->Items[i]-
>SubItems[8]->Text; //"OutpurGold";
                                        exWs->Cells[row, col+9]   = listView->Items[i]-
>SubItems[9]->Text; //"DeltaAfbAfbG";

                                        row++;
                                }

                                // set column width

                                exWs->Range["A1",    Type::Missing]->EntireColumn->ColumnWidth
= 6;
                                exWs->Range["B1:J1", Type::Missing]->EntireColumn->ColumnWidth
= 14;

                                // Show the Workbook
                                exApp->Visible = true;

                                //exWb->SaveAs(saveFileDialog-
>FileName,nullptr,nullptr,nullptr,nullptr,nullptr,Excel::XlSaveAsAccessMode::xlNoChange,
                                //    nullptr,nullptr,nullptr,nullptr,nullptr);

                                //exApp->Save(saveFileDialog->FileName);
                                //exApp->Quit();
                                //}
                        }
        private: void onOpen(System::Object^  sender, System::EventArgs^  e) {

                                if (openFileDialog->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
                                        // to do
                                        MessageBox::Show("Not available yet!");
```

```
                        }
                    }
        private: void onNew(System::Object^  sender, System::EventArgs^  e) {
                        dlgPrInfo->clearFields();
                        if (dlgPrInfo->ShowDialog() ==
System::Windows::Forms::DialogResult::OK){
                            dlgPrInfo->getProjectInformation(prInfo);


                            tbPrName->Text = prInfo->ProjectName;
                            tbPrDescription->Text = prInfo->ProjectDescription;
                            tbInputDescription->Text = prInfo->InputDescription;
                            tbOutputDescription->Text = prInfo->OutputDescription;
                            tbAmplificationDescription->Text = prInfo-
>AmplificationDescription;
                            tbBetaDescription->Text = prInfo->BetaDescription;
                            tbDeltaTDescription->Text = prInfo->DeltaTDescription;

                            switch (prInfo->ProjectType)
                            {
                            case 0:{
                                cbPrType->SelectedIndex = 0;
                                break;
                                    }
                            case 1:{
                                cbPrType->SelectedIndex = 1;
                                break;
                                    }
                            }

                            dtPrDate->Value = prInfo->ProjectDate;
                        }
                    }
        private: System::Void openToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
                        onOpen(sender,e);
                    }
        private: System::Void tsbOpen_Click(System::Object^  sender, System::EventArgs^  e) {
                        onOpen(sender,e);
                    }
        private: System::Void trbDIU_Scroll(System::Object^  sender, System::EventArgs^  e) {
                        tbDIUVal->Text = (trbDIU->Value).ToString();
                    }
        private: System::Void trbVii_Scroll(System::Object^  sender, System::EventArgs^  e) {
                        tbViiVal->Text = (trbVii->Value).ToString();
                    }
        private: System::Void trbDeltaA_Scroll(System::Object^  sender, System::EventArgs^
e) {
                        tbDeltaAVal->Text = (1.0*trbDeltaA->Value/100.0).ToString();
                    }
        private: System::Void trbDeltaT_Scroll(System::Object^  sender, System::EventArgs^
e) {
                        tbDeltaTVal->Text = (trbDeltaT->Value).ToString();
                    }
        private: System::Void trbBeta_Scroll(System::Object^  sender, System::EventArgs^  e)
{
                        tbBetaVal->Text = (1.0*trbBeta->Value/100.0).ToString();
                    }
        private: System::Void trbAmplification_Scroll(System::Object^  sender,
System::EventArgs^  e) {
                        tbAVal->Text = (1.0*trbAmplification->Value/10.0).ToString();
                    }
```

```
	private: void clearControls(){
				listView->Items->Clear();

				for(int i=0; i<chartView->Series->Count; ++i){
					chartView->Series[i]->Points->Clear();

				}
			}
	private: System::Void btStart_Click(System::Object^  sender, System::EventArgs^  e) {
				onStart(sender,e);
			}
	private: System::Void btStop_Click(System::Object^  sender, System::EventArgs^  e) {
				onStop(sender,e);
			}
	private: System::Void tsbStart_Click(System::Object^  sender, System::EventArgs^  e)
{
				onStart(sender,e);
			}
	private: System::Void tsbStop_Click(System::Object^  sender, System::EventArgs^  e) {
				onStop(sender,e);
			}
	private: System::Void stopToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
				onStop(sender,e);
			}
	private: System::Void startToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
				onStart(sender,e);
			}
	private: System::Void exitToolStripMenuItem_Click(System::Object^  sender,
System::EventArgs^  e) {
				onStop(sender,e);
				Application::Exit();
			}
	private: void onStart(System::Object^  sender, System::EventArgs^  e) {
				clearControls();

				// get values form UI
				Viin = Convert::ToDouble(tbViiVal->Text);
				A = Convert::ToDouble(tbAVal->Text);
				beta = Convert::ToDouble(tbBetaVal->Text);

				delta_t = Convert::ToInt32(tbDeltaTVal->Text);
				dwTimeout = Convert::ToInt32(tbDIUVal->Text);

				delta_A = Convert::ToDouble(tbDeltaAVal->Text);

				homeostaticState = FALSE;

				if(hTesterThread!=NULL){
					TerminateThread(hTesterThread, -10);
					CloseHandle(hTesterThread);
					hTesterThread = NULL;
				}
				if(hDecisionThread!=NULL){

					TerminateThread(hDecisionThread, -11);
					CloseHandle(hDecisionThread);
					hDecisionThread = NULL;
				}
				if(hIterationThread!=NULL){
					TerminateThread(hIterationThread, -12);
```

```
                                CloseHandle(hIterationThread);
                                hIterationThread = NULL;
                        }

                hTesterThread = CreateThread(NULL, 0, TesterThreadProc, NULL, 0, NULL);
                hDecisionThread = CreateThread(NULL, 0, DecisionThreadProc, NULL, 0, NULL);
                hIterationThread = CreateThread(NULL, 0, IterationThreadProc, NULL, 0, NULL);
                        }
        private: void onStop(System::Object^  sender, System::EventArgs^  e) {


                        if(hTesterThread!=NULL){
                                TerminateThread(hTesterThread, -10);
                                CloseHandle(hTesterThread);
                                hTesterThread = NULL;
                        }
                        if(hDecisionThread!=NULL){
                                TerminateThread(hDecisionThread, -11);
                                CloseHandle(hDecisionThread);
                                hDecisionThread = NULL;
                        }
                        if(hIterationThread!=NULL){
                                TerminateThread(hIterationThread, -12);
                                CloseHandle(hIterationThread);
                                hIterationThread = NULL;
                        }
                }
        };

}

[STAThreadAttribute]
int main()
{
        // init events
        hCloseEvent =  CreateEvent(NULL,TRUE,FALSE,NULL);
        hContinueEvent = CreateEvent(NULL,TRUE,FALSE,NULL);
        hEventObject = CreateEvent(NULL,TRUE,FALSE,NULL);
        ResetEvent(hCloseEvent);
        ResetEvent(hContinueEvent);
        ResetEvent(hEventObject);

        using namespace ProjectGUI;
        Application::EnableVisualStyles();
        Application::SetCompatibleTextRenderingDefault(false);
        MainForm ^f = gcnew MainForm;
        f->ShowDialog();


        // Store Thread handles in Array of Thread
        HANDLE hTempHandles[] = { hTesterThread, hDecisionThread, hIterationThread};
        int nNumHandles = sizeof(hTempHandles) / sizeof(hTempHandles[0]);
        int iHandles = 0;


        // fine active handles
        for( int n = 0; n < nNumHandles; n ++ ){
                if(hTempHandles[n]!=NULL){
                        iHandles++;
                }
        }
```

```
	// Wait until all threads have terminated.

	if (iHandles > 0)
	{
		// There are threads to shut down - set the closure event.
		SetEvent (hCloseEvent);

		switch (WaitForMultipleObjects (iHandles, hTempHandles, TRUE,
dwCloseWaitMillies))
		{
		case WAIT_FAILED:
			MessageBox::Show("WaitForMultipleObjects failed, " + GetLastError());
			break;

		case WAIT_OBJECT_0:
			MessageBox::Show("All COM threads closed OK");
			break;

		case WAIT_ABANDONED:
			MessageBox::Show("Wait abandoned");
			break;

		case WAIT_TIMEOUT:
			MessageBox::Show("Timed out waiting for COM threads");
			break;
		}

		if (ResetEvent (hCloseEvent))
			;// Reset close event for next use
		else
		{
			MessageBox::Show("Attempt to reset close event failed " +
GetLastError());
		}
	}
	else
	{
		//No active threads, exiting
	}

	// Close all thread handles upon completion.

	for( int n = 0; n < nNumHandles; n ++ ){
		if(hTempHandles[n]!=NULL){
			CloseHandle( hTempHandles[n] );
		}
	}
	//DWORD result1, result2; // these variables will receive the return values
	//rc = GetExitCodeThread(threadArray[0],&result1);
	//rc = GetExitCodeThread(threadArray[1],&result2);

	// close events
	CloseHandle(hEventObject);
	CloseHandle(hContinueEvent);
	CloseHandle(hCloseEvent);

	return 0;
}
```

```
#pragma once
using namespace System;
using namespace System::Timers;

public enum ProjectKind {Enumerative, Reconstructive};

public ref class ProjectInformation
{
public:
	String^ m_prName;
	String^ m_prOwner;
	String^ m_prDescription;
	String^ m_inputDescription;
	String^ m_outputDescription;

	String^ m_amplificationDescription;
	String^ m_betaDescription;
	String^ m_deltaTDescription;

	ProjectKind m_prType;
	DateTime m_prDate;
	String^ m_outputFileName;

	property String^ ProjectName{
		String^ get() {return m_prName;}
		void set(String^ val) {m_prName = val;}
	}

	property String^ ProjectOwner{
		String^ get() {return m_prOwner;}
		void set(String^ val) {m_prOwner = val;}
	}
	property String^ ProjectDescription{
		String^ get() {return m_prDescription;}
		void set(String^ val) {m_prDescription = val;}
	}

	property String^ InputDescription{
		String^ get() {return m_inputDescription;}
		void set(String^ val) {m_inputDescription = val;}
	}
	property String^ OutputDescription{
		String^ get() {return m_outputDescription;}
		void set(String^ val) {m_outputDescription = val;}
	}
	property String^ AmplificationDescription{
		String^ get() {return m_amplificationDescription;}
		void set(String^ val) {m_amplificationDescription = val;}
	}
	property String^ BetaDescription{
		String^ get() {return m_betaDescription;}
		void set(String^ val) {m_betaDescription = val;}
	}
	property String^ DeltaTDescription{
		String^ get() {return m_deltaTDescription;}
		void set(String^ val) {m_deltaTDescription = val;}
	}
	property ProjectKind ProjectType{
		ProjectKind get() {return m_prType;}
		void set(ProjectKind val) {m_prType = val;}
	}
	property DateTime ProjectDate
```

```
	{
		DateTime get() { return m_prDate; }
		void set(DateTime val) {	m_prDate = val; }
	};

	property String^ OutputFileName{
		String^ get() {return m_outputFileName;}
		void set(String^ val) {m_outputFileName = val;}
	}

public:
	ProjectInformation(void)
	{
		setDefaults();
	}
	void setDefaults(){
		m_prName = "New project";
		m_prOwner = "";
		m_prDescription = "";
		m_inputDescription = "Input";
		m_outputDescription = "Output";

		m_amplificationDescription = "";
		m_betaDescription = "";
		m_deltaTDescription = "";

		m_prType = ProjectKind::Enumerative;
		m_prDate = DateTime::Now;
		m_outputFileName = "output.txt";
	}
};

#pragma once
#include "ProjectInformation.h"

namespace ProjectGUI {

	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;

	/// <summary>
	/// Summary for ProjectInfoDlg
	/// </summary>
	public ref class ProjectInfoDlg : public System::Windows::Forms::Form
	{
	public:
		ProjectInfoDlg(void)
		{
			InitializeComponent();
			cbPrType->SelectedIndex=0;

		}

	protected:
		/// <summary>
		/// Clean up any resources being used.
		/// </summary>
		~ProjectInfoDlg()
```

```
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::Button^  btnOk;
	private: System::Windows::Forms::DateTimePicker^  dtPrDate;
	private: System::Windows::Forms::ComboBox^  cbPrType;
	private: System::Windows::Forms::TextBox^  tbPrName;
	private: System::Windows::Forms::Label^  label1;
	private: System::Windows::Forms::Label^  label2;
	private: System::Windows::Forms::TextBox^  tbPrDescription;
	private: System::Windows::Forms::Label^  label3;
	private: System::Windows::Forms::Label^  label4;
	private: System::Windows::Forms::TextBox^  tbInputDescription;
	private: System::Windows::Forms::Label^  label5;
	private: System::Windows::Forms::TextBox^  tbOutputDescription;
	private: System::Windows::Forms::Label^  label6;
	private: System::Windows::Forms::GroupBox^  groupBox1;
	private: System::Windows::Forms::TextBox^  tbBetaDescription;
	private: System::Windows::Forms::Label^  label8;
	private: System::Windows::Forms::TextBox^  tbAmplificationDescription;
	private: System::Windows::Forms::Label^  label7;
	private: System::Windows::Forms::Label^  label9;
	private: System::Windows::Forms::TextBox^  tbDeltaTDescription;
	protected:

	private:
		/// <summary>
		/// Required designer variable.
		/// </summary>
		System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
		/// <summary>
		/// Required method for Designer support - do not modify
		/// the contents of this method with the code editor.
		/// </summary>
		void InitializeComponent(void)
		{
			System::ComponentModel::ComponentResourceManager^  resources = (gcnew
System::ComponentModel::ComponentResourceManager(ProjectInfoDlg::typeid));
			this->btnOk = (gcnew System::Windows::Forms::Button());
			this->dtPrDate = (gcnew System::Windows::Forms::DateTimePicker());
			this->cbPrType = (gcnew System::Windows::Forms::ComboBox());
			this->tbPrName = (gcnew System::Windows::Forms::TextBox());
			this->label1 = (gcnew System::Windows::Forms::Label());
			this->label2 = (gcnew System::Windows::Forms::Label());
			this->tbPrDescription = (gcnew System::Windows::Forms::TextBox());
			this->label3 = (gcnew System::Windows::Forms::Label());
			this->label4 = (gcnew System::Windows::Forms::Label());
			this->tbInputDescription = (gcnew System::Windows::Forms::TextBox());
			this->label5 = (gcnew System::Windows::Forms::Label());
			this->tbOutputDescription = (gcnew System::Windows::Forms::TextBox());
			this->label6 = (gcnew System::Windows::Forms::Label());
			this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
			this->tbDeltaTDescription = (gcnew System::Windows::Forms::TextBox());
			this->label9 = (gcnew System::Windows::Forms::Label());
			this->tbBetaDescription = (gcnew System::Windows::Forms::TextBox());
			this->label8 = (gcnew System::Windows::Forms::Label());
```

```
                    this->tbAmplificationDescription = (gcnew
System::Windows::Forms::TextBox());
                    this->label7 = (gcnew System::Windows::Forms::Label());
                    this->groupBox1->SuspendLayout();
                    this->SuspendLayout();
                    //
                    // btnOk
                    //
                    this->btnOk->DialogResult = System::Windows::Forms::DialogResult::OK;
                    this->btnOk->Location = System::Drawing::Point(425, 444);
                    this->btnOk->Name = L"btnOk";
                    this->btnOk->Size = System::Drawing::Size(75, 23);
                    this->btnOk->TabIndex = 0;
                    this->btnOk->Text = L"OK";
                    this->btnOk->UseVisualStyleBackColor = true;
                    this->btnOk->Click += gcnew System::EventHandler(this,
&ProjectInfoDlg::btnOk_Click);
                    //
                    // dtPrDate
                    //
                    this->dtPrDate->Location = System::Drawing::Point(285, 93);
                    this->dtPrDate->Name = L"dtPrDate";
                    this->dtPrDate->Size = System::Drawing::Size(206, 20);
                    this->dtPrDate->TabIndex = 1;
                    //
                    // cbPrType
                    //
                    this->cbPrType->FormattingEnabled = true;
                    this->cbPrType->Items->AddRange(gcnew cli::array< System::Object^  >(2)
{L"Enumerative", L"Reconstructive"});
                    this->cbPrType->Location = System::Drawing::Point(90, 91);
                    this->cbPrType->Name = L"cbPrType";
                    this->cbPrType->Size = System::Drawing::Size(115, 21);
                    this->cbPrType->TabIndex = 2;
                    //
                    // tbPrName
                    //
                    this->tbPrName->Location = System::Drawing::Point(90, 59);
                    this->tbPrName->Name = L"tbPrName";
                    this->tbPrName->Size = System::Drawing::Size(401, 20);
                    this->tbPrName->TabIndex = 3;
                    //
                    // label1
                    //
                    this->label1->AutoSize = true;
                    this->label1->Location = System::Drawing::Point(12, 62);
                    this->label1->Name = L"label1";
                    this->label1->Size = System::Drawing::Size(72, 13);
                    this->label1->TabIndex = 4;
                    this->label1->Text = L"Project name:";
                    //
                    // label2
                    //
                    this->label2->AutoSize = true;
                    this->label2->Location = System::Drawing::Point(10, 23);
                    this->label2->Name = L"label2";
                    this->label2->Size = System::Drawing::Size(52, 13);
                    this->label2->TabIndex = 5;
                    this->label2->Text = L"Scenario:";
                    //
                    // tbPrDescription
                    //
```

```
                    this->tbPrDescription->Location = System::Drawing::Point(10, 40);
                    this->tbPrDescription->Multiline = true;
                    this->tbPrDescription->Name = L"tbPrDescription";
                    this->tbPrDescription->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
                    this->tbPrDescription->Size = System::Drawing::Size(469, 50);
                    this->tbPrDescription->TabIndex = 6;
                    //
                    // label3
                    //
                    this->label3->AutoSize = true;
                    this->label3->Location = System::Drawing::Point(211, 95);
                    this->label3->Name = L"label3";
                    this->label3->Size = System::Drawing::Size(68, 13);
                    this->label3->TabIndex = 7;
                    this->label3->Text = L"Current date:";
                    //
                    // label4
                    //
                    this->label4->AutoSize = true;
                    this->label4->Location = System::Drawing::Point(12, 94);
                    this->label4->Name = L"label4";
                    this->label4->Size = System::Drawing::Size(66, 13);
                    this->label4->TabIndex = 8;
                    this->label4->Text = L"Project type:";
                    //
                    // tbInputDescription
                    //
                    this->tbInputDescription->Location = System::Drawing::Point(10, 111);
                    this->tbInputDescription->Name = L"tbInputDescription";
                    this->tbInputDescription->Size = System::Drawing::Size(469, 20);
                    this->tbInputDescription->TabIndex = 9;
                    //
                    // label5
                    //
                    this->label5->AutoSize = true;
                    this->label5->Location = System::Drawing::Point(10, 94);
                    this->label5->Name = L"label5";
                    this->label5->Size = System::Drawing::Size(34, 13);
                    this->label5->TabIndex = 10;
                    this->label5->Text = L"Input:";
                    //
                    // tbOutputDescription
                    //
                    this->tbOutputDescription->Location = System::Drawing::Point(10, 152);
                    this->tbOutputDescription->Name = L"tbOutputDescription";
                    this->tbOutputDescription->Size = System::Drawing::Size(471, 20);
                    this->tbOutputDescription->TabIndex = 11;
                    //
                    // label6
                    //
                    this->label6->AutoSize = true;
                    this->label6->Location = System::Drawing::Point(10, 135);
                    this->label6->Name = L"label6";
                    this->label6->Size = System::Drawing::Size(42, 13);
                    this->label6->TabIndex = 12;
                    this->label6->Text = L"Output:";
                    //
                    // groupBox1
                    //
                    this->groupBox1->Controls->Add(this->tbDeltaTDescription);
                    this->groupBox1->Controls->Add(this->label9);
```

```
                    this->groupBox1->Controls->Add(this->tbBetaDescription);
                    this->groupBox1->Controls->Add(this->label8);
                    this->groupBox1->Controls->Add(this->tbAmplificationDescription);
                    this->groupBox1->Controls->Add(this->label7);
                    this->groupBox1->Controls->Add(this->label6);
                    this->groupBox1->Controls->Add(this->tbOutputDescription);
                    this->groupBox1->Controls->Add(this->label5);
                    this->groupBox1->Controls->Add(this->tbInputDescription);
                    this->groupBox1->Controls->Add(this->tbPrDescription);
                    this->groupBox1->Controls->Add(this->label2);
                    this->groupBox1->Location = System::Drawing::Point(12, 128);
                    this->groupBox1->Name = L"groupBox1";
                    this->groupBox1->Size = System::Drawing::Size(488, 310);
                    this->groupBox1->TabIndex = 13;
                    this->groupBox1->TabStop = false;
                    this->groupBox1->Text = L"Descriptions";
                    //
                    // tbDeltaTDescription
                    //
                    this->tbDeltaTDescription->Location = System::Drawing::Point(10, 275);
                    this->tbDeltaTDescription->Name = L"tbDeltaTDescription";
                    this->tbDeltaTDescription->Size = System::Drawing::Size(471, 20);
                    this->tbDeltaTDescription->TabIndex = 18;
                    //
                    // label9
                    //
                    this->label9->AutoSize = true;
                    this->label9->Location = System::Drawing::Point(10, 258);
                    this->label9->Name = L"label9";
                    this->label9->Size = System::Drawing::Size(41, 13);
                    this->label9->TabIndex = 17;
                    this->label9->Text = L"Delta t:";
                    //
                    // tbBetaDescription
                    //
                    this->tbBetaDescription->Location = System::Drawing::Point(10, 234);
                    this->tbBetaDescription->Name = L"tbBetaDescription";
                    this->tbBetaDescription->Size = System::Drawing::Size(469, 20);
                    this->tbBetaDescription->TabIndex = 16;
                    //
                    // label8
                    //
                    this->label8->AutoSize = true;
                    this->label8->Location = System::Drawing::Point(10, 217);
                    this->label8->Name = L"label8";
                    this->label8->Size = System::Drawing::Size(32, 13);
                    this->label8->TabIndex = 15;
                    this->label8->Text = L"Beta:";
                    //
                    // tbAmplificationDescription
                    //
                    this->tbAmplificationDescription->Location = System::Drawing::Point(10,
193);
                    this->tbAmplificationDescription->Name = L"tbAmplificationDescription";
                    this->tbAmplificationDescription->Size = System::Drawing::Size(469,
20);
                    this->tbAmplificationDescription->TabIndex = 14;
                    //
                    // label7
                    //
                    this->label7->AutoSize = true;
                    this->label7->Location = System::Drawing::Point(10, 176);
```

```
                    this->label7->Name = L"label7";
                    this->label7->Size = System::Drawing::Size(69, 13);
                    this->label7->TabIndex = 13;
                    this->label7->Text = L"Amplification:";
                    //
                    // ProjectInfoDlg
                    //
                    this->AcceptButton = this->btnOk;
                    this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                    this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
                    this->ClientSize = System::Drawing::Size(517, 479);
                    this->Controls->Add(this->groupBox1);
                    this->Controls->Add(this->label4);
                    this->Controls->Add(this->label3);
                    this->Controls->Add(this->label1);
                    this->Controls->Add(this->tbPrName);
                    this->Controls->Add(this->cbPrType);
                    this->Controls->Add(this->dtPrDate);
                    this->Controls->Add(this->btnOk);
                    this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedSingle;
                    this->Icon = (cli::safe_cast<System::Drawing::Icon^  >(resources-
>GetObject(L"$this.Icon")));
                    this->MaximizeBox = false;
                    this->MinimizeBox = false;
                    this->Name = L"ProjectInfoDlg";
                    this->ShowInTaskbar = false;
                    this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterParent;
                    this->Text = L"Project Information";
                    this->groupBox1->ResumeLayout(false);
                    this->groupBox1->PerformLayout();
                    this->ResumeLayout(false);
                    this->PerformLayout();

             }
#pragma endregion

       public: void getProjectInformation(ProjectInformation^ prInfo){
                           prInfo->ProjectName = tbPrName->Text;
                           prInfo->ProjectDescription = tbPrDescription->Text;
                           prInfo->InputDescription = tbInputDescription->Text;
                           prInfo->OutputDescription = tbOutputDescription->Text;
                           prInfo->AmplificationDescription = tbAmplificationDescription-
>Text;
                           prInfo->BetaDescription = tbBetaDescription->Text;
                           prInfo->DeltaTDescription = tbDeltaTDescription->Text;

                           switch (cbPrType->SelectedIndex)
                           {
                           case 0:{
                                  prInfo->ProjectType = ProjectKind::Enumerative;
                                  break;
                                     }
                           case 1:{
                                  prInfo->ProjectType = ProjectKind::Reconstructive;
                                  break;
                                     }
                           }

                           prInfo->ProjectDate = dtPrDate->Value;
```

```
	}
	public: void clearFields(){
				tbAmplificationDescription->Clear();
				tbBetaDescription->Clear();
				tbDeltaTDescription->Clear();
				tbInputDescription->Clear();
				tbOutputDescription->Clear();
				tbPrDescription->Clear();
				tbPrName->Clear();
				dtPrDate->Value = DateTime::Now;
				cbPrType->SelectedIndex=0;

			}

	private: System::Void btnOk_Click(System::Object^  sender, System::EventArgs^  e) {
	//				this->Close();
			 }
};
}

#pragma once

namespace ProjectGUI {

	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;

	/// <summary>
	/// Summary for AboutDlg
	/// </summary>
	public ref class AboutDlg : public System::Windows::Forms::Form
	{
	public:
		AboutDlg(void)
		{
			InitializeComponent();
			//
			//TODO: Add the constructor code here
			//
		}

	protected:
		/// <summary>
		/// Clean up any resources being used.
		/// </summary>
		~AboutDlg()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::Button^  btnOK;
	private: System::Windows::Forms::Label^  label1;
	private: System::Windows::Forms::PictureBox^  pictureBox1;
	private: System::Windows::Forms::Label^  label2;
	private: System::Windows::Forms::Label^  label3;
	private: System::Windows::Forms::Label^  label4;
```

```
        private: System::Windows::Forms::LinkLabel^  linkLabel1;
        protected:

        protected:


        private:
                /// <summary>
                /// Required designer variable.
                /// </summary>
                System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
                /// <summary>
                /// Required method for Designer support - do not modify
                /// the contents of this method with the code editor.
                /// </summary>
                void InitializeComponent(void)
                {
                        System::ComponentModel::ComponentResourceManager^  resources = (gcnew
System::ComponentModel::ComponentResourceManager(AboutDlg::typeid));
                        this->btnOK = (gcnew System::Windows::Forms::Button());
                        this->label1 = (gcnew System::Windows::Forms::Label());
                        this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
                        this->label2 = (gcnew System::Windows::Forms::Label());
                        this->label3 = (gcnew System::Windows::Forms::Label());
                        this->label4 = (gcnew System::Windows::Forms::Label());
                        this->linkLabel1 = (gcnew System::Windows::Forms::LinkLabel());
                        (cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(this-
>pictureBox1))->BeginInit();
                        this->SuspendLayout();
                        //
                        // btnOK
                        //
                        this->btnOK->Anchor = System::Windows::Forms::AnchorStyles::None;
                        this->btnOK->DialogResult = System::Windows::Forms::DialogResult::OK;
                        this->btnOK->Location = System::Drawing::Point(555, 144);
                        this->btnOK->Name = L"btnOK";
                        this->btnOK->Size = System::Drawing::Size(75, 23);
                        this->btnOK->TabIndex = 0;
                        this->btnOK->Text = L"OK";
                        this->btnOK->UseVisualStyleBackColor = true;
                        //
                        // label1
                        //
                        this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                                static_cast<System::Byte>(0)));
                        this->label1->Location = System::Drawing::Point(194, 8);
                        this->label1->Name = L"label1";
                        this->label1->Size = System::Drawing::Size(436, 53);
                        this->label1->TabIndex = 1;
                        this->label1->Text = L"Proof of Concept Application for Non-Algorithmic
Cybernetic Modelling of Living S"
                                L"ystems";
                        this->label1->TextAlign =
System::Drawing::ContentAlignment::MiddleCenter;
                        //
                        // pictureBox1
                        //
```

```
                    this->pictureBox1->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^  >(resources-
>GetObject(L"pictureBox1.BackgroundImage")));
                    this->pictureBox1->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Zoom;
                    this->pictureBox1->Location = System::Drawing::Point(2, 8);
                    this->pictureBox1->Name = L"pictureBox1";
                    this->pictureBox1->Size = System::Drawing::Size(186, 159);
                    this->pictureBox1->TabIndex = 2;
                    this->pictureBox1->TabStop = false;
                    //
                    // label2
                    //
                    this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                           static_cast<System::Byte>(0)));
                    this->label2->Location = System::Drawing::Point(192, 105);
                    this->label2->Name = L"label2";
                    this->label2->Size = System::Drawing::Size(439, 39);
                    this->label2->TabIndex = 3;
                    this->label2->Text = L"Copyright: “Lucian Blaga” University of Sibiu,
Faculty of Engineering";
                    //
                    // label3
                    //
                    this->label3->AutoSize = true;
                    this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 9, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                           static_cast<System::Byte>(0)));
                    this->label3->Location = System::Drawing::Point(364, 123);
                    this->label3->Name = L"label3";
                    this->label3->Size = System::Drawing::Size(84, 15);
                    this->label3->TabIndex = 4;
                    this->label3->Text = L"Sibiu, 2013 ";
                    //
                    // label4
                    //
                    this->label4->AutoSize = true;
                    this->label4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans
Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                           static_cast<System::Byte>(0)));
                    this->label4->Location = System::Drawing::Point(354, 72);
                    this->label4->Name = L"label4";
                    this->label4->Size = System::Drawing::Size(106, 13);
                    this->label4->TabIndex = 5;
                    this->label4->Text = L"by Ralf D. Fabian";
                    //
                    // linkLabel1
                    //
                    this->linkLabel1->AutoSize = true;
                    this->linkLabel1->Location = System::Drawing::Point(352, 87);
                    this->linkLabel1->Name = L"linkLabel1";
                    this->linkLabel1->Size = System::Drawing::Size(111, 13);
                    this->linkLabel1->TabIndex = 6;
                    this->linkLabel1->TabStop = true;
                    this->linkLabel1->Text = L"ralf.fabian@ulbsibiu.ro";
                    //
                    // AboutDlg
                    //
                    this->AcceptButton = this->btnOK;
                    this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                    this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
```

```
                    this->ClientSize = System::Drawing::Size(638, 172);
                    this->Controls->Add(this->linkLabel1);
                    this->Controls->Add(this->label4);
                    this->Controls->Add(this->label3);
                    this->Controls->Add(this->label2);
                    this->Controls->Add(this->pictureBox1);
                    this->Controls->Add(this->label1);
                    this->Controls->Add(this->btnOK);
                    this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedSingle;
                    this->Icon = (cli::safe_cast<System::Drawing::Icon^  >(resources-
>GetObject(L"$this.Icon")));
                    this->MaximizeBox = false;
                    this->MinimizeBox = false;
                    this->Name = L"AboutDlg";
                    this->ShowInTaskbar = false;
                    this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
                    this->Text = L"About";
                    (cli::safe_cast<System::ComponentModel::ISupportInitialize^  >(this-
>pictureBox1))->EndInit();
                    this->ResumeLayout(false);
                    this->PerformLayout();

                }
#pragma endregion
        };
}
```