Fuzzy Dynamics in Software Project Simulation and Support

J. Fernández Ramil and M. M. Lehman

Imperial College of Science, Technology and Medicine
Department of Computing
London SW7 2BZ
[JCF1,MML]@doc.ic.ac.uk

Abstract. Established simulation techniques require quantification of relevant aspects of the entity whose behaviour is being investigated. The data expressing the quantification provides a disciplined link between the simulation model and the domain being modelled. In this connection, domains such as software projects or processes that are, for example, fuzzy, uncertain or lack precise data quantification, may present difficulties. Alternative paradigms must be established to capture, in some sense, the behaviour to be simulated. One such paradigm, complementary to existing techniques, is provided by *fuzzy dynamics*. Though appearing promising, this suggestion awaits validation in real-world situations.

1 Introduction

Over the last decade widespread interest has developed in the use of *quantitative* models to support decision making in, for example, policy formulation, planning, controlling and general management of software projects[1] and management of software project resources. The impact of decisions on project *behaviour* can be investigated by simulating it as is, that is without change, and after implementation of a change. Simulation may also be used to forecast behaviour under varying external circumstances, for management training [21] and for software process improvement [2] for example.

An early approach to computer based simulation relied on *discrete-time*, *discrete-state* modelling [3, 26]. For modelling real-world systems in which time flows continuously, the *system dynamics* [5] and the *discrete-event* [6] approaches have been used. The use of simulation to resolve software process issues has been facilitated, at least in part, by:

- the wide availability of quantitative simulation languages and tools
- the relative success of simulation in replicating behaviour of real-world software projects [21, 22]
- the potential of simulation for modelling processes involving humans [2]

[1] Software project is used here in its widest sense, to include development, maintenance and enhancement of operational software systems.

[2] More generally, for reflecting processes that for any one of a number of reasons are not readily representable or easily studied in analytic form.

Quantitative approaches, such as discrete-event simulation and system dynamics modelling require that all aspects to be included in the simulation must be reduced to numeric form. In many cases, however, quantification will prove difficult or even infeasible. This may, for example, be due to phenomenological fuzziness, uncertainty, lack of data or to a process that is relatively undisciplined or immature. Circumstances such as these are likely to lead to unreliable models.

Linking model parameters to measuring mechanisms can help ensure model validity or to detect departure from such validity. More generally, establishing explicit links between process observations and the simulation model can help overcome such difficulties. Implementing the links is likely to improve the long term validity, applicability and usefulness of the model [19]. To be generally effective, this requires application of a paradigm which in addition to numerical data accepts other data classes. The ability to processing uncertain *a priori* information is considered a further powerful facility. This leads to the question "what type of simulation paradigm may be expected to be effective, given the nature of software projects, and data on such projects?". After references to work relevant to this question, the application of *fuzzy dynamics* [7] to model, simulate and analyse the behaviour of software projects is considered. This approach is seen as complementary to and supportive of established quantitative simulation paradigms.

2 Simulation and Fuzziness

Simulation models which combine *deterministic* and *stochastic* elements have long been used [26], even in the software domain. The approach may be based on events triggered by randomly generated inputs with known distributions or known frequency of occurrence. However in many cases, such as when the number of real world observations is small or noisy due to the actions of individual humans, meaningful distributions are difficult to estimate. Moreover, uncertainty, however it arises, may not be adequately reflected in the probability distributions [23]. It may, for example, be non-stochastic [7].

Knowledge based techniques termed *semi-quantitative* and *qualitative*, for modelling of the dynamics of a system under uncertainty, have appeared in the literature [20, 23]. This work has focussed, in the main, on the diagnosis of physical systems. Reference has also been made to their application to business and other non-physical applications [20]. It has, however, not reached the state of practical exploitation, having been hindered, *inter alia*, by the inherent difficulty of representing time-varying phenomena, multiple interacting feedback loops, probabilistic interactions [23] and/or activity involving humans [1]. The reader will notice that all these properties are very relevant to the software process [10].

Levary proposed to deal with uncertainty by the incorporation of fuzzy logic [25] components within system dynamics models [14]. In particular, he suggested exploring the impact of fuzzy variables such as *work pressure* in the process model of a software project. Plans to combine a system dynamics simulation and fuzzy logic elements with explicit linkage of metric data have been reported by Fatehi [4].

A further approach to dealing with uncertainty and fuzziness in simulation models arises from the linking of quantitative simulation with knowledge based techniques. Widman and Loparo describe various ways and different levels by which this can be achieved [23]. In the software domain, Levary and Lin added a front-end fuzzy logic based expert systems to support, respectively, the consistency checking of inputs and the analysis of system dynamics simulation outputs [15]. More recently, Madachy has implemented a COCOMO extension in the form of a front-end expert system linked to system dynamics simulations for risk assessment and cost estimation [16].

3 Fuzzy Dynamics

The fuzzy dynamics paradigm, based on fuzzy logic and mentioned above as a complement to quantitative approaches, have recently been described, along with its antecedents, by Friedman and Sandler [7, 8]. These authors discuss simulation of the behaviour of "...systems with limited and incomplete information on the real state of the system and only vague ideas on the rules how these states are changed during time..."[7]. They propose that the technique discussed be used in economic, social and biological systems and have, indeed, applied it with encouraging success in the third of these areas [8]. Their work indicates that some of the problems identified by Levary [14] have now been overcome.

The technique is based on linguistic rules along with an estimation of system initial state. The rules may be obtained from a variety of sources and provide the base for the fuzzy dynamics simulation. The many possible sources of rules include human expertise, human interpretation of observed phenomena, human or computer analysis of empirical data and discovery of patterns and mechanisms in metric and other indicators. More generally, *data mining*, *knowledge discovery* and related techniques may be used [24]. It is hoped to explore application some of these techniques in an investigation now being proposed [12].

4 Long Term Vision

To achieve the hybrid models envisaged above requires the capture of relevant sources of expertise. This, in turn, requires association, encapsulation and appropriate encoding of identified patterns, mechanisms, empirically derived rules and human expert knowledge, and their aggregation into a rule base. This would be available for immediate or later use by process engineers, project managers, model builders and on-line simulation to facilitate the reuse of all available expertise, organisationally, *intra*-organisationally and at project level, for example. This vision represents a goal that may be attainable in the not too distant future and is presented here to encourage others to undertake the appropriate research and development. We ourselves are hoping to be enabled to include aspects of such development, including also the use of multi agent models [17] in the immediate future [12].

In the context of the software process, simulation models at the organisational or global level may be of special interest. Recent results from the FEAST/1 project appear to confirm a hypothesis that organisational dynamics may be a major influence in determining the evolutionary trends of long lived software systems [13]. This observation is confirmed by rules (termed *laws of evolution*) which appear compatible with the evolutionary behaviour of, at least, the systems studied to date [9, 11]. If the laws and related conclusions are upheld they provide a theoretical framework, a *theory of software evolution*, for the simulation support envisaged. This will surely prove of wide relevance to software production and evolution, to software process improvement.

5 Final remark

The implicit message underlying this contribution is that there is an ever increasing need to study, understand and master the software development and evolution process and its management. The approach discussed is seen as providing the means for achieving such understanding and mastery. Development of appropriate paradigms, languages and tools for *behaviour description* may be at least as important as the continued search for improved languages for software process *description* and *enactment* [18]. The role of multi-loop feedback interaction and its implications [10], the implications of fuzziness in data and phenomena and the human role and its description also deserve attention.

Acknowledgements. Many thanks are due to Profs. Y. Fridman and U. Sandler for having drawn our attention to the relevance of fuzzy dynamics to our software evolution studies. Many thanks are also due to Prof. Adam Porter for his comments on an initial version of this paper and to Dr. Paul Wernick for many productive discussions; also to the anonymous referees for their comments. This work has been supported by the EPSRC under grant GR/K86008.

References

1. Aulin A.: Foundations of Mathematical System Dynamics, Pergamon Press, Oxford, UK (1989)
2. Christie A.: Simulation in Support of Software Process Improvement, Proc. Workshop on Software Process Simulation and Modelling, ProSim'98, June 22-24, Silver Falls, OR (1998)
3. Eshed, R., Lehman, M. M. and Netter, Z.: The Checking of Computer Logic Via Simulation on a Computer, The Computer Journal **6** (1963), 154–162
4. Fatehi, K.: Modelling the Impact of Quality Improvement Initiatives, Proc. ICSE'98, Vol. 2, April 19-25 (1998) Kyoto, Japan, 164–165
5. Forrester, J.W.: Industrial Dynamics, The MIT Press, Cambridge, MA (1961)
6. Fishman G.S.: Concepts and Methods in Discrete Event Simulation, Wiley, New York (1963)

7. Friedman Y. and Sandler U.: Evolution of Systems Under Fuzzy Dynamic Laws, Fuzzy Sets and Systems, **84** (1996), 61–74
8. *id.*: Dynamics of Fuzzy Systems, Invited Paper, to appear, Int. J. Chaos Theory and Application (1998)
9. Lehman M.M. and Belady L.A.: Program Evolution - Processes of Software Change, Academic Press, London (1985)
10. Lehman M.M.: Process Modelling - Where Next?, Invited Talk, the Most Influential Paper of ICSE 9 Award, Proc. ICSE'97, May 17-23 (1997) Boston, MA, 549–552
11. Lehman M.M., Perry D.E., Ramil J.F., Turski W. and Wernick P.: Metrics and Laws of Software Evolution - The Nineties View, Proc. Fourth Int. Soft. Metrics Symp., Nov. 5-7, Albuquerque, NM, (1997) 20–32. Also as Process Improvement - The Way Forward, in El Emam, K. and Madhavji, N.H., (eds.), Elements of Software Process Assessment and Improvement, IEEE CS Press (1998)
12. Lehman M.M.: FEAST/2 Project: Case for Support, in preparation, Dept. of Comp., Imp. Col., London (1998)
13. Lehman M.M. and Ramil J.F.: The Impact of Feedback in the Global Software Process, Invited Paper, Proc. Workshop on Software Process Simulation and Modelling, ProSim'98, June 22-24, Silver Falls, OR (1998)
14. Levary R.R.: System Dynamics with Fuzzy Logic, Int. J. Systems Sci., **21** (1990), 1701–1707
15. Levary R.R. and Lin C.Y.: Modelling the Software Development Process using an Expert Simulation System Having Fuzzy Logic, Software - Practice and Experience, **21** (1991), 133–148
16. Madachy R.J.: Knowledge-Based Risk Assessment and Cost Estimation, Automated Software Engineering, **2** (1995), 219–230
17. McCabe F.G. and Clark K.L.: Programming in April: An Agent Process Interaction Language, in Intelligent Agents, Springer (1995)
18. Osterweil L.: Software Processes are Software Too - Revisited, Invited Talk, the Most Influential Paper of ICSE 9 Award, Proc. ICSE'97, May 17 – 23 (1997) Boston, MA, 540 – 548
19. Pfal D.: Integration of System Dynamics Modeling with Descriptive Process and Goal Oriented Measurement, Proc. Workshop on Software Process Simulation and Modelling, ProSim'98, June 22-24, Silver Falls, OR (1998)
20. Travé-Massuyès L. and Milne R.: Application Oriented Qualitative Reasoning, The Knowledge Eng. Rev., **10** (1995) 181 – 204
21. Waeselynk H. and Pfal D.: System Dynamics Applied to the Modelling of Software Projects, Software - Concepts and Tools , **15** (1994) 162 – 176
22. Wernick P. and Lehman M.M.: Software Process White Box Modelling for FEAST/1, Proc. Workshop on Software Process Simulation and Modelling, ProSim'98, June 22-24, Silver Falls, OR (1998)
23. Widman L.E. and Loparo K.A.: Artificial Intelligence, Simulation, and Modeling - A Critical Survey, in Widman L.E., Loparo K.A. and Nielsen N.R. (eds.), Artificial Intelligence, Simulation and Modeling, Wiley, New York, (1989)
24. Wu X., Kotagiri R. and Korb K. (eds.): Research and Development in Knowledge Discovery and Data Mining, Proc. Second Pacific-Asia Conference, PAKDD-98, Melbourne, Australia, April (1998), Lec. Notes. in Artif. Intelligence, Nr. 1394, Springer, Berlin
25. Zadeh L.A.: Fuzzy Sets, Information and Control **8** (1965), 338–353
26. Zeigler B.P.: Theory of Modelling and Simulation, Wiley, New York, (1976)