Mind out of Programmable Matter: Exploring Unified Models of Emergent System Autonomy for Collective Self-Regenerative Systems - *Extended Abstract*

M. Randles, A. Taleb-Bendiab, P. Miseldine

School of Computing and Mathematical Science, Liverpool John Moores University, Byrom St. Liverpool, L3 3AF, UK

{cmsmrand, a.talebbendiab, cmppmise}@livjm.ac.uk

Abstract: *This article advocates the need for a radical rethink of software agent technology by investigating the mechanisms through which knowledge, deliberation, action and control interact to form truly intelligent autonomous agents, be they deliberative, intentional or purely reactive automata/particles/actors. Using sound logical formal modelling techniques, this work attempts to propose a unified model of multi-agency, which integrates and consolidates various proposed software agent models including; deliberative, cognitive, collectivist and individualist agent perspectives. In particular, the paper focuses on the formal semantics of model-based and emergent regulatory structure of autonomic self-regenerative systems, agents or particles (swarm intelligence). In addition, the paper uses our new scripting language – Neptune and associated development environment, which transforms formal requirements models of a given agency to executable code.*

1. Introduction: The Current Situation

Over the last 3-4 decades, many noteworthy advances have been made extending our understanding and application of Distributed Artificial Intelligence (DAI) to develop models of reactive, proactive, deliberative and/or emotional agents. Many AI systems have been produced that perform well within the domain they have been designed for, "yet they could not learn concepts that most children know by the time they are 3 years old."[1]. It is the aim of AI research to make computer systems perform "intelligent" tasks to make life easier for humans. This may be in the realm of pervasive computing [2], hidden in the users' environment, or in autonomic computing [3], seeking to alleviate the burden of system management. In either case it is very useful to conceive of the tasks as being carried out by agents, which are allowed the freedoms to be reactive, proactive and deliberative [4] as a generic feature, rather than in any domain specific way. Unfortunately AI has suffered from grand, unfulfilled promises and unsubstantiated claims. These have mostly arisen from separate isolated areas of research that seldom interact with each other. At the most fundamental level there is a divergence of method between the top-down, centralised management approach of policy-based systems, for instance, and the bottom-up devolved control models, capable of exhibiting self-organisation. At a more involved level various models are used to imbue agents with intelligent-like behaviour. Researchers work on producing Artificial Immune Systems (AIS) [5], Autonomic Systems [3], Sensor Networks [6] and many other features, of a biological entity, which can be seen as possessing intelligent responses to stimuli, whether through reaction, anticipation or deliberation. These can all be realised to some extent using agent technology, in particular autonomous agent methods, but they are developed separately with no overall model of coordination. This is, in part, due to the huge variety of environmental information and potential events that can occur. A pre-defined model cannot possibly capture the requisite system varieties. A collective of self-regenerative and self-adaptive agents can use concepts of diversity, cognitive immunity, scaleable redundancy and threat reasoning to change and adapt to circumstances in response to the variety of external or system stimuli.

2. Artificially Intelligent Agents

To model the complex structure of an intelligent agent a blueprint can be applied to model the system as a single agent with embedded subagents conforming to the same blueprint, in a similar way to humans comprising of many interacting lower level systems [7]. In turn these subagents contain their own embedded subagents giving a fractal like structure to model the system complexity. This has been represented in the Viable Intelligent Agent Architecture (VIA-A) architecture [8], which is based on an extension of the well-established cybernetics representation the Viable System Model (VSM) [9] combined with elements of the IRMA [10] model. The model identifies five subsystems within a running system: The effectors (S^1), coordination (S^2), management (S^3), an audit component (S^{3*}), deliberation on future scenarios (S^4) and system identity (S^5). Each of these systems also embeds its own copy of the S^1-S^5 structure to any level required. Also, alternatively to separately developed, single concept agent software systems, human agents function as a complex, yet coordinating, hybrid of all sorts of systems. There is an autonomic system that self-regulates the body; there are the five senses that provide environmental input and so on. But these do not act in isolation but overlap and coordinate in a complex way to deliver the complete autonomously acting system. In a similar way a computer system based on the VIA-A design ought to be viewed as a holistic entity where the various systems of agents and methods of control interact and no one paradigm is used to enforce a total modelling methodology throughout the blueprint or at any level of the subsystem. The overall model ought to include both a top down normative intentional approach to deliberation and a bottom up distributed control mechanism with methods such as Chance Discovery, Novelty Detection or Danger Theory [11,12] used to detect emergence via suitable partial observation conditions.

3. A Hybrid Agent Design: Top-down versus Bottom-up

Using a blueprint, such as the VIA-A architecture, provides the requisite means of reasoning about complexity in the system. It is proposed to model the various systems as hybrid centralised/decentralised control agents. This is achieved through the use of a centralised deliberative module, acting as an observer, with control abstracted out through environmental semiotics (Stigmergy [13]) and self-organisation through swarming. By studying such models, self-organisation of computer systems can be arranged, where centralised governance is replaced with control distributed across all the participating agents. Such systems are better modelled by a bottom-up approach where the basic agents and their interactions in the system, and the environment, are defined. In this way federated behaviour is not prescribed but rather emerges from the system. So although, as a software entity, the system controller is still extant, acting as a top down observer, the control is actually abstracted by distributed knowledge. The observer views and appreciates emerging patterns of organised behaviour, unknown to the low level collective participants. In this way the system control is divested from a central controller to the active participants within the process. This leaves the central controller with a less complex (easier implemented) monitoring observer's role in the system, yet retaining deliberative capabilities based on its normative state.

4. Formalising Knowledge and Control

In order to provide the formalism that allows reasoning about and verification of the system it is necessary to consider the emergent nature of swarm-like behaviour. The detection of novelty and emergent behaviour is very difficult. What is sought is a theoretical and implementation method for autonomous software agents to sense, reason and act in transitional, only partially observable, non-predictable environments where the local interactions of the participants abstract out the need for centralised control. This leaves only detection to promote safe behaviour and proscribe undesirable emerging properties as a central (observer) role. Thus agents' beliefs and the ensuing actions prompted by these beliefs are paramount. These beliefs are best represented as logical sentences [14] that condition behaviour. This behaviour affects other members of the collective and in such a way it is hoped to provide a formal and computational account of deliberation leading to action in a swarm type system (Swarm Calculus). Thus there is an emphasis on beliefs and how they condition individual behaviour, to influence group behaviour. This, in turn, means that logical consequences of emergent behaviour, whether inferred or detected, can be ascertained, evaluated and utilised within the system. As an example, using dynamic logic in this case, consider a system where the number of agents, in a team, needs to be maintained above a certain number N. A mutual goal, p, for a team (T) can be stated as:

$$G(T,p) \Leftrightarrow B(T,\neg p) \wedge G(T,p) \wedge [UNTIL[B(T,p) \vee B(T,\neg p)] \quad LTG(T,p)$$

where B is the modal belief operator [15] and LTG is a lesser team goal that the team mutually believes that all the members have a team commitment to p

So we have $\models G(T, numberofmembers(T)>=N)$

From this we can infer, as a logical consequence, that team members as individuals have a commitment to maintain the number of colleagues above a specified level so that when one believes that the number of fellow members is less than required and believes that this is not mutually believed by the team and believes it is not impossible to establish mutual belief for the team then it has an individual commitment to bring about this mutual belief.

i.e. $\models G(T,p) \Rightarrow \forall t \in T\ [B(t,\neg p \wedge \neg C(T,\neg p)) \wedge \neg B(t, \square\neg C(T,\neg p)) \Rightarrow G(t,C(T,\neg p) \wedge B(t,\neg p))]$

where p is $, numberofmembers(T)>=N$ and C is the common belief operator [15] which is proven by assuming:

$$\forall t \in T\ [B(t,\neg p \wedge \neg C(T,\neg p)) \wedge \neg B(t, \square\neg C(T,\neg p))$$

Then from the definition of team goal:

$B(t, \neg p) \wedge \neg C(T,\neg p) \Rightarrow G(t, C(T,\neg p))$

and $G(t, C(T,\neg p))$ is satisfied because if one member of the team does not believe there is mutual belief then there is no mutual belief. So since the consequent of an implication must remain true until the antecedent or the implication statement becomes false

$G(t, C(T,\neg p))$ holds until

$t \in T\ B(t,\neg p \wedge \neg C(T,\neg p)) \wedge \neg B(t, \square\neg C(T,\neg p))$ doesn't;

that is until $\neg B(t,\neg p) \vee C(T, \neg P) \vee B(t, \square\neg C(T,\neg p))$ is true.

So all the conjuncts in the definition of $G(t,C(T,\neg p) \wedge B(t,\neg p))$ are satisfied proving the result. Similar results follow as a matter of logical consequence. Knowledge producing actions, whether sensory, inferential or discovered, lead to consequences for the system that can be established via this formal representation.

5. The Story So Far – Calculus to Code

5.1 Formalisation

As can be seen from the brief example above; logic based formalism works well in using system knowledge to extract necessarily true system properties as logical consequences of the specification. The Situation Calculus [16] further provides knowledge capturing and deliberation techniques to more than adequately reason about complex, dynamical agent systems. It is a first order predicate calculus with some second order features giving excellent expressiveness with good reasoning methods. In this way the emergent logical consequences of the specification can be considered and verified whilst emergent behaviour arising from the inevitable incompleteness of the specification can be handled via a dynamic observer monitoring for novel system aspects. The further introduction of stochastic actions to the Situation Calculus representation allows extraneous and natural events to be included. The Stochastic Situation Calculus therefore provides a mapping from knowledge to stochastic actions, including environmental actions, based both on sensing mechanisms and logical reasoning. As a representational tool it does not require an explicit enumeration of the state space, thus permitting unknown situations. In this way it can be used as a Swarm Calculus, for a bottom-up approach, and as a behavioural deliberative model in norm based systems, for a top-down, model-driven approaches. Its use here is to provide a unifying design model, within the recursively embedded VIA-A structure, to represent composite swarm-based/deliberative agents and agent systems that have the capacity to regenerate themselves. This encourages fault tolerance and redundancy through diversity, cognitive immunity by adapting and generating agents, to learn from past failures, and deliberation for adapting to threats.

5.2 Implementation

A new scripting language, based on logical statements, has been developed that is very useful in taking the logical sentences and transforming them into run time adaptable and addressable objects. This language, Neptune [17] thus provides the means to realise, in practice, the desirable system properties that are evident in the logical specification. Thus agent systems exhibiting human-like behaviour can be constructed from autonomous subagents acting in teams to provide distributed control following as logical consequences from the specifications. Additionally chance discovery can be performed by a "stripped down" system controller assessing emergence and providing a norm based deliberative agent connotation.

6. Conclusions

Work has already been completed on an agent-norm based approach to system self-governance. This has been implemented as a medical decision support system exhibiting deliberation on knowledge for both application and system level control. It is implemented in a grid computing environment using autonomous agents in a top-down system controlled norm-based application. The work on Swarm Calculus is more recent. The programming model is seeking to make use of a calculus to code design so that the logically developed methods of modelling swarming and emergent behaviour can have a verifiably correct implementation into code. This forms part of the ongoing work, on many fronts, to develop a unifying model of an intelligent, generic agent. At a nanotech level such work has implications for the control of small components (agents) to be used for the construction of larger agents (e.g rendering a claytronic entity). It is hoped, by this holistic approach to bridge the gap between the learning ability of the human mind and that of software agents by modelling a software system as an agent consisting of a complex arrangement of adaptable agent subsystems utilising different organisational and deliberative schemes within a unified representational structure.

References

1. M. Minsky, "Building Intelligent Machines", Speech at Boston University, May 16, 2003, http://wiredvig.wired.com/news/technology/0,1282,58714,00.html
2. A. Schmidt, K. van Laerhoven, "How to Build Smart Appliances" *IEEE Personal Communications,* Vol.8(4): 66-71, 2001
3. P. Horn, "Autonomic Computing: IBM Perspective on the State of Information Technology", *IBM T.J. Watson Labs,* New York, 2001, http://www.research.ibm.com/autonomic
4. N. Jennings, M. Wooldridge, "Agent-Oriented Software Engineering" *Handbook of Agent Technology,* ed. J. Bradshaw, AAAI/MIT Press, Cambridge MA, 2000
5. M. Burgess 'Computer Immunology' *In Proceeding of 12th Systems Administration Conference* (LISA'98), 283-298
6. A. Lim "Distributed Services for Information Dissemination in Self-Organizing Sensor Networks," ***Special Issue on*** *Distributed Sensor Networks for Real-Time Systems with Adaptive Reconfiguration, Journal of Franklin Institute,* Elsevier Science Publisher, Vol. 338, 2001, pp. 707-727
7. M. Randles, A. Taleb-Bendiab, P. Miseldine, A. Laws "Adjustable Deliberation for Self-Managing Systems", In proc. *12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based* Systems (ECBS 2005): 449-456, Washington MD, April 2005
8. A.G. Laws, A. Taleb-Bendiab, S. J. Wade, D. Reilly (2001) 'From Wetware to Software: A Cybernetic Perspective of Self-adaptive Software'. IWSAS 2001: 257-280
9. S. Beer, "The Viable System Model: its provenance, development, methodology and pathology", in R. Espejo and R. Harnden, *The Viable System Model: Interpretations and Applications of Stafford Beer's VSM,* John Wiley and Sons Ltd., Great Britain, 1989
10. M.E. Bratman, , D.J. Israel, and M.E. Pollack, *Plans and Resource-Bounded Practical Reasoning,* Computational Intelligence, Vol. **4**, No. 4, pp. 349-355, 1988.
11. Y. Ohsawa (editor), "Proceeding of the 1st International Workshop on Chance Discovery", *Japanese Society for Artificial Intelligence,* 2001
12. M. Randles, A. Taleb-Bendiab, P. Miseldine," Using Stochastic Situation Calculus to Formalise Danger Signals for Autonomic Computing" Proceedings of PGNet 2005, Liverpool John Moores University, 2005
13. Hadeli, P. Valckenaers, M. Kollingbaum, H. van Brussel, "Multi-agenr Coordination and Control using Stigmergy" *Computers in Industry* 53(1): 75-96, 2004
14. B.C. Smith "Reflection and Semantics in a Procedural Language" *PhD. Thesis,* MIT, Cambridge Mass., 1982
15. J.Y. Halpern, Y. Moses (1990) "*Knowledge and Common Knowledge in a Distributed Environment*" Journal of the ACM, 37(3) pp: 549-587
16. H. J. Levesque, F. Pirri and R. Reiter (1998) 'Foundations for the situation calculus'. Linköping Electronic Articles in Computer and Information Science, Vol. 3(1998): nr 18. http://www.ep.liu.se/ea/cis/1998/018/
17. P.Miseldine, A. Taleb-Bendiab "Rainbow: An Approach to Facilitate Restorative Functionality within Distributed Autonomic Systems" Proceedings of PGNet 2005, Liverpool John Moores University, 2005