УДК 681.3

MULTI-AGENT TECHNOLOGY FOR DESIGNING ADAPTIVE BUSINESS PROCESSES

George Rzevski
Complexity Science and Design Group
The Open University
Milton Keynes, UK
E-mail: george@rzevski.net

Ключевые слова: Multi-Agent Technology, Adaptive Business Processes, Complexity.

Аннотация: Complexity of the Internet based global market is increasing relentlessly. To survive and prosper under new conditions business processes need to be adaptable, which means capable of positively reacting to frequent disruptions. A methodology and technology for designing adaptable business processes is outlined and examples are given of solutions supplied to commercial enterprises.

1. Ease of use

As complexity of our economic environment increases the need for Adaptability of business processes becomes more and more obvious. Adaptability is a new Critical Success Factor that is far more important under conditions of complexity than the Economy of Scale, which, in many cases, is now harmful, being responsible for rigidity of a business and its inability to react positively to frequent disruptive events now prevalent in the Internet based Global Market.

2. The concept of complexity

A system is complex if it consists of a large variety of autonomous components engaged in interaction. The global behaviour of such a system is unpredictable and yet not random – it emerges from the interaction of local behaviours of constituent components. Complex systems are nonlinear - a small disturbance may cause large changes in their global behaviour (butterfly effect) whilst large disturbances may be unnoticeable.

The following three paragraphs from Wikipedia are a good introduction to the concept of complexity.

“Complexity has always been a part of our environment, and therefore many scientific fields have dealt with complex systems and phenomena. Indeed, some would say that only what is somehow complex – what displays variation without being random – is worthy of interest.

The use of the term complex is often confused with the term complicated. To understand the differences, it is best to examine the roots of the two words. “Complicated” uses the Latin ending “plic” that means, “to fold” while “complex” uses the “plex” that means, “to weave.” Thus, a complicated structure is one that is folded with hidden facets and stuffed into a smaller space. On the other hand, a complex structure uses interwoven components that introduce mutual dependencies and produce more than a sum of the parts… This means that complex is the opposite of independent, while complicated is the opposite of simple.

While this has led some fields to come up with specific definitions of complexity, there is a more recent movement to regroup observations from different fields to study complexity in itself, whether it appears in anthills, human brains, or stock markets."

Table 1. Complex versus Complicated.

COMPLEX SYSTEMS	**COMPLICATED SYSTEMS**
Free market	Centrally planned economy
Flexible manufacturing system	Mass production line
Management team	Management hierarchy
Real-time schedule	Batch schedule
Multi-agent system	Large conventional computer program
Human being	Jumbo Jet
Society	Traditional army

The key consequence of complexity is Uncertainty. If we take uncertainty as a demarcation factor we can classify systems in three categories: Deterministic, Complex and Random, as shown in the table below.

Table 2. Complex versus Deterministic versus Random.

RANDOM SYSTEMS	**COMPLEX SYSTEMS**	**DETERMINISTIC SYSTEMS**
Uncertainty = 1	1>Uncertainty>0	Uncertainty = 0
Full autonomy	Partial autonomy	No autonomy
Disorganised	Self-organising	Organised

The often-quoted statement made by Stephen Hawking at the end of the 20th century best illustrates the importance of the increasing complexity of our environment:

"I think the next century will be the century of complexity".

3. The science of complexity

The Nobel Laureate in chemistry, 1977, Ilya Prigogine is generally recognized as the father of Complexity Science. In his books "Is Future Given?" [1], and "The End of Certainty: Time, Chaos and the new Laws of Nature" [2], he contrasts his discoveries with notions of classical science as follows.

Classical science, as represented by Newtonian dynamics, is deterministic. The world is, it always was, and always will be, the same; just as argued by the classical Greek philosopher Parmenides. All natural laws are permanent and reversible and independent of time and space. The key scientific concepts are equilibrium, stability and predictability.

Prigogine argued that this view of the world in which we live is rather restrictive. It is valid only for a narrow domain of the world, such as Mechanics. Newtonian science does not concern itself with major domains of interest - what we need is the science that can explain behaviour of systems consisting of physical, chemical, biological and socio-technical systems. A broader perspective is required to incorporate into the science readily observable phenomena of change, evolution and unpredictability of events that may or may not occur.

According to Prigogine, Future is not given; it is constructed in front of our eyes by billions of actions and decisions made by natural or artificial Agents operating in the Universe. The Universe is perpetually changing and constituent systems co-evolve affecting each other.

Fundamental concepts of Complexity Science, as formulated by Rzevski [3], [4], [5], are described below.

1) CONNECTIVITY - A system consists of a large number of diverse components, referred to as Agents, which are richly interconnected.
2) AUTONOMY - Agents are not centrally controlled; they have a degree of autonomy but their behaviour is always subject to certain laws, rules or norms. Agent behaviour is never random.
3) EMERGENCE - Global behaviour of a complex system emerges from the interaction of agents and is therefore unpredictable but not random; it generally follows discernible patterns.
4) NONEQUILIBRIUM - Global behaviour of a complex system is far from equilibrium because frequent occurrences of disruptive events do not allow the system to return to the equilibrium between two disruptive events.
5) NONLINEARITY - Relations between agents are nonlinear, which occasionally causes an insignificant input to be amplified into an extreme event (butterfly effect).
6) SELF-ORGANISATION - A system is capable of self-organizing in response to disruptive events, a feature termed Adaptability. Self-organisation may also be initiated autonomously by the system in response to a perceived need, a feature termed Creativity.
7) CO-EVOLUTION - A system irreversibly co-evolves with its environment.

4. Managing complexity

If we accept that “to control” means to specify a desirable behaviour of a system and to steer the system towards achieving it, then complex systems cannot be controlled. The very concept of “emergent behaviour” precludes controllability.

Fortunately we are not helpless in the presence of complexity, the complexity science gives us tools for analysing complexity issues, for planning how to react to unpredictable disruptive events, for designing adaptability into social, business or technological processes at hand, and for “tuning” complexity using experimentally derived heuristics.

Let us call these activities “Managing Complexity” and let us focus first on “coping’ with complexity, which is defined here as a means of achieving desirable results under conditions of complexity that is not under our control (in other words, external complexity). The ability to cope with external complexity is extremely important, for example, for businesses that sell to global markets.

The best strategy for coping with complexity is to develop a capacity for self-organization that will neutralize or reduce consequences of disruptive events when they occur.

Processes with capacity for self-organization are Adaptive, that is, capable of achieving their goals when operating in complex environments.

For self-organization to be possible we must have the following elements in place:

1) A range of optional actions that may be necessary to undertake when a disruptive event occurs.
2) Decision-making technology capable of autonomously and rapidly choosing which action to undertake when a disruptive event occurs, and implementing the selected action before the next event.
3) Strategic redundancy of resources to support planned options.

Dynamic forecasting of the occurrence of disruptive events, based on learning and instantaneous updating of forecasts with what actually happened, may help.

Building the capacity for self-organization into systems in which we live and work amounts to designing complexity into our life, which is counterintuitive. Common sense suggests we should attempt to simplify the complexity of the environment, which is not practical because by definition our environment is not under our control.

Complexity of our social, economic, political or technological environments is, by definition, External Complexity. And so is complexity of natural systems governed by the laws of nature. We cannot simplify external complexity or change it in any way. But we can learn how to cope with external complexity.

The best way of coping and, indeed, of surviving and prospering in a complex environment is to become adaptive.

To be adaptive means to be able to achieve desired goals under conditions of complexity. Adaptability can be achieved by anticipating and positively reacting to an unpredictable disruptive event before the next one occurs.

Let us consider an example – a supply chain. An unpredictable disruptive event in a supply chain is a failure of a delivery truck. A real-time supply chain scheduler that can rapidly reschedule the supply chain in order to organize repair or replacement of the faulty truck before the next disruption occurs (that is, "in real time"), without disturbing unaffected parts of the schedule, makes supply chain adaptive.

An important part of adaptability is resilience. To be resilient means to be able to achieve desired goals under attack (hacking, fraud, spam, virus). Resilience can be achieved by real-time monitoring critical data and rapidly detecting a change in data that looks like a malevolent behaviour. Resilience is increased by real-time risk management.

In practice, Adaptability and Resilience can be achieved only by employing real-time schedulers and dynamic data mining systems, respectively, incorporating multi-agent technology.

Building the capacity for adaptability and/or resilience into systems in which we live and work amounts to designing complexity into our life, which is counterintuitive. Common sense suggests we should attempt to simplify the complexity of the environment, which is of course not possible, because, by definition, our environment is not under our control.

5. Designing for adaptability

Let us consider adaptability requirements. To be adaptive we have to:

1) Continuously monitor changes in our environment in order to spot the occurrence of a relevant disruptive event as early as possible.
2) Decide how to react to an event before the next one occurs.
3) Make changes only to those parts of the system that are affected by the event.
4) Delay the implementation of the decision as long as practical in order to improve the decision and make it as near as possible to the optimal.

These requirements can be met only by employing a very advanced technology. The scope of the problem and required speed of decision-making is beyond human capabilities.

To continuously monitor changes in our environment we need technology capable of (1) monitoring changes of diverse data sources and (2) dynamic data mining.

To rapidly react to disruptive events we need to have decision-making elements capable of comparing a large number of options and selecting the best option under current conditions in a very short time (typically in few seconds).

6. Complex adaptive software

Self-organization in business processes is feasible only if decisions how to respond to disruptive events are made and implemented rapidly – the decision what to do as well as the action, which implements the decision must be completed in between two consecutive disruptive events.

Here are some examples. In production of cars the frequency of disruptive events is one in two hours whilst in a large-scale metropolitan taxi operation (say, 2,000 vehicles) – one every few seconds.

It is obvious that for scheduling decisions to be done with such a speed we cannot rely on humans. We need complex adaptive software. Conventional software is not of much help because it requires a re-start from scratch whenever a disruptive event occurs and requires 8 to 10 hours to re-schedule a typical factory.

To exhibit adaptability software must have an extensive Knowledge Base and built-in emergent intelligence. At present, only the Multi-Agent Technology is capable of providing complex adaptive behaviour.

Let us outline a methodology for developing adaptive software [6], [7], [8], [9].

1) Collecting and organising domain knowledge.

The first step is to collect and organise knowledge on the domain of the real world that is being investigated. The most effective method of representing knowledge on a complex domain is to construct a network in which nodes are domain Concepts and links are Relations between Concepts. For example, for an airline relevant Concepts include: Flight, Passenger, Aircraft, Pilot, Maintenance, Seat Price, Route and Network. Each Concept is characterised by attributes and rules constraining its behaviour. Such domain knowledge representation is called Ontology.

2) Constructing a Virtual World.

The next step is to build a Virtual World consisting of instances of Concepts and their Relations from domain Ontology. For an airline, a Virtual World will be a network in which nodes are Passenger P1, Passenger P2, … Flight F1, Flight F2, … Seat S1, Seat S2, … Aircraft A1, Aircraft A2, … etc., and links are “S1 is allocated to P3”, “A1 is allocated to F2”, etc. Complex systems, such as supply chains of large international organisations, and Virtual Worlds that represent them, may contain millions of objects, attributes, rules and relations. To construct Virtual Worlds for such complex problems one requires powerful multi-agent software tools.

3) Connecting Virtual World to the Real World.

The Real World (i.e., a complex situation that is being modelled) is perpetually changing. The changes are represented by the occurrence of events exemplified, in an airline, by: Seat booking, Flight departure, Flight delay, Flight Cancellation, Airspace Closure, Aircraft Failure, etc. The occurrence of every Real Event must be communicated instantly to the Virtual World where an equivalent Virtual Event is created, causing the affected part of the Virtual World to adapt to changes originated in the Real World. Every change (adaptation) of the Virtual World must be communicated to the Real World before the occurrence of the next Event.

4) Empowering Virtual World to manage the Real World in real time.

An Agent (a computational object) is assigned to every node of the Virtual World with responsibility to maintain its integrity. For example, if a Virtual Aircraft breaks down, the Aircraft Agent sends messages to Agents of all affected nodes letting them know that this Virtual Aircraft does not exists for a time being. The message provokes a flurry of activities among affected Agents who try to accommodate the failure by searching for a replacement. As soon as a solution is found, it is conveyed to the Real World for implementation, ensuring that the two worlds co-evolve (change in unison).

5) Investigating behaviours of the Real World by experimenting in the Virtual World.

Once a suitable Virtual World is constructed in software, it could be used to simulate behaviours of the Real World under different states of its environment, e.g., studying behaviour of a supply chain under varying market conditions.

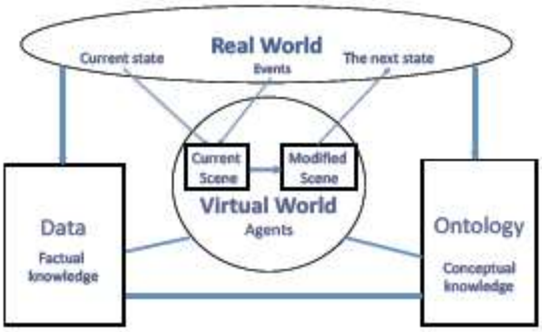

Fig. 1. A Virtual World manages the Real World.

7. Examples of applications

During the twelve-year period, 1999 – 2011, the author with his team have developed a very large number of complex adaptive agent-based software products, which are in commercial use. All these systems have one feature in common – they have succeeded in solving problems, which were considered too complex for generally available conventional methods and tools.

Examples of successfully developed and implemented complexity management systems, include:

- Managing in real time a fleet of 2,000 taxis, for a transportation company in London [10].
- Managing in real time a large fleet of car rentals, for one of the largest car rental operators in Europe [11].
- Managing in real time 10% of the world capacity of crude oil sea-going tankers, for a tanker management company in London [12].
- Real-time scheduling of a large fleet of trucks transporting parcels across the UK [13], [14].
- Selecting relevant abstracts for a research team using agent-based semantic search, for a genome-mapping laboratory in the USA [15].
- Creating contract templates for car insurance using multi-agent based text understanding and dynamic clustering, for a logistics company in the UK [16].
- Dynamic pattern discovery for a logistics company in the UK [17].
- Managing social benefits for citizens supplied with electronic id cards, for a large region in Russia
- Agent-based simulator for modelling the airport and in-flight, RFID-based, catering supply chain, luggage handling processes, and passenger processing, for a research consortium in Germany.
- Resolving clashes in aircraft wing design for the largest commercial airliner in Europe.

- Real-time scheduling of cargo delivery to the International Space Station.
- Real-time scheduling of a supply chain for LEGO [18].

Majority of the above applications have in fact converted a rigid business process into an adaptive one.

8. Comparing multi-agent systems with conventional software

Conventional programs allocate resources to demands following pre-programmed algorithms in a sequential manner and therefore, when dealing with a large number of resources and demands, they require a long time to find the optimal allocation. Whenever resources or demands change, these programs start the allocation process from the beginning and if changes are frequent, they "oscillate" and cannot reach the optimal solution. Centralized intelligent systems are more flexible since they are normally driven by heuristics (rules derived from experience). Nevertheless they still solve the allocation problem sequentially and therefore cannot handle frequent changes effectively.

In contrast, high granularity multi-agent systems execute the allocation of resources in parallel (quasi-parallel, in sequential machines). Typically, hundred of thousands of agents located on a single server or workstation work concurrently, and if the problem is distributed over many servers and workstations, the number of concurrent allocation processes can rise considerably. This explains how multi-agent systems can rapidly arrive at a near-optimal allocation of resources in real time. In cases where changes are infrequent and therefore the optimal allocation is possible, agents will systematically reconsider each concluded resource-demand matching with a view to reducing the overall cost function. This is a time consuming process, which agents will carry out in addition to any re-allocations due to changes in market conditions. Agents work solving client's problems 24 hours a day and will continue re-negotiating partially matched deals until the best possible match is achieved or time runs out.

Agents do not have to wait for instructions. They plan and execute tasks autonomously and are capable of deciding when to compete and when to co-operate with each other. They react to any change in demand or supply without being prompted. Agents representing resources will pro-actively try to place them by searching for potential customers, offering discounts, cross-selling, making special offers and/or co-operating with other agents. Agents representing customers will actively search for resources that match their requirements and will ring their clients or send them emails when they obtain satisfactory allocations.

9. Conclusions

The methodology for managing complexity described in this paper has been developed experimentally, by step-wise improvements. The team has learned from experience of developing and commissioning practical, commercial software products.

References

1. Prigogine, I. (2003). "Is Future Given?" ISBN 9789812385086.
2. Prigogine, I. (1997). "The End of Certainty: Time, Chaos and the new Laws of Nature", ISBN 9780684837055.
3. Rzevski, G., "A practical Methodology for Managing Complexity". Emergence: Complexity & Organization – An International Transdisciplinary Journal of Complex Social Systems. Volume 13, Nos. 1-2, 2011, ISSN 1521-3250, pp. 38-56.
4. Rzevski, G., "Using Tools of Complexity Science to Diagnose the Current Financial Crisis". ISSN 8756-6990, Optoelectronics, Instrumentation and Data Processing, 2010, Vol.46, No. 2.

5. Rzevski, G., “Using Complexity Science Framework and Multi-Agent Technology in Design”. In Alexiou, K., Johnson, J., Zamenopoulos, T. (eds.), Embracing Complexity in Design, Routledge, 2010, ISBN 978-0-415-49700-8, pp. 61-72.
6. Rzevski, G., “Modelling Large Complex Systems Using Multi-Agent Technology”. Kyoto Conf. August 2012.
7. Rzevski, G. “A New Direction of Research into Artificial Intelligence”, Invited Paper, The Annual Conference of Sri Lankan Association for Artificial Intelligence, Colombo, 2008.
8. Rzevski, G., Skobelev, P., “Emergent Intelligence in Large Scale Multi-Agent Systems”. International Journal of Education and Information Technology, Issue 2, Volume 1, 2007, pp. 64-71.
9. Rzevski, G., Skobelev, P. Andreev, V. “MagentaToolkit: A Set of Multi-Agent Tools for Developing Adaptive Real-Time Applications”. Lecture Notes in Computer Science, Volume 4659, Holonic and Multi-Agent Systems for Manufacturing. Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2007, Regensburg, Germany, Springer, ISBN 978-3-540-74478-8, pp. 303-314.
10. Glaschenko, A., Ivaschenko, A., Rzevski, G., Skobelev, P. “Multi-Agent Real Time Scheduling System for Taxi Companies”. Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), Decker, Sichman, Sierra, and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary. ISBN: 978-0-9817381-6-1, pp. 29-35.
11. Andreev, S., Rzevski, G., Shveykin, P., Skobelev, P., Yankov, I. “Multi-Agent Scheduler for Rent-A-Car Companies”. ”. Lecture Notes in Computer Science, Volume 5696, Holonic and Multi-Agent Systems for Manufacturing: Forth International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2009, Linz, Austria. Springer, ISBN 978-3-540-74478-8, pp. 305-314.
12. Rzevski, G., Himoff, J., Skobelev, P., “Magenta Technology: A Family of Multi-Agent Intelligent Schedulers”. Proceedings of the Workshop on Software Agents in Information Systems and Industrial Applications (SAISIA). Fraunhofer Verlag. February 2006, pp. 55-67.
13. Himoff, J., Rzevski, G., Skobelev, P., “Magenta Technology: Multi-Agent Logistics i-Scheduler for Road Transportation”. International Conference on Autonomous Agents: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems, Hakodate, Japan, 2006. ISBN 1-59593-303-4, pp. 1541-1521.
14. Andreev, M., Rzevski, G., Skobelev, P., Shveykin, P., Tsarev, A., Tugashev, A. “Adaptive Planning for Supply Chain Networks”. Lecture Notes in Computer Science, Volume 4659, Holonic and Multi-Agent Systems for Manufacturing. Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2007, Regensburg, Germany. Springer, ISBN 978-3-540-74478-8, pp. 215-225.
15. Rzevski, G., Skobelev, P., “Agent Based Semantic Web”, Semantic Universe Website, 2009.
16. Minakov, I., Rzevski, G., Skobelev, P., Volman, S., “Creating Contract Templates for Car Insurance Using Multi-Agent Based Text Understanding and Clustering”. Lecture Notes in Computer Science, Volume 4659, Holonic and Multi-Agent Systems for Manufacturing. Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2007, Regensburg, Germany, September 2007, Springer, ISBN 978-3-540-74478-8, pp. 361-371.
17. Rzevski G., Skobelev, P., Minakov, I., Volman, S., “Dynamic Pattern Discovery using Multi-Agent Technology”. Proceedings of the 6th WSEAS International Conference on Telecommunications and Informatics (TELE_INFO ’07), Dallas, Texas, USA, March 22-24, 2007, ISBN: 978-960-8457-60-7, pp. 75-81.
18. Madsen, B., Rzevski, G., Skobelev, P., Tsarev, A. “Real-time Multi-Agent Forecasting & Replenishment Solution for LEGO Branded Retail Outlets”. Kyoto Conf. August 2012.

Published: George Rzevski. Multi-Agent Technology for Designing Adaptive Business Processes // XII Всероссийское совещание по проблемам управления ВСПУ-2014, Москва,16 – 19 июня 2014 г.: Труды. [Электронный ресурс] М.: Институт проблем управления им. В.А. Трапезникова РАН, 2014. – С. 9022-9029.