COMPLEX ADAPTIVE LOGISTICS FOR THE INTERNATIONAL SPACE STATION

G. RZEVSKI[1], V. SOLOVIEV[2], P. SKOBELEV[3] & O. LAKHIN[3]
[1]The Open University and Multi-Agent Technology Ltd, UK.
[2]Moscow State University and S. P. Korolev Rocket and Space Corporation Energia, Russia.
[3]Samara Aerospace University and Smart Solutions Ltd, Russia.

ABSTRACT
Whilst launching of astronauts is always in the news, little is known about logistics support that must be in place for the successful spaceflight to be possible. This paper describes some details of the Intelligent Logistics Management System, conceived by the authors, and developed by two sister companies, one in the UK and the other in Russia, which supports Russian contribution to the international space exploration. The System is designed as a complex adaptive network of interacting real-time schedulers, believed to be the first of its kind in the world. At present, five real-time schedulers cooperate or compete with each other, depending on the context. They schedule flights, cargo flow, storage allocation, scientific experiments, and resource allocation within the international space station (ISS). Further, schedulers can be developed and easily connected to the network, as the need arises. When two cargo vehicles were lost in 2015, the Logistics Management System rapidly re-scheduled deliveries, ensuring that astronauts were not left short of food, water, healthcare material, and laboratory equipment for space exploration. The System is based on multi-agent technology and exhibits Emergent Intelligence.
Keywords: complex adaptive logistics, complexity, international space station, space exploration.

1 INTRODUCTION

Russian Soyuz rockets are currently used for all manned flights to the International Space Station (ISS), as well as for cargo deliveries and waste disposals from the Station [1].

The logistic support for ISS, provided by the Russian corporation Energia, is based on advanced multi-agent systems developed by two sister companies, Multi-Agent Technology Ltd, London and Smart Solutions Ltd, Samara, Russia. The lead author initially created multi-agent technology on which ISS logistics is based.

2 THE PROBLEM

Logistics for the ISS is a complex business process consisting of many interrelated processes and tasks, including:

- Scheduling spaceflights (with distinct operations of launch, docking and un-docking)
- Scheduling delivery of fuel, spare parts and extravehicular activity (EVA) gear
- Supporting ISS crew life by scheduling the deliveries of food, water and healthcare items and collection of waste
- Supporting ISS scientific experiments by scheduling the deliveries of laboratory equipment, materials and instruments and the returning of experimental results to Earth
- Allocating delivered supplies to storage zones within ISS

WIT CONFERENCES
This paper is part of the Proceedings of The New Forest Conference on Complex Systems 2016 (Complex Systems 2016)
www.witconferences.com

ISSN: 1755-7437 (paper format), ISSN: 1755-7445 (online), http://www.witpress.com/journals
DOI: 10.2495/DNE-V11-N3-459-472

- Ordering of required resources from suppliers
- Managing projects on the design and development of resources that will be produced in-house
- Scheduling production of in-house developed resources

All activities are carried out under conditions of severe space, weight and time constraints, and frequent occurrences of disruptive events. Even a small change in a schedule leads to a wave of consequences in other schedules, which in turn create disruptions in all linked systems. A large number of different stakeholders that need to be involved in decision-making contribute to the complexity of the problem.

The key logistics management goal is to achieve the best possible allocation of resources to demands, in time and space, under ever-increasing conditions of uncertainty.

The allocation was, until recently, done manually by a large number of scientists, engineers and managers, who had to make thousands of iterations and interactions to find a compromise solution using simple tools such as spreadsheets.

2.1 Logistic support tasks

Logistic support for ISS can be partitioned into the following tasks:

1. **Flight Program Design**, which produces a schedule of dockings of spacecraft to ISS modules under a variety of constraints, i.e., minimal period of time between operations of docking and undocking; permanent presence of at least one piloted spacecraft docked to the station and different preferences for docking between different spacecraft types.
2. **Strategic Planning and Operational Scheduling**, based on the approved Flight Program, consists of the following:
    - Cargo flow – deliveries of units, blocks and systems for cargo flights and piloted (manned) spacecraft
    - Fuel deliveries, based on the forecast of ISS orbital corrections and the fuel capacity of spacecraft types
    - Water, food and other human life support items deliveries, based on the length and composition of each expedition
    - Deliveries of cargo from ISS back to Earth
    - Flight crew work hours
    - Scientific experiments
    - Purchases from the external suppliers
    - The internal production facilities
3. **Project Management** – for projects concerned with the in-house design and development of resources.

2.4 Logistic stages and scheduling horizons

Designing the Flight Program and scheduling deliveries and resources for ISS consist of several stages with different scheduling horizons.

First, a strategic cargo flow model is used to calculate the number of launches required each year based on the total mass of projected cargo. Typically, the payload is divided between four Progress cargo vehicles each year in addition to four Soyuz rockets, which, in addition to astronauts, also carry a small amount of cargo.

The next stage is the interactive design of the Flight Program. The task is to achieve an agreement between involved parties on the number and date/time of spacecraft dockings and un-dockings to ISS modules, while considering possible launch timeframes, the solar activity, configuration, and expected position of ISS, etc. Several versions of the Flight Program are created and examined at this stage before settling on the final plan.

After the Flight Program is approved, begins the scheduling of cargo flow and deliveries of fuel and water. Cargo shipments are distributed between cargo flights and manned flights to ensure an emergency supply reserve in case of a missed delivery.

The number of astronauts on board of each flight and frameworks of launches and docking depend on the approved version of the Flight Program. Fuel and water deliveries are calculated on the basis of the ISS orbital corrections and data on consumption for various operations. Similar to the cargo flow, a schedule for return of various cargo units from the ISS back to Earth is composed.

The main logistics problem here is the interdependence of all decisions, which requires a considerable co-ordination effort. The cargo capacity of spacecraft is limited and when an unexpected demand for additional cargo arrives, fuel or water volumes may need to be reduced, and vice versa.

3 THE SOLUTION

It is quite obvious from the above description that we are dealing with a *complex problem* and therefore a solution must be sought with the help of *complexity science* [2], whose key fundamental assertion is:

- *Only a complex system with requisite complexity can effectively interact with a complex environment*

In accordance with this assertion, our team designed and implemented a complex adaptive logistics management system, SMART LOGISTICS™, which is supporting launching of spacecraft delivering astronauts and cargo to ISS since 2010. In addition, we have developed a complex adaptive project management system, SMART PROJECTS™, which is used by the client for managing the development of resources for logistics and production.

3.1 Brief overview

The solution is, in fact, a unique *adaptive network (or a swarm) of adaptive real-time multi-agent schedulers*. Constituent schedulers co-operate or compete with each other depending on the context.

The interdependence of schedulers is illustrated in Fig. 1.

Each scheduler consists of one or more swarms of software agents, assigned to individual demands and resources, which also co-operate or compete with each other depending on the context.

To the best of our knowledge, this is the first complex adaptive logistic management system in the world that is in full operation.

The following constituent schedulers are currently allocating resources to demands in time and space.

- Adaptive Real-Time Scheduler for Flight Program Planning
- Adaptive Real-Time Scheduler for Cargo Flow
- Adaptive Real-Time Scheduler for Cargo Storage Allocation
- Adaptive Real-Time Scheduler for Scientific Experiments
- Adaptive Real-Time Scheduler for Resource Management

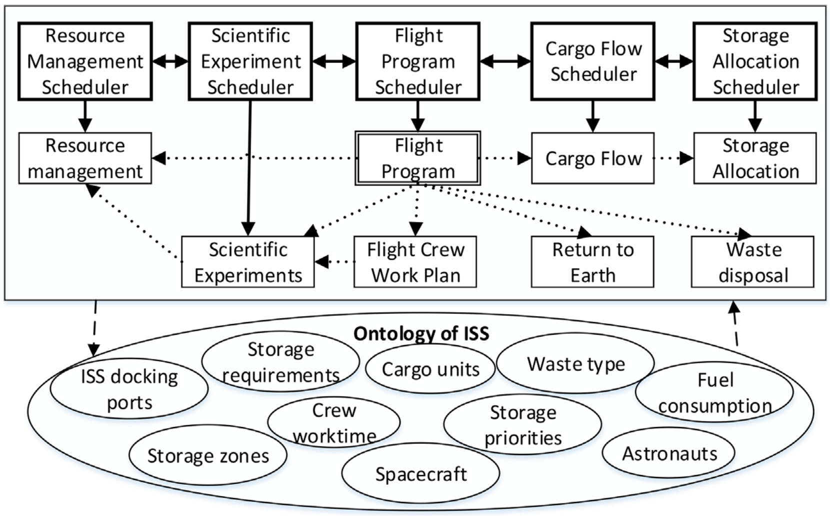

Figure 1: An ontology-based swarm of adaptive real-time schedulers.

Additional schedulers can be developed on demand and easily connected to the Network of Schedulers. In addition to schedulers, the logistic management system includes complex adaptive real-time project management systems, which are used for developing hardware and software resources for ISS logistics.

Individual agents, schedulers, project management systems and the whole network, provide services to users at the appropriate levels, reacting in real time to every new demand, disruption in services, or change in resources [3].

Each scheduler has three key subsystems – *Ontology*, *Virtual World,* and *Interfaces* between the Virtual World and the Real World, as illustrated in Fig. 2.

The Virtual World manages the Real World by performing the following functions.

- Monitoring key variables in the Real World (e.g. demands and the availability of logistics resources)
- Detecting rapidly a disruptive event (e.g. non arrival of an expected order, modification or cancellation of previously accepted order or failure of a resource)
- Identifying parts of the Real World which will be affected by the detected disruptive event
- Re-scheduling of the affected part of the Real World (e.g. redeploying resources assigned to a cancelled order, replacing a failed logistics resource)

3.2 Ontology

All ISS logistics domain knowledge is captured in ontology and factual databases, which together represent the logistics Knowledge Base. A fragment of ontology is represented in Fig. 3.

Domain Object Classes represented in ontology include: ISS, ISS Module, Flight, Spacecraft, Spacecraft Engine, Expedition, Crew Member, Cargo Unit, Fuel, Port, Flight Program, Schedule, etc.

Object Classes are connected by Relations to a network, which can be modified by users with the help of a user interface that allows ontology to be edited without system shutdown. Users can intro-

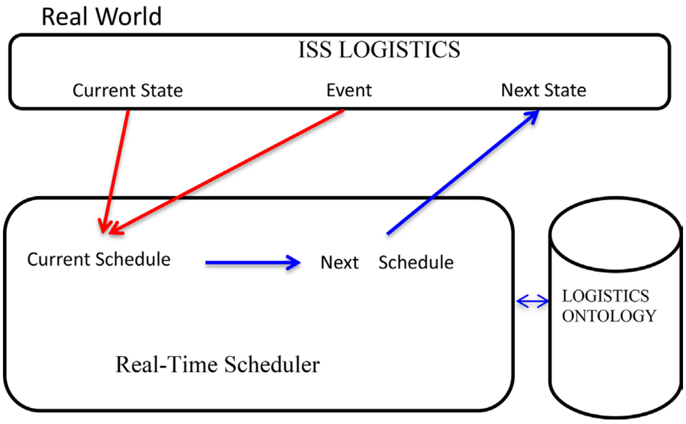

Figure 2: Two worlds, virtual and real, interact and produce a cargo schedule in real time.

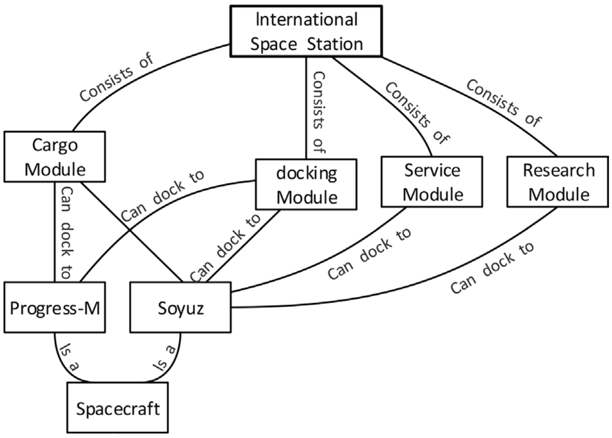

Figure 3: A fragment of flight program ontology.

duce, if necessary, new types of spacecraft (and specify to which ISS ports they can be docked); new types of ISS modules; and new types of cargo and flight operations.

The user interface for the interactive Flight Program Design will adapt appropriately, providing new appropriate capabilities.

The relations between ontology Object Classes and their Attributes can describe both preferences and restrictions used by agents during their negotiation with each other. For example, the relations can indicate which spacecraft types can be used for waste disposal; their maximum payload; different time intervals between launch and docking, based on the chosen approach manoeuvre; spacecraft preferences for ISS docking ports and cargo units; ISS Port preferences for different spacecraft; the required emergency reserve on board the ISS; and the minimal payload of the certain cargo type required to be on each flight.

Ontology provides a basis for constructing instantaneous models of domain, known as Scenes. A scene is, in fact, the current state of the logistics network. Any change in the real world will initiate a series of actions aimed to change the affected parts of the scene. The emerging new scene is the next state of the network.

3.3 Virtual world of agents

Virtual World of each scheduler consists of one or more swarms of software agents. Agents are rather small self-contained computational objects capable of composing messages appropriate to the task in hand, sending messages to each other, interpreting received messages and composing answers to received messages. Agents are triggered into existence when required and removed from the virtual world when not required. Before acting, agents consult knowledge on how to act stored in ontology.

Each agent represents a logistics stakeholder – an order, a production, transportation or storage resource, or a cargo unit. Agents work in swarms (teams) – negotiating with each other (by exchanging messages) how to match logistics resources to demands.

The advantage of agent-based software over conventional programs is due to agents' ability to replace lengthy computational searches with rapid exchanges of messages.

In addition to generic types, such as Order Agent and Resource Agent, schedulers have specialized types of agents. For example, the Flight-Program Scheduler has Flight Agent, ISS Port Agent and Docking Agent types, whilst the Cargo-Flow Scheduler has Cargo Agent and Flight Agent types. Some agent types exist in two or more Virtual Worlds.

To facilitate interaction between schedulers, every two Virtual Worlds should have agents of at least one common type. For example, the Flight Agent is introduced into both the Flight-Program Scheduler and Cargo-Flow Scheduler. If, due to a spacecraft preparation delay, the launch date of the spacecraft is postponed, the agent of this flight, resident in the Flight-Program Scene, will change its status, i.e. it will shift the dates of launch, docking and undocking. Because it acts in both the Flight-Program and Cargo-Flow Scenes, its message about the changes will alert Cargo Agents in the Cargo-Flow Scene to the delay of their flight, and give them an opportunity to negotiate moving into another flight, if necessary.

Conversely, if some cargo deliveries are reduced, the capacity of the flight can become underutilized, which will be represented in the Cargo-Flow Scene by a "partially unsatisfied" Flight Agent. This agent will try to increase its satisfaction rate by searching for new Cargo Agents, which could join its flight.

Virtual Worlds of the ISS servicing system are populated with three types of agents:

- Decision-Maker Agents, representing those who participate in the real world negotiations: system curators, engineers and scientists, top management, etc.
- Agents of Things, representing physical resources that can act as independent entities and have own objectives and constraints, such as spacecraft, flights, expeditions, cargo units, systems, fuel, water, etc.
- Specific Agents, representing abstract entities that negotiate on behalf of groups of the agents, for example, flight-program options or restrictions, cargo-flow schedules, fuel and water tactical calculations, etc.

Virtual World of each scheduler forms a separate swarm of agents, where agents of different types can communicate. Although each constituent agent of a swarm has its own preferences and constraints, it can interact with agents belonging to another swarm using preferences and constraints of the whole swarm.

Here is an example of a typical agent interaction.

- When a new request to allocate a cargo unit to a certain flight arrives from an ISS system curator, a new Cargo Agent is created
- The new Cargo Agent interacts with the Flight Agents that have launch dates in the current Flight Program, searching for an allocation opportunity. This interaction involves sending messages between the Cargo-Flow Scheduler and Flight-Program Scheduler
- Available options are prioritized taking into the account cargo requirements and transaction values
- If there is enough free space on the chosen flight and cargo fits in terms of its weight and size, the new delivery is included into the schedule
- Alternately, the following actions are performed:
- The new cargo is added to the current payload of the chosen flight, causing overload
- Flight Agent sends a request to Cargo Agents to reduce the payload on this flight. Cargo Agents are sorted by their Priority Ranking and by their Preferable Delivery Date attributes and the lowest ranking are taken off the flight
- If the current payload still exceeds the maximum capacity, the previous step is repeated until requirements are met
- Cargo Agents taken off their preferred flight attempt to assign themselves onto the next suitable flight, possibly pushing off some other Cargo Agents
- If the Cargo Agent is unable to assign itself onto any flight in the current Flight Program, its priority is deemed too low, and it will remain unassigned until the next change in the Flight Program, such as moving of a cargo spacecraft launch date from January of the next year into December of the current one.

Two points here are of high importance:

Not all cargo units are independent – two or more cargo units can be linked by being parts of the same set or placed in the same box. In such a case, before one cargo unit is moved to another flight, the system considers interests of other linked cargo units as well as the overall profitability of the move.

When one or several cargo units move, they may free space that is greater than it is required to allocate a new cargo. In this case, it is necessary to initiate rescheduling of oxygen, water, and/or fuel. If Oxygen, Fuel and/or Water Agents previously had to reduce their volumes, they have now an opportunity to restore them. In the above negotiation process, it is impossible to predefine any sequence of steps or logic. Negotiation process is not an algorithm. It adapts to its environment.

The user interface is highly adaptive and customizable, enabling each user to personalize their workplace and reduce work stress.

4 INNOVATIVE FEATURES

The unique solution to the problem of logistics support for the ISS, as described above, has the following distinguishing features.

4.1 Knowledge-driven rather than data-driven

Logistics domain knowledge is collected and stored together and is available to authorized users for inspection and updating without any interruption of the operation.

Agents consult logistics domain knowledge stored in ontology before making any decision. Agents are designed to switch to trial-and-error approach to problem solving, like humans, whenever they find that knowledge is partially incomplete – consequently the system never stalls.

4.2 Real-time rather than batch processing

Both the schedulers and project management systems are sufficiently intelligent (1) to rapidly detect the smallest change in monitored data (early warning about a disruptive event), (2) to identify parts of ISS logistics or project that will be affected by this change and (3) to re-schedule it to eliminate or, at least to reduce, consequences of the disruption before the next disruptive event occurs. In other words, scheduling is performed in real time.

In contrast, the standard way of scheduling is in batch-processing mode, which generally means that scheduling is done once a day. Under current volatile market conditions, when disruptive events typically occur once per hour, it follows that a certain percentage of deliveries will be always out of date.

4.3 Self-organising (adaptive) rather than deterministic

The key advantage of a complex system is its ability to self-organize, in other words, to change its behavior and/or configuration autonomously (rather than under instruction) to neutralize consequences of disruptive events or to improve performance. A system is adaptive if it self-organizes in response to a disruption in a way that eliminates, or at least reduces, consequences of a disruption. Both SMART LOGISTICS™ and SMART PROJECTS™ are adaptive.

In contrast, conventional logistics systems currently on the market, including Enterprise Resource Planning (ERP) systems, are deterministic and therefore have difficulty in coping with volatility of the market and unpredictability of supply and demand.

4.4 Creative rather than programmed

Both SMART LOGISTICS™ and SMART PROJECT™ are capable of autonomously initiating and carrying out performance improvement activities during periods free from disruptions through agent negotiation (rather than by following given algorithms). Therefore, in a limited sense, they are creative.

4.5 Resilient

Complex systems are by definition more resilient to attacks than deterministic systems because of their ability to self-organize, leave aside normal work and concentrate on defence when attacked.

5 RESULTS

The client has used SMART LOGISTICS™ for the design of several Flight Programs for a period 2010–2019 and for scheduling of cargo flows and resources for 2011–2017.

SMART PROJECTS™ has been used for managing several mission critical projects in 2014–2015.

The scope of the scheduler network continues to expand and now also covers the scheduling of scientific experiments and energy consumption, among others.

Integration with the ISS inventory management system constantly updates schedules with the current data, which allows re-scheduling in real time. For example, if an onboard unit reaches its expiration date, it requires the system to deliver a replacement on a nearest preceeding flight, to dispose of the expired unit on the nearest departing cargo vehicle and to reassign one or more of the arriving units into the storage space made available by the removal of the expired unit.

The key result is a situation-driven, flexible, and efficient decision-making process and a reduction in time spent on scheduling and the consequent ability to simulate different schedule options and to support negotiations between the involved departments aimed at finding better reactions to external events.

The total amount of time saved annually varies between different scheduling areas from 30 to 540 man-hours. The saved time is now used for research and comparison of Flight Program options and for analyzing feasible reactions to disruptive events.

In 2015 for the first time in spaceflight history, two consecutive cargo deliveries to ISS have resulted in a failure. Russian vehicle Progress M-27M was lost on April 28th, followed by SpaceX CRS-7 on June 28th. These incidents presented no threat to the safety of ISS or the crew, however. Because of the failures, it was required to rapidly re-schedule all flights and deliveries.

The speed with which our logistics systems accomplished this task was truly remarkable.

Here are some consequences of failures. The next manned flight had to be postponed for two months to allow time for establishing causes of failures. To ensure continuing human presence aboard the station, the departure of then docked Soyuz TMA-15M had also to be delayed for a month, inadvertently helping to set records for the longest continued female spaceflight and for the European Space Agency (ESA) astronaut spaceflight.

While the astronauts on board of ISS had no shortage of supplies due to the properly determined emergency reserve, the depleted stock had to be replenished, and therefore the next Progress cargo delivery mission had to be moved one month forward. To maintain the necessary amount of cargo deliveries after the loss of one vehicle, the launch of Progress M-29M had to be moved from early 2016 into late 2015.

The cargo payload of the next resupply mission had to be completely reworked. Some of the lost 2,500 kg of cargo could be easily replaced (oxygen tanks, food and medical equipment), but the amounts had to be re-scheduled to accommodate both the longer stay of the departing crew and the shorter stay of the next arriving crew, as well as replenishing the depleted emergency stocks. Cargo units, which were deemed to be of least importance (tissue rolls, spare manuals, trash bags, etc.) had to be excluded from the cargo flow completely without affecting the ISS crew. Several of the lost cargo units, which were important and existed only as unique specimens (air absorption filter shell, electrical current converter battery etc.), had to be re-scheduled for later flights until the replacements could be manufactured. The loss of SpaceX CRS-7 also led to cancellation of several EVA, reducing the need to replace several single-use components.

Finally, the scientific equipment for multiple experiments of each space agency was lost between the two launch failures. Like with other cargo, some of that equipment could be replaced on the next supply delivery, while others required a lengthy process of manufacturing a replacement, which affected the experiment schedule. As a consequence, several experiments scheduled for later dates

Table 1: Demarcation of complexity.

Random	Complex	Deterministic
Uncertainty = 1	1> Uncertainty >0	Uncertainty = 0
Components have full autonomy	Components have partial autonomy	Components have no autonomy
Disorganized	Self-organized	Organized
Unpredictable	Emergent	Predictable

had to be moved forward to ensure the maximum effectiveness of the flight crew during their extended stay.

6 CONCLUSIONS

The successful use of a complex adaptive network of schedulers in mission-critical logistics for space exploration validates the thesis that only complex systems with requisite complexity can successfully interact with a complex environment. *Complexity can be managed only by Complexity.*
Authors have learned to design complex adaptive systems and organisations by conjecture-refutation method, to use Karl Popper's terminology [4] – by proposing new designs and subjecting them to severe test of daily use in commercial and space exploration environments. During the last 20 years, the method yielded excellent results: a large number of robust, well performing practical applications and new insights into the science of complexity [2].

APPENDIX

Complexity Fundamentals

Complexity is an inherent property of many systems that constitute the environment in which we grow, develop, live and work. The most important ones are: ecological, cultural, educational, social, scientific, technological, economic and political.

Until recently, levels of complexity of our environment were low and consequently complexity was largely ignored. However, with the rapid development of digital technology the situation has changed, particularly when the Internet transformed the world into a genuine global village and linked regional and national markets into a single global market. The trend is for the complexity of our environment to continue increasing.

Many researchers have contributed to the understanding of complexity, notably the pioneers: Prigogine [5, 6], Kaufman [7], Holland [8] and many others.

This appendix is based on the original work by Rzevski and Skobelev [2, 9] who have built, during the last 20 years, large-scale complex software systems for business clients and investigated these systems with the aim of developing science and art of Managing Complexity.

Researching Complexity

As complexity increasingly affects our capability to carry out our work and cope with our life, it becomes more and more necessary to intensify effort on developing a coherent body of knowledge about complexity; in other words, to develop Complexity Science.

But how do you start assembling a coherent body of knowledge on complexity?

The most productive way is by *building complex systems and then conducting experiments aimed at gaining insight into their behavior.*

This paper outlines research results obtained by experimenting with very large-scale complex systems built for industrial clients for many diverse applications, including: real-time supply chain management in Denmark and Germany, real-time scheduling of road transport in the UK and Russia, real-time scheduling of 2,000 taxis in London, real-time scheduling of car rentals for Avis, real-time scheduling of railways, real-time logistics for delivery of crew and cargo to the ISS, real-time management of aircraft lifecycle, adaptive data mining for an insurance company, adaptive semantic processing for an USA research client.

Defining Complexity

There is no generally agreed precise definition of Complexity. This is to be expected – similar complex concepts, such as Intelligence, do not have precise definition.

Our definition is: *Complexity is a property of open systems that consist of diverse, interacting components, often called Agents and is characterized by the seven key features: connectivity, autonomy, emergence, nonequilibrium, nonlinearity, self-organization and co-evolution* [2, 9].

Connectivity

Agents are interconnected. Complexity of the system increases with the number of links that connect agents to each other. The strengths of agent links also affect system complexity; the weaker the links, the easier is to break them and form new ones, which increases system complexity. Adjusting agent connectivity is an effective method for tuning complexity. Complex systems often consist of regions of high connectivity (and high complexity) interconnected by low-connectivity (and low complexity) links, as exemplified by clustering of activities in the human brain.

Autonomy

Agents have certain freedom of behavior (autonomy), which is always limited by norms, rules, regulations, and/or laws. The increase in autonomy of agents increases complexity and if all constraints on agent behavior are removed the system switches from complex to random behavior. Inversely, if autonomy of agents is reduced (by tightening of laws and/or regulations), the system complexity will decrease, and in the extreme, the system will become deterministic. Complex systems have no central control.

Emergence

Behavior of complex systems emerges from the interactions of agents and is not predictable and yet it is not random. Uncertainty about the outcome of agent interactions is always between 0 and 1. Emergence, in general, denotes a property of a system that is evident in the system as a whole but it is not present in any of its components.

Nonequilibrium

Complex systems are subjected to perpetual change experienced either as a succession of discrete disruptive events or as a slow, imperceptible drift into failure. Frequency of disruptive events varies with complexity. In systems of high complexity disruptive events occur so frequently that the system has no time to return to stable equilibrium before the next disruption occurs. When complexity levels are very high the system is said to be at the edge of chaos because the uncertainty of behavior is close to 1.

Nonlinearity

Relations between agents are nonlinear and may include amplification, acceleration, and even autocatalytic properties [10]. Nonlinearities may amplify a small, insignificant disruptive event and cause a catastrophic outcome (an extreme event), the property called butterfly effect. The butterfly effect increases with complexity. In complex systems outcomes are, as a rule, consequences of numerous interacting causes, and therefore the cause-effect analysis is inappropriate.

Self-organization

Complex systems have a propensity to react to disruptive events by autonomously self-organizing with the aim of eliminating or, at least, reducing consequences of the disruption. This property is called *Adaptation*. Self-organization may be also caused by a propensity to improve own performance, the property called *Creativity* or *Innovation*. To initiate and perform adaptive and creative activities the system must be *Intelligent*. Intelligence, adaptation, and creativity are emergent properties exclusive to complex systems; their levels increase with complexity. The Artificial Intelligence (AI) found in complex adaptive software is normally referred to as *Emergent Intelligence* [10].

Co-evolution

With time, complex systems change as their environments change and, in turn, they affect their environments. Co-evolution is irreversible.

Examples of Complex Systems

Examples of complex systems that affect how we work and live include: ecology, climate, space, geopolitical system, terrorist networks, poverty, the Internet-based global market, national economies, businesses, supply chains, logistics, transportation systems, production systems, services, cities, communities, but also atomic explosions, laser behavior, and tsunami.

Let us consider the Internet-based global market as a representative example.

The global market consists of billions of agents (suppliers, consumers, producers, service providers, investors, bankers, insurers, retailers, traders, consultants, advisers, inspectors, repairers, etc.) engaged in negotiating, agreeing, changing or cancelling commercial transactions.

The Internet enables every market agent to connect to every other agent. These connections are weak, and they can be easily changed. Therefore, the market is characterized by high *connectivity* and, consequently, by high complexity.

Agents, although constrained by national and international laws, market regulations and norms of behavior, enjoy considerable freedom of choice in selecting trading partners and negotiating deals. In other words agent *autonomy* is high and, consequently, complexity of the market is high.

There is no centralized control of the market and its global behavior *emerges* from agent interactions. The supply and demand are therefore unpredictable but not random.

Trading transactions are made, changed, or cancelled with such a speed that the system has no time between two consecutive disruptive events to return to supply–demand equilibrium. The state of the market most of the time is *far from equilibrium*.

Individually insignificant actions of agents (such as arranging subprime loans) are slowly accumulating and, due to the *nonlinearity* of connections, when the *tipping point* is reached, the market experiences *butterfly effect* (such as the global financial crisis).

Every new commercial deal, as well as a change or cancellation of the previously agreed deals, represents a *disruptive event*, prompting the market to *self-organize* with the aim of neutralizing the effects of disruption. As a result, the global market is highly volatile and supply and demand are unpredictable.

The market, society, and technology *co-evolve* and the co-evolution progresses in steps. Transitions from agricultural to industrial and then to information economy are well covered in literature.

Complexity Mindset

Let us use uncertainty of behavior to distinguish complex systems from deterministic and random, as illustrated in Table 1. The term deterministic implies that uncertainty is equal to zero, whilst the term random means that uncertainty is equal to one. Complex Systems have uncertainty value between zero and one.

Uncertainty is a consequence of complexity, and it increases as complexity increases. Low complexity systems have uncertainty close to 0, and their behavior differs little from the behavior of deterministic systems. The behavior of highly complex systems with uncertainty close to 1 is said to be at the edge of chaos. These systems have increased propensity for self-organization, generation of unpredictable extreme events and co-evolution.

The distinction between complex and deterministic systems is very important and has philosophical repercussions.

For centuries eminent philosophers and scientists have believed that the world is deterministic – that it behaves in accordance with natural laws in a predictable manner and that any uncertainty of outcomes is a result of our lack of knowledge how the world works. In other words, for supporters of determinism the world is complex only for those who do not understand it.

A more plausible alternative view has been put forward recently by Prigogine [5, 6]. Prigogine proposes that the world is inherently complex, and it evolves irreversibly with time. Future is not given; it *emerges* from the interaction of billions of activities performed by constituent agents, including people, animals, plants, as well as natural forces such as climate, erosion, volcanic eruptions, and solar spots.

Prigogine's hypothesis how the world works is an essential part of the *Complexity Mindset,* or the *Complexity Worldview,* which consists of the set of assumptions, concepts, principles and methods, which represents an effective toolset for addressing complex issues.

Modelling Complexity

Simplifying complex reality by representing it with deterministic models, such as Newton's laws, or laws of linear control theory, works only in a limited number of situations where uncertainty of behavior is very close to 0. It works, say, for determining the movement of planets but it is totally inappropriate for modelling of systems where the uncertainty of behavior is high, as exemplified by ecological, cultural, social, economic, political and business systems.

Experimental evidence supports the assertion that: *Models of complex systems must be complex.* Moreover, *Complexity of models must be the same, or approximately the same, as complexity of the system that is being modelled*, the principle known as *Requisite Complexity*.

The rationale of the above statements is obvious. Complex systems frequently self-organize (change) in response to external or internal disruptive events and therefore, to be continuously representative, their models must autonomously change in the same way and with the same frequency.

The most appropriate model of a complex system is therefore complex adaptive software designed as a multi-agent system. Agent-based software can be *tuned* to be complex to any degree that is required to meet requisite complexity criterion.

REFERENCES

[1] Soloviev, V., Lubinsky, V. & Guk, E., Current state and perspectives of developments of management of human flights in space. FSBO "Gagarin R&T CTC", Star City, Russia, **1**(1), pp. 27–37, 2011.

[2] Rzevski, G. & Skobelev, P., *Managing Complexity*, WIT Press, UK, 2014, ISBN-13: 978-1845649364.

[3] Ivashenko, A., Khamits, I., Skobelev, P. & Sychova, M., Multi-agent system for scheduling of flight program, cargo flow and resources of international space station. *Proceedings of the 5-th International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS 2011)*,Springer Verlag: France, Toulouse, pp. 165–174, 2011. http://dx.doi.org/10.1007/978-3-642-23181-0_16

[4] Popper, K., *Conjectures and Refutations: The Growth of Scientific Knowledge*, Routledge & Kegan Paul, 4th edn, 1972, ISBN 0-7100-6508-6.

[5] Prigogine, I., *The End of Certainty: Time, Chaos and the new Laws of Nature*, Free Press, 1997, ISBN 0-684-83705-6.

[6] Prigogine, I., *Is Future Given?* World Scientific Publishing Co., 2003, ISBN 981-238-508-8.

[7] Kaufman, S., *At Home In the Universe: The Search for the Laws of Self-Organization and Complexity*, Oxford Press, 1995, ISBN 0-19-511130-3.

[8] Holland, J., *Emergence: from Chaos to Order*, Oxford University Press, 1998, ISBN 0-19-850409-8.

[9] Rzevski, G. & Brebbia, C.A. (eds), *Complex Systems: Fundamentals and Applications*, WIT Press: Southampton, Boston, 2016, ISBN 978-1-78466-155-7.

[10] Rzevski, G. & Skobelev, P., Emergent intelligence in large scale multi-agent systems. *International Journal of Education and Information Technology*, **1**(2), pp. 64–71, 2007.