A Strategy for Managing Complexity of the Global Market and Prototype Real-time Scheduler for LEGO Supply Chain

Bjorn Madsen
LEGO System, Billund, DK
bm@multiagenttechnology.com

George Rzevski
Multi-Agent Technology Ltd, London, UK
gr@multiagenttechnology.com

Petr Skobelev
Multi-Agent Technology Ltd, Samara, RU
ps@multiagenttechnology.com

Alexander Tsarev
Smart Solutions Ltd, Samara, RU
tsarev@smartsolutions-123.ru

ABSTRACT

The paper describes main features of a strategy for managing complexity of the global market and real-time scheduling multi-agent system designed for the LEGO Company. The design is based on Multi-Agent Technology Group (MATech) own strategy blueprint and multi-agent platform, which provide real-time adaptive event-driven scheduling to replenish products to LEGO Branded Retail stores.

The prototype system has been used to schedule 20 US-based LEGO retail outlets for a yearlong trial period and has achieved the following results:

- Reduction of lost sale from 40% to 16%;
- Increase in service level from 66% to 86%;
- Increase in profitability 56% to 81%.

The results show a considerable potential value for full scale LEGO supply chain multi-agent solution which would be able to dynamically and adaptively re-schedule deliveries in real time.

Keywords: *LEGO, supply chain, multi-agent systems, forecasting, scheduling, real-time*

INTRODUCTION

LEGO is known worldwide for its famous LEGO bricks (LEGO web-site, 2012). In addition to supplying over 50,000 retailers worldwide, LEGO also has about 100 own branded retail outlets, which provide the LEGO brand experience. As this retail operation is built to provide a unique shopping experience for consumers, lost sales and service level are considered of paramount importance.

Since the conception of the LEGO Brand Retail (LBR) outlets, the process of ordering stock to the retail outlets has been managed by the LEGO System, the organizational unit that purchases stock from the sole supplier. To create orders, the LBR inventory management team uses an in-house developed Visual Basic / Excel tool which is loaded with point-of-sales data summaries for the past four weeks of sale, inventory position and buying budget for each store. Based on this the LBR inventory management team creates orders for each outlet for each stock keeping unit (SKU), which are submitted to LEGO System.

As the molding process of LEGO bricks is of very high quality, constraints on the lead-time of molds for special plastic bricks propagate into product packaging and subsequently provide constraints on supply. As some products are more popular than others LEGO System has to make a decision on how to allocate the stock amongst its retail customers, and this also determines how large or small will be a share of LEGO Brand Retail.

In this paper we shall consider characteristics of the Internet-based global market within which LEGO System operates (section II), key problems with the LEGO as-is business processes (section III), a strategy for managing global market complexity for LEGO supply chain (section IV), the selection of technology for meeting the requirements specification (section V), the architecture of the prototype solution (section VI), the results of the prototype evaluation using data from 20 US-based outlets (section VII), and, finally, the conclusion (section VIII).

LEGO AND THE INTERNET-BASED GLOBAL MARKET

LEGO is a global business that sells LEGO bricks literally all over the world. The Company operates within the Global Market, which is characterized by a high level of complexity (Beinhocker, 2007) with prominent 7 key features (Rzevski, 2011):

(1) INTERACTION – The Market consists of an exceedingly large number of participants, i.e., suppliers, consumers, service providers and service consumers, who make and change previously agreed deals with a high frequency thus generating disruptive events that affect all participants.

(2) AUTONOMY - Global Market players have considerable autonomy, since they are not subject to a central control, which makes any prediction of demand and supply unreliable.

(3) EMERGENCE – The global market behavior emerges from the interaction of market participants and is therefore unpredictable but not random – it follows discernable patterns.

(4) NONEQUILIBRIUM - The Global Market operates far from equilibrium because the frequency of disruptive events is too high for the market to return to equilibrium between two consequent disruptions.

(5) NONLINEARITY - The relations between market participants are nonlinear and even an insignificant disturbance may be occasionally amplified to cause an extreme event, such as a global financial crisis.

(6) SELFORGANIZATION - the dynamics of the market is very high as it self-organizes in response to disruptive events.

(7) COEVOLUTION – The Market irreversibly coevolves with political, social and technological systems.

The steep increase in complexity of the Internet based Global Market is a new phenomenon attributed to the genuine explosion of digital technology by the end of the 20th and beginning of the 21-st century. Our century has been recently described as the “age of complexity”.

Since complexity of the Global Market is increasing with time, the survival and prosperity of all Global Market players, including LEGO, depends on their capability to recognize the need and develop a Strategy for Managing Complexity. Multi-Agent Technology Ltd (MATech) has created a blueprint for such a strategy and is making it available to its customers, including LEGO.

PROBLEMS WITH THE LEGO AS-IS BUSINESS PROCESSES

Lack of Transparency & Validity of Ordered Quantities

LBR does not revise orders after the stocks have been allocated to them, as the allocation is forwarded automatically to LEGO Systems for picking, packing and dispatch. What determines the stock allocation is the sequence in which LEGO ERP system (a SAP ECC 6.0) receives the orders from LBR. The common procedure is that the orders of "the most important outlets" are processed early in the week, and "the less important outlets" later, therefore the queue by which stock is assigned generates the self-fulfilling prophecy that well performing outlets always will perform well as they are assigned stock early, whilst poorer performing outlets are assigned stock later

What makes the problem worse is that the queue of orders is not being processed until week-end whereby the outlets which were assigned stock first have longer time-lag from the latest demand signal, than those outlets whose orders are processed just before week-end processing.

The constraints of supply, aggregated usage of point-of-sale information, transfer of unresolved problems to suppliers and usage of in-house developed spreadsheets to overcome workload are, by experience, typical for human centric supply chain scheduling processes. The positive perspective is that LBR is aware of them and know that change is required to deliver its promise to the consumers.

Physical Bottlenecks

LBR does not revise orders after the stocks have been allocated to them, as the allocation is forwarded automatically to LEGO Systems for picking, packing and dispatch. What determines the stock allocation is the sequence in which LEGO ERP system (a SAP ECC 6.0) receives the orders from LBR. The common procedure is that the orders of "the most important outlets" are processed early in the week, and "the less important outlets" later, therefore the queue by which stock is assigned generates the self-fulfilling prophecy that well performing outlets always will perform well as they are assigned stock early, whilst poorer performing outlets are assigned stock later.

This storage is costly and accounts for ~ 12% of the distribution cost. If LEGO Systems warehouse operation would be flexible, so that only the receivable quantities would be dispatched on a day to day basis, for example in a pallet-network, this would not significantly increase the total logistic cost.

A STRATEGY FOR MANAGING COMPLEXITY OF THE GLOBAL MARKET FOR LEGO

The Complexity Management Strategy developed by MATech is based on concepts and methods of Complexity Science (Prigogine, 1997; Holland, 1998), and it has been tested in a very large number of commercial implementations (Rzevski, 2008; Rzevski, 2010; Glaschenko &

Ivaschenko & Rzevski & Skobelev, 2009; Andreev & Rzevski & Shveykin & Skobelev & Yankov, 2009; Rzevski & Skobelev & Andreev, 2007; Andreev & Rzevski & Skobelev & Shveykin & Tsarev, 2007).

The key idea behind this strategy is that to survive and prosper under conditions of complexity it is necessary to ensure, in the first instance, that critical business processes are Adaptive. Once Adaptability is in place it is necessary to improve security under conditions of uncertainty created by complex Global Market by ensuring that critical business processes are Resilient.

Adaptability requires distributed and rapid decision making to enable the business process to react positively to an unpredictable disruptive event before the next event occurs. The appropriate distribution and speed of decision making can be realized in practice only by real-time scheduling systems incorporating multi-agent technology.

Resilience requires distributed and rapid dynamic data mining of critical data sources in order to discover a malicious attack or fraud as early as practical. Dynamic data mining systems are much more advanced and are usually developed in the second phase of the Complexity Management Strategy.

The requirement specification for a Real-Time Scheduler for LEGO Supply Chain, developed using principles elaborated in MATech Strategy for Managing Complexity, was as follows:

- The system must be capable of dealing with 100+ outlets, thousands of SKUs and weekly, monthly and annual fluctuations in demand, including merger of belief-based long-term forecast with data-driven short-term forecasting.
- The system must be able to scale up (and down) with the size of the business as it evolves with time.
- The system must be able to optimally exploit any given moment in assigning the limit supply of stock to outlets, so that lost sales are minimized, and service level & profit are maximized.
- The system must propose replenishment orders automatically and respond to any change in data. This is to be both interactive and to move away from batch processing of information, which is considered an inhibitor of transparency of the business.
- The system must allow users to override its decisions when required. However whenever users override the system they must be informed of the consequences to the rest of the business.

As LEGO Brand Retail has no experience with this type of systems the leadership team decided to initiate a pilot project under the management of an internal researcher. The pilot project revealed additional problems. LBR and LEGO Systems usage of enterprise wide applications are batch-based, which means that the transition to real-time information processing is a large development step. Other alternatives, such as SAP Forecasting & Replenishment (F&R) was evaluated, but due to SAP F&R's architecture, which generates orders under the assumption that the supplier has infinite capability to respond, the orders which SAP F&R creates are not revised after it has been decided how much stock is available, whereby the problem persists. In addition SAP F&R is based on batch information processing, which inhibits learning as all interactions require a batch run before the user may learn the consequences of his/her action

To minimize the risks in the development process a stand-alone proof-of-concept model was developed over 6 months, with outlook for the full scale ERP integration in the following 6 months.

SELECTING TECHNOLOGY FOR LEGO SUPPLY CHAIN SOLUTION

As mentioned the main challenge was to respond to any changes in demand based on point of sales data and compute the optimal solution to the time-variant sequential multiple knapsack problem created by constraints of movement (inbound to outlets, outbound from supplier), costs of all activities, lack of knowledge about future demand, present rate of sale and utilization of the inventory in the outlets.

This requires a continuously ongoing optimization process, which is evolutionary (as events take place and data is added) and permits a more efficient adaptive method of identifying solutions in the solution landscape, so that the system does not have to compute every solution top-down, whenever a minor update is made to the data set.

The above requirement eliminates methods such as mixed integer programming and similar other methods (Leung, 2004; Vos, 2000; Rego & Alidaee, 2005), up front as inefficient, and points to a preference for multi-agent systems, where number of orders and resources is not known in advance and decisions are need to be made under conditions of uncertainty and high dynamics.

In the category of multi-agent systems (Bonabeau & Theraulaz, 2000; Wooldridge, 2002; Brussel & Wyns & Valckenaers & Bongaerts, 1998), ontology based optimizers were preferred ahead of generic particle swarm optimizers, as ontology based systems attempt to assess the consequence of mutation of the existing solution, prior to mutating, whilst generic PSO's mutate and then assess the fitness of the solution in the solution landscape. For practitioners this means that ontology based systems have fewer mutations though the run-time is comparable with PSO's. Finally amongst the different categories of ontology based systems, negotiating resource-demand-networks have shown to be most efficient (Rzevski & Skobelev, 2007; Skobelev, 2011; Multi-Agent Technology web-site (2012).

MULTI-AGENT SOLUTION

To replicate the environment in which data is to be transformed into allocation and order decisions the following architecture was developed on the Microsoft .Net-platform in which four conceptual elements are present (Fig. 1).

The "Real World" is captured in a Microsoft SQL server 2008 R2, with import through.

The "Data" is imported to the multi-agent virtual world by "day-end" with all point-of-sale records (location, material sold, quantity, etc). However the architecture permits that the data from the point-of-sales database could be forwarded to the MAS in real-time, if needed.

The "Virtual World" contains agents as autonomous objects triggered by events or messages from other agents.

The "Ontology" contains a XML-based construct of "how the supply chain world works". Visual representation of ontology for LEGO supply chain network is given in Fig. 2.

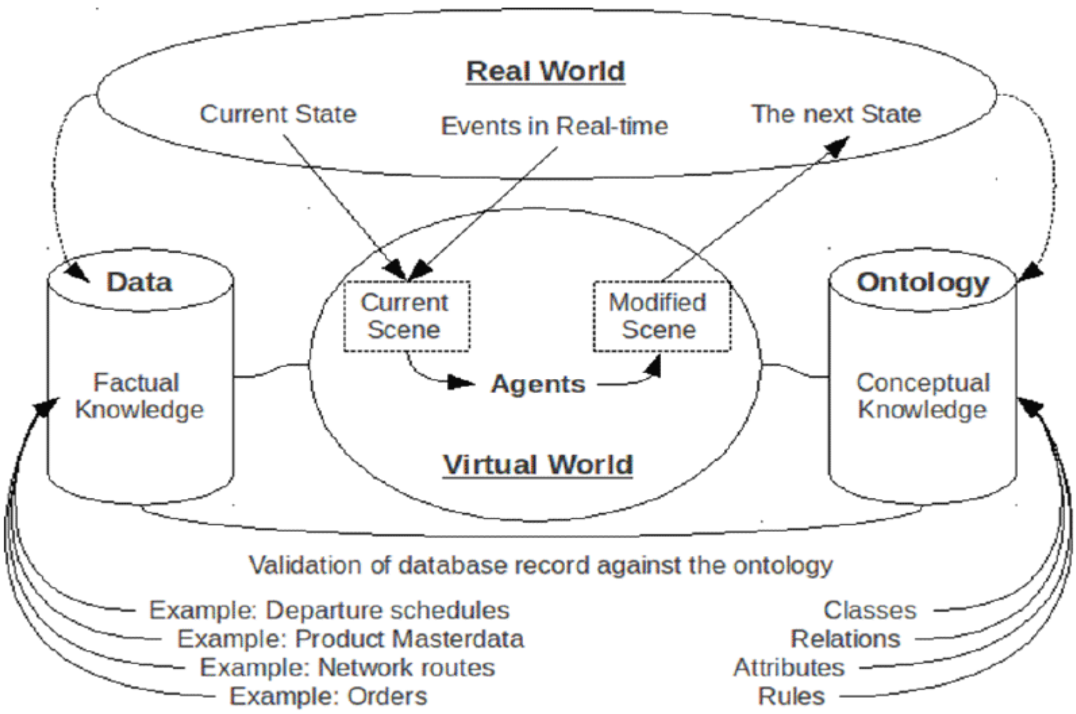

Figure 1. Conceptual architecture of multi-agent solution for LEGO

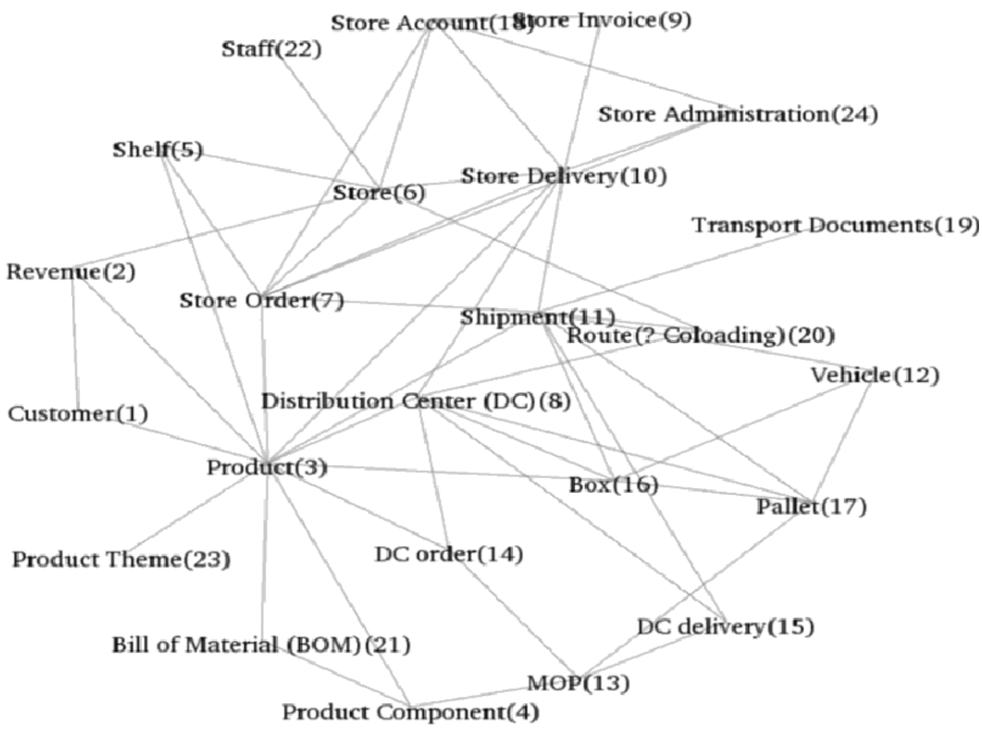

Figure 2. Semantic network of LEGO Supply Chain ontology presented as Fruchterman-Reingold graph

The examples of classes, relations, attributes and rules are presented in Tab. 1.

Table 1. Examples of classes, relations, attributes and rules

Class (node)	**Relations** (node)	**Attributes**	**Rules**
Customer (1)	Revenue (2), Product (3),	Customer type {unknown, club member}	Pays for products. Gets refund when returns product. Probable to select alternative product if wanted SKU is not there.
Revenue (2)	Customer (1), Product (3),	currency {GBP, USD, EUR, ...}	Created when paid
Product (3)	Customer (1), Shelf (5), Store Order (7), Distribution Center (8), Store Delivery (10), Shipment (11), DC order (14), Box (16),	product id, height, width, length, price, FMC-value, theme, barcode	Must be packed into a box before shipping

How the adaptive scheduling works. Though the final scheduler will contain all the conceptual elements, the proof-of-concept included only essential elements for the autonomous forecasting & scheduling, which could be managed in a single swarm governing deliveries and orders as a resource-demand network [18].

This permits incremental import of each event, which triggers adapted forecasting and repeated rescheduling following a plan/commit/execute protocol, which reflects the flexibility of real-world conditions. For example if a delivery has been planned, it may be changed until such point in time where it is necessary to commit the orders to the warehouse operation for picking, packing and subsequent dispatch, or for example a truck has to be booked a day in advance of the warehouse operation (Fig. 3).

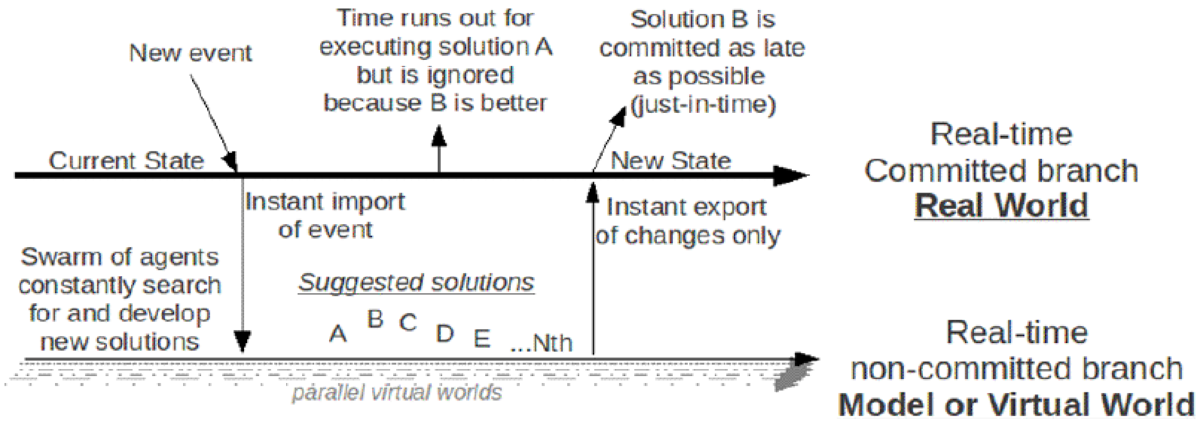

Figure 3. Illustration of the incremental adaptive re-scheduling as events are imported

The scheduling process is based on two steps for every event. First the event signals that a product has been consumed, through point-of-sales records. This triggers a revision of the forecast for that particular product, based on the virtual worlds current state, containing attributes such as current inventory level, current rate of sale and stochastic variation. The computation may show that an agent should be initialized to coordinate the delivery of a product. The “negotiation power” is determined by the agent profitability based on a trade-off between value of a lost-sale and profit of a sale at the point in time when the product is expected to be sold.

Virtual World of Agents. The whole processing of the initial scene and each individual event is performed by a community of agents called the Virtual World (VW). Each event represents a set of changes happened in real world, and triggers the activity of agents associated with the changed objects. The deviation from the stable result provoked by the changes is VW and the propagating changes in the scene. In this way the system reacts adaptively and in real time while maintaining the optimal KPIs.

Agent types. The multi-agent world consists of several types of agents:

- Consumption Agent;
- Replenishment (Delivery) Agent;
- Stock Agent;
- Product Agent;
- Site (Location) Agent;
- Transportation Agent.

Consumption agent is a demand in the supply-and-demand network and responsible for making the consumption of a specific product at a specific moment of time possible. It can represent a forecasted consumption or a consumption that has really happened. The consumption demand is fully satisfied if there is enough stock for it at the scheduled time of consumption. If there is not enough stock, the consumption demand negotiates with Replenishment agents to deliver more product items by this time.

Replenishment agent is also a demand and represents the delivery of products to a location. Replenishment agent negotiates with the Transportation agent, Product agent, and Site agent to get the restrictions and cost of delivery for a specific volume of products. Replenishment agents charge Consumption agents for putting the products into the delivery and for changing the time of delivery. Replenishment agents produce additional stock levels. Stock agent represents the main resource in the swarm. The Stock agents charge consumption demands for keeping product items in stock and provide information on the availability. If the stock level changes unexpectedly the Stock agent pushes the Product agent to re-consider the forecast.

Product agent is mainly responsible for maintaining the forecast of consumptions up-to-date. It knows the specifics of the Product and changes the forecasted consumptions if the situation changes (e.g. if they are sold faster).

Site agent is responsible for tracking site restrictions (storage size, delivery processing power) and knows the cost of storage.

Transportation agent knows the limitations of a specific transportation channel (number of pallets) and cost function.

Events. The solution supports the following list of events:

- Expected Occurrence of Consumption;
- Unexpected occurrence of Consumption;
- Nonoccurrence of expected Consumption;
- Change in Consumption quantity;

- Unexpected change in Stock level;
- An occurrence of Replenishment;
- Change in current time.

Any event can produce a chain of negotiations inside the VW. The length of the chain depends very much on the situation and can lead to a complete rescheduling in the worst case. Sometimes several events are processed at once. The possible negotiation relations and protocols between agents are presented in Fig. 4.

The processing of events can affect: time of delivery; allocation of consumptions to replenishments; consolidations of products in deliveries; size of consumptions; size of deliveries; cost of product storage and transportation; and/or company profit.

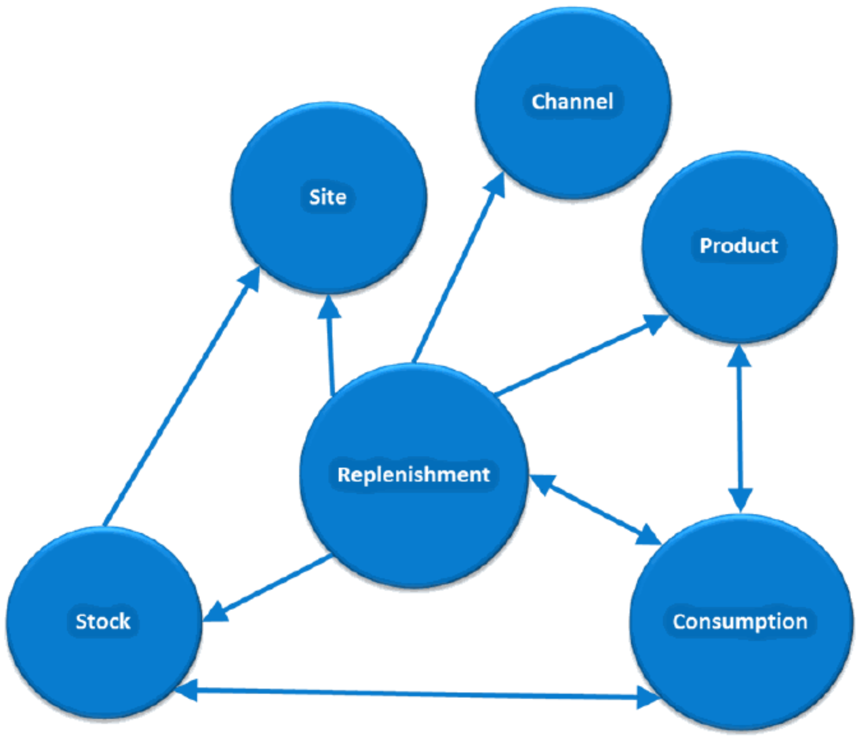

Figure 4. Basic Proptocols of Agents Negotiations

Logic of forecasting mechanism is presented in Fig. 5.The main idea here is that each new sale can trigger re-scheduling of delivery with the goal to support service level or make more profit.

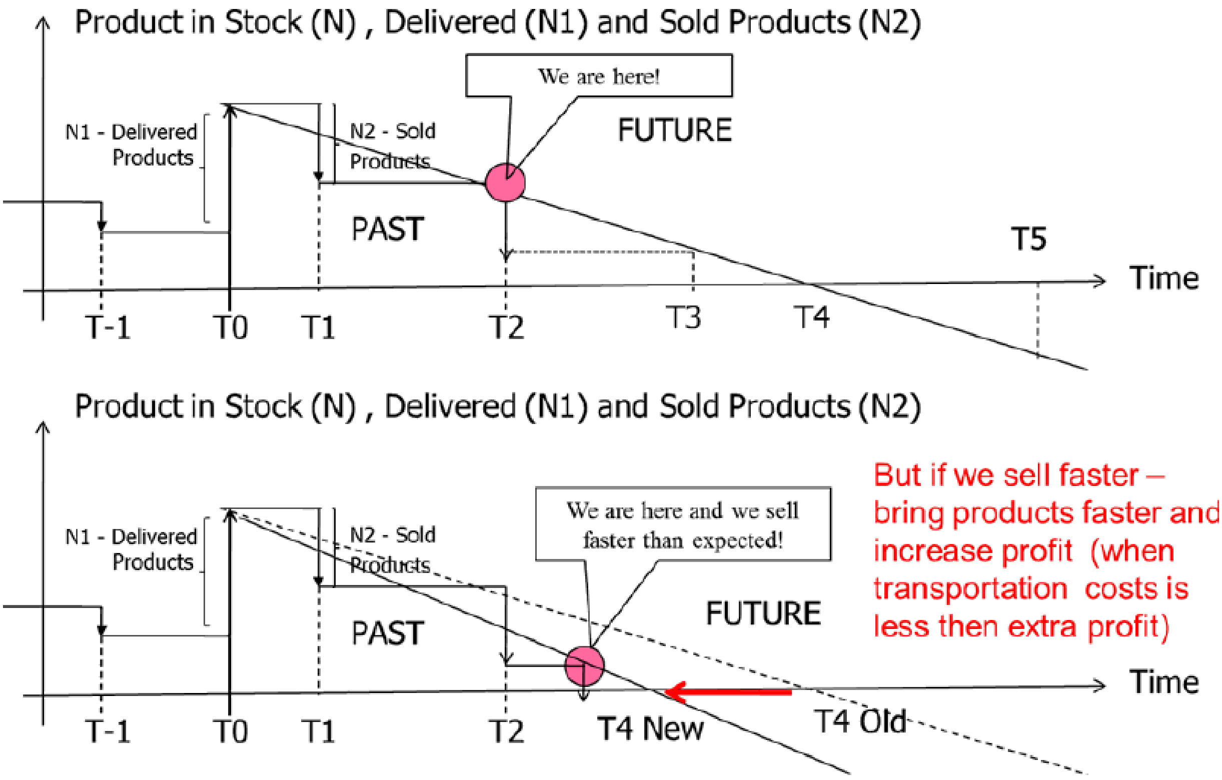

Figure 5. Example of forecast recalculations

THE KEY RESULTS

We designed and implement multi-agent solution for real time adaptive re-scheduling of deliveries in LEGO supply chain.

User interface of solutions is presented on Fig. 6, which represents current sales, forecasts, etc.

As the results were produced using the point-of-sales data (to represent the demand signal) the key decision was to apply LBRs existing processes (i.e. current practice) once more on the same data. This does not give the full picture but provides an indication of the effect of relaxing the constraints, which the business faces at present on real data (Fig.7). In addition profit (of potential based on the assumption that the POS data is the real demand) was calculated for the relaxation of each constraint. The constraints were relaxed as follows, starting from the ideal case, then added layers of constraints to match current practice. The combinations were:

A. Real-time scheduling with flexible business processes (idealistic future).

B. Real-time scheduling with fixed business processes (realistic future).

C. Fixed scheduling scheme & rigid business processes (current practice).

We have also considered the following different mechanisms of forecasts:

1. “Perfect forecast” – in case if we fully know reality in advance.

2. Stochastic forecasting – in case we know history and adaptively changed probabilities of next sales.

3. Trendline based forecasting (current practice).

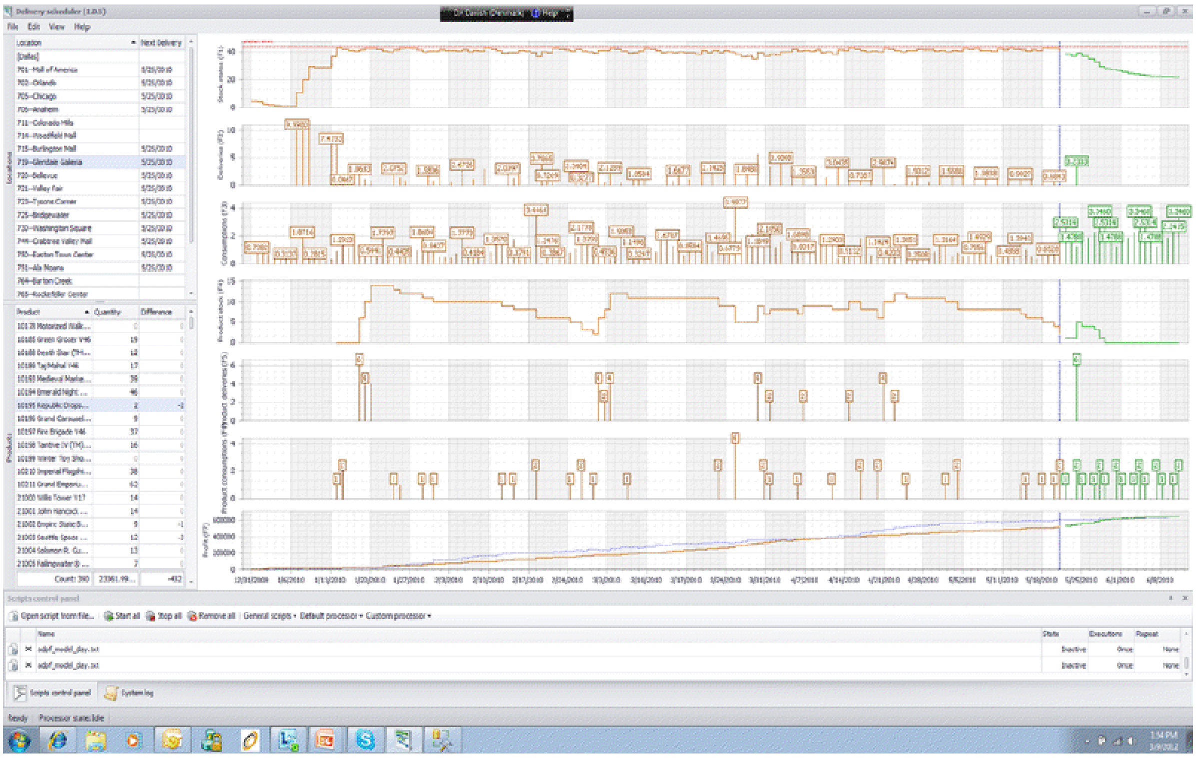

Figure 6. Example of User Interface

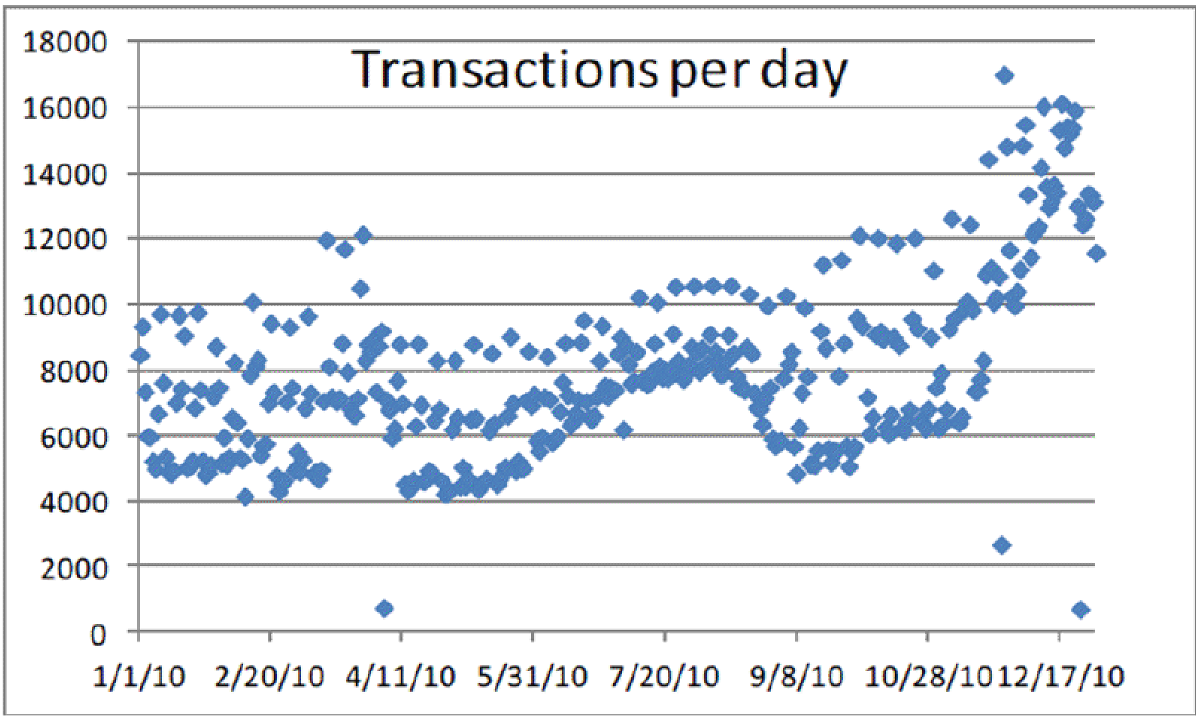

Figure 7. Example of real sales data from outlets

The results are summarized in Tab. 2. The orange line (bottom) indicates current practice, which is contrasted with the green line (third row) that indicates an achievable state with real-time scheduling. Using the designed multi-agent solutions for selected US-based 20 outlets for one-year trial period time LEGO has achieved the results:

- Reduction of lost sale from 40% to 16%;
- Increase in service level from 66% to 86%;
- Increase in profitability 56% to 81%.

The achieved results are exceptionally positive and show the value of a full scale LEGO supply chain multi-agent solution, which will be able to dynamically and adaptively re-schedule not only outlets transportation deliveries but also manufacturing and managing cross-docs inbound and outbound in real time.

Table 2. Results achieved by the prototype scheduler

Scenario	**Profit**	**Service Level**	**Lost Revenue**	**Cost**
Theoretical Ideal	100%	100%	0%	100%
(A1) Real-time scheduling with flexible business processes + “perfect forecast”	88%	90%	10%	102%
(A2) Real-time scheduling with flexible business processes + Stochastic forecasting	81%	86%	16%	105%
(A3) Real-time scheduling with flexible business processes + Trendline based forecasting	76%	86%	20%	105%
(B1) Real-time scheduling with fixed business processes + “perfect forecast”	82%	83%	17%	96%
(B2) Real-time scheduling with fixed business processes + Stochastic forecasting	76%	79%	22%	96%
(B3) Real-time scheduling with fixed business processes + Trendline based forecasting	61%	71%	35%	96%
(C1) Fixed scheduling scheme & rigid business processes + “perfect forecast”	81%	82%	17%	96%
(C2) Fixed scheduling scheme & rigid business processes + Stochastic forecasting	66%	69%	31%	95%
(C3) Fixed scheduling scheme & rigid business processes +	56%	66%	40%	95%

Trendline based forecasting				

CONCLUSIONS

The results of the first stage of designing Adaptability into LEGO supply chain, as described above, confirm the value of MATech Strategy for Managing Complexity and its multi-agent platform for building real-time schedulers.

Next step will be focused on integrating LEGO Brand Retail with existing SAP system. Future improvements will include the support for product lifecycle, removing non-selling products, merging with belief-based forecast realized deliveries and further expansion of the real-time scheduling.

REFERENCES

LEGO web-site (2012). Retrived December 12, 2012, from: http://www.lego.com/en-us/Default.aspx

Beinhocker, E. (2007). *The Origin of Wealth: Evolution, Complexity and the Radical Remaking of Economics*. Random House Business Books.

Rzevski, G. (2011). A practical Methodology for Managing Complexity. Emergence: Complexity & Organization. *An International Transdisciplinary Journal of Complex Social Systems*. 13(1-2), 38-56.

Prigogine, I. (1997). *The End of Certainty: Time, Chaos and the new Laws of Nature*. Free Press.

Holland, J. (1998). *Emergence: from Chaos to Order*. Oxford University Press.

Rzevski, G. (2008). Investigating Current Social, Economic and Educational Issues using Framework and Tools of Complexity Science. *Journal of the World University Forum*. 1(2), 75-84.

Rzevski, G. (2010). Using Tools of Complexity Science to Diagnose the Current Financial Crisis. *Optoelectronics, Instrumentation and Data Processing*. 46(2).

Glaschenko, A. & Ivaschenko, A. & Rzevski, G. & Skobelev, P. (2009). Multi-Agent Real Time Scheduling System for Taxi Companies. In Decker, Sichman, Sierra, and Castelfranchi (Eds.), *AAMAS 2009, 8th Int. Conf. on Autonomous Agents and Multiagent Systems* (pp. 29-35).

Andreev, S. & Rzevski, G. & Shveykin, P. & Skobelev, P. & Yankov, I. (2009). Multi-Agent Scheduler for Rent-A-Car Companies. In Lecture Notes in Computer Science: Vol. 5696. *HoloMAS 2009, Holonic and Multi-Agent Systems for Manufacturing: 4th Int. Conf. on Industrial Applications of Holonic and Multi-Agent Systems* (pp. 305-314). Linz, Austria: Springer.

Rzevski, G. & Skobelev, P. & Andreev, V. (2007). MagentaToolkit: A Set of Multi-Agent Tools for Developing Adaptive Real-Time Applications. In Lecture Notes in Computer Science: Vol. 4659. *HoloMAS 2007, Holonic and Multi-Agent Systems for Manufacturing: 3rd Int. Conf. on Industrial Applications of Holonic and Multi-Agent Systems* (pp. 303-314). Regensburg, Germany: Springer.

Andreev, M. & Rzevski, G. & Skobelev, P. & Shveykin, P. & Tsarev, A. (2007). Adaptive Planning for Supply Chain Networks. In Lecture Notes in Computer Science: Vol. 4659. *HoloMAS 2007, Holonic and Multi-Agent Systems for Manufacturing: 3rd Int. Conf. on Industrial Applications of Holonic and Multi-Agent Systems* (pp. 215-225). Regensburg, Germany: Springer.

J. Y-T. Leung (Ed.). (2004). *Handbook of Scheduling: Algorithms, Models and Performance Analysis*. Chapman & Hall: CRC Computer and Information Science Series.

Vos, S. (2000). Meta-heuristics: The State of the Art. In A. Nareyek (Ed.), *Local Search for Planning and Scheduling: ECAI 2000 Workshop*. Germany: Springer-Verlag.
Rego, C. & Alidaee, B. (Ed.). (2005). *Metaheuristic Optimization Via Memory and Evolution: Tabu Search and Scatter Search*. Boston-London, Kluwer Academic Publishers: Operational Research & Computer Science Series.
Bonabeau E. & Theraulaz G. (2000). Swarm Smarts. What computers are learning from them? *Scientific American*. 282(3), 54-61.
Wooldridge, M. (2002). *An Introduction to Multi-Agent Systems*. John Wiley & Sons.
Brussel, H.V. & Wyns, J. & Valckenaers, P. & Bongaerts, L. (1998). Reference architecture for holonic manufacturing systems. *PROSA, Computer in Industry*. 37(3), 255-274.
Rzevski, G. & Skobelev, P. (2007). Emergent Intelligence in Large Scale Multi-Agent Systems. *Education and Information Technologies Journal*. 1(2).
Skobelev, P. (2011). Bio-Inspired Multi-Agent Technology for Industrial Applications. In F. Alkhateeb, F. & Maghayreh, E. & Abu Doush, I. (Ed.). *Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications* (pp. 495-522). Austria-Croatia: InTech Publishers.
Multi-Agent Technology web-site (2012). Retrived December 20, 2012, from: http://www.multiagenttechnology.com