Design & Deployment of an Enterprise Grade Real-time Multi Agent System for Supply Chain Synchronization

Bjorn Madsen
Multi-Agent Technology Ltd, London, UK
bm@multiagenttechnology.com

Abstract: To respond to customer demand businesses invest in capacities and supply. Any mismatch results in obsolescent stock, wasted resources and lost sales. In this paper the considerations for design & deployment of an Enterprise Grade Real-time Multi Agent System for supply chain synchronization is presented, so that each and every business involved in the supply chain can adjust their activities to minimize the wasted resources.

I. INTRODUCTION

Businesses must invest in capacities and supply to respond to their customers' demand. Any mismatch between supply, capacities and demand results in obsolescent stock, wasted resources and lost sales. By having the right tools to rapidly transform available information about changes in the business environment, into decisions; managers can perform a sequence of interventions that render themselves as the most productive interventions possible in their part of the complex economic system. The concept that multiple independent decision-makers share information so that each and every business involved in the supply chain can adjust their activities to minimize the wasted resources is called supply chain synchronization. The unrealistic ideal of perfect synchronization is what every chief supply chain officer (CSO) pursues. [1]

II. PROBLEM

The key problems which need to be overcome to reduce present date's lack of synchronization are bound in the complex network through which each piece of information has to propagate before it becomes available to the decision makers who assist the CSO in achieving his objective. This makes factors that delay usable information from arriving at its users the focal problem:

- Propagation time from source to user.
- Translation between heterogeneous platforms.[2]
- Batch-processing of information. [3]
- Periodic Routines such as weekly or monthly decision routines. [4]
- Human interaction causing errors & bias. [5]

III. CURRENT STATE OF RESEARCH

A system suitable for supply chain synchronization must be capable of incorporating the signals produced by the complex socio-economic system which the supply chain is a part of with minimum delay.

We define the complex economic system as a complex socio-technical transactional network, in which interactors transmit, transform and receive information. All interactors are both autonomous and interdependent which as a distributed system is propagating transactional information that leads to self-organization (through actions as response to information) and co-evolution (competition for scarce resources). These features produce a system which is far from equilibrium, and exhibits non-linear response to events, which is both emergent and unpredictable. [6]

Previous research revealed that timely information sharing could lead to:

- An increase in vehicle utilisation for transport companies from 43% to 62%.
- An increase in for a retailer in:
  - sales (+24%)
  - service (+20%) and
  - profit (+25%)
- An increase in flexibility to deliver products from 90% to 98%.

The value of the potential for the involved megabrands are in the range of billion $US. [7]

IV. METHOD PROPOSED

To be synchronized, any updates which have been received by any business in the supply chain from the complex economy should be incorporated immediately and transformed into a decision that defines which intervention to make (if any at all) and communicate its conclusion immediately to its supply chain peers, using the communication concept of "maximum delayed commitment". For this purpose the designed system has three chronological time horizons:

- Planned,
- Committed,

978-1-4799-0172-2/13 $26.00 copyright 2013 IEEE
ICIS 2013, June 16-20, 2013, Niigata, Japan

- Executed.

Using differentiated commitment horizons the system can operate from the perspective that the plans may change with every single update and only at the point in time when commitment must occur, the reduction of degrees of freedom is accepted through commitment. [8]. It is posited that three generic processes (Transformation, Storage & Relocation) suffice to imitate any supply chain:

Transformation is the conversion of inputs to outputs in coherence with an object ratio model, commonly referred to as "bill of material", that requires a certain amount of time, and consumes a certain amount of resources.

Storage is the process of keeping an object at the same location in the supply chain. Because storage is keeping distinct objects distinct, the process of storage is maintaining a certain level of entropy, which also requires consumption of resources.

Relocation is the process of moving of objects from one location to another, which also require a certain amount of time and consumes a certain amount of resources.

In combination and at different levels of scale these three generic processes may imitate any system, ranging from the storage, transformation and relocation of bits of information in an electronic circuit to large scale physical processes such as assembly lines, continuous production and international container logistics. To make the supply chain context explicit I shall refer to locations that hosts processes as "sites" and the link that connects sites as "channels".

I will also refer to the objects which the generic supply chain handles as "stock keeping units" (SKUs) disregarding whether there are services or physical objects:

The logic is that at certain levels of abstraction it becomes unclear whether a person is delivering a service (for example providing software for a machine to operate) or whether that software is simply an "object" that has to be installed correctly and ready to use at a certain point in time. In this view a service is the availability of a capability which is equally important as the object or objects which it is exerted on. Finally I will refer to consumed and gained resources respectively as "costs" and "revenue".

V. SOLUTION FUNCTIONALITY

In the complex economic system information is associated with the state of the system, in theory as a finite state machine: Either an object is in a given state or it is not. In conventional planning systems assumptions are made about objects where the state is unknown and sustained until proven right or wrong. A system suitable for supply chain synchronisation may benefit from forecasting about unknown elements, but must trigger a review of assumptions based on both occurrence and the non-occurrence of an expected (planned) event.

As review of assumptions always is connected to the deployment of resources and capacities, the algorithmic challenge is to solve two inter-dependent assignment problems:

A. Assignment of resources to supplies and capacities, and
B. Assignment of supplies and capacities to orders that return revenue.

The theoretical problem is NP-hard [9] but solvable to a satisfactory level, as long as asynchronous updates may be incorporated effectively. The latter requires computation with time-variant datasets, which experience shows that only 4 classes of algorithms are suitable for:

- Monte Carlo simulation
- Genetic algorithms
- Particle swarm optimizers
- Multi-agent systems

The three first classes operate with unnecessary overhead as they are based on random initial state, where multi-agent systems, in contrast exploits the disturbance caused by the update-event itself. Using this trigger, propagation of changes will quickly refine the solution in the solution landscape.

As supply chain synchronization is the objective, the system designer should be conscious that if a commitment (decision) has to be made and there only is time to generate a single alternative solution, then the better of the two is the best choice. However if there is sufficient time to evaluate more options, then, a continued evaluation of the solution landscape should be performed. This confirms the suitability of multi-agent systems to incorporate updates "effectively" in a manner very similar to the characteristics describing the complex economic system. Based on these considerations a system capable of computing the

right solution to the two assignment problems is a real-time multi-agent system that uses the generic supply chain model as ontology.

VI. ARCHITECTURE

The practicality of connecting a system designed for supply chain synchronization requires thorough consideration of *which other systems* that may connect to it. In the illustration below [Figure 1] an overview is provided which seeks to highlight the difference between data received from the complex network the installation is a part of (unidirectional), and the systems with which it is interactively engaged (bidirectional).

The illustration also highlights that there are system elements which augment the supply chain synchronization system, but which may not be necessary if the organisation cannot execute more detailed levels of planning.

An example hereof is the ability to consider real-time feedback from the cargo routing system (CRS), which determines the optimal route of each delivery in the supply network. The supply chain synchronisation system will have to operate with master data, such as for example rates for relocation (transportation) of cargo between its sites. These transportation rates (costs) are based on past routing decisions.

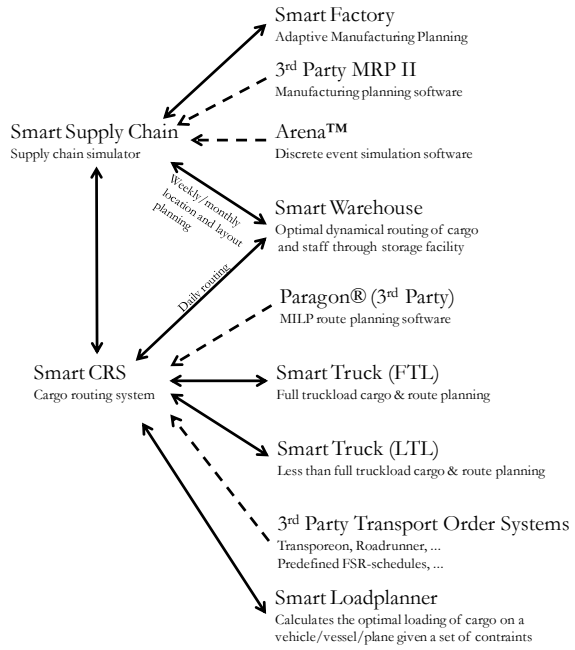

Figure 1 Smart Supply Chain and external systems.

The cargo routing system may query the actual transportation plans from a FTL or LTL system, which will calculate the exact routes, driving and waiting time, express and cross-docking handling fees, and update the cargo-routing system, so that it may consider the exact condition. This may result in the update that the real costs of performing an action which was proposed, based on historic master data, needs revision because the historic rates are inappropriate for the particular case. Hereby the assignment decision for the particular "order" is revised anew in the supply chain simulator. This does not mean that the system is computing in vain. Rather the opposite: The combinatorial solution landscape of the complex economy is larger than the number of particles in the universe [10], so it is a computationally parsimonious strategy to initiate planning based on generic assumptions, and then refine the decisions in iterations.

In the research performed, no other system, proprietary or open source, were identified that can update its own master data using this interactive methodology.

Architecture - the core MAS Engine

The Multi Agent Systems is limited by the ontology. This permits a lightweight engine to be developed that natively uses "channels" as communication lines and hereby becomes regionalized between "sites". So by imitating the coordination channels of the real world, our system complies to the Rzevski Thesis [6]:

"*Complex situations can be effectively investigated only by using models that are of similar complexity to situations that are being modelled.*"

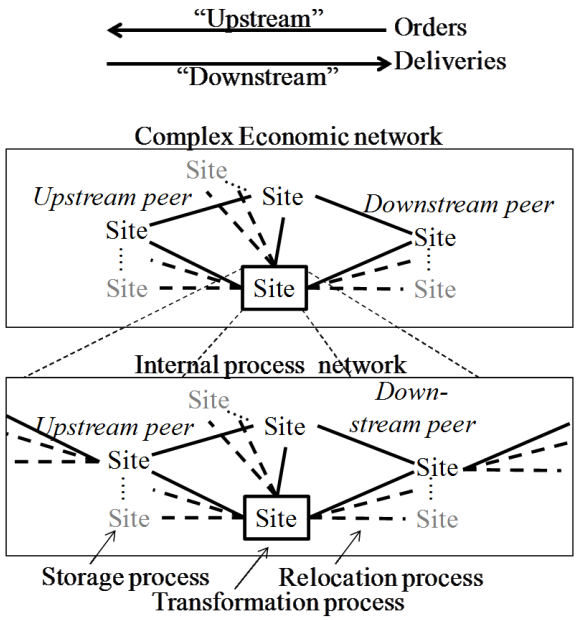

Figure 2 Conceptual illustration of scale free architecture.

The complexity of the activities within a site must be modelled as "a swarm of agents" within the "swarm of the network" so that the application can operate incrementally throughout the system.

The system therefore contains the following classes of interacting agents:
ConsumptionDemandAgents that represent orders for products and "consumes" stock from storage processes and pays the revenue when the transaction is satisfied. A consumption demand may contain request for multiple SKUs, so that it will only pay if completely satisfied. For this purpose the ConsumptionDemand event has a lifetime determined by a time period setting. If this is set to "0" or "instantaneous" then only orders for already available stock will be fulfilled.
DeliveryDemandAgents that govern the logistic journey of a SKU from its origin to its destination; so that process constraints are complied to. Its objective is to negotiate the route through the network, seeking to assure that a minimum contracted service levels is complied to and afterwards maximising profitability.
StockAgent, that governs each SKU in each storage location, transformation and relocation process. It objective is to minimize storage costs (number of SKUs stored) and circulate the stock by FIFO principle.
SKUagent which governs the acceptable relationships to other SKUs (for example product compatibility in storage locations) and manages the profit & loss account of the SKU at each step in its journey.
SiteAgent, which acts as gateway for communication to peers. It is also place holder for the generic processes, which are linked to the site. All key performance indicators are governed by entry and exit through the Site so that incremental computation of statistics is possible. This prevents that update of KPI's require query of all productpackcollections (swarms of agents in relationships).
TransportationAgent which governs the relocation process of each channel, and allocates costs to SKUs as the "share costs of transport" when relocated.

In addition all agents have calendars, which determine their state in time and permit parsimonious usage of available CPU power. An example hereof is the usage of shifts in a factory, where the workforce is represented as a SKU that is used in a process. By having a calendar that notifies that the workforce only is available in daytime, the planning constraint becomes enforced, by all queries for the workers (SKU) at non-available hours, to be in vain.
Likewise bills-of-materials determine if certain SKUs that were used as input may be removed from the active pool of objects in order to pack them as Structs and reduce memory footprint. The ability to use calendars switch components on and off in the memory context may also be used to simulate future changes in, for example, infrastructure.

Architecture - Information exchanged
A key question for all communication processes concerns what actually needs to be exchanged?
Supply chain synchronization requires that information is exchanged about a sequence of events: The schedule of actions each of the interactors in the complex economy intends to perform. In absence of commitment to events is present, the forecast is used with acceptance that it may change at any time. Classical problems with communicating forecasts is that they are driven by each business budgeting process, in which "hope" of monthly or quarterly revenues are translated into aggregate transactions. However as transaction periods, such as retail sale days per month are not uniform (some months have 4 weeks whilst others have 5 weeks), the disaggregation of the forecast into transactional events is difficult. This means that forecasts must be generated based on simulations that generate data that imitates the transactional profile of the given period for which a forecast is present. When the operational level of granularity is present the non-occurrence of an expected event also becomes an update which may trigger a review of the assignment problem. As there are no technological problems in performing this creation of events, the next is the content of the schedule.
To cover a wide range of systems the research unit defined a benchmark, where total transactional volume exceeds any real-world client studied:

365 days * 100k SKUs * 3 transactions per SKU per location per day * 80 sites = 8,760,000,000 records.

Using transaction time as primary key the unique lookup value is based on a 92 bit sequence, which may hold every second of 365 days for 80 sites for 100k SKUs that are stored 's', received 'i' and dispatched 'o' the same second for the whole time period. The first authentication between the sending and receiving system transaction will describe start time, i.e. the first microsecond in the model, from which all events are calculated.

Any order or delivery results in the transaction "stock update" which must contain the following information:

Header	Format	Size
Date Time	Date time	32 bit
Site id	Integer	8 bit
SKU id	Integer	20 bit
Quantity	Integer	32 bit
State change direction	In, Out or Stored.	4 bit

Table 1. Required information for synchronisation

With this 12 byte record the data volume aggregates to 105.12 Gb. This volume of data may appear relatively small to modern database systems, but for solvers of the assignment-problem this is truly huge problems.

Common practice may also propose to load data of this magnitude into enterprise grade database systems, but some caveats must be considered:
If the database induces random disk access of the otherwise sequentially accessible data it would reduce the performance with up to 150,000 times. Even though the costs of modern solid state discs (SSD) are decreasing, SSD only improves access times with about one order of magnitude. What could reduce performance even further is a database that normalises the data completely during import whereby subsequent join operations are required to get the data out of the database. This would lead to need for additional memory and CPU power, which forces the access speed to perform random memory access operations, which are a factor of 10 lower than the sequential access to magnetic disk [11].
By duplicating the data on disk for sequential access with the key-sequence: (Site → SKU → Datetime → direction : quantity) a simple 33 step B-tree will permit random access, using distinct location pointers for Sites, SKUs and Datetime. The duplicate of the dataset would therefore permit O(n) for sequential access for time-periods, and O(log(n)) random access [9].

Architecture - Connecting systems

As users of the system may need to intervene at short notice a segmentation of the network that mirrors the geographical segmentation is beneficial as information updates will occur in the part of the system which will need to incorporate them first. Processes which have little influence on each other, for example due to different topology, may also be segmented effectively. This emphasises co-location of systems that require higher levels of communication whilst a synchronizing entity (like Smart Supply Chain, Figure 1) should connect to its own category, acting as a gateway for subsystems. To avoid additional overhead for this operation a pre-authenticated network is preferred.
The distributed architecture, in which other installations will have to be made, requires authentication protocols which ensure the integrity of the information of both sender and receiver. However as the volume of data which needs to be exchanged is substantial and requires quick data exchange; the only feasible model is peer-to-peer. To be able to manage this in practice, a combination of authentication and transmission systems is required. A well proven model is Skype™ and Cisco®'s centralized service that uses 2048-bit RSA/DSA for authentication and P2P data-exchange using 256-bit encryption.

VII. NEXT STEPS

A wide range of experience has been gained with deployment of present agent-based systems:

- Warehousing (Smart WMS)
- Job-shop scheduling (Smart Factory)
- Truck-load scheduling (Smart Truck)
- Rail-network scheduling (Smart Railway)
- Supply chain simulator & synchronizer (Smart SCSS)

These multi-agent systems are engineered for real-time operation, but may also be deployed for the purpose of simulating improvements through removal of constraints. Instead of using real-time information and replicating the whole system a more financially viable alternative is to perform the same investigation using historical data and accept non-real-time run-time using processing strategies which simulate the behaviour of the distributed real-time cluster[1]. An example of performing such simulation with measurement of the impact of removal of constraints was done in [7].

[1] How this is done is subject for another paper.

Once the potential opportunities for improvement have been identified through simulation, the investigation of the costs of seizing each opportunity will become a subject for external analysis and determine which of the potential interventions will be the most productive.

Experience from the development of these systems reveal that the exercise of identifying opportunities for reduction of delay, becomes similar by concept as run-time profiling, performed in computer science. The key difference is that the analysis uses delay in the supply chain as metric instead. This observation points towards future research of a supply chain synthesiser which analytically identifies opportunities in the supply chain and autonomously requests missing information from business analysts and search engines. Such a "supply chain synthesiser" would permit continuous revision of the longer commitment horizons, that are commonly referred to as strategic decisions and permit executives to extend their analytical abilities to make even more productive interventions through pre-emptive analysis of the complex economic system.

VIII SUMMARY

Factors that delay propagation of information are focal for supply chain synchronization. The system described in this paper is designed to permit a business to imitate the complex economic system it is a part of, and to operate with delayed commitment to respond adaptively to changed information obtained from the economy. This is done using a scale free combination of 3 generic processes (transformation, relocation and storage) in a network of channels and sites, in which agents pursue solution of the time-variant dual-assignment problem, of matching resources to capacities & supplies in order to minimize costs, whilst matching orders to supplies & capacities in order to maximize revenue. A compact format for data exchange was presented and suitable principles for connection of systems were highlighted. Finally methods for validation of benefits of real-time systems were presented so that the business may investigate which interventions permit delay of commitment of resources until the last possible moment, whilst also exploiting the resources already invested.

REFERENCES:

[1] J. Gattorna, *Living Supply Chains: How to Mobilize the Enterprise Around Delivering What Your Customers Want*. Pearson Education, 2006, p. 337.

[2] J. Kleinberg and D. Easley, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010, p. 744.

[3] P. A. Laplante and S. J. Ovaska, *Real-Time Systems Design and Analysis: Tools for the Practitioner [Hardcover]*. Wiley-Blackwell; 4th Edition edition, 2011, p. 584.

[4] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Springer; 2nd Edition. edition, 2011, p. 376.

[5] A. Tversky and D. Kahneman, "Judgement under uncertainty: heuristics and biases," *Science*, vol. 185, no. 4157, pp. 1124–1131, 1982.

[6] G. Rzevski, "A Practical Methodology for Managing Complexity," *Complexity*, vol. 13, no. 1–2, pp. 38–56, 2011.

[7] B. Madsen, P. Skobelev, G. Rzevski, and A. Tsarev, "Real-Time Multi-agent Forecasting and Replenishment Solution for LEGOs Branded Retail Outlets," in *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2012, pp. 451–456.

[8] G. Rzevski, "Planning under conditions of uncertainty," *Complexity*.

[9] S. Arora and B. Barak, *Computational complexity: a modern approach*. Cambridge University Press, 2009, p. 579.

[10] E. D. Beinhocker, *The origin of wealth: evolution, complexity, and the radical remaking of ...* Random House Business Books, 2007, p. 526.

[11] A. Jacobs, "The Pathologies of Big Data," *Queue*, vol. 7, no. 6, p. 10, Jul. 2009.