2

A Synergistic Approach towards Autonomic Event Management in Supply Chains

Roy Oberhauser
Aalen University
Germany

1. Introduction

Supply Chains (SCs), due to their very nature and intent (e.g., embracing change in markets, products, manufacturing, partners, globalization) in conjunction with market pressures, will face ongoing challenges that are necessarily reflected on the Information Technology (IT) infrastructure used to manage and optimise their operations.

Supply Chain Management Software (SCMS) typically covers the various functional aspects of SCs, including integration technology. The result of the IT integrations is a form of an information supply chain, including computational representations of physical SC entities. For purposes of this chapter SCMS will be considered to incorporate any ERP solutions and/or IT infrastructure utilized to enable the information integration required to support the SCs. Current SC IT challenges include decision making, collaboration, and attaining qualities such as scalability, performance, integratability, correctness, and reliability in the face of the perpetual dynamics and increasing complexity of SCs.

To avoid disruptions to SCs, Supply Chain Event Management (SCEM) considers the set of possible event scenarios and plans solutions. Events can be either representations of real-world events or can be introduced as a side-effect of the Information Systems (IS) supporting the SC (IT events). SCs can achieve their goals for optimal management of operations only to the extent and degree that they manage and automate the necessary information flow, especially with regard to managing unexpected events. The effective handling of potentially disruptive events is vital to achieving the aforementioned qualities, yet the ongoing change (mirrored in the IT systems) in entities and the properties and relations thereof, necessarily limits the sufficiency and totality of predefined solutions. A synergistic approach that leverages various computing paradigms can provide improved SCEM solutions.

In the face of potentially disruptive SC and IT events (referred to as SCEs in this chapter), autonomic computing (AC), inspired by the human autonomic nervous system, with its stated goals of self-configuration, self-optimization, self-healing, and self-protection (also known as self-X), would appear to be a synergistic candidate for improving SCEM. While some properties defined for autonomic systems[1] may not be applicable to SCEM, others will be beneficial. A partial application of AC techniques to achieve improved reactive event

[1] http://www.research.ibm.com/autonomic/overview/elements.html

Source: Supply Chain, The Way to Flat Organisation, Book edited by: Yanfang Huo and Fu Jia,
ISBN 978-953-7619-35-0, pp. 404, December 2008, I-Tech, Vienna, Austria

management might be both practical and beneficial to SCEM. However, the changeability, heterogeneity, distribution, internationalization/localization, support, governance issues, and partner interdependencies in SCMS (both from an IT and linguistic/cultural viewpoint) makes SCEM and self-X attainment in SC and SCMS far more challenging compared to that of a self-contained rigid system.

Granular Computing (GC) is a paradigm that concerns itself with the processing of complex information entities called information granules, recognizing that at different abstraction levels of data, different relationships can be inferred (Pedrycz, 2001), (Bargiela et al., 2003), (Pedrycz et al., 2008). The meaning and impact of an SCE is also dependent on the granularity at which it is viewed, and other implications and trends may be detected at various abstraction levels.

To enable internationalization and decoupled SC partner agents to autonomically collaborate to address SCEs, it is imperative that the meaning for shared concepts be defined. The Semantic Web (SemWeb) adds machine-processable semantics to data (Berners-Lee et al., 2001). SemWeb computing (SWC) allows for greater and improved automation and integration of information in large information SCs due to its formal structuring of information, clearly defined meanings and properties of domain concepts, and standardized exchange formats. One of the issues facing SemWeb is the creation and adoption of standardized ontologies in OWL (Web Ontology Language) (McGuinness et al., 2004) for the various industry domains to precisely define the semantic meaning of the domain data – standardization is laborious and adoption is slow. However, to address both the challenge of SCEM to avoid disruptive impacts and the challenge of SCMS to achieve self-X and other qualities in a heterogeneous, changing, loosely-coupled and global environment, a transitional hybrid stage is proposed. A high-value event-specific subset tailored to the SCMS is tackled first that enables the collaborative involvement of partner agents (computing or human). In other words, if the partners have no agreement on a common meaning of an event, the concepts necessary to diagnose the indicative problem, and the meanings of the actions required in a solution, then the required collaborative and (partially to completely) automatable solutions for interdependent and non-trivial situations will continue to be elusive.

Additionally, to enable collaboration, partner exchangeability, and sharing across heterogeneous IT partner services and data, standardized access protocols for SCMS and SCEM is desirable if not essential. Service-oriented Computing (SOC), with its reliance on Web Services (WS), provides platform-neutral integration for arbitrary applications (Alonso et al., 2003).

Furthermore, Space-Based Computing (SBC) is a powerful paradigm for coordinating autonomous processes by accessing a distributed shared memory (called a tuple space) via messaging, thereby exhibiting linear scalability properties by minimizing shared resources. Tuple spaces implement a shared data repository of tuples (an ordered set of typed fields) that can be accessed concurrently in a loosely-coupled way based on the associative memory paradigm for parallel and distributed computing first presented by (Gelernter, 1985).

This chapter explores the potential for SCs that a synergistic approach to SCEM (SASCEM) that leverages various computing paradigms provides for improving the qualities of SCEM, especially with regard to approaching self-X properties and automation.

The rest of the chapter is organized as follows: Section 2 presents a review of the literature. In Section 3 the solution approach is presented. Section 4 presents initial implementation work based on the solution approach. In Section 5 preliminary results which evaluated certain performance and scalability characteristics are discussed, followed by a conclusion.

2. Literature review

(Mischra et al., 2003) describes an agent-based decision support system for a refinery SC, where agents collaborate to create a holistic strategy using heuristic rules. (Bansal et al., 2005) present a model-based framework for disruption management in SCs, generalizing the approach of (Mischra et al., 2003).

Related to SCs, Value-Added Networks (VANs) are hosted service offerings that add value to common networks by acting as an intermediary between business partners for sharing proprietary or standards-based data via shared business processes. As such they can be viewed as supporting informational SCs. Work on modelling collaborative decision making in VANs includes MOFIS (Naciri et al., 2008) and could be applied to improving SCEM, e.g., via integration of the concepts in a SASCEM.

Complex Event Processing (CEP) (Luckham, 2002) is a concept to deal with meaningful event detection and processing using pattern detection, event correlation, and other techniques to detect complex events from simpler events. Besides the research work that considers various aspects of CEP (e.g., high volume, continuous queries), commercial products include the TIBCO Complex Event Processing Suite.

The Resource Event Agent (REA) model aims at providing a basic generic shared data model that can describe economic phenomena of several different systems, both within and between enterprises of many different types (McCarthy, 1982). Work includes (Haugen et al., 2000) who present a semantic model for SC collaboration, (Hessellund, 2006) discusses SC modelling extensions to REA, while (Jaquet et al., 2007) presents a semantic framework for an event-driven operationalization and extension of the REA model that preserves flexibility and heterogeneity. An extended REA approach and hybrid/partial semantic formalization of events are congruent with a SASCEM.

Multi-Agent Systems (MAS) have been researched extensively, as has MAS in combination with SCs. Agent-based event management approaches includes Sense, Think & Act (ST&A), which exhibits function-driven, goal-driven (local goals), and collaborative goal-driven (global goals) behaviours (Forget et al, 2006). Agent-oriented supply-chain management is explored in (Fox et al., 2000) among others. (Adla, 2008) proposes an integrated deliberative and reactive architecture for SCM for supporting group decision making. Although this work has typically not utilized SOC and SWC, enabling and leveraging the integration of such problem-solving approaches is one goal of a SASCEM.

Work on semantic enhancement of tuple spaces includes sTuples (Khushraj et al., 2004), which extends the object-oriented JavaSpace implementation (Freeman et al., 1999) with an object field of type DAML-OIL Individual. (Tolksdorf et al., 2005) and (Tolksdorf et al., 2005a) describe work on Semantic Tuple Spaces. The Triple Space Computing (TSC) project[2] aims to develop a communication and coordination framework for the WSMX Semantic Web Service platform (Bussler et al., 2005) (Simperl, 2007). However, there has been insufficient exploration of the application of semantically-enhanced tuple spaces for collaborative event-based problem solving in general, and for SCEM in particular.

[2] http://tsc.deri.at

With regard to partner communication interoperability, the issue of scalable server-side push notification protocol over HTTP for Space-based Computing (SBC) is explored in (Kahn et al., 2007) but lacks standardization. Agent-interoperability via Web Services has been explored, e.g., JADE WSIG (Greenwood, 2005), but its application to SCs is still hampered due to a lack of standardization, e.g., by FIFA (Greenwood et al., 2007).

3. Solution

To achieve improved and more holistic solutions for SCEs while exhibiting AC and other expected qualities, the SASCEM is a synthesis of various areas of computing, specifically granular (GC), semantic web (SWC), service-oriented (SOC), space-based (SBC), event-based (EBC), context-aware (CAC), multi-agent (MAC), and autonomic computing (AC) as shown in (Fig. 1).

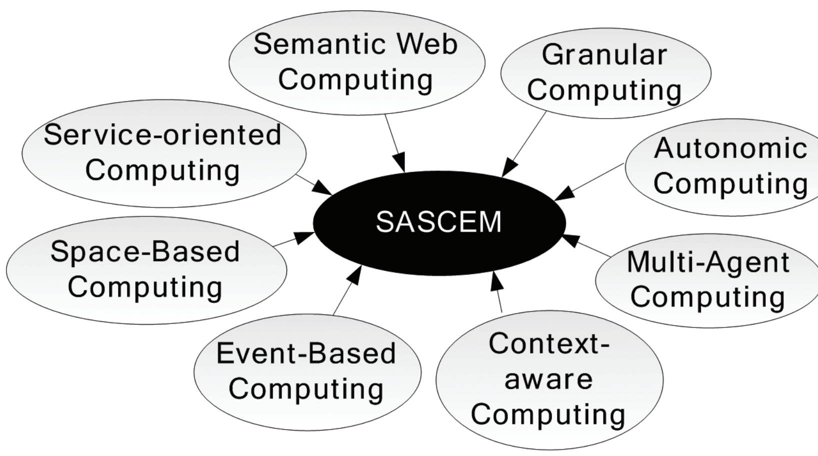


Fig. 1. Synergistic Solution Approach to SCEM

Solution constraints include heterogeneity, e.g., in partner agent implementations, rule-based techniques, platform software, and the adaptive and dynamic specialization of problem-solving for SCs. Additionally, it is assumed that for non-trivial SCs, no complete autonomic problem-solving for SCEM is as yet practical, thus the involvement of humans to the necessary degree is subsumed.

Principles that guided the solution approach include shared-nothing, decentralization, loose-coupling, standards-based communication, exchangeability (e.g, of collaborative decision making agent techniques), and enabling hybrid subsets for practical collaborative problem solving in SCs.

A simplified distributed SC solution infrastructure is shown in (Fig. 2). Using the SBC paradigm, tuple spaces are used to store event and event-relevant data, without deciding on meaning. Separate Semantic Web-aware tuple spaces are then used for collaboration on event diagnosis, problem prescription, and prognosis. Proactions or reactions are then initiated by partner agents and may involve the invocation of Partner or Infrastructure Services. Infrastructure Services and Partner Services provide the integration and access to SC (partner) functionality in accordance with the SOC paradigm. Heterogeneous interoperability and accessibility is supported via standards-based Web Services protocols, such as SOAP and REST (zur Muehlen et al., 2005). While an Enterprise Service Bus (ESB) is possible, its use depends on the SCMS and SCEM needs. In place of WS, Semantic Web Services (SWS), which envisions enabling automatic and dynamic interaction between software systems (Studer et al., 2007), might be a consideration; however, since the data repository can be readily accessed using simpler WS interfaces, a pragmatic approach utilizing the minimal amount of SemWeb to the extent needed to enable partner collaboration is currently preferable until SWS maturity and adoption has progressed.

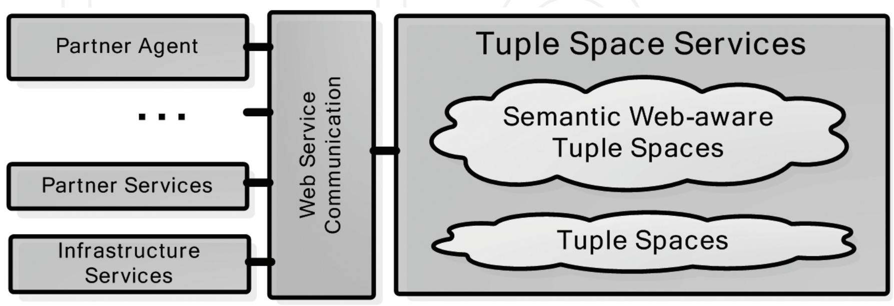


Fig. 2. Solution Infrastructure of the SASCEM (simplified)

The details of the solution approach will follow the event process steps shown in (Fig. 3).

Fig. 3. Event Process Steps in the SASCEM

3.1 Event acquisition

The acquisition of SCEs can come from sensors, partner machines and IT systems or services, and other event producers. In accord with EBC, the functionality of SCEM is triggered and invoked in response to the generation of SCEs. The events can be simple events to complex events inferred from simpler events, as considered in CEP. To enable the advantages of GC, these SCEs should be retained in their original state and supplementary complex events generated when these are detected via pattern matching or other CEP techniques by partner agents or other components. CEP and GC can be incorporated in (Partner or Infrastructure) Services or Agents.

3.2 Event storage

The event data is stored as a tuple in a tuple space following the SBC paradigm. This allows decoupled partner agents to flexibly subscribe to and be notified of relevant events. The tuple can be retrieved over time by various partner agents.

The data model is a hybrid that keeps data-only SCE tuples separate from the SemWeb tuple space. The SASCEM uses a hybrid transitional approach of communication between agents, supporting a blend of SemWeb and other data exchange in the tuple spaces. This allows the original event data to be viewed at different times, at different granularity levels, and to have multiple and even contradictory interpretations by diverse partner agents.

Registration for notifications by partner agents can be based on event data arrival, event data changes, etc., independent of semantic events. Thus partner agents without semantic awareness but, e.g., with viable event handling rules and heuristics, can participate and support SCEM. Those partner agents with SemWeb capabilities can collaborate in the SemWeb tuple space and create and adjust the semantic meaning of the event data, type, attributes, and relations at different levels of abstraction and perhaps in different ontologies. This includes analysis and processing with regard to the event's relation to a problem (if any), diagnosis, prognosis, prescription, actions, etc. necessary to resolve it.

3.3 Contextual annotation

Contextual annotation of the event supports the retrieval of relevant data close to the occurrence of the event, and helps to determine its meaning and implications as well as infer complex events. As events are diagnosed over time, it may be determined by partner agents that certain information which is applicable and relevant should be gathered and other information may be determined to be irrelevant. CAC is thus utilized to annotate contextual and environmental information with the event, and those services registered for the event are notified. If no RDF(S)[3] (Brickley et al., 2004) information is provided with the event, then this too could be annotated to provide a uniform way of describing information resources associated with the event.

3.4 Event diagnosis

The correct diagnosis of SCEs is dependent on appropriate knowledge and rules, and due to the partner interdependency of SCs, collaborative effort to achieve AC is necessary. Diagnostic MAC enables the various partner agents to specialize in their particular knowledge without the limitations that a centralized single agent would incur. In order for heterogeneous partner agents to collaborate to achieve (semi-)autonomic behaviour, SWC is utilized to allow for a standardized and extensible approach for giving meaning to the events. A SemWeb-enabled tuple space (SWETS) provides a shared data storage where the meaning of the data types is defined and collaborative event analysis and interpretation is thus enabled. SemWeb-aware agents using inference engines can collaborate at various abstraction levels using GC paradigms. Complex events can be inferred from simple events, e.g., regarding their timing, sequence, patterns, or trends, and CEP could be utilized. If the collaborative diagnosis relates the event to a(n) (unknown) problem, processing continues, otherwise it is completed. Multiple and even contradicting diagnoses are allowed and may occur. Note that this situation may in turn create a new event which in turn goes through the processing steps.

Ontologies are minimally necessary for the intersection set of concepts necessary for SCEM between partner agents. In this regard, full ontologies that cover all possible concepts in the SC can - but must not necessarily, be avoided. A partial application of SemWeb appears practical and reasonable at this time, given some current practical limitations with regard to payoff vs. effort, standardization, maturity, industrial usage, training, tooling, etc. Yet the intersection of concepts between partners requires a formal definition and agreement in order for collaborative and automated SCEN to be enabled.

---

[3] Resource Description Framework (Schema)

While agents are often considered to be artificial computational entities that perform tasks with a degree of autonomy, in the SASCEM agents include the set of human agents as well for problem solving, supporting a hybrid spectrum from completely manual to automatable diagnosis and solutions, since each SC is unique and for non-trivial dynamic SCs new events and problems may occur that require human intervention before they become automatable.

3.5 Problem prescription

Using the SWETS, the agents, based on the possible diagnoses, collaboratively decide on a prescription consisting of a set of actions, e.g. using (Adla, 2008) or other decision techniques, and incorporating AC techniques where applicable.

3.6 Problem prognosis

Separately from the prescription, the forecasted impact, side-effects, and success chances of the diagnosis and/or the prescription in the form of a prognosis could optionally be (collaboratively) determined and placed in the SWETS, perhaps triggering new events.

3.7 Proactions and reactions

Based on the prescription and/or prognosis, the reactions are executed by the appropriate agent(s), using partner or infrastructure services as needed, and preventative proactions can be executed to limit the impact of side effects, repeated problems, etc.

4. Solution implementation

The prototype implementation of the SASCEM currently includes an adaptation of an open source tuple space implementation (XSpace[4]). Hybrid support for SWETS is currently dependent on the outcome of a tsc++[5] evaluation and integration. Apache Axis2[6], which supports asynchronous WS, was used for WS communication.

To illustrate the SASCEM implementation and for prototype testing purposes, an ontology (Fig. 4) for a software SC was created using Protege 3.3.1. First it will be described in prose, followed by OWL abstract syntax. Work on SCM ontologies includes (Haller et al., 2008).

BusinessObjects can depend on other BusinessObjects and have Suppliers, Consumers, and Producers. A Service is a BusinessObject with a Protocol, including human and organizational services, and can be specialized as a WS or a SWS.

Products and Information are Artifacts, which are BusinessObjects. Systems, Hardware, and Software are Products and Products may have a Configuration. A Patch is Software. A Document is Information.

Events may refer to one or more BusinessObjects and be associated with one or more Problems. Problems refer to a Quality that is affected, may include a Diagnosis and a Prognosis. A Diagnosis may include a Prescription that may refer to a set of Actions and may refer to a Patch and/or Configuration.

---

[4] http://xspacedb.sourceforge.net/
[5] http://tsc.sti2.at/
[6] http://ws.apache.org/axis2/

Fig. 4. Partial Software Supply Chain Event Management Domain Ontology

An alphabetical listing in OWL abstract syntax follows (Listing 1):

```
Class(Action partial owl:Thing)
Class(Artifact partial BusinessObject)
Class(BusinessObject partial restriction(hasEvent minCardinality(0))
         owl:Thing
         restriction(version cardinality(1))
         restriction(name cardinality(1))
         restriction(depends minCardinality(0))
         restriction(hasConsumer minCardinality(0))
         restriction(hasProducer minCardinality(1))
         restriction(hasSupplier minCardinality(0)))
Class(Configuration partial owl:Thing)
Class(Consumer partial restriction(hasBusinessObject minCardinality(1))
       owl:Thing
       restriction(name cardinality(1))
       restriction(homepage cardinality(1)))
```

```
Class(Diagnosis partial restriction(hasPrescription minCardinality(0))
       owl:Thing)
Class(Document partial Information)
Class(Event partial restriction(hasProblem minCardinality(0))
       restriction(hasBusinessObject minCardinality(0))
       owl:Thing)
Class(Format partial owl:Thing)
Class(Hardware partial Product)
Class(Information partial Artifact
       restriction(hasFormat cardinality(1)))
Class(Patch partial Software)
Class(Prescription partial restriction(hasPatch maxCardinality(1))
       restriction(hasAction minCardinality(0))
       restriction(description cardinality(1))
       owl:Thing)
Class(Problem partial restriction(hasEvent minCardinality(0))
       owl:Thing
       restriction(description cardinality(1))
       restriction(hasImpact minCardinality(0))
       restriction(hasSolution minCardinality(0))
       restriction(hasQuality minCardinality(0)))
Class(Product partial restriction(hasConfiguration minCardinality(0))
       Artifact)
Class(Prognosis partial owl:Thing
       restriction(description cardinality(1)))
Class(Protocol partial owl:Thing)
Class(Quality complete oneOf(Functionality
       Reliability
       Usability
       Efficiency
       Maintainability
       Portability))
       SubClassOf(Quality owl:Thing)
Class(SemanticWebService partial WebService)
Class(Service partial restriction(hasProtocol cardinality(1))
       BusinessObject)
Class(Service partial restriction(hasProtocol cardinality(1))
       BusinessObject)
Class(Software partial Product)
Class(Supplier partial owl:Thing
       restriction(name cardinality(1))
       restriction(homepage cardinality(1)))
Class(System partial Product)
Class(WebService partial Service)
```

Listing 1. Partial Software Supply Chain Event Management Domain Ontology

5. Results

Preliminary results considered the viability of the solution architecture and prototype implementation used for this peer-based middleware combination of a tuple space, relational database, message broker, and asynchronous Web Services infrastructure for addressing the SC qualities in scalability on a per-agent and a system level before integrating true SemWeb-aware problem-solving agents. For this, two key throughput scenarios were measured consisting of the event message into the tuple space (put scenario) and the notify scenario to other agents (notify scenario).

The test configuration consisted of 2,4 GHz Dual Core Opteron 180 PCs running Windows XP Pro SP2, 3.3GB RAM, 100 Mbit LAN, JRE 1.6.0_07, and Apache Axis2 0.93. One server PC ran Xspace 1.1, Jboss 4.0.3, and HSQLDB 1.8.0. The averages over three runs were used for all results (Fig. 5).
For the notify scenario, 1000 SOAP messages containing an event to put into the tuple space were sent from a single producer PC to the server, with a Message-Driven Bean, upon receiving the put, notifying agents (via asynchronous SOAP messages) on either 1, 2, 4, or 8 consumer PCs. Note that all the throughput results exclude and ignore the server and the producer PCs, but only consider the notification throughput on the consumers. The results show that asynchronous notifications by the tuple service to 1 to 2 and 4 peers regarding the put allowed an almost linear scalability, with a reduction at 8 peers due to full CPU utilization on the server.
For the put scenario, 1000 SOAP messages containing an event to put into the tuple space were sent from either 1, 2, 4, or 8 producer PCs to the same tuple space on the server PC. The results show a significant reduction in cumulative throughput with each added peer, which can be explained by the transactional bottleneck of the puts to the relational database on the server. These results and storage options, including persistence requirements on a per tuple basis, will be taken into account and optimization opportunities considered in future work.

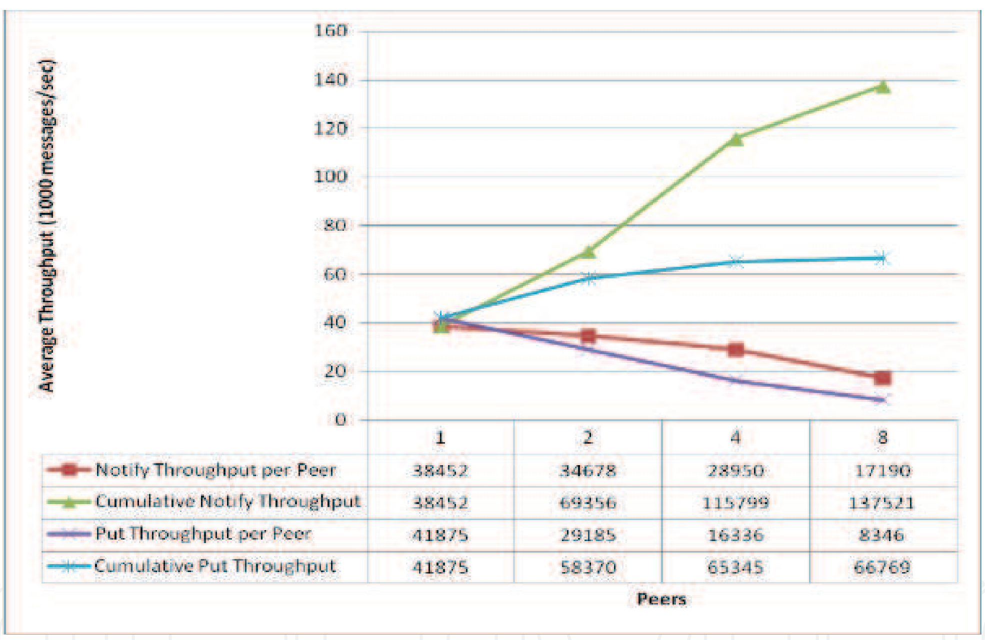


Fig. 5. Average throughput vs. number of peers for web service notifications

Since for SASCEM the number of notifications is expected to be much higher than the number of generated events, the nearly linear scalability for notifications show that the SBC and EBC foundation for SASCEM is viable for SCEM.

6. Conclusion

The increasing reliance on SCs, coupled with increasing complexity, dynamism and heightened quality expectations, are necessarily reflected in the SCMS and implicitly in the need for improved SCEM to limit disruptions and achieve self-X qualitites. A novel synergistic approach to SCEM, as presented in this chapter (SASCEM), leverages the computing paradigms of granular, semantic web, service-oriented, space-based, event-based, context-aware, multi-agent, and autonomic computing to create a holistic solution approach that can change how SCEM is approached. Within the SASCEM, the hybrid approach to SWC makes adoption practical and viable in the near term. Preliminary results show sufficient performance and scalability qualities for such an SBC infrastructure to address SCEM.

The scope of applicability for this approach goes beyond SCEM, and could be applied to event management in general outside of SCs. Moreover, SCMS might be architected differently where a SASCEM adopted.

Future work includes integrating SemWeb-based problem-solving agents with Semantic Web-aware tuple spaces and evaluating the solution with regard to real-world problem-solving scenarios.

7. Acknowledgements

Thanks to Tobias Gaisbauer for his assistance with the experiments and implementation.

8. References

Adla, A. (2008). Agent-Based Group Decision Making, *Proceedings of The Third International Conference on Software Engineering Advances (ICSEA 2008)*, pp.146-151, ISBN 978-0-7695-3372-8, Sliema, Malta, October 2008, IEEE Computer Society

Alonso, G.; Casati, F.; Kuno, H. & Machiraju, V. (2003). *Web Services – Concepts, Architectures and Applications*, Springer Verlag, Berlin

Bansal, M.; Adhitya, A.; Srinivasan, R. & Karimi, I.A. (2005). An Online Decision Support Framework for Managing Abnormal Supply Chain Events, 38th European Symposium of the Working Party on Computer Aided Process Engineering: ESCAPE-15, 29 May - 1 June 2005, Barcelona, Spain, Elsevier Science B.V., ISBN 0444519912

Bargiela, A. & Pedrycz, W. (2003). *Granular Computing: An Introduction,* Dordercht: Kluwer Academic Publishers

Berners-Lee, T.; Hendler, J. & Lassila, O. (2001). The Semantic Web, *The Scientific American*, May 2001.

Brickley, D. & Guha, R.V. (2004). RDF Vocabulary Description Language 1.0, RDF Schema, W3C Recommendation, February 2004, http://www.w3c.org/TR/rdf-schema/

Bussler, C.; Kilgarriff, E.; Krummenacher, R.; Martin-Recuerda, F.; Toma, I. & and Sapkota, B. (2005) WSMX Triple-Space Computing, http://wsmo.org/TR/d21/v0.1, June 2005, D21 v0.1

Forget, P. ; D'Amours, S. & Frayret, J.-M. (2006). Collaborative event mangement in supply chains: an agent-based approach, *Information Technology for Balanced Manufacturing Systems*, volume 220, *IFIP International Federation for Information Processing*, 2006, Springer, Boston, USA, pp. 89–98

Fox, M. S.; Barbuceanu M. & Teigen, R. (2000). Agent-oriented supply-chain management. *International Journal of Flexible Manufacturing Systems*, 12(2-3), pp. 165–188

Freeman, E.; Hupfer, S. & Arnold, K. (1999). *JavaSpaces Principles, Patterns and Practice*, The Jini Technology Series, Addison-Wesley, Reading MA, USA.

Gelernter, D. (1985). Generative communication in Linda, *ACM Transactions on Programming Languages and Systems*, volume 7, number 1, January 1985, pp. 80-112

Greenwood, D. (2005). JADE Web Service Integration Gateway (WSIG), JADE Tutorial, AAMAS 2005

Greenwood, D.; Lyell, M.; Mallya, A. & Suguri, H. (2007). The IEEE FIPA approach to integrating software agents and web services, *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, Honolulu, Hawaii, 2007, Article No. 276, ISBN:978-81-904262-7-5

Haller A.; Gontarczyk, J. & Kotinurmi P. (2008). Towards a complete SCM ontology: the case of ontologising RosettaNet, *Proceedings of the 23rd Annual ACM Symposium on Applied Computing*, ACM, Fortaleza, Ceará, Brazil, 2008, pp. 1467-1473

Hessellund, A. (2006). Supply Chain Modeling with REA. *TR 2006-80*, 2006, The IT University of Copenhagen, Denmark

Jaquet, M. & Østerbye, K. (2007). A Property Driven Approach Towards Creating an Adaptive REA Framework, *Proceedings of the REA-25 Conference*, June 2007, Delaware, USA

Kahn, E.; Riemer, J. & Lechner, L. (2007). XVSMP/Bayeux: A Protocol for Scalable Space Based Computing in the Web, *Proceedings of the 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (WETICE 2007), pp. 68-73

Khushraj, D.; Lassila, O. & Finin, T. (2004). "sTuples: Semantic Tuple Spaces," First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), pp.268-277

Luckham, D. (2002). *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, Addison-Wesley Professional, ISBN 0-201-72789-7

McCarthy, E. W. (1982). The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. *The Accounting Review*, July 1982, pp. 554-78

McGuinness, D. L. & van Harmelen, F. (2004). OWL Web Ontology Language: Overview, W3C Recommendation, February 2004, http://www.w3c.org/TR/owl-features/

Naciri, S.; Yoo, M. & Glardon, R. (2008). MOFIS: new conceptual modeling framework for handling value adding networks complexity, *Proceedings of The Third International Conference on Software Engineering Advances (ICSEA 2008)*, pp.516-522, ISBN 978-0-7695-3372-8, Sliema, Malta, October 2008, IEEE Computer Society

Pedrycz, W. (2001). *Granular Computing: An Emerging Paradigm*, Heidelberg: Physica-Verlag

Pedrycz, W.; Skowron, A. & Kreinovich, V. (2008). *Handbook of Granular Computing*, Wiley-Interscience, ISBN: 978-0-470-03554-2

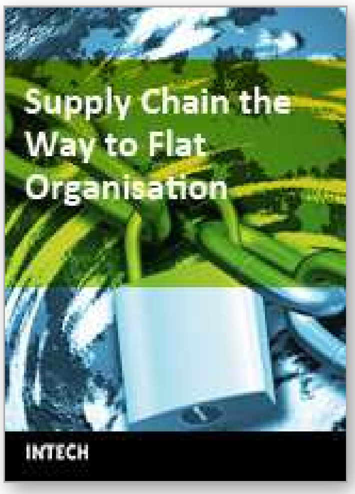
Simperl, E.; Krummenacher, R. & Nixon, L. (2007). A coordination model for triplespace computing, *Proceedings of the 9th International Conference on Coordination Models and Languages (Coordination)*, Springer Verlag, June 2007

Studer, R.; Grimm, S. & Abecker, A. (2007) *Semantic Web Services: Concepts, Technologies, and Applications*, Springer, ISBN-10: 3540708936

Tolksdorf, R.; Bontas, E. P. & Nixon, L. J. (2005). Towards a Tuplespace-Based Middleware for the Semantic Web, *Proceedings of the 2005 IEEE/WIC/ACM international Conference on Web intelligence*, (September 19 - 22, 2005). Web Intelligence. IEEE Computer Society, Washington, DC, pp. 338-344

Tolksdorf, R.; Nixon, L.; Bontas, E. P.; Nguyen, D. M. & Liebsch, F. (2005a). Enabling real world Semantic Web applications through a coordination middleware, *Proceedings of the 2nd European Semantic Web Conf.* ESWC'05, 2005

zur Muehlen, M.; Nickerson, J. V. & Swenson, K. D. (2005). Developing Web Services Choreography Standards - The Case of REST vs. SOAP, *Decision Support Systems*, Volume 40 , Issue 1, July 2005, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, ISSN:0167-9236



Supply Chain the Way to Flat Organisation
Edited by Julio Ponce and Adem Karahoca

ISBN 978-953-7619-35-0
Hard cover, 436 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

With the ever-increasing levels of volatility in demand and more and more turbulent market conditions, there is a growing acceptance that individual businesses can no longer compete as stand-alone entities but rather as supply chains. Supply chain management (SCM) has been both an emergent field of practice and an academic domain to help firms satisfy customer needs more responsively with improved quality, reduction cost and higher flexibility. This book discusses some of the latest development and findings addressing a number of key areas of aspect of supply chain management, including the application and development ICT and the RFID technique in SCM, SCM modeling and control, and number of emerging trends and issues.