Agent-based Transactional Framework for the Supply Chain

R. Anane[1], M. Younas[1], C-F Tsai[2], K-M. Chao[1],
[1]*School of Mathematical and Information Sciences, Coventry University, UK*
{r.anane, m.younas, k.chao}@coventry.ac.uk
[2]*Department of Industrial Management, Aletheia University, Taiwan*
Tsai@email.au.edu.tw

Abstract

We propose a framework for supporting manufacturing activities as part of the supply chain management. The manufacturing process is seen as a network of processing nodes where products are either transformed or assembled. Each node is endowed with some autonomy that enables it to enhance the agility and the 'fluidity' of the manufacturing process. Nodes are able to search for the best deals on the market and to respond dynamically to changes in internal or external conditions. This requires transactional knowledge as well as some knowledge of the supply chain (embodied into the degree of availability at a particular node). This framework is modelled by a combination of agent technology and extended transaction models in order to facilitate dynamic activities in supply chain.

1. Introduction

In their attempt to optimise manufacturing processes and increase competitiveness, organisations are focussing their efforts in the reduction of a number of factors: product development time, production costs, production lead times, etc. This concern has led to innovative approaches to supply chain management in general. Specifically, organisations are faced with the challenge of establishing and maintaining efficient material flows along product supply chains [1]. Current trends in the supply chain management aim to: increase responsiveness to market changes, ensure a wide base of supply of material, increase the fluidity of the supply chain and enhance the agility of the manufacturing systems [2, 3].

In this paper we propose an agent-based transactional framework for the construction of dynamic models where supply chain entities are able to play autonomous roles in the identification and selection of the best options available at a particular processing point on the chain. In the framework there is no pre-determined relationship between the supply chain entities. For instance, when an order comes, a virtual supply chain may emerge through the negotiation of agents (representing supply chain entities).

In the proposed framework, supply chain entities are represented by software agents, which are autonomous, and proactive entities that facilitate the introduction of decentralised, emergent and concurrent manufacturing systems. Agents make use of their knowledge of the supply chain in order to tailor the granularity of the transactions to the needs of manufacturing systems.

Transactions (originated from database technology [4]) can play a crucial role in the management of supply chain activities. For instance, supply chain activities need to automatically tailor products and services to fulfil customer's needs. These activities are required to perform several operations, which are logically linked together such that either all of them are executed or none of them is executed. Further, the concurrent execution of these activities is required to be co-ordinated so that they can efficiently and consistently use common system resources. Transactions enforce certain correctness criteria to ensure the correct, consistent, and reliable execution of these activities. The classical criteria, called the ACID (Atomic, Consistent, Isolated, and Durable) criteria, have widely used in database transactions [4]. However, ACID criteria are too inflexible to accommodate the nature of the current supply chain management. We therefore propose the incorporation of extended transactions [5, 4] that relax ACID criteria and enable awareness between different transactions. These transactions integrate well with agent technology in enhancing the agility and 'fluidity' of manufacturing systems.

The paper is structured as follow. Section 2 provides the context for the supply chain and issues associated with it. Section 3 presents the proposed framework, and Section 4 describes agent-based transaction management. Section 5 describes related work, and Section 6 concludes the paper.

2. Supply Chain Context

A good definition of the supply chain is given in [6]: "The supply chain of a manufacturing enterprise is a world-wide network of suppliers, factories, warehouses, distribution centres and retailers through which raw material are acquired, transformed and delivered to customers".

The wider perspective on the supply chain leads to the consideration of a number of issues that can be represented by transactional knowledge and knowledge of the supply chain [7]:

- *Products*: with the proliferation of products as well as of the manufacturers, locating suppliers with needed products is a substantial challenge, especially since the specification of a part can be very detailed.
- *Complexity*: complex and flexible set-ups are required to satisfy demand for customised products and services.
- *Transaction management:* Once suitable products are located, transactions can be initiated. This involves negotiation and may cover price, quantity, etc.
- *Co-ordination:* with potentially geographically dispersed participants, co-ordination of activities becomes a crucial function in the supply chain. The task of buying parts can be centralised or distributed where each processing node negotiate/co-operate with other nodes.
- *Knowledge acquisition and use in the supply chain:* The need for knowledge that goes beyond transactions is required in order to make the supply chain responsive.

In the supply chain, the production system plays a crucial role. Raw material and components are either transformed or assembled at processing nodes. These processing nodes correspond to a network that represents the dependency between components and the requirements for sub-components from components at each processing node.

Traditional methods like Just-in-Time (JIT) approach to production is aimed at eliminating waste and making optimal use of manufacturing processes, and also integrates manufacturing and suppliers into a production line. The JIT approach is demand-driven but it suffers from a number of weaknesses that include:

- the relationship between the nodes of supply chain network is static
- the chain is closed and driven from one end and control is more or less centralised
- it assumes complete knowledge of and control of the process and reliable supply
- the different nodes in the network are passive
- static allocation of resources is the norm
- transactions are to some extent fixed and centralised

3. The Proposed Framework

First, we present architecture of the framework (Section 3.1), and then describe agents' functionality (Section 3.2). Section 3.3 devises different formulas to determine the degree of availability in the supply chain.

3.1 The Architecture

The proposed framework is presented at a higher level of abstraction through the generalised architecture shown in Figure 1. This integrates multi-agents and extended transactions in the supply chain environment. Multi-agents provide a framework for decentralised, emergent (rather than planned) and concurrent systems.

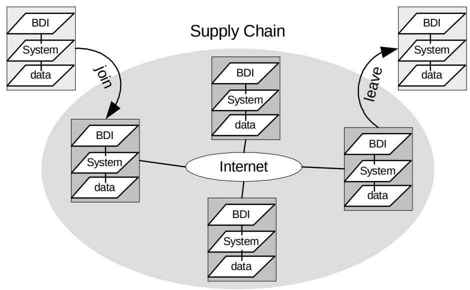

Figure 1. Architecture of the Proposed Framework

In Figure 1, shaded area represents the supply chain in which multiple agents communicate with each other via Internet. These agents also coordinate with each other to execute transactions of a supply chain order. Each agent acts as a proxy for company and represents a processing node in the supply chain. Further, agents can dynamically join and leave the supply chain depending on the prevailing conditions. Moreover, in the proposed framework, there is no central or coordinating agent that controls the execution of transactions in the supply chain. Instead they apply a peer-to-peer approach to perform various activities such as transactions management. Our framework allows transactions to start at one node and may terminate at some other nodes.

3.2 Functionality of Agents

In the above architecture agents represent processing nodes of a supply chain network. Each node possesses some autonomy that enables it to enhance the agility and the 'fluidity' of the manufacturing process of a supply chain. Each node is able to search for the best deals on the market and to respond dynamically to changes in internal or external conditions. Processing nodes are therefore focal points of decision. The distributed nature of control over the supply chain is closely linked to knowledge of the supply chain. The proposed framework is capable of displaying the following properties:

- Ability to cope with incomplete knowledge
- Processing nodes are open and are active
- Control is distributed as each node is autonomous

- Processing nodes respond dynamically to changes in the environment. Thus situations can be re-evaluated.
- Each level may be opportunistic in exploiting new situation by building a portfolio of available resources.
- Supplies can be ordered from internal/external sources
- Entities may join the supply chain or leave it depending on their objectives.

At each processing node, a search for the best components is conducted. If this requires procuring components from different suppliers, many options may be available. In order to facilitate the selection of the best option, a measure of availability, called degree of availability, is associated with each component at the processing nodes. This degree of availability represents one crucial element of the knowledge of the supply chain. The degree of availability is determined by combining and weighting a number of factors:

- Importance of component to processing (% of assembly etc.)
- Level of complexity (number of steps required, sequential or parallel)
- existence of substitutes (how widely available is this component)
- level of genericity (can it support a family of components)
- supplier (reliability of supplier)
- tolerance (precision level)

This measure can be generated dynamically in response to new circumstances. Logically, we shall assume that each processing node takes one or more sub-components as input and output a single component, after either transformation or assembly. At each processing node, a number of operations may be performed:

- Negotiation with lower level and upper level processing nodes *(social ability)*
- Each component requires one or more sub-components
- Search for alternatives/substitutes *(proactiveness)*
- Determination of list options
- Determination of degree of availability based on component characteristics
- Selection of component (or components) with the highest degree of availability *(autonomy)*
- Reallocation of resources in response to changes *(reactivity)*

Agents use knowledge of the supply chain to effect transactional knowledge. Knowledge of the supply chain provides information about the sources of supply and the options available at a particular processing node. Each option has a degree of availability associated with it. Once options are selected and committed then transactions can be initiated.

3.3 Determination of the Degree of Availability

Agents incorporate a reasoning mechanism such as Belief, Desire and Intention (BDI) model — which is a procedural reasoning method that enables agents to deal with highly dynamic activities. An agent can be aware of its environment and can refer to its own goals to select appropriate plans and carry out related actions in order to reach a desired state. Thus, the BDI model provides agents with a mechanism for representing knowledge (beliefs) and for modelling its behaviour (semantics). The following descriptions are the main BDI procedures in pseudo code.

```
'Initialise-state
B←B0;  D←D0; I←I0
While ¬empty (I) ∨ Succeeded (D) ∨ Impossible (D)
  B← perceive (B, state) ;gathering the external and internal information
  D← option(B, I)        ;generating desires
  I← filter(B, D, I)     ; selecting intentions
  P←plan(B, I)           ; selecting a list of plans to fulfil the intention
  p←MAX(P)               ; selecting the highest utility plan p
  B←p(actions)           ;starting the action
End –while
```

In most agent applications, the process of filtering intentions is simplified by assuming that only one intention or none will be selected. The agent, however, does not have a generic decision making system for deciding in a situation where more than two different intentions or plans can possibly achieve the same desire. In order to overcome this difficulty, we propose a generic and novel decision making system that includes multiple attributes to enable the agent to generate the overall availability over multiple factors. This process consists in determining the individual degree of availability for each factor, and then summing up the weighted degrees of availability in order to produce the overall degree of availability. These values range over the interval [0,1].

Tolerance: If the component A requires high precision from other components such as B, C, D, then the degree of availability is low. $\overline{X}_B \pm 1\sigma$ = 73.27% for B, $\overline{Xc} \pm 2\sigma$ = 95.45%, $\overline{X}_D \pm 3\sigma$ = 99.73% and $\sigma = \frac{\sqrt{(Xi - \overline{X})^2}}{i}$. The tolerance level of component A is β_1 = 1-(73.27% + 95.45% + 99.73%)/3

Existence of Substitutes: If the component A can be replaced with component A1, A2, or A3 its degree of availability is relatively high. The more substitutes the higher the degree of availability. This can be formulated as $\beta_2 = 1-(1-X_{A1}) \times (1-X_{A2}) \times (1-X_{A3})$ where X_{Ai} represents the compatibility with component A.

Level of Genericity: If the component A is a generic part to a number of products, other suppliers could supply it. The degree of availability can be represented as $\beta_3 = 1-(1-X_1) \times (1-X_2) \times (1-X_3) \times (1-Xi)$; where $(1-Xi)$ represents the possible shortage of component A for supplier *i*. The more generic the higher the degree of availability.

Reliability of Supplier: The degree of availability of a supplier depends on his/her reliability (denoted as β_4). The evaluation of supplier A can be carried out through its historical records. If the supplier A falls within the range of $\overline{X}_A \pm 1\sigma$, his reliability is 73.27%. If there is no previous data, the default value, β_4, will be derived from $\overline{\beta}_4$. The value of his reliability is revised after each transaction. The higher the reliability the higher the degree of availability.

Importance of Component to Processing Time: The importance of a component A in terms of processing time can be evaluated as $\beta_5 = t/T$; where *t* and *T* denote required processing time for component A and total processing time for product P, respectively. The higher the percentage the lower the degree of availability.

Level of Complexity: Two factors such as vertical (*v*) and horizontal (*h*) determine the level of complexity of component A. The *V* is the number of processing steps and *H* is the number of dependent components in the same step. AR_A is the area of $(v \times h)/2$ that means the component A's dependence area. So, the significance of component A to the product P can be formulated as $\beta_6 = AR_A / AR$, and $AR = V \times H$ (Fig. 2). The higher the complexity the lower the degree of availability.

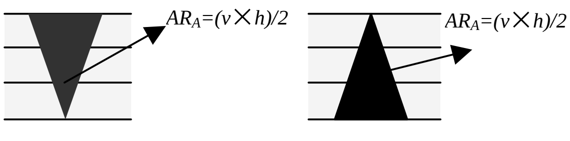

Figure 2. The degree of complexity

The overall degree of availability is determined as follows:

$$A = w_1 \times \beta_1 + w_2 \times \beta_2 + w_3 \times \beta_3 + ... + wk \times \beta_k$$

where w_k denotes the weights for factor *i* and $\sum_{i=1..k} wi = 1$.

The resulting value, A, represents the availability of the component A, and the higher its value the better. So, the agent can utilise this to evaluate each possible intentions or plans in order to select the best option.

4. Agent-based Transaction Management

Transaction is defined as a unit of work such that all the operations are successfully completed or none of them is completed if there is any failure. Classical transactions are characterised by the ACID properties [5, 4]. However, ACID remains inappropriate for the current supply chain activities. In particular their isolation and atomic policy does not suit the characteristics of supply chain, which are described earlier. We therefore propose extended transactions for the supply using multi-agent approach. Extended transactions in our framework include: split, join, open-nested and closed nested transactions. These transactions relax the atomicity and isolation properties of the ACID criteria [5]. The aim is to effectively handle the supply chain activities, which are cooperative, dynamic, long-lived, interactive, and non-prescriptive in nature. Extended transactions are briefly described as follow:

Split transaction model slits an active transaction, T_i, into multiple transactions, T_i and T_j, by delegating the actions of T_i to T_j (T_i is assumed to preserve its identity after the split). T_i and T_j can commit or abort independently. Join transaction model merges multiple independent transactions, T_i and T_j, into a single transaction, T_k.

Nested transaction allows transactions to be constituted from hierarchically structured subtransactions. But the commit or abort of the subtransactions is suspended until their parent or root transaction is committed or aborted. Open-nested transactions allow the unilateral commit of subtransactions. Further, we also incorporate alternative transactions so as to perform alternative activities.

In the proposed framework, agents process orders through the execution of transactions using different transaction models as described above. When an order is arrived a virtual supply chain is created in which multiple autonomous agents are used to process an order. An agent is called an *originating agent* (denoted $Agent_o$) if it starts the execution of a *main transaction* (denoted T_m) of a supply chain order. Other agents are called participating agents represented as $Agent_p$. All these agents ($Agent_o$ and $Agent_p$) are autonomous and there is no central or controlling agent that controls the execution of transactions.

Depending on the situations, agents execute transactions using different criteria. The following cases illustrate some of the possible criteria, which agents use in the execution of transactions of a supply chain order.

Case-1: *During the execution of a transaction Agent$_o$ splits T_m into independent transactions (T_i and T_j) which can be executed by Agent$_o$ and/or Agent$_p$.*

This criterion is used to model different situations. For instance, an order is made for acquiring two components A (CPU) and B (memory chip) at a particular processing node. These components are assumed to be supplied by the same supplier. An agent therefore executes a single transaction to purchase these components. However, it is possible that the supplier may fail to supply component B. Occurrence of this situation will trigger the search for another supplier. Thus it is required to split the original transaction into two separate transactions, one for A, and the other for B. Moreover, splitting a transaction is also required to deal with the situation where agents dynamically leave the supply chain, as they may not get the desired profit. Thus (sub)transactions belong to the withdrawing agents can be split and terminated independently without causing the termination of T_m.

Case-2: *During the execution of T_m, Agent$_o$ joins T_m with another independent transaction, T_j, thus forming a new transaction T_k.*

This criterion is also used to model different situations. For example, Agent$_o$ launches two separate transactions to acquire components A and B from different suppliers. If at any time during the lifetime of a product, one of the suppliers is unable to supply a component, or falls behind schedule or raises its prices, then the situation is reassessed. This situation triggers a number of operations for the search of potential replacements. If, for instance, the other supplier is able to supply components A and B, at acceptable terms, then the two separate transactions can be merged into one single transaction. This criterion can also be used when agents dynamically join the supply chain.

The split and join operations may lead to the generation of new options. The selection of the best operation to proceed and the best option to select is again dynamically performed by determining the degree of availability (see Section 3.3) for each option.

Case-3: *Before/During the execution of T_m, Agent$_o$ associates alternative and compensating transactions with transaction, T_m.*

This criterion models the following situation. For instance, an order is made for acquiring two components A and B at a particular processing node. An agent therefore executes a transaction to purchase these components. Such transactions may comprise different subtransactions. For example, one subtransaction purchases component A and other purchases component B. Further, some other alternative subtransactions are defined that perform alternate jobs in case the original subtransactions are unable to fulfil the requests. Similarly, compensating transactions are defined to cancel the affects of completed transactions, which are not required.

The basic technique used by agents in choosing an appropriate criterion for the execution of transactions is to determine the dependence between the components of a supply chain order.

If at a particular processing node, given that a product requires a number of components for its assembly, then for any two components it is possible to determine whether they are dependent or independent. Some of the possible dependencies are defined as follow:

1. Two components c_i and c_j are said to be independent if they are supplied by different suppliers S_i and S_j, or if they are manufactured by different processes P_i and P_j.

$$(S_i\,(c_i) \wedge S_i\,(c_j)) \vee ((P_i\,(c_i) \wedge P_i\,(c_j)) \Rightarrow \text{Independent}\,(c_i, c_j)$$

2. Two components c_i and c_j are said to be dependent if they are supplied by the same suppliers S_i, or if they are manufactured by same process P_i.

$$(S_i\,(c_i, c_j)) \vee (P_i\,(c_i, c_j)) \Rightarrow \text{Dependent}\,(c_i, c_j)$$

3. Two components c_i and c_j are said to be independent if their sub-components c_{im} and c_{jn} are independent according to (1).

$$(S_i\,(c_{im}) \wedge S_i\,(c_{jn})) \vee ((P_i\,(c_{im}) \wedge P_i\,(c_{jn})) \Rightarrow \text{Independent}\,(c_{im}, c_{jn}) \Rightarrow \text{Independent}\,(c_i, c_j)$$

4. Two components c_i and c_j are said to be independent if they do not share sub-components c_{im} and c_{jn}.

$$(c_{im} \notin c_j) \wedge (c_{jn} \notin c_i) \Rightarrow \text{Independent}\,(c_i, c_j)$$

5. Two components c_i and c_j are said to be dependent if they share sub-components c_{im} and c_{jn}.

$$(c_{im} \in c_j) \vee (c_{jn} \in c_i) \Rightarrow \text{Dependent}\,(c_i, c_j)$$

5. Related work

A number of the current approaches employ multi-agents techniques in the supply chain management [7, 3, 2]. For example, [7] underline the importance of the domain knowledge for agents. Agents can deal with issues such as managing outsourcing, decision-making, information sharing and various coordinative activities of the supply chains. This approach also suggests ways in which XML can be used to represent the needed knowledge. Similarly, [3] proposes an agent-based framework, called SC-Web-CS, for the supply chain management. The aim is to provide a generic, and open framework that supports supply chain functions such as collaborative supply chain planning, scheduling and optimization. Further, [2] proposes hybrid agent-based architecture for manufacturing enterprise integration and supply chain management such that the activities of manufacturing enterprise can be integrated with those of their suppliers, customers, and partners within wide supply chain networks.

Multi-agent is a very promising technology for effectively handling the supply chain activities. However, transaction-like facilities (e.g., ensuring concurrent and consistent access to shared resources) are not well addressed in multi-agents. Unfortunately, a limited research has been made to integrate multi-agents with transactional facilities in the supply chain. For example, the approach in [8] applies transaction techniques to multi-agent systems in scheduling production orders in a manufacturing environment. An open-nested transaction model is used to schedule different requests made for a particular product. Further, [9] also applies open nested transactions to the INTERAP architecture of the agents. Among the advantages claimed are the increased robustness and enhanced performance of the multi-agent applications. These approaches are also associated with various limitations. First, the open-nested transaction model compromises data consistency especially if the transactions are not compensatable. Second, transactions are not modelled dynamically according to different criteria. Our framework provides support for multitude of transaction models such as split, join, open-nested, and nested transactions. Thus depending on the nature of the environment, an appropriate transaction model is adopted. Third, current approaches [8,9] follow traditional techniques in which one agent is controlling the transactions. In our approach agents are autonomous, and they execute the transactions on peer-to-peer basis. There is no central coordinator. Thus our framework provides flexibility where a transaction can be initiated by one agent and terminated at other agent.

6. Conclusion

In this paper we have presented a framework for supporting the functions of manufacturing systems in supply chains. The framework owes its flexibility to a combination of agent technology and extended transaction models. The framework is designed to capture both transactional knowledge and knowledge of the supply chain. These features are missing from the current research on the supply chain. Transactions ensure the correctness and consistency of the supply chain activities despite failures of computer systems or network or the overlapped or parallel executions of these activities. Further, our framework automatically tailors various transaction models according to the needs of supply chain. Another interesting feature of our framework is that it enhances the responsiveness and agility of manufacturing systems.

It is worth noting, however, that the flexibility afforded by the framework raises a number of issues. The framework requires an open architecture that may expose it to security breaches. Furthermore, the reliance on autonomous agents may lead to some duplication of resources and potentially high communication overheads. Agents are sophisticated entities that embody a lot of expertise. Their reliability is therefore crucial to the operation of the underlying system.

7. References

[1] J.M. Swaminathan S.F. Smith. N.M. Sadeh, "Modelling Supply Chain Dynamics: A Multi-Agent Approach", *Decision Sciences*, Vol. 29, No 3, Summer, 1998.

[2] W. Shen & D.H. Norrie "An Agent-Based Approach for Manufacturing Enterprise Integration and Supply Chain Management" In G. Jacucci, et al (eds.), *Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility, and the Virtual Enterprise*, Kluwer Academic Publishers, 1998, pp. 579-590.

[3] J. Wu, M. Cobzaru, M. Ulieru, D. Norrie, SC-web-CS: Supply Chain Web-Centric Systems, *IASTED International Conference on Artificial Intelligence and Soft Computing (ASC2000),* Banff, July 24-26, 2000.

[4] A.K.Elmagarmid, *Database Transaction Models for Advanced Applications*, Morgan Kaufman*, 1992.*

[5] M. Younas, B. Eagelstone, R. Holton, A Review of Multidatabase Transactions on the Web: From the ACID to the SACRED, *Proc. of British National Conference on Databases (BNCOD),* Exeter, UK, Springer, LNCS, 2000

[6] M.S. Fox, J.F. Choinglo, M. Barbuceanu, *The Integrated Supply Chain Management System,* Internal Report, Department of Industrial Engineering, University of Toronto, 1993.

[7] K.D. Fenstermacher and D. Zeng, Know Your Supply Chain: Transactional Knowledge and Beyond, *The AAA-I Workshop on Knowledge-based Electronic Markets*, July 2000, Austin, TX, USA:www.igec.umbc.edu/kbem/Fenstermacher.pdf

[8] K. Nagi, Transactional Agents: A Robust Approach for Scheduling Orders in a Competitive Just-in-Time Manufacturing Environment, *Proc. of Workshop on MAS in Logistic and Economical Perspectives of Agents on Conceptulisation*, Germany, Sept., 1999

[9] K. Nagi, Implementation Model for Agents with Layered Architecture in a Transactional Database Environment, *1st Int. Bi-Conference on Agent-oriented Information Systems*, June 1999, Italy.