A CSP-based Model for Integrated Supply Chains

Rongming Sun[1]
Bei-Tseng (Bill) Chu[2,3]
Robert Wilhelm[1,2]
Jian Yao[2]

[1]Department of Mechanical Engineering
[2]School of Information Technology
[3]Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC 28223

{rsun, billchu, rgwilhel, jyao}@uncc.edu

Abstract

Supply Chain Integration is a very important problem for business to business electronic commerce. An integrated supply chain allows businesses to share real-time information and drastically reduce transaction costs. This paper describes our efforts to model the order selection and negotiation process as a multi-agent system based on Constraint Satisfaction Problems (CSP). A negotiating agent can represent each company along the supply chain. The core capabilities of such agents can be modeled as a set of CSPs. These agents can generate a purchase plan to meet the company's demands. Negotiation will be triggered when no satisfactory plan can be found. Strategies are identified to relax some of the constraints to generate counter proposals.

1. Introduction to Supply Chain Integration

Retailing and manufacturing activities can be viewed as a set of supply chains. For example, a home center may stock lawn mowers for consumers. A lawn mower manufacturer may assemble lawn mowers from parts, some of which are bought from its suppliers. Their supplier, may, in turn, buy parts from their suppliers. Companies along the supply chain, the home center, lawn mower manufacturer and its suppliers must carry some inventory. Common reasons for incurring the necessary inventory costs include natural fluctuations in market demand beyond their ability to forecast, and lead times for certain products / parts may be too long to satisfy peak demands in a just-in-time manner. In 1997, U.S. consumer spending was about $3 trillion. According to one study, retailers, manufacturers, and their suppliers must carry an inventory that is worth about $1 trillion to support these consumer sales.

The objective of Supply Chain Integration is to use Information Technology so that companies may better share information and achieve significant reduction in inventory carrying costs. For example, use of Internet communications can enable a lawn mower manufacturer to work with more suppliers and obtain supplies much more quickly, thus reducing the need to carry inventory. The retail store may have arrangements to share better sales forecasts and information with lawn mower manufacturers. The manufacturer in turn can use the Internet to share better forecasts with their suppliers and bring further reduction in inventory carrying costs.

Two extreme models for Supply Chain Integration have been proposed. The first is a heavy supply chain integration model[1] which is typically employed by large companies, e.g. Boeing and its engine manufacturer. In the heavy supply chain model, a manufacturer commits to buy a fixed amount of products from a supplier over a fixed time horizon. Information sharing, typically through private networks, may allow small variations (e.g. 10%) around this negotiated level. In a light Supply Chain model, on the other hand, a company may choose to buy products on the open market in a way similar to consumers buying products from the web. A more common practice, employed by companies of all sizes, is a hybrid approach where a company may have a set of selected suppliers. The company may have negotiated purchase agreements with the suppliers but ordering is based on demand. That is to say, the company will order from a supplier based on its own demand and the capacity for that supplier to service this demand.

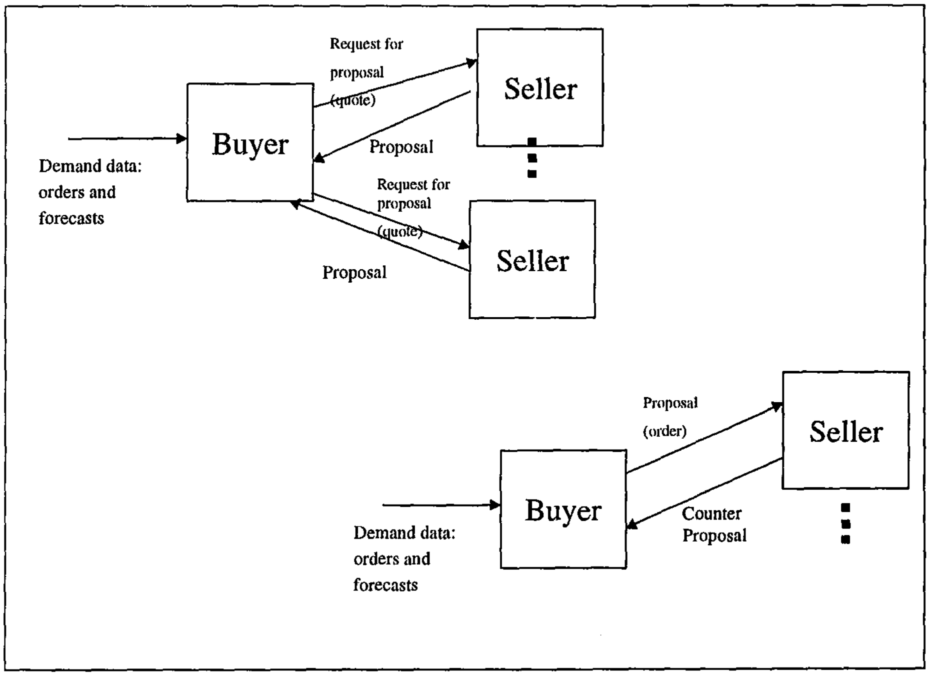

Figure 1. The Supplier selection and negotiation process

In this paper we look at a common problem of the hybrid model: supplier selection and negotiation. Because of the recursive structure of a supply chain, without loss of generality, we will focus on a manufacturer (also referred to as the buyer) and its suppliers (also referred to as sellers). We assume that the buyer has selected a set of sellers (which may change over time) with negotiated standard purchase agreements. The buyer also may have a ranking system of the sellers along dimensions such as quality and reliability of delivery. Our objective is to use a multi-agent system to aid or automate buyer selection and negotiation. This is important because:

- This is a difficult task for people to carry out when one is confronted with even a modest number of choices (e.g. 10 suppliers each with 10 options). With the advent of Electronic Commerce, a buyer will have many more alternative sellers on a global scale. Agents can be used to carry out routine selections and negotiations so that people are only involved in exceptional cases. This capability will enable a buyer to examine more options and find a more cost-effective set of sellers that meet requirements.
- Agents can also monitor the execution of the committed orders. They can inform each other early about potential changes and make real-time adjustments. As a result the buyer and sellers can reduce their inventory levels.

A typical selection and negotiation process is illustrated in Figure 1.

A buyer sends out requests for proposal of quotes to a set of sellers. A seller may respond with a general business rule such as "if your order is placed at least 4 weeks ahead of the due date and less than 8 weeks ahead of the due date, the unit price is $20". After getting such quotes from different suppliers, the buyer will decide, based on its demand, which seller(s) to order from. The buyer will then send order proposals to selected sellers. For example, the buyer may order 500 pieces by the 15th of March from a seller, s1.

Buyers may sometimes combine demands from different time periods into a single order to take advantage of volume prices offered by a supplier. For example, s1 may offer a discounted price for orders of 500 or more.

The seller may not be able to satisfy a particular order at a given time. For example, an unaccounted for machine maintenance schedule may prevent on-time shipment of an order. In this case, the seller may counter-offer: "ship 300 by the 15th of March, and 200 by the 25th of March, no extra shipping charge, and you can enjoy the same discount price". After considering actual scheduled demands, the buyer may reply and accept the counter offer.

2. Modeling the supplier selection and negotiation process

Buyers and sellers can be represented by respective agents. The demand input for a buyer takes two forms: a detailed demand, and an aggregated demand. By using industry standard algorithms (such as Wagner-Whitin [2]), we can further generate dynamic lot sizes that minimize order costs and inventory costs. The following tables illustrate inputs to a buyer agent using the lawn mower example:

Suppose Acme Hardware chain serves multiple cities. Daily lawn mower demand forecasts from Charlotte are:

Week 1	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Delux mower	180	140	98	178	130	0	0
Electric M&M	200	100	94	260	85	0	0

Week 2	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Delux mower	96	130	137	145	182	0	0
Electric M&M	244	233	102	210	97	0	0

Daily forecasts from Raleigh are:

Week 1	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Delux mower	120	100	242	82	230	0	0
Electric M&M	130	160	296	190	45	0	0

Week 2	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Delux mower	114	160	173	215	138	0	0
Electric M&M	96	127	188	140	233	0	0

The aggregated forecast from Acme North Carolina is as follows:

Week 1	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Delux mower	300	240	340	260	360	0	0
Electric M&M	330	260	390	450	130	0	0

Week 2	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Delux mower	210	290	310	360	320	0	0
Electric M&M	340	360	290	350	330	0	0

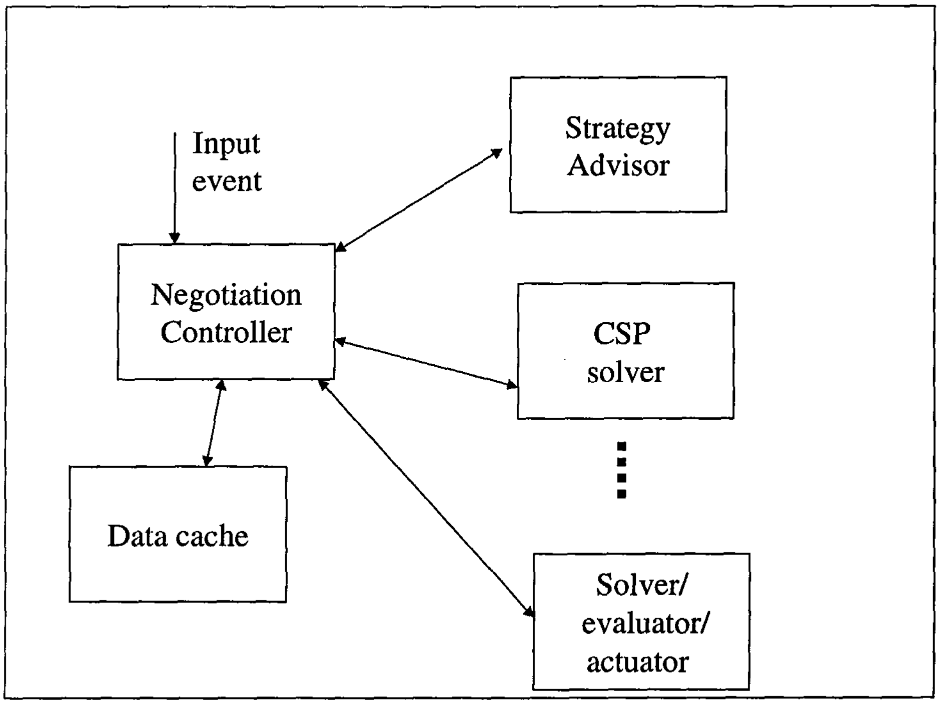

Figure 2. Components of a buyer / seller agent

After considering the initial stock level, the dynamic lot size for Delux mower in the first week may come out to be 410 on Thursday of week 1, 500 on Monday of week 2, 310 on Wednesday of week 2 and 680 on Thursday of week 2.

Both the buyer agents and seller agents are structured as illustrated in Figure 2.

The negotiation controller manages the execution of the agent. The controller is event driven. Events are messages sent either by people or other agents. We assume agent communication will be based on KQML[11] and its extensions for negotiations[4]. The KQML messages for negotiation are summarized in Table 1.

The strategy generator provides strategies to be followed by the controller at different steps of the selection / negotiation process. A number of strategies have been identified. New strategies can be incorporated. For example, constraint relaxation in constraint satisfaction problems [5] can be modeled as a strategy. The agent also need actuators to carry out actions such as: interfacing to legacy applications and sending messages to other agents.

We believe that the Constraint Satisfaction Problem (CSP) [5] offers a very good framework for representing the knowledge and information needed for integrated supply chain management. A CSP consists of a set of variables and a set of constraints that these constraints must be satisfied. In the supply chain domain, many business rules

Table 1. KQML Messages for Negotiation

accept-proposal	the action of accepting a previously submitted proposal to perform an action
CFP	the action of calling for proposals to perform a given action
Proposal	the action of submitting a proposal to perform a certain action, given certain preconditions
reject-proposal	the action of rejecting a proposal to perform some acting during a negotiation
Terminate	the action to finish the negotiation process

Table 2. Negotiation Steps in CSP Paradigm

Possible Terms and conditions	CSP variables and their domains
Offer	CSP variables with value assignments
Acceptable offer	A set of variable assignments that satisfy all constraints
Unacceptable offer	A set of variable assignments that does not satisfy all constraints
Negotiating counter offer	Relax some of the constraints to make an unacceptable offer acceptable.

can be easily represented as constraints. For example, a price quote business rule: " if your order is placed at least 4 weeks ahead of due date and less than 8 weeks ahead of the due date, the unit price is $20" can be represented as a constraint among variables: shipping date, lead time, and price.

Negotiation can also be naturally modeled using the CSP paradigm as illustrated in Table 2.
Many of the classical constraint satisfaction problems (e.g. graph coloring, resource allocation) are known to be NP-complete or NP hard problems [5,7] There is a very large literature on how to utilize heuristics to solve constraint problems. The complexity of CSP for negotiation is likely to vary according to what is being negotiated. One part of our current research focus is to systematically exam the computation complexity of negotiation problems.

The strategy component is responsible for formulating the CSP to be solved by the CSP solver. Other solvers / evaluators may be employed for specific tasks. The agent also needs actuators to take actions, such as sending out a message.

3. Negotiation steps

A buyer agent will follow the steps of

- Opening the bid by sending requests for proposals to prospective sellers. Sellers will respond with proposals.
- In response to proposal messages, the buyer agent will evaluate proposals and generate a purchase plan.
- Upon approval (of human decision makers) of the purchase plan generated by the buyer agent, the buyer agent can negotiate with prospective sellers. During the negotiation process, buyer agents and seller agents may exchange counter-offers. The buyer agent may also, at various points, present options to be evaluated by human decision-makers. Users can configure these intervention points.

A seller agent will follow the steps of:

- Response to request for proposals from prospective buyers.
- Evaluate (proposed) orders from buyers.
- If seller cannot fulfill a proposed order, the seller may wish to generate a counter proposal. The seller agent may also, at various points, present options to be evaluated by human decision-makers. Users can configure these intervention points.

4. Negotiation Strategies

During each negotiation step, multiple strategies may be followed. For example, splitting orders into multiple shipments may be a strategy for a seller agent to offer counter proposals. An alternative strategy at this point may be to propose a set of orders to be delayed. We envision that these strategies can be configured in a variety of ways to be followed by buyer and seller agents.

In a simple case, a default strategy will be associated with each step. If this strategy fails (e.g. the counter-offer was rejected), a human decision-maker may be called in to select one of the alternative strategies. In the following section, we discuss some example strategies we have implemented.

Strategies for opening bid / response to request for proposal

A buyer agent may employ this strategy in response to a "start negotiation" event. The inputs to this event are demand data described earlier. The agent will solicit ordering rules from a set of suppliers. A typical seller's response may be:

Supplier s1 can supply as many as 500 pieces at the unit price of $20 with a lead time no greater than 4 weeks.

Strategies for generating purchase plan (proposal):

CSP may be used to generate a (proposed) purchase plan. A configuration option may indicate whether human approval of such a plan is needed.

This process can be formulated in constraint logic programming (CLP) [6]. For ease of explanation, we restrict to the case of one product. The approach can be easily generalized to multiple products.

An order can be represented as a predicate *Order(S,Q,D,P)* where S is the supplier, Q is the quantity ordered, D is the due date, and P is the unit price.

Often, sellers may offer price quotes with conditions attached. These conditioned price quote can be represented as the following rule / constraint, assuming we have the appropriate mechanism to do date arithmetic. The following constraint means that seller s1 can supply 500 pieces at the unit price of $20 given a lead time of 4 weeks.

Order(s1,Q,D,$20) :- Q<=500, D-today>=4weeks.

The problem of generating a purchase plan can be cast into a CLP problem of finding a set of orders that satisfy a total quantity before a given due date and under a given price cap. This can be represented as a predicate *OrderProposals (SL, QL, DL,PL,TQ,DD,PC)* where SL is a list of suppliers, QL is a list of quantities to be ordered, DL is a list of due dates, PL is a list of unit prices, TQ is the total quantity needed, DD is the due date TQ is needed by, and PC is the price cap. For example, the problem of generating a purchase plan for 1000 pieces by May 15 with total price not exceeding $20,000 can be viewed as solving for:

OrderProposals(SL,QL,DL,PL,1000,May15,$20000).

The answer to this question may be:

OrderProposals([s1,s2],[500,500],[May13,May14],[20, 18],1000,May15,$20000)

It represents the following orders:
Order(s1,500,May13,$20) and *Order(s2,500,May14,$18)*

Purchase plan generation may be governed by the following rules / constraints:

OrderProposals([S],[Q],[D],[P],TQ,DD,PC) :-
*Order(S,Q,D,P), Q<=TQ,D<=DD,P*Q<=PC.*

It means that if we can find a single supplier who can satisfy all the requirements, then this supplier will be selected. However, if we cannot find a single supplier, we must split the total order among more than one supplier. This strategy can be represented as:

*OrderProposals([S|RS],[Q|RQ],[D|RD],[P|RP],TQ,DD, PC): -Order(S,Q,D,P), Q<TQ, D<=DD,P*Q<PC, OrderProposals(RS,RQ,RD,RP,TQ-Q,DD,PC-P*Q).*

Here the notation [H|T] indicates list concatenation, commonly found in Prolog systems, where H represents the head of the list and T represents the rest of the list.

It should be pointed out that, the problem presented here represents a "bare-bone" case for purchase plan generation. The CLP/CSP approach may not offer the most efficient solution. An important reason for us to pursue the CLP/CSP approach is that additional constraints will be involved for generating a purchase plan in practice. For example, certain components must arrive at the same time, if one is delayed, the other one needs to be delayed as well. The constraint formulation gives an open framework for such constraints to be represented. Further research is needed to come up with computation strategies that can explore special types of constraints and compute solutions efficiently.

Strategies for order proposal evaluation

When a seller receives an order proposal from a buyer, the seller may need to check it against its current manufacturing capacity to see if the order can be met. CSP / CLP has been clearly demonstrated as an effective tool to model and solve scheduling and capacity estimation problems [7]. Therefore, evaluation of these proposals based on CSP /CLP is one of the possible strategies to be employed for this step.

Assume that predicate *Capacity(TQ,D)* solves the scheduling/capacity problem with a CSP. It means that a seller is able to deliver a total of TQ by date D. Again we choose to focus on one product for ease of illustrations. The problem of determining whether an order is feasible can be expressed as the rule:

FeasibleOrder(Q,D) :- Capacity(TQ,D), Q<=TQ.

If the order is not feasible, the seller may want to come up with counter-proposals and negotiate with the buyer. The process of generating counter-proposals and negotiation can be modeled as contrast relaxation. In our framework, constraint relaxation can be represented as a library of predefined negotiation strategies. These strategies can be invoked at appropriate times.

Strategies for counter proposal generation

Many strategies can be explored to offer counter proposals when a seller cannot meet a proposed order from a prospective buyer. Here we discuss order splitting as a specific example that we have implemented. The idea of order splitting is to split an order into multiple shipments with the first shipment on or very close to the date

Table 3. Generated Purchase Plan

Order ID	SO-1	SO-2	SO-3	SO-4	SO-5	SO-6	SO-7	SO-8	SO-9
Part ID	X	X	X	X	X	X	X	X	X
Supplier ID	S1	S2	S3	S4	S5	S6	S7	S8	S9
Quantity	410	500	310	680	770	760	570	240	690
Release date	0	0	0	0	1	4	8	10	11
Ship date	2	5	7	8	12	15	19	21	22
Deliver date	4	8	10	11	15	18	22	24	25

requested by the buyers. This can be naturally modeled in CSP/CLP.

Predicate *Split(OriginalOrder(Q,D),Orderl(Q1,D1), Order2(Q2,D2))* means to split an original offer *OriginalOrder(Q,D)* into two orders *Orderl(Q1,D1)* and *Order2(Q2,D2).*

The seller needs to first evaluate whether this order splitting is feasible. This can be achieved by solving:

Split(OriginalOrder(Q,D),Orderl(Q1,D1), Order2(Q2,D2)) :-Q1+Q2=Q, D1<=D, D2>D, FeasibleOrder(Q1,D1), FeasibleOrder(Q2,D2).

If the two-way-splitting does not yield an answer, 3 way splitting may be tried in a similar fashion. This strategy needs a cut off criterion; for example, the splitting strategy fails when 3 way splitting fails to generate a feasible solution. At this point another strategy may be used to suggest a partial shipment as a counter offer.

Strategies for order negotiation

Order negotiation is the step in which the buyer agent negotiates with seller agents. We describe a strategy we have implemented. In this strategy the buyer agent sends messages to each seller, one at a time. A more parallel negotiation strategy may be developed later. For each order proposal sent, the seller may accept, reject, or counter-propose (e.g., using the splitting strategy discussed earlier). If the seller accepts the order, the demand information is updated to reflect the commitment. If the seller rejects the proposal, the *generate order proposals* strategy may be employed again with this particular seller deleted as a possible supplier. If the seller counter -proposes, the buyer will check to see if the counter-proposal can satisfy the buyer's constraints. If that is acceptable, it is treated as an accepted proposal, otherwise it is treated as a rejection with this particular supplier's rule (seller constraint) modified based on the counter proposal. The *generate purchase plan* step is then invoked again.

We have presented here a very simple model of evaluating proposals. Further research is needed to provide risk/benefit analyses and determine if counter-counter-offers need to be made when confronted with an unacceptable offer. Furthermore, a similar strategy can be employed to deal with the situation where a seller might report that committed order cannot be fulfilled.

5. An example

We use an example to illustrate the interaction between a buyer agent and a number of seller agents. The purchase plan shown in Table 3 was generated by the buyer agent.

where SO-1, SO-2, ... are orders. Each order specifies the part being ordered , supplier id, order quantity, and dates related to the order.

Suppose the suppliers S1, S2, and S3 accept the first three proposals (SO-1, SO-2, SO-3). The buyer agent will modify demand data to reflect the expected receipt of the products as illustrated in Table 4 where the demands for dates 4, 8, and 10 are revised to 0 as they will be supported by the accepted orders.
Suppose supplier S4 rejects order SO-4. Constraint

Table 4. Demands Modified per Expected Receipt of Product

Time (date)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Q				410 $\Rightarrow$0				500 $\Rightarrow$0		310 $\Rightarrow$0	680			
time	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Q	770			760				570		240	690			

Table 5. Revised Demand

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Q				410 ⇒0				500 ⇒0		310 ⇒0	680 ⇒0			
t	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Q	770			760				570		240	690			

Table 6. Counter-proposal

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14
D	0	0	0	50	360	0	0	210	290	310	360	320	0	0
t	15	16	17	18	19	20	21	22	23	24	25	26	27	28
D	*290*	*340*	140	370	390	0	0	230	340	240	380	310	0	0
	300	***470***												
	301													

Table 7. Updated Demand from Rolled Up Order

T	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Q				410 ⇒0				500 ⇒0		310 ⇒0	680 ⇒0			
T	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Q	290 ⇒0	340 ⇒0	140 ⇒0	760				570		240	690			

"(supply S4 X)^(delivery date == 11) => (quantity ==0) " has to be added. The *generate purchase plan* step will be invoked again to find a new supplier, S11. The new order SO-4 becomes "Supplier S11 supplies part X on date 11 with quantity 680". Suppose this order is accepted by S11. Demand on date 11 is revised from 680 to 0 as shown in Table 5.

Suppose order SO-5 is sent to supplier S5 and the buyer agent receives a counter proposal from S5: "Supplier S5 can supply part X 300 at day 15th and 470 on date 16th" as shown in Table 6. The buyer agent will evaluate this counter offer against detailed demands (see the tables in section 2).

The evaluation result is positive because this particular order was "rolled up" from multiple demand sources to take advantage of the discount for ordering a large shipment. Again, demand is updated as shown in the Table 7.

6. Implementation

We have prototyped a set of buyer and seller agents for a large industrial consortium of companies developing solutions for integrated supply chain. We found that CLP offers a convenience means for problem formulation and knowledge representation. However, integration with legacy systems is a key requirement from our industrial partners. Our current prototype is implemented using ILOG Solver [10] which is a C++ based implementation of a constraint solver. From the application programmer's point of view, ILOG Solver is a set of C++ classes that can be used to implement a mix of different problem solving strategies.

7. Conclusion

The main contributions of this work are in the following areas.

First, we have developed a generic approach that allows the supplier selection and order negotiation process to be modeled and implemented within a multi-agent framework. This extends our previous statement of using a multi-agent framework for enterprise integration [3] to multiple enterprises as in a supply chain. The framework, though incomplete at this point, is extensible and allows new negotiation strategies to be added dynamically.

Second, while many researchers have proposed the use of CSP representations to model supply chain integration and negotiation [8,9], this work demonstrates the concept via concrete implementation. In the process we have learned many valuable lessons. The most important lesson being that CSP by itself is not a strong enough paradigm to model complex negotiation activities such as the order negotiation problem. A framework, such as the one we have suggested, is needed. One very important reason for the need of such a framework is to apply domain specific knowledge when relaxing constraints in the negotiation process (e.g. the order splitting strategy discussed in this paper).

Acknowledgement: This work is supported in part by a grant from NIST/ATP under the number: H980513-00. The authors also wish to thank the reviewers for their valuable suggestions.

References

[1] IBM Corporation, Dec 17, 1998, "Extended Enterprise Coalition for Integrated Collaborative Manufacturing Systems" (EECOMS) Version 1, Revision 1. White Paper

[2] H. M. Wagner, T. M. Whitin, 1958, "Dynamic Version of the Economic Lot Size Model". *Management Science*. 5(1):89-96

[3] Peng, Y., Finin, T., Labrou, Y., Chu, B. , Long, J. Tolone, W.J. & Boughannam, A., 1998, "A Multi-Agent System for Enterprise Integration", *International Journal of Agile Manufacturing*, Vol. 1 No. 2 pp. 201-212.

[4] Ye Chen, Yun Peng, Tim Finin, Yannis Labrou, Scott Cost, Bill Chu, Rongming Sun, Bob Wilhelm. "A Negotiation-based Multi-agent System for Supply Chain Management". Workshop on Agent Based Decision-Support for Managing the Internet-Enable Supply-Chain, The International Conference on Autonomous Agents (Agents '99), Seattle, May 1999

[5] Tsang, Edward, 1995, *Foundations of Constraint Satisfaction*, Academic Press Limited.

[6] Kim Marriott and Peter J. Stuckey, 1998, "Programming with constraints: An introduction". *MIT press*.

[7] J. Chris Beck, Andrew J. Davenport, Ed M. Sitarski and Mark S. Fox., 1997, "Beyond Contention: Extending Texture-Based Scheduling Heuristics". *Proceeding of the National Conference on Artificial Intelligence (AAAI-97)*

[8] Amol Dattatraya Mali, July 27, 1997, Constraint-based specification of reactive multi-agent systems, *AAAI-97 Workshop on Constraints and Agents.*

[9] Kazuo Miyashita, July 27, 1997, Iterative Constraint-based Repair for Multiagent Scheduling. *AAAI-97 Workshop on Constraints and Agents.*

[10] ILOG Optimization Suite, White paper, April, 1998, http://www.ilog.com

[11] Finin, T., McKay, D., Fritzson, R., & McEntire, R. , November, 1994, "The KQML Information and Knowledge Exchange Protocol". *Third International Conference on Information and Knowledge Management.*

[12] Makoto Yokoo, Seika-cho Hikaridai, Nov 1993, Constraint Relaxation in Distributed Constraint Satisfaction Problems, *Proceeding of the 1993 IEEE Int.'l Conf. on Tools with AI*, p56-63

[13] Susan E. Conry, Kazuhiro Kuwabara, Victor R. Lesser, Robert A. Meyer, Nov/Dec 1991, Multistage Negotiation for Distributed Constraint Satisfaction, *IEEE Transactions on System, Man, and Cybernetics*, Vol 21, No 6, p1462-1477

[14] Douglas J. Thomas, Paul M. Griffin, 1996, Coordinated supply chain management (invited review), *European Journal of Operational Research*, v94, n1, p1~15

[15] Asron Armstrong, Edmund Durfee, 1997, Dynamic Prioritization of Complex Agents in Distributed Constraint Satisfaction Problems, *Proceeding of the Fifteenth International Joint Conference on Artificial Intelligence*, 620-625

[16] Makoto Yokoo, Edmund H. Durfee, Toru Ishida, Kazuhiro Kuwabara, Sep/Oct 1998, The Distributed Constraint Satisfaction Problem: Formalization and Algorithms, *IEEE Transaction on Knowledge and Data Engineering*, Vol 10 No 5, p673-685