

Chains of Commitment Software Architecture

Rich Ervin

2MX Consulting, NanoGuys.com

richervin@nanoguys.com

As eCommerce becomes more competitive and continues to expand into and merge with traditional supply-chain based businesses, a significant competitive advantage can be achieved through improved automation of the commitment process. The “chains of commitments” approach to this problem exploits the fact that a commitment from a vendor is backed up by a chain of commitments from a chain of vendors, a.k.a. the supply chain. This paper presents a high-level architecture to support this automation. It draws heavily on principles of collaborative and value chain analysis as examples, but is applicable to any commitments-based supply chain management approach.

1 INTRODUCTION

The development of successful chains of commitment applications calls for the development of flexible, scalable architectures for the communication and negotiation of orders and commitments for those orders that maintain security and anonymity of that information. The greatest area of opportunity appears to be in the eCommerce realm, where web-based privacy and statistical methods may be applied to guarantee statistical anonymity and privacy of otherwise sensitive corporate data. This article discusses the application of a particular software architecture for the management of chains of commitment with primary application in the eCommerce area, but also directly relevant to any chain of supplier-customer relationships.

This discussion does not cover implementation specific details except for illustrative purposes, and it purposefully avoids detailed discussion of numerous technical challenges one would encounter in the implementation of a “real” system (i.e. rollback, deadlock detection, resource locking, verification, error detection and recovery, etc). To approach these issues would enter the realm of product design and specification [Garlan and Shaw 1997]. It generally assumes a chains-of-commitment management approach that is based on statistical methods and predictive analysis, but sufficiently flexible to directly support a variety of other management methods, such as optimized, constraint-based, or

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted with-out fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works, requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept, ACM Inc., 1515 Broadway, New York, NY 10036 USA, fax +1 (212) 869-0481, or permissions@acm.org.

simple commitment-delivery approaches. The article does not discuss how this will be accomplished, but it should not require significant modification of the general architecture or principles contained herein.

The article assumes an infrastructure model of a TCP/IP enabled network with open-architecture operating systems widely connected via the Internet to a variety of vendors and application service providers (ASPs). However, the proposed architecture is sufficiently flexible that it may be applied to a wide range of potential infrastructure models.

1.1 Definitions

- Quality (of product): The “quality” of a product in this document refers to the conformance of the end product to the customer’s original request and expectations. For example, if the customer orders a green sweater and receives a blue sweater, the “quality” of the product is below perfect. The same is true if she receives a green sweater that does not meet her expectations of material or workmanship quality, or receives a sweater of a different style. There will be different measures of quality, some objective and some subjective. In this paper, we refer only to a non-specific aggregate property we simply call the “quality” of the product. A variety of methods for measuring consumer satisfaction in a quantitative way are readily available. In this document, we only assume that some quantitative method is available and that it will yield some scalar representation of product “quality”.
- Commitment: In this context, we define a “commitment” as a promise from a vendor to deliver a specific product or service to a specific customer in a specific time frame at a specific level of quality for a specific price. In our examples, a commitment is made against currently unallocated inventory or capacity of the vendor, or else against commitments from the vendor’s vendors that are backed up with their inventory, capacity, or vendor commitments. However, this is not a necessary feature, and vendors are free to make commitments on whatever basis they choose or are able. Recent publications about collaboration [Forger 2000] and value chain analysis [Schiebel 2000] represent some modern approaches and amply outline the challenges and proposed solutions of evaluating vendor capabilities.
- Quality of Commitment: A key element in a chain of commitment analysis is the “quality of commitment” (QoC). This is a measure of a given vendor’s ability to deliver a given product of a given quality in a given time at a given price. The formula for this measure depends on the needs of the customer involved – the average consumer is primarily concerned that the product will be delivered at the agreed upon price. A manufacturer, however, may be primarily concerned that the products be delivered on a timely manner, or at a very specific level of quality. The QoC can be expressed in terms of a “score” that is applied to a specific “commitment” being offered by a particular vendor. This score will generally be computed as a multi-dimensional vector, but

will likely be of the most value to the end user when reduced to a scalar.

2 CHAIN OF COMMITMENT

In a supply chain, the ability of the end provider to make a delivery commitment to an end consumer depends on the quality of his supply commitments – that is, the ability of a consumer to get a good quality sweater delivered on time and at a good price depends on the sweater maker’s ability to get wool yarn, dye, and thread in a timely and cost effective manner. Late or expensive supplies make for late or expensive end products. The yarn maker’s commitment quality is based on his wool supplier’s timeliness and cost effectiveness, and his commitment quality is dependent on his ability to manage his supply of raw materials. This chain of dependencies is what we refer to as the “chain of commitment” or “CoC”.

2.1 Managing the Quality of Commitment

The simplest and most common means of managing the CoC problem is inventory. The sweater e-Tailer may keep a supply of sweaters on hand, or more likely will require that the sweater maker maintain a supply. The sweater maker will maintain an inventory of yarns, dyes, thread, and other materials and supplies to insulate against variations in the quality of commitment from his suppliers. The yarn maker will keep more wool on hand than is needed to meet immediate production requirements because he cannot be certain that the raw materials will be available at the required price, quality, and quantity at the time it is needed.

This system of inventories is costly to business overall, and drives down customizability of end products because a provider only wants to maintain inventories of the most popular products with the most popular combinations of features.

2.2 Leveraging the Chain of Commitment

A simple inventory management system, such as that suggested in the sweater example above, is a very poor solution because of the extreme cost of maintaining a very wide variety of product configurations and the preference of customers for very specific configurations over “common” configurations. The focus of SCM technology in the last several years has been improving the supply chain to be more reactive to consumer demands, and the primary driver for that has been the cost of maintaining these inventories in an environment of shifting consumer preferences [Hammond 2001].

The automobile market also provides a very clear and intuitive example of this problem [Verespej 2001]. Automobile buyers typically prefer highly customized vehicles. Given the opportunity, most would like to choose from a selection of wheels, radios, colors, paint schemes, interior materials, drive-train, and so on. The automobile industry has traditionally attempted to meet this demand by producing an incredibly range of models, colors, and configurations of new vehicles at the great cost of dealer returns, drastic markdowns of

unpopular combinations, and the reduced satisfaction of the customer who must compromise on their choice of color and configuration preferences.

In principle, there's no reason why the automotive industry couldn't provide that level of customizability. However, business logistics interfere. Because there is tremendous variation in the supply chain process, the quality of commitment for custom ordered automobiles is very poor. Though the dealer can guarantee the price, he cannot guarantee the timeliness or quality (meaning conformance to the specification) of the end product.

To solve this problem, it would be necessary for the automobile dealer to have access to information about all of the various vendor's abilities to "commit" to delivering their respective products on time and on bid, and he would need to have it in real-time because end consumers are not going to stand for costing systems that take weeks or months.

In such a system, a dealer might access a computer system at his manufacturer's plant on which he would specify the vehicle he wants to order with all its various options. The manufacturer's computer would then, in real time, project a possible completion date and make a series of "commitment requests" to its own warehouses and/or vendors. That is, the system would examine current inventories and scheduled production of produce-on-demand resources and dynamically allocate those current and future inventories against the production of this one particular vehicle. The system might even understand cost bases and be able to obtain real-time automated quotations for products based on requested delivery date and specifications [Camstar Systems, Inc. 2001]. For materials and parts not maintained or produced locally, the system would make real time commitment requests from external vendors.. At the end of this process, the dealer (or consumer, in an online scenario) would be presented with a final set of specifications that indicates a final price, a scheduled delivery date, and any possible substitutions that might have been made by the system. When the consumer accepts the offer, the "commitment requests" become firm orders and the supply chain is put into motion.

Diagram A shows, from a high level, what a single chain of commitment looks like. The request starts with the customer (solid black arrow) and flows downward throughout the entire tree of vendors, ending at the bottom of the supply chain (the raw material provider, in this somewhat optimistic example). The commitment itself flows back up the chain, and is processed at each level until the consumer is ultimately presented with a commitment date, price, and product. The quality of the commitments have been determined in the CoC Analysis Engine (not depicted) at each level of the chain, so the final commitment to the customer has a known "quality of commitment", which is an aggregate of all the QoCs downstream from her. Based on the overall QoC score, the customer, within a known level of confidence, can expect that the desired product will be delivered on time and on price. The actual delivery of this product (and of all the components that go into it) will be tracked against the commitment offered by the CoC Data Warehouse to be used in the evaluation of future commitments from the same vendors.

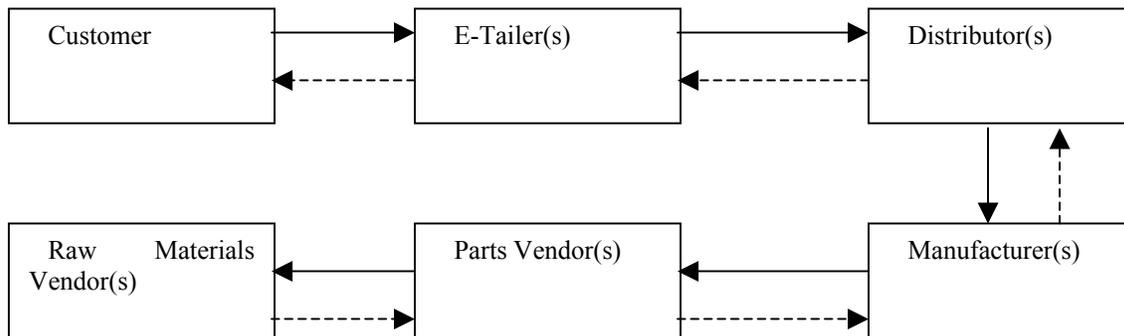


Diagram A: An example chain of commitment
Solid arrows are “commitment requests”. Dashed arrows are vendor commitments.

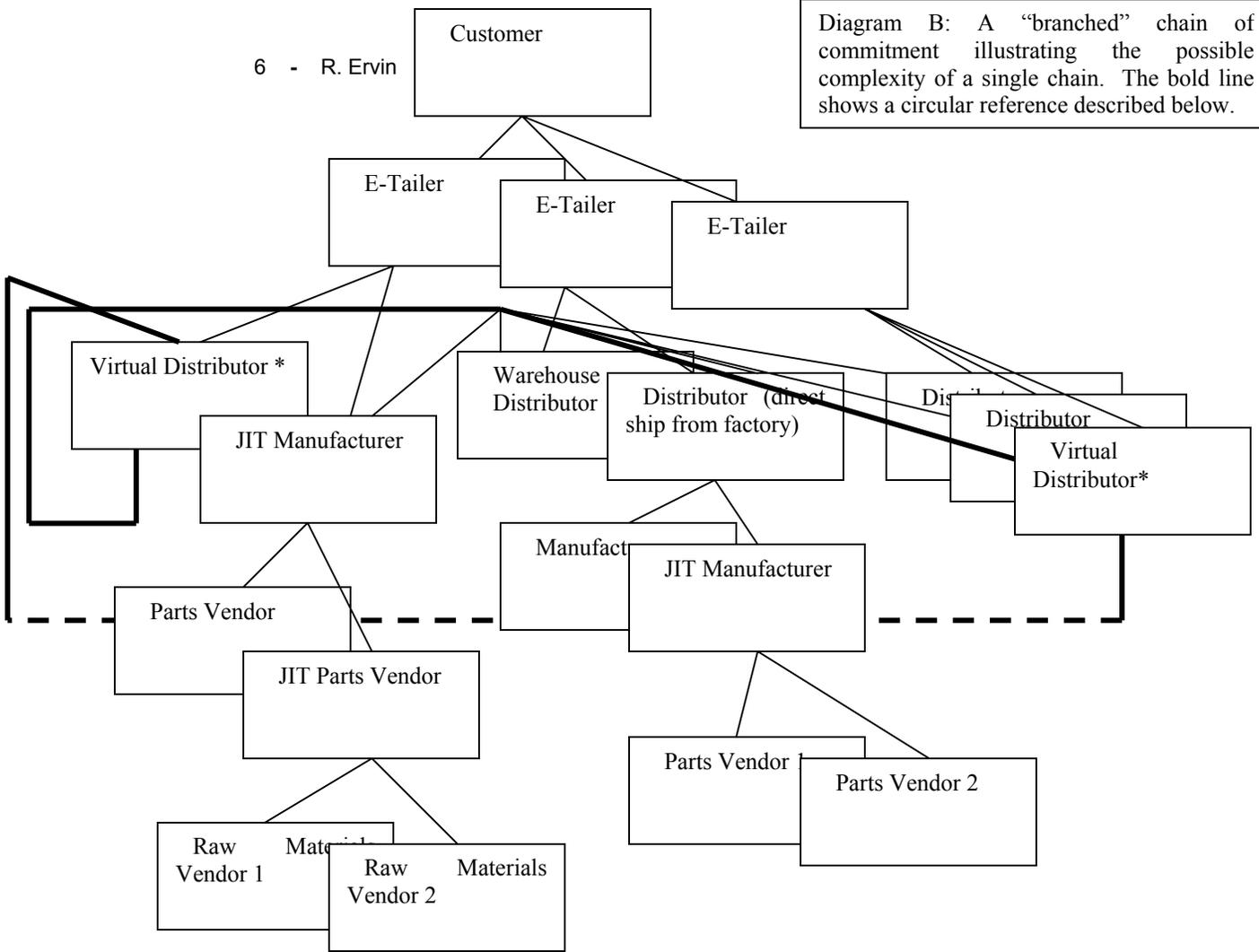
This view is deceiving in that it makes this transaction appear to be much simpler than it really is. In reality, each downstream (solid) arrow represents requests for 1 or more products to one or more vendors.

Diagram B gives a flavor for the potential complexity of a transaction that might be supported by this architecture. The complexity derives not from architectural complexity, but from the availability of multiple vendors, the complexity of the product, and the variety of ways in which a vendor might fulfill a particular request. Despite the potentially high complexity of the transaction, the underlying architecture remains relatively simple because only part of the transaction is being managed at any one time.

From the point of view of any single partner in this “branched chain”, the transaction appears to be simple to modestly complex. On the demand (upstream) side, the requests “fan out” to a tree of providers. The replies (commitments), however, are narrowed at each level of the chain by the CoC engine so the end customer receives a small list of vendor choices that have been pre-evaluated to be among the best available options.

Diagram B emphasizes the dual-role of a vendor as both a vendor and consumer of products with possible interdependencies with other vendors and virtual vendors [Nøkkentved 2000] (introducing the possibility of a recursive loop, which must be accounted for in an actual implementation.)

Diagram B: A “branched” chain of commitment illustrating the possible complexity of a single chain. The bold line shows a circular reference described below.



This part of the branched train (or “tree”) illustrates one possible scenario for a circular chain. The circular chain begins with the virtual distributor who does not directly distribute products, but consolidates requests from multiple sources, then places orders against other virtual distributors. In this example, two virtual distributors are attempting to source the product from one another. Pricing dynamics normally guards against this type of circular dependency, but the possibility should be considered in any CoC design.

3 A CHAINS OF COMMITMENT SOFTWARE ARCHITECTURE

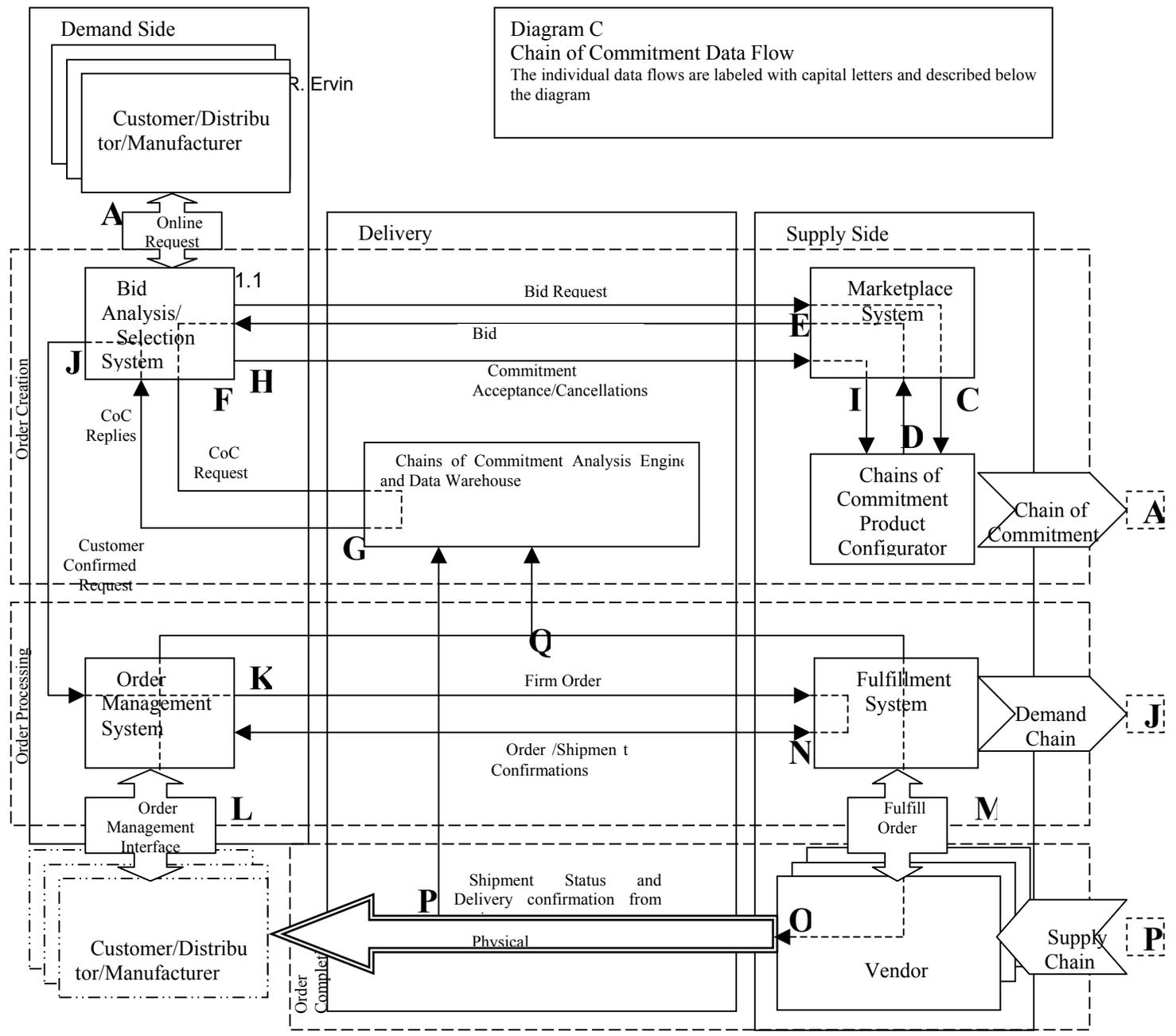
The proposed architecture has 6 primary components:

- Order System: A company’s ordering system tracks all currently outstanding orders for products from vendors and the internal status of those orders. Through links with vendor’s fulfillment systems, it may also be track the status of the shipment from the vendors.

- Fulfillment System: A vendor's fulfillment system receives and tracks incoming orders from original receipt and acceptance to the final delivery.
- Marketplace System: Usually a subscribed service, the marketplace application maintains all of the necessary rules, pricing, inventory, and capacity information for the products a vendor is offering for automated commitment. It is up to the vendor to ensure that the information in the marketplace is accurate and correct. In a non-automated environment, the vendor's sales group performs this function.
- Chains of Commitment Product Configurator: For sophisticated products that require custom bidding, the product configuration creates a new commitment request, thus adding one link to the "CoC". In a non-automated environment, the sales department fulfills this function, which can take weeks to perform for a complex transaction.
- Bid Request/Analysis System: This system automates requests for bids for products to one or more vendors, then analyses the results of the bids with a scoring system that is based on the customer's preferences, with attributes such as vendor preference, time to deliver, availability of certain features or colors, etc. This system chooses the best available combination of vendor and product, then presents it to the order system for final acceptance and placement of the order.
- Chains of Commitment Analysis Engine: This component contains the historical information and scoring rules that are required to compute the bid scoring that is performed in the Bid Request/Analysis system. The key the CoC Analysis Engine is wide connectivity and a rich set of information related to a vendor's ability to deliver a quoted product within the quoted parameters of time, cost, and quality. The exact type and amount of data that's required will depend upon the particular scoring method chosen by the implementer, and is the subject of current research in collaborative supply chains [National Textile Center.

A fundamentally complete data flow diagram is illustrated in Diagram C. Minor transactions, the details of status updates, and other minor data flows are excluded for visual clarity. This data flow diagram is intended to be representative of common supply chain and demand chain approaches, and may be adapted to suit particular implementations or designs. Extending the details will be the subject of several research and/or product specification efforts. The author apologizes for the complexity of this diagram, but has been unsuccessful at devising a better means of presenting it in the prescribed format and in the time allowed.

Diagram C
Chain of Commitment Data Flow
 The individual data flows are labeled with capital letters and described below the diagram



- A. The demand side partner (customer) generates a request for bid.. This could be in response to an internal or an external request (i.e. a customer placing an order).
- B. A bid request is submitted to one or more marketplace systems. This initiates a large chain of requests, but only the immediate request is illustrated here.
- C. Marketplace system requests COC Product Configurator to analyze the request, determine the chain of commitment requirements, then submit one or more commitment requests, creating a new link in the chain. That is, this is spawning a new instance of data flow (A).
- D. After receiving all the vendor requests.
- E. The marketplace systems respond with a number of bids and their accompanying commitment keys (these are temporary commitments until a firm order is placed).
- F. The bid analysis engine submits a request for a CoC quality analysis from the CoC Analysis Engine for each bid under consideration.
- G. The CoC Engine computes a score for each vendor bid based on historical information and customer determined criterion.
- H. The Bid Analysis Engine selects from the available bids based on the score and sends commitment accepts and rejections to the marketplace applications.
- I. The commitment acceptance or rejection is forwarded to the product configurators for complex bids, which then send a chain of acceptances or cancellations for all the requests being held by them. This protocol remains to be developed as the subject of a separate research or development effort.
- J. The bid analysis engine sends the customer request to the Order Management System, where it is managed for the remainder of its lifecycle.
- K. The OMS sends the approved order to the winning vendor's fulfillment system.,triggering a series of (J) data flows throughout the chain of commitment.
- L. On the customer side, the order is tracked in the OMS by the customer or the customer's purchasing agents.
- M. On the vendor side, the order is managed in the Fulfillment System until it is shipped.
- N. All status updates (acceptance, rejection, ship confirmation, etc) are shared between the OMS and FS for the life of the order.
- O. When the order has been completed, the product is shipped via the designated carrier by the vendor.
- P. When possible, information from the package carrier is either sent to or pulled by the CoC Analysis Engine as part of the quality of commitment historical information.
- Q. The OMS and FS share data with the CoC Data Warehouse. This data is crosschecked and stored for future use.

The illustration suggests that all the parties in this complex transaction subscribe to the same application service provider. A common hosted app is desirable to better predict the probability that commitments will be met, but not essential to constructing a working system. Security practices must guarantee the privacy and statistical anonymity of the data, however, before common service providers will gain wide acceptance .

The greater number of partners that share information with a common CoC analysis database, the greater the potential of the system to provide value to its participants [Sturium 1999]. Even if each vendor uses a different set of systems for evaluation quality of commitment, however, the architecture is still valid.

3.1 Migrating Existing Architectures

Some components of this architecture exist today, but vary widely in implementation, quality, security, speed, and interoperability. For example, a particular seat manufacturer might have an online order system with a graphical web interface, but no interface to support automated ordering. A carpet vendor supports EDI transactions, but can only accept requests for bid and purchase orders. The paint vendor can accept online orders, but cannot provide any commitment information through the online system. None of these systems provides enough information to support an online chain-of-commitment process, and the variability between them makes it extremely difficult to develop unified measures of commitment quality.

One solution to this problem is to develop highly standardized protocols for exchanging chains of commitment information and standard “functionality” sets that each component of the architecture must or should support in order to maximize the value of the CoC approach.

Another solution is to simply develop the architecture into one or more standardized and deeply integrated product suites. In many ways, this may be faster and easier than attempting to develop widely adopted standards, but would be quite expensive and potentially difficult to market to data-protective clients.

3.2 Developing the Chains of Commitment Components.

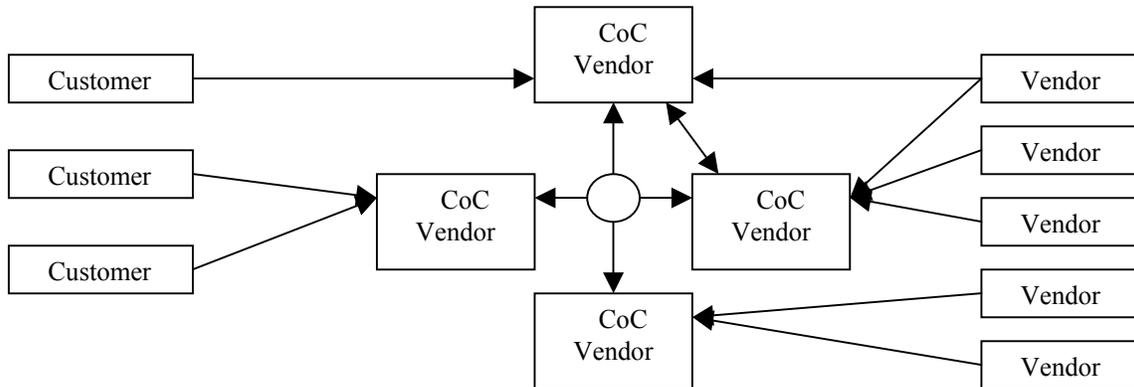
The new components – the CoC Analysis Engine and the CoC Product Configurator, in particular, need to be further analyzed and developed as new products. In all likelihood, they will be developed as extensions to existing configure-to-order and data warehousing/analysis products [Adrian.

The CoC analysis engine is clearly the newest and most significant enabling technology in this architecture. Since it benefits most by having information from the widest possible variety of data sources, it makes sense to consider the creation of public, rather than private, CoC Analysis engines operated by CoC “vendors”. (The case of privately managed databases held closely by partners was referenced earlier in this document. Any public or semi-public database will raise privacy and statistical anonymity concerns, which will have to be dealt with during the design of a specific implementation. The relative benefits of centralized and de-centralized data warehousing schemes should be the subject

of future analysis, and may well be dependent on the scoring schemes to be employed). By networking these CoC Vendors, it becomes possible to create a loosely integrated, potentially redundant CoC information network that is apparently seamless to the end users.

The variety and number of CoC paradigms, vendors, and protocols that spring up dictate the size, number, and complexity of these networks. If we interconnect these networks via protocol shifting mechanisms (giving rise to yet another type of potential service vendor), end users would have real-time access to CoC quality information for any vendor in any of the world wide interconnected CoC network (the WW-CoC network).

To help modulate the growth of this sub-industry, it would be valuable to establish, in advance, a combination of operating (application functionality) standards and integration (communication protocols) standards. For example, a standard CoC operations set (CoC-OS) and chains of commitment markup



language (commitML) might be defined to encourage interoperability and consistency of practices. This language might be an extension of existing supply chain markup languages, such as Value Chain Markup Language (VCML) proposed by Vitria Technology, Inc [Kain 2001].

The single ended arrows represent subscription from CoC subscribers (i.e. customers and vendors) to CoC vendors. The network linkage between the CoC vendors suggests a fee-sharing network that allows subscribers of different CoC vendors to access information from other network participants. The double ended arrow suggests the possibility that CoC vendors could establish direct information sharing agreements that bypass the information brokers. CoC Vendors may track orders, shipments, or both. The more direct the pathway between a customer and vendor, the more efficient the transaction will be in terms of transaction cost, thus providing incentive to CoC vendors to pursue market share, to maintain quality and standards of ethics, and to see relationships with other major vendors of CoC quality information.

Another option is to maintain the CoC engine within the demand side of the overall architecture. This limits its exposure to information that is available from within the company and, possibly, that might be downloaded from outside information sources. For many customers that might feel uncomfortable about

sharing potentially critical business information with an outside vendor, this might be a viable option. To enhance the overall architecture, it may be possible to develop a P2P protocol for sharing this critical “historical” information amongst multiple CoC users. However, continued research and development of collaborative networks will need to develop security and anonymity protocols to protect against unauthorized or illicit use of the information. In addition, all provided information should be cross checked between multiple sources to limit the possibility of falsified or inaccurate reporting by vendors.

4 CONCLUSIONS

A highly flexible and intelligent CoC system can be developed largely using existing components and a small number of new, fairly sophisticated components. Each part of the system may be distributed or centralized, and the overall value and quality of the complete CoC system will depend on the quality of communications between the component systems and the ability of the component systems to support the required operations.

A number of opportunities exist for eCommerce researchers, integrators, application service providers, B2B application vendors, and software developers, among others. The large number of options for distribution and operation of the components may be difficult for end users, but will provide systems architects with a wide range of options for integrating multiple diverse partners with highly heterogeneous internal environments.

The greatest benefit of a system such as this will like be to those businesses that provide highly configurable items, or businesses that want to maximally leverage a wide base of vendors for products. The ability to dynamically provide real-time quotes and accept online orders for even the most complex products will be viewed as a great benefit to the end consumer, and is likely to affect the vendor’s ability to make online sales. Also, any vendor that wants to increase their potential market exposure will find a hosted environment dramatically beneficial because they automatically become a potential “preferred” vendor every time a business or individual signs onto the system to make an online transaction. After the initial effort of setting up and maintaining the costing rules, the system runs fully automated.

Many of the risks of as system like this are readily apparent. Because quotations and delivery promises are made based on information provided by the vendor, there is always the risk that information might be inaccurate, out of date, or even falsified, which could result in late deliveries and shifting production schedules all the way up the CoC. The CoC software can help manage this by crosschecking independent sources of information such as the vendor, the customer, and the package carrier.

On the supply side, a very serious risk is the possibility that an accidental misquotation might result in hundreds of sales before the error is discovered. Because of this risk, it may be desirable to have a “review” period following the actual transaction during which any vendor has the option of refusing to honor the commitment. This review period would likely be configured by the supplier, and should be communicated to the end consumer based on the suppliers that are

in the chain. Functionality of this sophistication will be easier to manage if all or most of the parties are sharing information with a common, central system.

REFERENCES

- GARLAN, D. AND SHAW, M. 1997. Architectures for Software Systems. http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/tinker-arch/www/html/Tutorial_Slides/.
- FORGER, D. 2000. The Problem with Collaboration. *Supply Chain Management Review*. <http://www.manufacturing.net>.
- SCHIEBEL, W. 2000. The Value Chain Analysis of ECR Europe, Interpreting A System Innovation in Supply Chain. <http://www.boku.ac.at/iao/ecr/>.
- HAMMOND, J. 2001. Managing the Apparel Supply Chain in the Digital Economy. *Sloan Industry Centers Meeting: Corporate Strategies for the Digital Economy*.
- VERESPEJ, M. 2001. Automakers Put Wheels On Supply Chains. *The Value Chain*. <http://www.iwvaluechain.com>.
- CAMSTAR SYSTEMS, INC. 2001. The Virtual Factory: Managing Distributed Manufacturing in a Distributed Economy. <http://www.camstar.com>.
- NØKKENTVED, C. 2000. Collaborative Processes in e-Supply Networks. <http://www.pwcglobal.com/extweb/pwcpublications.nsf>.
- NATIONAL TEXTILE CENTER. Business-to-Business Collaboration in a Softgoods E-supply Chain. <http://www.ntcresearch.org/current/year10/Projects/i01-s01final.htm>
- STURIUM, R. 1999. Achieving Competitive Advantage through Supply Chain Integration. <http://www.vitria.com>.
- ADRIAN, M. .Making Sense of Collaborative Commerce. *Research Digest Vol. 4 Issue 3*.
- KAIN, R. 2001. Vitria Introduces VCML to Standardize Business Collaboration Across the Extended Enterprise. <http://www.vcml.net>