LOGICRUNCHER
A Logistics Planning and Scheduling Decision Support System for Emerging EMS and 3PL Business Practices

Raymund J. Lin, Jack Huang
Institute for Information Industry, Taipei, Taiwan, R.O.C.

Norman Sadeh-Koniecpol, Benjamin Tsai
School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213, USA

Keywords: Logistics Planning and Scheduling, Web-based Supply Chain Management, Electronic trading, Mixed Initiative Decision Support, Collaborative Planning.

Abstract: LogiCruncher is a dynamic logistics planning and scheduling module developed to support emerging third party logistics practices. Given information about inventory profiles for different product types at different locations, a set of transportation assets as well as a variety of quotes and contractual arrangements with logistics service providers, the system is capable of generating or revising transportation plans and schedules that meet changing customer requirements. These requirements are expressed in the form of demands for delivering different types of SKUs in different quantities to different locations. The system is capable of capturing a rich set of domain constraints and costs. It can be used to support the development and dynamic revision of solutions as well as to support requests for quotes from prospective customers. This includes support for "what-if" analysis through the creation and manipulation of solutions in different contexts, each corresponding to possibly different sets of assumptions. This paper provides an overview of LogiCruncher and summarizes results of initial evaluation.

1 INTRODUCTION

As manufacturing companies focus on their core competencies and attempt to further reduce supply chain costs and improve delivery performance, they increasingly turn to third party logistics (3PL) providers to manage both their inbound and outbound logistics. Examples of current day 3PLs include Fedex, UPS, DHL as well as many others. Original Equipment Manufacturers (OEMs) such as Cisco, Dell or Nokia go one step further and also outsource key manufacturing activities to Electronics Manufacturing Services (EMS) contractors such as Flextronics, Sanmina-SCI or Solectron to name just a few. EMS contractors are faced with the delicate task of concurrently coordinating manufacturing activities with inbound and outbound logistics for a number of OEMs. This involves dynamically orchestrating the activities of hundreds, if not thousands, of suppliers with operations at multiple manufacturing facilities and with logistics and warehousing activities around the globe - all to keep up with OEM demands that are typically adjusted on a daily basis. To satisfy demand from their OEM customers, EMS contractors tend to rely on a combination of both in-house and external warehouses and transportation assets. Within such environments, supporting high levels of product customization, increasingly short product life cycles and tight delivery commitments (all while minimizing costs) requires unprecedented levels of supply chain visibility and coordination (Figure 1). In this paper, we summarize ongoing work on LogiCruncher, a logistics planning and scheduling decision support tool aimed at supporting tight integration between procurement, manufacturing and logistics activities across the global supply chain environments spawned by emerging EMS and 3PL practices (Alp, 2003). In particular, we detail the overall architecture of LogiCruncher, focusing on the way in which it

J. Lin R., Huang J., Sadeh-Koniecpol N. and Tsai B. (2006).
LOGICRUNCHER - A Logistics Planning and Scheduling Decision Support System for Emerging EMS and 3PL Business Practices.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - AIDSS*, pages 176-181
DOI: 10.5220/0002464201760181

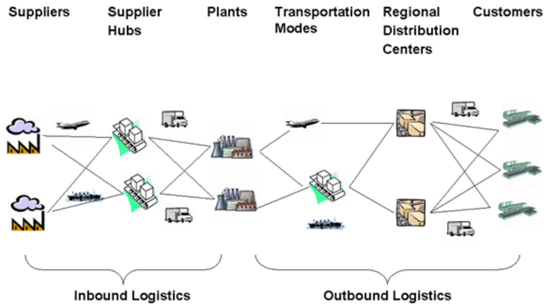

Figure 1: Effective supply chain management in emerging OEM/EMS practices requires unprecedented levels of visibility and coordination across global logistics networks.

supports dynamic coordination between manufacturers, suppliers and logistics providers under constantly changing conditions. This includes a description of the system's mixed initiative functionality to enable users to collaboratively explore alternative supply chain arrangements. We also detail the system's powerful modeling framework, which enables it to capture both in-house logistics and warehousing resources as well as quotes obtained by third party providers.

Specifically, the remainder of this paper is organized as follows. Section 2 provides a brief review of the literature and highlights key innovative aspects of LogiCruncher. Section 3 gives an overview of the system's overall architecture, including a discussion of different ways in which it can be configured to capture different possible business practices. An overview of the LogiCruncher logistics and warehousing model is provided in Section 4. Section 5 focuses on heuristic search procedures developed to support the rapid generation and revision of large-scale logistics and warehousing solutions under dynamic business conditions. Empirical results obtained with these procedures are summarized in Section 6. Section 7 contains some concluding remarks.

2 RELATED WORK

Traditionally, operations research has focused on somewhat stylized models of logistics planning and scheduling problems, favoring models that lend themselves to the computation of optimal or near-optimal solutions (e.g. (Cordeau, 2002; 2004; Li, 2005)). Over the past ten years, in parallel with this work, a number of research efforts have attempted to increasingly relax many of the assumptions made in more classical models. This has included looking at larger-scale problems (e.g. (Sadeh, 1996; Kott, 1998; 1999; Smith, 2004)), more dynamic models (e.g. (Sadeh, 1996), (Kott, 1999; Smith, 2004)), more complex constraints(e.g. (Sadeh, 1996; Kott, 1998; Smith, 2004)) along with support for more flexible mixed initiative decision models (e.g. (Kott, 1999; Becker, 2000; Sadeh, 2003)).

LogiCruncher is a logistics planning and scheduling decision support system that builds on our own work on a mixed-initiative decision support tool for collaborative supply chain planning and scheduling in the context of the MASCOT system (Sadeh, 2003), as well as our earlier research on developing iterative improvement techniques to build and dynamically update large-scale planning and scheduling solutions (Sadeh, 1997). LogiCruncher is unique in the way in which it combines these techniques within a flexible modeling framework capable of capturing a rich set of emerging EMS/3PL practices. This includes the ability to model hybrid networks of plants, warehouses, distribution centers and multi-modal transportation assets that include a mix of assets directly under the control of an EMS organization and assets made available by third party partners under different contractual arrangements.

3 OVERALL ARCHITECTURE

LogiCruncher is a decision support shell aimed at supporting mixed initiative planning and scheduling functionality required by emerging EMS/3PL business practices. The shell, which can be deployed at the level of an EMS or a third party logistics provider, aims to support users as they interact with other participants across the supply chain. This includes provisions for developing and revising logistics plans and schedules that cut across multiple suppliers, plants, warehouses and transportation assets. Some of these assets may be directly under the control of the user organization, while others may be provided by third party organizations subject to different types of contractual arrangements. This includes both long-term arrangements as well as more dynamic arrangements identified by issuing Requests for Quotes (RFQs – or more generally RFxs) and evaluating bids– see Figure 2. In particular, the shell gives its user access to a number of problem solving services, ranging from solution generation and revision services to services aimed at submitting RFQs, evaluating bids and even submitting bids (e.g. in the case of a large third party logistics provider). Using these services,

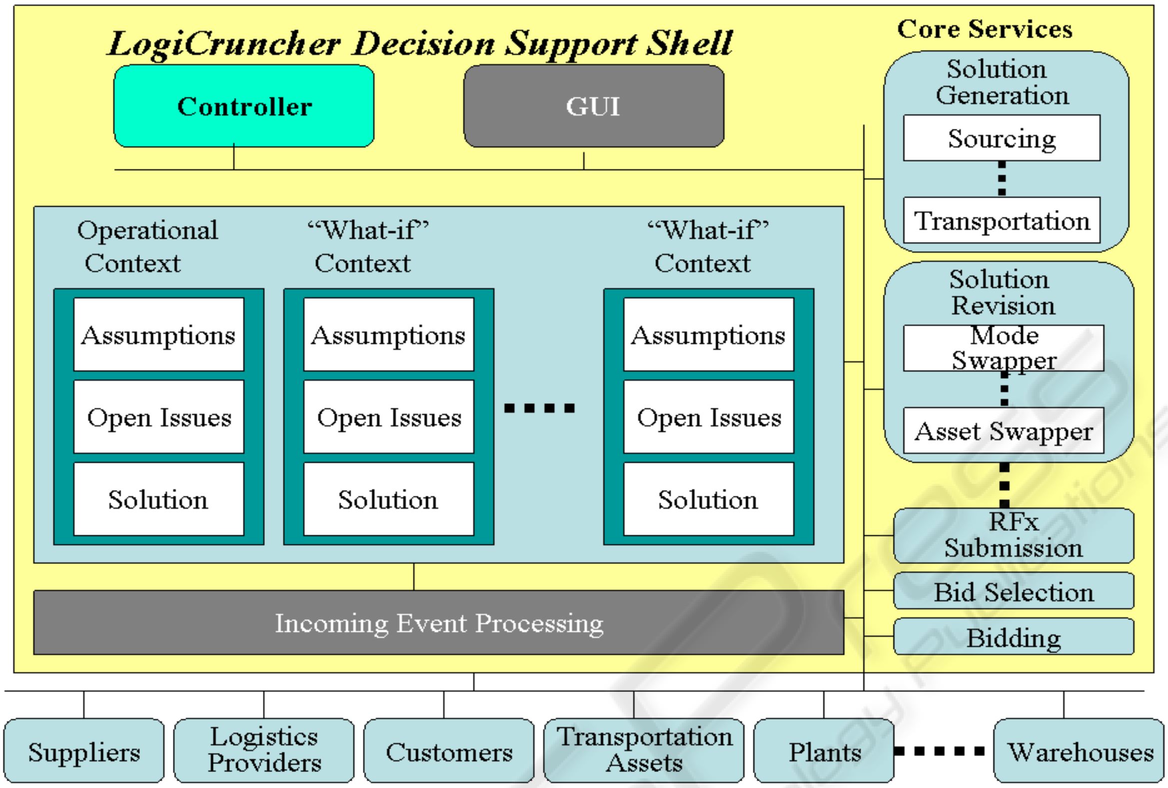

Figure 2: LogiCruncher decision support shell: overall architecture.

LogiCruncher users can concurrently develop, refine and evaluate multiple solutions, using "what-if" contexts that can differ in terms of working assumptions and solutions (e.g. different customer demand assumptions as well as different sourcing or logistics arrangements). As new developments unfold (e.g. changes in customer demands, arrival of new bids from logistic service providers, transportation contingencies, etc.), they are handled by an *incoming event processing* module that selectively updates relevant contexts (e.g. taking an incoming bid and posting it in the context that generated the corresponding request for quote). This can be done either automatically or manually by the user (e.g. if the user wants to selectively control events that are visible within a given "what-if" context). As they get posted into a given context, events result in the generation or updating of "open issues". An "open issue" is a flag that is used to identify aspects of a working solution that is either incomplete, inconsistent or unsatisfactory (see (Sadeh, 1998; 2003) for further details). Open issues can be used to help the user compare contexts and identify areas of a given solution that require further work. They can also be used to support automated decision support functionality that directly maps problem solving services onto different sets of open issues. Such mapping can be implemented through a control module ("controller") and can range from providing suggestions to the user to automatically invoking one or more services (e.g. in the form of scripts) – see (Sadeh, 1998; 2003) for further details.

4 LOGISTICS PLANNING AND SCHEDULING MODEL

A LogiCruncher **Context** corresponds to a set of assumptions and possibly a planning and scheduling solution developed under these assumptions. Assumptions include customer demands (expressed as "customer orders") to be satisfied, a set of available storage nodes, a set of transportation assets as well as quotes obtained from providers of transportation and warehousing services.

A **Solution** is an allocation of goods to customer demands, including a selection of sourcing nodes (e.g. plants, warehouses, distribution centers or some combination of the above), a selection of transportation modes and/or transportation assets to move these goods ("transportation plan") and a schedule for when each move is to take pace ("transportation schedule"). A solution may be complete or incomplete and may possibly include some assumption and/or constraint violations, in which case it is said to be inconsistent.

An **Order** represents a requirement for delivering some quantity of a given SKU type to a point of destination by a given date. An order will contain information on due time, earliest acceptable delivery time, latest acceptable time, and various penalty costs. Each order is broken down into one or more jobs that inherit the properties of the order, and which together will satisfy the requirements of the order. The fashion in which orders are split into jobs may reflect an organization's policy or contractual arrangements or may be the result of limiting sourcing and transportation conditions. Orders can originate directly from actual customer requirements or may reflect a company's policies such as safety stock policies.

LogiCruncher can capture constraints associated with different transportation modes and be used to model internal transportation assets as well as assets operated by third party providers under different contractual arrangements. This includes quotes obtained from third party logistics providers as well as longer-term contractual relationships.

Transportation modes as well as different classes of transportation assets vary in terms of their available routes, costs, speeds and capacities as well as other idiosyncratic constraints (e.g. type of products they can accommodate, setup constraints, etc.).

Inventory Storage Resources represent resources that can store components. These resources can be used to model warehouses, suppliers, plants and customer delivery sites. They can have capacity constraints and restrictions on the different types of SKUs they can accommodate. Each inventory storage resource also has an inventory profile for each SKU it can store. This profile indicates how many units of each SKU is expected to be available at that resource at any point in time, given existing problem assumptions and decisions made in the current context.

5 ITERATIVE SEARCH TECHNIQUES

In LogiCruncher, solutions are developed and refined through the activation of core problem solving services, either manually or automatically. Core problem solving services include (see Figure 2):

- Solution generation services such as services to help optimize the nodes from which to source SKUs required by different orders, transportation planning and scheduling services, etc.
- Solution revision services, which take an existing solution (possibly partial or inconsistent), and revise it to either resolve some inconsistencies, complete the solution or improve it. This includes services aimed at exploring alternative sourcing options, the selection of alternate transportation modes or of different bids from logistics service providers. It also includes finer revision services such as services to swap resource allocations between different orders in hope of producing a better quality solution (e.g. lower inventory costs, lower transportation costs, lower delivery penalties, etc.)
- RFQ submission services that can be used to issue requests for quotes to prospective business partners (e.g. RFQs sent to third party logistics providers)
- Bid selection services implementing logic that can be used by LogiCruncher to select among multiple bids (e.g. through what-if analysis in multiple contexts and/or through the use of solution revision services)
- Bidding services, in configurations where LogiCruncher is deployed to assist a 3PL and help the organization decide what to bid on and help it determine optimal bid parameters.

This flexible architecture enables a number of possible system configurations as well as a variety of mixed initiative problem solving styles, where solution construction and revision is interactively controlled by an end-user, while tedious or complex problem solving steps can selectively be delegated to automated functionality (e.g. through control heuristics embedded in the shell's controller). Experience with this mixed initiative architecture in the context of collaborative supply chain planning and scheduling scenarios is detailed in (Sadeh, 2003).

In its simplest configuration, LogiCruncher can be used to generate initial logistics planning and scheduling solutions using its solution generation services and can then be used to revise these solutions as events unfold (e.g. as contingencies occur, as demand changes or as new bids from prospective business partners arrive). In its current implementation, the LogiCruncher solution generation services rely on a set of greedy heuristics to rapidly generate initial solutions. Iterative improvement techniques embedded in the form of multiple neighborhood search heuristics can selectively be invoked to further refine or improve these solutions, whether in support of what-if scenarios or to reflect changing conditions. The following section briefly summarizes empirical results obtained with these techniques.

6 EMPIRICAL EVALUATION

An initial version of LogiCruncher has been implemented in Java. Experiments conducted on scaled down logistics problems involving 600 customer requests, 10 to 20 warehouses, distribution centers and customer delivery centers, multiple classes of transportation assets, each with between 10 and 100 transportation units (e.g. 50 trucks, 100 vans, etc.) appear rather promising. Initial solutions are generated in a matter of a few seconds. Iterative improvement heuristics have been shown to typically converge towards seemingly high quality solutions within 30 to 120 seconds, though admittedly additional experimentation is needed to further evaluate the quality of these solutions. Figure 3 summarizes results obtained on eight problem sets

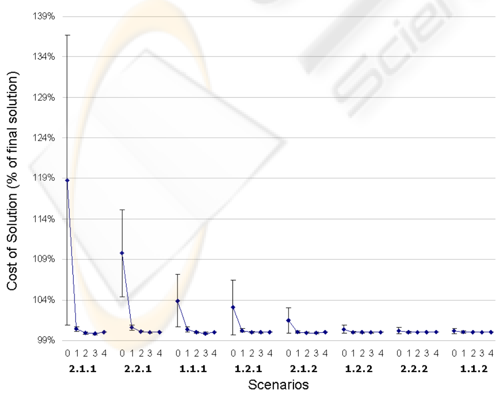

Figure 3: Performance of iterative improvement heuristics on eight categories of logistics planning and scheduling problems. A total of 6 problems were generated in each category.

that differ in terms of logistics network layout, mix of transportation assets and tightness of customer requests. The results show improvement in solution quality, starting from an initial solution generated by the LogiCruncher solution generation heuristics, and applying multiple rounds of iterative improvement search.

7 CONCLUDING REMARKS

In this paper, we introduced LogiCruncher, a logistics planning and scheduling decision support shell aimed at supporting emerging EMS and 3PL logistics scenarios. We focused in particular on key elements of the LogiCruncher mixed initiative decision support architecture as well as on its unique modeling capabilities. These capabilities enable the system to capture complex constraints and costs under which EMS and 3PL service providers need to operate, with logistics networks consisting of a mix of transportation and warehouse assets, some operated by these companies themselves and others operated by third party players. Initial evaluation of the system's heuristics appear promising. As part of our future work, we plan to further refine elements of our model and evaluate an enhanced set of heuristics on yet larger sets of problems to be identified jointly with prospective end-user organizations based in the Republic of China (Taiwan).

ACKNOWLEDGEMENTS

The work reported herein has been supported in part by the R.O.C.'s Institute for Information Industry and in part by the US National Science Foundation under ITR Grant 0205435.

REFERENCES

Alp, O., N.E. Ekrip, R. Gullu (2003). Outsourcing Logistics: Designing Transportation Contracts Between a Manufacturer and a Transporter, Transportation Science 37(1) 23-29.

Becker, M. and S.F. Smith (2000, April). "Mixed-Initiative Resource Management: The AMC Barrel Allocator", Proceedings 5th International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2000), Breckenridge, CO

Cordeau, J.-F., M. Gendreau, A. Hertz, G. Laporte, and J.-S. Sormany, (2004). New heuristics for the vehicle routing problem. In: Logistics systems: Design and optimization. Eds.: A. Langevin and D. Riopel, Kluwer Academic Publishers (to appear - earlier version as Technical Report G-2004-33, GERAD, Université de Montreal, Canada).

Cordeau, J.-F., M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet, (2002). A guide to vehicle routing heuristics. Journal of the Operational Research Society 53(5) 512–522.

Kott, A., and V. Saks (1998, June). A Multi-Decompositional Approach to Integration of Planning and Scheduling – An Applied Perspective, in Proc. of Workshop on Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments. International Conference on Artificial Intelligence Planning Systems, AAAI Press TR WS-98-02

Kott, A., V. Saks, and A. Mercer (1999, Spring). A New Technique Enables Dynamic Replanning and Rescheduling of Aeromedical Evacuation, AI Magazine, Vol. 2, No. 1, pp. 43-54

Li, F., B. Golden, E. Wasil, (2005). Very large-scale vehicle routing: New test problems, algorithms, and results. Computers & Operations Research 32(5) 1165–1179.

Smith, S.F., M.A. Becker and L. Kramer (2004). "Continuous Management of Airlift and Tanker Resources: A Constraint-Based Approach", Mathematical and Computer Modeling: Special Issue on Defense Transportation

Sadeh, N.M., D.W. Hildum, and D. Kjenstad (2003). "Agent-based e-Supply Chain Decision Support", Journal of Organizational Computing and Electronic Commerce, Vol. 13, No. 3

Sadeh, N.M., D.W. Hildum, T.J. Laliberty, J. McA'Nulty, D. Kjenstad, and A. Tseng (1998, June). "A Blackboard Architecture for Integrating Process Planning and Production Scheduling". Concurrent Engineering: Research and Applications, Vol. 6, No. 2

Sadeh, N.M. and A. Kott (1996). "Models and Techniques for Dynamic Demand-Responsive Transportation Planning", Technical Report CMU-RI-TR-96-09, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3891.

Sadeh, N.M., Y. Nakakuki, and S. Thangiah (1997). "Learning to Recognize (Un)Promising Simulated Annealing Runs: Efficient Search Procedures for Job Shop Scheduling and Vehicle Routing". Annals of Operations Research, 75, pp.189-208