DECIDE: AGENTS CONTROLLING A BHS OF AN AIRPORT HUB

Received 2nd April 2007; accepted 30th April.

Kasper Hallenborg, Claus Risager

Abstract:

It is an ingrown fear to most travelers, frequent or not, to lose their baggage while flying from one destination to another, but if you take a look behind the scene you will experience a system with a complexity unlike most others.

The baggage handling system (BHS) of an airport is just one link in a long chain of processes your bag is going through. The BHS stretches from check-in or unloading of a plane onto the BHS (connecting flights), to it is collected at the gate and transferred to your flight.

The conventional control software uses a strategy primarily based on a shortest path algorithm, not taking into account dynamical changes or utilizing less packed areas of the system to increase capacity.

We changed that perspective towards a decentralized multi-agent based solution by developing strongly collaborating agents, to replace the original control soft-ware. The hierarchy of agents horizontally spans the system through local agents capable of altering the flow of totes, and vertically through mediator agents, which have a global perspective and support decisions of the local autonomous agents.

The local agents monitors their own local neighborhood for rising queues and other delaying factors, which aggregate to a status of the node, influencing the decision of other agents trying to route totes through this node. Interactions are handled as a combination of contract-nets, queries, and subscriber interaction schemes, which generalized the agent interfaces from the specific application. An agent-based approach not only improves robustness of the system, and utilize the entire BHS in a more convenient and dynamical way, it also include strategies for maximizing capacity of the system, which follows a work in-progress against capacity curve (WIPAC).

In this paper we present the agent-based design of the control-software, and elaborate on the mutual collaboration compliant with the FIPA interaction schemes. We present ontologies used for sharing information and services among the agents as well as the results of our work for a major airport hub in Asia, and compare it against the traditional centralized strategies. We complete the paper by drawing some conclusions and present ideas for future work.

Keywords: *control, optimization, aircraft transportation*

1. Introduction

The baggage handling system (BHS) for transfer baggage in major airports is similar in setup and functionality to many job shop and manufacturing systems baggage enters the systems through various channels, undergoes various processes (mainly routing), before it leaves the BHS at the departure gate.

A BHS is a huge mechanical system, usually composed of conveyor-like modules capable of transferring totes (plastic barrels) carrying one bag each. We have researched a BHS for a major airport hub in Asia, with more than 5000 of these modular components each with a length between 2-9 meters, which run at speeds from 2-7 meters per second. The BHS alone can easily be up to 20 km in total length and may cover an area of up to 600.000 square meters. A BHS should be capable of handling more than 100.000 pieces of baggage every day, and for the airport we have researched the maximum allowed transfer time for a suitcase is 8-11 minutes for a distance of up to 2.5 km. The figure below shows a tiny part of a BHS.

1.1 The core task

The overall task of a BHS could be described as simple as transferring bags from A to B, but the highly dynamic environment complicates the control and optimization of capacity in the system. Changes in flight schedules, lost baggage information, and breakdowns are some of the factors, which in combination with peak loads on the system result in queues and delayed baggage.

In many airports the BHS only takes care of transit baggage for connecting flights. All check-in baggage is distributed to the correct flights using a traditional sorter system, unless the passenger checks in long before departure, so the bags need temporary storage. In general the BHS could be described as a production system with a number of input facilities (toploaders) receiving baggage from arriving planes (or early check-in baggage), which is the procurement to the production system [1].

Apart from toploaders, the system has a dischargers at the departure gates, which temporarily are allocated to one or more flights. If a bag enters the BHS before a discharger has been allocated to the flight, it must be sent to temporary storage (EBS Early Baggage Storage) or claimed by the passenger.

BHSs are typically built by conveyor-like systems with totes* or DCVs transporting the baggage. As totes or DCVs in many systems have to stop or slow down when discharging or unloading, respectively, the capacity for that lane section goes down, and a queue can accumulate behind the discharger, thus more dischargers are often allocated to the same flight to distribute the load.

Identity and destination of the bags is unknown until scanned at the input facility, as no reliable model exists for arriving bags, which together with complexity and time constraints makes exact off-line scheduling impossible.

Traditionally the control software of the BHS is built on a rather simple centralized, but reliable, approach based on static shortest path calculations of the system. Each pre-calculated route between toploaders and dischargers are given a route number and when the destination of a bag is known by the system, it follows that route until it reaches the destination – no redirections during the transfer, unless there has been a lane breakdown.

The structure, complexity and task of the BHS make is an appropriate candidate for an agent-based control system.

1.2 MAS technologies

MAS technology, which spawned from artificial intelligence as DAI (Distributed Artificial Intelligence) [2], offers an approach to decompose complexity of systems into a number of collaborating autonomous agents. System-wide tasks are solved partly by subtasks in the individual agents, which are coordinated and aligned through their interaction patterns.

Interaction schemes and communication protocols for agents can be specified or programmed in an ad-hoc or domain specific manner, but to increase common understanding and platform independence, FIPA** provi-des a set of specifications for interaction protocols sup-porting both negotiation and co-ordination between agents [3]. The protocols have analogies in human societies and are adopted or extended for software agents. Examples of such interaction patterns are auctions, publisher-subscriber relationships, and queries. Also the terminologies in agent messages can be generalized using ontologies, which have roots in knowledge engineering, and it decreases the dependencies among the agents and the underlying platform.

1.3 Decide

Our research case of the BHS for the airport in Asia is conducted in collaboration with the company installing and producing the BHS, FKI Logistex. This case is part of a larger research project called DECIDE, which focus on promoting and proving the appropriateness of multi-agent based control in production and manufacturing systems.

Major Danish manufactures are among the other partners of the consortium: Lego, Grundfos, Bang and Olufsen (B&O), and Odense Steel Shipyard. The project has been supported by The Ministry of Science, Technology and Innovation of Denmark.

2. Approach

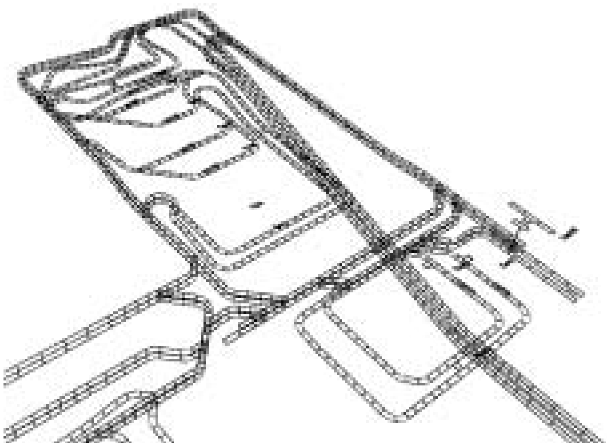
Recent year's advancement in computer and graphics performance has made it possible to do realistic real-time simulations of very complex environments, including productions systems like the BHS. The ability to continuously interact with the simulation model during operation creates a perfect off-site test-suite for the control-software, which emulates the real BHS.

2.1 Emulation model

Together with another consortium partner, Simcon, the BHS company FKI Logistex has created an emulation model of the researched BHS using the AutoMod simulation and modeling package [4]. AutoMod is a de-facto-standard for systems analysis of manufacturing and material handling systems.

One of the strong advantages of using AutoMod is that you can communicate with the model over a standard socket connection, which is almost identical to the connection between the control server and the PLCs in the real hardware. Thus the control software cannot see the difference, if it is connected to the emulation model or the real hardware. The same protocol and telegrams are used, which simplifies the development process, and makes the emulation model reliable, whenever the basic communication has been tested correct.

A snapshot of the emulation model is shown in the figure below. It shows the area with input facilities for terminal 3 of the airport.



In our agent-based approach the connection is handled through a special agent name, AdapterAgent, which act as a gateway between the agents and the emulation model. So the agents are still virtual collaborating processes running on a centralized server, therefore the agent approach can replace the traditional control software with no hardware changes.

*) A tote is a container into which your baggage is inducted upon entering the system, and a DCV (Destination Coded Vehicle) is a small train-like wagon running on a rail-system.
**) FIPA is an IEEE standards organization that promotes agent-based technologies.

2.2 Agent based design

Given the layout of the BHS the most intuitive approach of decomposing and decentralization the decision logic, is to add agents anywhere the flow of bags can change. The main reason for this approach is that no practical arguments exists for changing decisions for bag on the conveyor lines between the nodes in the graph spanning the routes of the BHS*, as no changes in direction can be made before it reaches a new node of the graph.

The node agents corresponds mainly to diverter and merger modules in the real setup, but other specialized modules exists, such as lifts and cross-transfers, which can direct a bag in 4 directions.

Ideally, with respect to generality and simplicity in the design of the agents and control software, the domain of observation for each agent should only span the edges from the node to the next node in the graph, which means the input and output lines of conveyors at each node. But for some decisions, agents require more information about the current status of the load on the entire system, e.g. when deciding between alternative dischargers far from the current location.

In order to support such kind of system-wide collaboration of agents, a number of mediator agents assist the node agents with dynamic information about the routes and load on the system. The RouteAgent is the most important mediator agent, which can be queried for information about routes from one node to others in the BHS. The route information contains information about the nodes along the route, which corresponds to the agents that the node agent should collaborate with, but it also informs about delays on different routes, so e.g. a dynamic shortest-path route can be chosen.

More details of the individual agents are given in the sections below. Only a subset of the most important agents is described.

ToploaderAgents

ToploaderAgent waits for arriving baggage, which can then be inducted (or toploaded) into an empty tote, and starts its journey on the BHS. Empty totes (the transport barrels on the BHS) come from a stacking storage and the bags comes from traditional conveyor belts.

DivertAgents

DivertAgent represents divert elements in the BHS, which basically can send a bag along one of two output directions. So here is an option for choosing between routes, if both output legs can lead to a legal destination. A special variant of the divert elements are the so-called cross-transfers, which can send a tote along all four directions (left, right, forward, and backward).

MergeAgents

When the graph has diverted elements, there must also be merging elements for the graph to be connected. The task of the merger is to forward totes from the two input legs and send them along the single output leg. Adjusting the ratio between the two input legs is the primary task for the MergeAgent.

DischargeAgents

DischargeAgent corresponds to a discharger in the system, which besides being allocated to different flights also has the task of emptying totes and sending the empty totes back to the tote stackers. Attention must be given to the task of the dischargers, because even though the job seems rather simple (discharge the present tote or not), the empty totes must be directed back to the tote stackers, which are located near the toploaders. The empty totes shares the conveyor lanes with the full totes. That is a focus point for optimal resource utilization, which complicates the control of the discharger and the decisions of choosing between alternative dischargers at the node agents.

RouteAgent

The RouteAgent is a mediator agent, which allows the node agents to request various informations about the topology of the system. Is a certain destination reachable, all functional routes to a given destination, and the list of nodes (diverter and mergers) along the route to a given destination, etc.

2.3 Agent interactions

In order to increase the generality of the agents, and prepare the software agents of the BHS to be used for other manufacturing systems with little or no modifications, it was decided to specify both agents and their interactions against the FIPA standard.

F IPA interactions schemes are based on human interaction models closely related to micro economic trade schemes. Negotiation between agents complying with the FIPA standards therefore follow common understandable interaction schemes such as auctions, publisher-subscriber relationship, contract-nets, queries, etc.

To illustrate the principle of interactions used in the implementation, a toploading task is described in the following section.

Toploading bags

Initialization of a transport on the BHS is coordinated between two consecutive, but independent, stimuli from the emulation model, the event of a new bag arriving and being scanned at the toploader, and another telegram when it is being inducted into an empty tote, and released from the toploader. These stimuli are named `item_detected` and `item_inducted`.

The purpose of paying attention to scanning of items serves to two purposes. First the identity of the bag and destination in the BHS is revealed, which must be tied to the tote at induction, but it also enables a pre-planning of the route, when the destination is known. Thus negotiation with the corresponding dischargers can be handled before the tote is launched at the `item_inducted` stimuli.

*) In theory the flow could be more fluent, thereby increasing the capacity, if bags can avoid stopping and minimize acceleration and deceleration, but in practice that should be controlled by the strategies of the node-agents.

As mentioned above each flight departure could be associated with a number of dischargers to spread the load on the system. A contract-net is an obvious choice for negotiating with the dischargers; the toploader has a task, dischargers can "solve" the task, and the toploader agent takes the final decision. After deciding which discharger to use, the toploader agent must co-ordinate with the node agents (diverters and mergers) along the path to route it towards the discharger. Depending on the strategy used for the routing, the toploader would either make an agreement with all agents along the path or alternatively just the first one. In the later case the route might be re-directed when the tote approaches that agent, as it re-calculates the best choice at the time. We call the two strategies *Routing by static shortest path* and *Routing on the way*. In general FIPA interactions form a unified approach to develop agents and simplify the development process. Each agent can be designed and implemented individually, if we can clarify its role in the community. It must support interaction mechanisms according to the agent's domain knowledge and understand how its role could be relevant to other agents.

2.4 Ontologies to generalize communication (relax the coupling)

Ontologies are mechanisms to provide meta-level communication between agents [5]. Ontology is similar to an object hierarchy and can be used at many levels of abstractions. Agents compliant with the FIPA standard follow the FIPA-ACL communication act ontology, which specifies both the syntax and semantics of the message structure. On top of that the FIPA-SL ontology provides standards for encoding the actual content of the message, including abstract descriptors used for queries. We have specialized the content language with an ontology that follows the concepts and terminology of the BHS system.

The strengths of using ontologies are to simplify message parsing, but more important the ability to send abstract messages used for queries, proposals, etc.

Query messages for the RouteAgent are good examples of ontology-based messages. Two concepts are understood by the RouteAgent:

- RouteBetween, a concept used for queries to other agents along parts of a route or a full route.
- LineBetween, a more fine-grained concept providing all details about elements in the line of connected nodes of the BHS.

An encoded message of a query could look like:

```
(iota
   :Variable (Variable :Name x :ValueType set)
   :Proposition (routeBetween
      :origin (element :elementID DFB01.TLA001)
      :destination (element :elementID DLA02.DIA023)
      :viaPoints (Variable :Name x :ValueType set)
      :numNodes 0
   )
)
```

which mean a proposition should be meet by finding only one (`iota`) list of points along the route between the origin and destination, as the variable (`x`) is set at the `viaPoints` entry.

3. Agent strategies

The primary reason for exchanging the conventional control software with an agent-based approach is to decrease the level of complexity in each decision node, which allows more advanced strategies to extend the routing of totes, in order to increase robustness and maximize capacity of the BHS. In the following sections two examples of strategies for route observations and capacity utilization are described.

3.1 Queue observations

Each node agent in the BHS collects information about the status of its local domain, which means the edges (conveyor lanes with no redirecting possibilities) to the previous and the following node of the graphs.

The information collected expose values for a edge/lane, such as the number of totes per element, the average delay for totes, and the average urgency of bags, which means how close a bag is to its departure (in time).

Queue observations are used in various decisions throughout the system, and queue statuses are exchanged by agents using the interaction schemes to support the decisions. Queue status helps an agent decide whether it is appropriate to send more totes along a route or alternative routes should be searched.

A special collaborating strategy between neighboring diverters and mergers has been used with success to speed up urgent baggage by sending other bags on a minor detour. Urgent baggage is lined up along the shortest path out of a diverter, and non-urgent baggage can be sent by an alternative route if the load of the area is high, indicating the delays and number of totes. Of course, if there is no pressure on the system, all totes follows the shortest path. When the routes merge again at the mergers, it will give higher priority to totes from the merging leg with the most urgent baggage; by this means we have established a method for totes to overtake others in the BHS, which is impossible using the conventional control software.

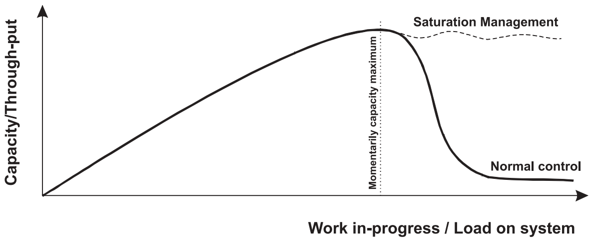
3.2 Saturation management and the WIPAC curve

Another important strategy is trying to avoid queues at all by minimizing the load on the system in critical areas. We assume everybody is familiar with slow starting queues of cars at an intersection, when the light turns green. Acceleration ramps and reaction times relative to drivers ahead accumulate to long delays in traffic queues, even though in theory all drivers should be able to accelerate synchronously (no reaction time). The same problem arises in the BHS, where reaction times correspond to the delay of the element head reporting clear*.

These matters result in the characteristics of the work in-progress against capacity curve (WIPAC), which is further described in [6], which in principles states that the capacity of the systems goes dramatically down,

*) In the mechanical setup of the BHS a tote can only be forwarded from one conveyor element to the next element, if that element is clear. A synchronized row of totes can then pass at full speed, from one element to another, but in queue situations acceleration ramps delays each element

if the load on the systems exceeds a certain threshold value, as indicated in the figure below.



The curve is dynamical, due to the various and changing load on the system, and the maximum cannot be calculated exactly. Thus the strategy is to quickly respond to minor observations, which indicates that the maximum has been reached, and then block new inputs to the area. We call this approach for saturation management, and currently we block a toploader if the routes from the toploader are overloaded. Queues close to the toploader are most critical, as the toploader have great impact on filling up those queues, whereas the parts of the route far from the toploader could easily have been resolved before the new totes arrive. Instead of blocking the toploader completely, we can just slow down the release of new totes using the following fraction of full speed for the toploader.

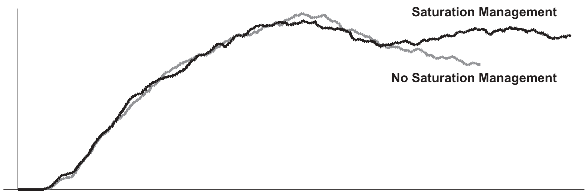
$$\nu_\tau \frac{\Sigma_i w_i q_i}{\Sigma_i w_i} = \nu_\tau \frac{\sum_i \frac{\alpha}{d_i} q_i}{\sum_i \frac{\alpha}{d_i}} \quad (1)$$

where ν_τ is the full speed of the toploader, and w_i are weights of the queue statuses, q_i , along the routes. The weight is given by a coeffi-cient α and the distance from the toploader d_i. Queue status is a number between 0 and 1, where 1 indicates no queue.

4. Results

Beside the more implicit results of simplifying the development process of new control software, and generalize control of the agents, our experiments have shown significant results regarding performance and throughput of the BHS.

Results related to overtaking totes and queue accumulations are easy to monitor in the emulation model. While the effect of the saturation management strategy is more appropriate to compare in the graph below.



5. Conclusion and future work

Is this paper we have presented important research contributions from the DECIDE project about multi-agent based control of a baggage handling system (BHS) in a major airport hub in Asia.

The agent-based approach has spread the decision and control logic of the system to a large number of collaborating agents and replaced a complex centralized control structure. It has enabled new strategies and observations in the local agents to increase robustness, capacity, and throughput of the BHS compared to the traditional approach.

5.1 Future work

We continue our research on the BHS and will develop more new strategies for the local agent, and increase their mutual collaboration to maximize the utilization of the BHS during peak times.

We will try to minimize the use of mediator agents by giving roles and profiles to the agents, as mediators become a bottleneck in communication between the agents. Ideally a swarm of local agents provide the most general setup, which easily can be ported to other manufacturing and material handling systems.

AUTHORS

Kasper Hallenborg* - Maersk Institute, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark, e-mail: hallenborg@mmmi.sdu.dk.

Claus Risager - Centre for Robot Technology, The Danish Technological Institute, Forskerparken 10, DK-5230 Odense M, Denmark.

* corresponding author

References

[1] S. Bussmann, N. R. Jennings, and M. Wooldridge. *Multiagent Systems for Manufacturing Control*. Springer, 2004.

[2] G. Weiss, editor. *Multiagent Systems – A modern approach to distributed artificial intelligence*. MIT Press, 1999.

[3] FIPA - The Foundation of Intelligent Physical Agents. http://www.fipa.org/, 2006.

[4] Brooks Software. Automod – simulation and modeling http://www.brookssoftware.com.

[5] S. Willmott, I. Constantinescu, and M. Calisti, *Multilingual agents: Ontologies, languages and abstractions*. In: Proceedings of First International Workshop on Ontologies in Agent Systems, Montreal, Canada, 2001.

[6] A. Kragh. *Kø-Netværksmodeller til Analyse af FMS Anlæg*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, 1990.

[7] S. M. Deen, editor, *Agent-Based Manufacturing – Advances in the Holonic Approac*, Springer, April 2003.

[8] I. McGregor, *The relationship between simulation and emulation*. In: E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, editors, *Proceedings of the 2002 Winter Simulation Conference*, pages 1683-1688.

[9] W. Shen, D. H. Norrie, and J.-P. A. Barthes, *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*, Taylor & Francis, 2001.