21

Decentralized scheduling of baggage handling using multi-agent technologies

Kasper Hallenborg
University of Southern Denmark
Denmark

1. Introduction

Agile and lean manufacturing are the trendy buzz-words among manufacturers today. Low-volume high-variety product series have radically changed the conditions for manufacturing systems.

To handle the new challenges and continuously optimize production flow, new requirements for flexible manufacturing systems have been introduced. Moving away from dedicated hardware and mass production additionally stress the disturbances of a dynamic environment. Whereas dedicated hardware are optimized system-wide and not expected to perform unless it is fully operational, flexible manufacturing systems should deal with these circumstances in the line of product variety, because the production usually allow the system to continue under restricted conditions.

It seems evident that not only are these systems harder to build in hardware, as they should be able to handle different products, but controlling them in an optimal manner is far from being a trivial development task.

In this chapter we will present novel approaches and strategies for developing a multi-agent based solution to control a baggage handling system of a larger airport hub.

The baggage-handling system (BHS) is similar in setup, complexity, and operation to many manufacturing systems. Items enter the system at input facilities, the *procurement* of the system, undergo some processing at various stations during its way to the output facilities – *shipping* in classic production terminology.

The research activities being described are conducted on a real case study of a BHS in Asia, and the project is part of a larger research project, called DECIDE, which seeks to prove and evaluate the feasibility of using multi-agent based control in large manufacturing systems. The project has been supported by the Ministry of Science, Technology, and Innovation in Denmark.

The chapter is structured as follows. We start by giving a detailed introduction to the evolution of manufacturing systems towards the flexible setup suitable for agent-based control. Next we provide an introduction to the FIPA specifications used to generalize agent interoperability and design.

Then we give a general introduction to the baggage handling problem, and briefly describe the case of the Denver International Airport, which among airport managers still may be a warning to experiment with intelligent baggage handling systems.

Source: Multiprocessor Scheduling: Theory and Applications, Book edited by Eugene Levner,
ISBN 978-3-902613-02-8, pp.436, December 2007, Itech Education and Publishing, Vienna, Austria

Before going into details about the strategies for the agents, we describe how the BHS has been agentificated (how agents have been mapped to the entities of the BHS) and how they are organized.
The strategies being documented give a limited, but not exhaustive, set of collaborative agent interactions to control important parts of the BHS. Some results and discussions follow before the general conclusions and our plans to continue work on the system in the future.

2. History

Beginning in the industrial ages, high-volume low-variety products were the new trend among manufacturers resulting in low-cost high-quality products. To begin with customers were satisfied by the new opportunities realized by mass production, even though customer requirements were not the driving forces in product design. Due to low competition in markets manufacturers were more concerned with production efficiency than customer requirements (Sipper & Bulfin, 1997).
Likewise dominating management theories of that time focussed on rationalization, such as Taylor's *scientific management* (Taylor, 1911).
Improvements in automation technologies led manufactures to see the possibilities of exchanging labour-intensive tasks with specialized machines and material handling systems to rationalize production. The automotive industry was among the first to take advantage of automation; Oldsmobile Motor introduced a stationary assembly line in 1907, followed by a moving assembly line in 1913 at Ford's new factory Highland Park in Michigan, even handling parts variety (Sipper & Bulfin, 1997).
For decades mass production, automation, rationalization, and scientific management were the dominating factors in manufacturing, but that gradually changed towards the end of the 20th century. Especially, due to the growth in international competition, market demands pushed forward new challenges for manufacturing – flexibility and customization. Japanese were the first to address the new conditions and changed from mass production to lean production. Instead of focussing on having high-volume and rationalization as the key drivers in developing of mass production environments, lean production focuses on the whole process of production; eliminating inventory, declining costs, increased flexibility, minimizing defects, and high product variety.
As trends in the automotive market changed customers were no longer satisfied by standard cars, but required customization (Brennan & Norrie, 2003). Lean production is focussed on not only production layout, but also on introducing flexibility in control and scheduling.

2.1 Flexible manufacturing

As flexibility was commonly accepted as one of the primary non-functional requirement to new manufacturing systems, research and development initiatives naturally concentrated on means and technologies to cope with the new demands.
The notion of a *flexible manufacturing system* (FMS) was born when Williamson in the 1960s presented his System24, a flexible machine that could operate 24 hours without human intervention (Williamson, 1967).
Computerized control and robotics were promising tools of the framework for automation with increased flexibility. Obviously not all products or systems would benefit from or require increased flexibility, but FMS was intended to close the gab between dedicated

manufacturing hardware and customization, as outlined by Swamidass (Swamidass, 1998) in figure 1.

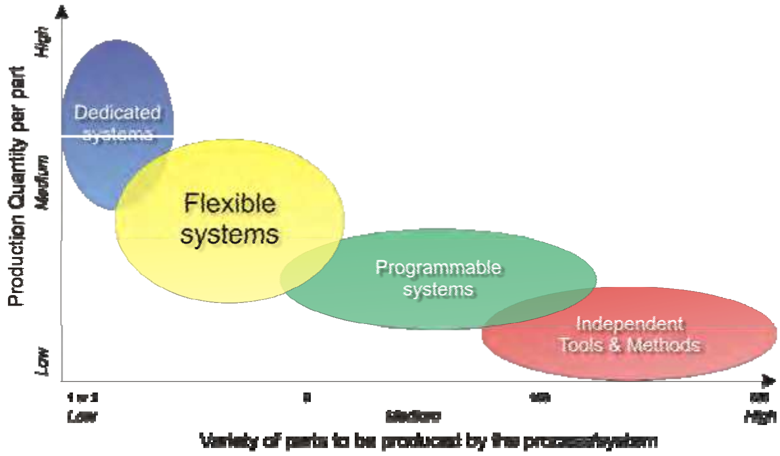


Figure 1. The manufacturing flexibility spectrum, adopted from (Brennan & Norrie, 2003)

FMS has the advantages of zero or low switching times, and hence is superior to programmable systems, but despite that FMS to its full extend has only had limited success in manufacturing setups.

Systems integration is the main issue for FMS to be successful and flexible hardware and manufacturing entities is only one part of the answer. The control software to handle and integrate the flexible entities in the overall process is equally important (Brennan & Norrie, 2003).

The control software is often regarded as the critical part, as it requires high expertise from developers. The complexity of the system and time-consuming process for reconfiguration often led to low understandability of the system, which is an important problem to manufacturers, who are not experts in manufacturing technologies.

The centralized control generally used in FMSs, which are based on principle and algorithms of classical control theories, which would not scale very well for such large systems was identified by Sandell (Sandell et al., 1978) as the main issue leading to new approaches for manufacturing control. Bussmann was even more specific and clear in his conclusion (Bussmann 1998):

> *"Manufacturing systems on the basis of CIM (Computer Integrated Manufacturing) are inflexible, fragile, and difficult to maintain. These deficits are mainly caused by a centralized and hierarchical control that creates a rigid communication hierarchy and an authoritarian top-down flow of commands."*

2.2 Distributed systems

The experienced problems with complexity and maintenance led to new approaches in the area of manufacturing control. Parunak states that traditionally a centralizing scheduler is followed by control (Parunak, 1995), which would generate optimal solutions in a static environment, but no real manufacturing system can reach this level of determinism. Even though scheduling of a shop floor environment could be optimized centrally, the system would fail in practice to generate optimal solutions, due to the dynamic environment caused by disturbances, such as failures, varying processing time, missing materials, or rush orders (Brennan & Norrie, 2003).

In general rescheduling and dissemination of new control commands are time-consuming and brings the centralized model to failure. Instead Parunak argues that manufacturing systems should be build from decentralized cooperative autonomous entities, which rather than following predetermined plans have emergent behaviour spawned from agent interactions (Parunak 1996). He lists them as three fundamental characteristics for a new generation of systems

- Decentralized rather than centralized
- Emergent rather than planned
- Concurrent rather than sequential

The area for intelligent manufacturing systems was born, and research was conducted in different directions. One of the major approaches was a project under the Intelligent Manufacturing Systems (IMS) programme, called *Holonic Manufacturing Systems* (Christensen, 1994), which settled as a new research area for manufacturing control. Holonic systems is composed of autonomous, interacting, self-determined entities called holons.

The notion was much earlier introduced by Koestler (Koestler, 1967), as a truncation of the Greek word *holos*, which mean *whole*, and the suffix *on* that means *part*, similar to the notion used for electrons and protons. Thus holons, or the manufacturing entities are *parts of a whole*.

The HMS project was initialized by a prestudy (Christensen, 1994), before the large-scale project was launched in the period from 1995 to 2000. A huge initiative with more than 30 partners worldwide, and the project not only focussed on applications, but 3 of the 7 work-packages concentrated on developing generic technologies for holonic systems, such as system architecture, generic operation (planning, reconfiguration, communication, etc.), and strategies for resource management. The application-oriented foci were organized in four work-package concerning manufacturing units, fixtures for assembly, material handling (robots, feeders, sensors, etc.), and holomobiles (mobile systems for transportation, maintenance, etc.).

The project was very successful regarding generic structures of the holons aimed at low-level real-time processing. The specification of the holons was even formally standardized by the International Electrotechnical Commission (IEC) 61499 series of standards.

The holonic parts of a system came to short in systems requiring higher level of reasoning (Brennan & Norrie, 2001), thus the term of holonic agents was introduced (Mařik & Pěchouček, 2001). Software agents that encapsulate the holon, and provide higher level decision-logic and reasoning, but also more intelligent mechanisms to cooperate with other holonic agents.

Generally agent technologies provides a software engineering approach to analyse, develop, and implement intelligent manufacturing control for distributed entities and holons. Whereas the holons were formally specified through the IEC standards, agent-based manufacturing control still lacks from having formal standards, even though various attempts have been taken, YAMS (Yet Another Manufacturing System) by Parunak (Parunak, 1987) or MASCADA (Bruckner et al., 1998) among others.

The most accepted and most promising initiative for standardizing agents are taken by FIPA (Foundation for Intelligent Physical Agents), which will be discussed in detail in the next section.

3. FIPA Standardization

The shortcomings of holons from being truly reasoning and having deliberate behaviour, but having only reactive capabilities were independently researched in agent communities. FIPA is a European based non-profit association that seeks to boost agent interoperability through standardized specifications.

Interoperability can be defined as the means for achieving agency (Mařik et al., 2003), where agency or agent-hood covers the concepts of autonomy and collaborative behaviour for agents. Whereas the HMS project had a strong focus on the internals of the holons, the FIPA specifications solely focus on external behaviour of the agents to handle interoperability. The specifications split into normative and informative specifications (Mařik et al., 2003).

- **Normative specifications** settle the external behaviour of agents to ensure interoperability not only among agents, but also with FIPA specified subsystems that make up a multi-agent system. These specifications are neutral with respect to both application domain, and the hardware and software platforms to be used.
- **Informative specifications** provide guidelines for developers and industries on how to use FIPA technologies for applications of that domain.

The FIPA subsystems mentioned above are supporting components that help to thigh an agent system together. Three subsystems are mandatory to a FIPA compliant platform and deals with the management of agents:

- **Agent Management System (AMS)**: is similar to a white-page service that maintains a list of all registered agents to a platform and their global identifiers. The AMS is also responsible for creating and deleting agents as running processes in the platform.
- **Directory Facilitator (DF)**: is a yellow-page service for agents, where agents can register their services or search for services using abstract queries.
- **Agent Communication Cannel (ACC)**: handles message transport to and from agents. The ACC is split into two components. MTS (message transport protocol) as the infrastructure for agents within an agent platform, and MTP (message transport protocol) for inter-platform communication.

The three subsystems that make up the Agent Management of a FIPA compliant platform covers one part of the FIPA Abstract Architecture for agent platforms, see figure 2 (FIPA 2002a).

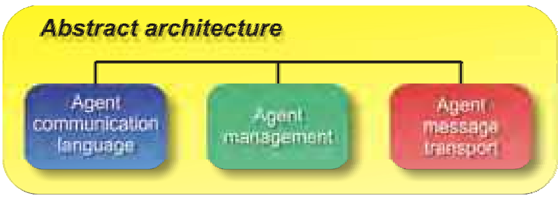


Figure 2. The FIPA Abstract Architecture

The Agent Management Component contains the formal models mentioned above, and the Agent Communication Language specifies the composition and semantics of an agent message, which comply with the FIPA-ACL specification that is based on the speech-act defined by Searle in the late sixties (Searle, 1969). Beside usual information, such as the receiver, sender, etc., the message container also holds information about the ontology used for encoding the content of the message, the time-out indicating the period the sender will

wait for a reply, and a performative for the message, which indicate which communication act the message follows.
A total of 22 performatives are specified by FIPA, but the list below only give examples of some of the most commonly used (FIPA, 2002b):

- **INFORM**: The sender informs the receiver that a given proposition is true.
- **QUERY REF**: The action of asking another agent for the object referred to by a referential expression.
- **AGREE**: The action of agreeing to perform some action, possibly in the future.
- **REFUSE**: The action of refusing to perform a given action, and explaining the reason for the refusal.

The content field of a message is also formalised and FIPA provides a semantic language (FIPA-SL) for that purpose. The Message Transport Component of the architecture contains formal specifications for the messaging service (ACC) explained above. It specifies the transport protocols used by the ACC; examples are IIOP (Internet Inter-Orp Protocol), WAP, or HTTP.
As mentioned in the beginning of the section, FIPA also provide guidelines for application domains through informative specifications. Currently 8 specifications are approved, which cover topics, such as personal travel assistance, audio-visual entertainment, and network management.
An agent platform must follow the structure of the abstract architecture in order to be truly FIPA compliant, and only a handful have reached that state yet. FIPA-OS and JADE are undoubtedly the most well-known, but others exist (Zeus and Grasshopper).
FIPA-OS is a component-based toolkit that simplifies the implementation of FIPA compliant agents and is JAVA-based. JADE developed by CSELT Laboratories at Telecom Italia is currently the most referenced FIPA compliant platform under active development with a huge developer community and has been used in a number of research projects.
JADE has also been our choice as an agent platform to implement the agent-based control solution for the baggage handling system discussed next.

4. Baggage handling

Handling of baggage in airports is shadowed by matters of complexity and uncertainty from the perspective of most passengers, similar to all other issues related to air traffic.
Many passengers, frequent or not, fell the moment of uncertainty when watching their bags disappearing behind the curtains at check-in counters. Will they ever see their bags again at the output of this "*black-box*".
Only few imagine which kind of complex system that handles the bags in major airport hubs. Small airports or charter destinations do not fall into this category, but airports with many connecting flights experience this huge sorting and distribution problem. Baggage from check-in is usually not the biggest problem, as the sorting can to some extend be handled by distributing flights correctly at the check-in counters, but bags from arriving planes that have not meet their final destination will arrive totally unsorted. So the core task of a baggage handling system (BHS) is to bring each piece of baggage from the input facility to its departure gate. The identity, and hence the destination, of the bags is unknown by the system until scanned at the input facility, which make the routing principle more attractive than scheduling and offline planning.

A BHS is a huge mechanical system, usually composed of conveyor-like modules capable of transferring totes (plastic barrels) carrying one bag each. The investigated BHS has more than 5,000 modules each with a length of 2-9 meters and run at speeds between 2-7 meters per second. The conveyor lanes of the modules that make up the BHS in the airport of Munich range 40 km in total length, and the system can handle 25,000 bags per hour, so the airport can serve its more than 25 million passengers yearly, and the BHS in Munich covers an area of up to 51.000 square meters. Thus the BHS of Munich is slightly larger than the investigated, as it has 13,000 modules and more than 80 different types of modules are used, but in setup and control they are very alike. We will return to the different types of modules when describing how agents have been mapped to the BHS. Figure 3 shows a snapshot into a BHS, where a tote containing a bag runs on the conveyors in the foreground.

Figure 3. Snapshot into a BHS with a moving tote in the foreground

A BHS often covers an area similar to the basements of the terminals in an airport, and tunnels with pathways connect the terminals. The system is rather vulnerable around the tunnels, because typically there are no alternative routes and the tunnels only contain one or two FIFO-based lanes that could be several kilometres long. Therefore the topology of the BHS could look like connected clusters of smaller networks, but within a terminal the network of conveyors is far from being homogeneous, as special areas to some degree serves special purposes.

Thus the BHS apparently shares several control characteristics with routing of packages in network traffic. Forced by both economical and architectural constraints of the airport, the layout of the BHS would usually have a rather low density of lanes and alternative routes compared to communication networks or traffic systems. The low density of connections in the graph of conveyors and the limited number of alternatives routes, makes the BHS less appropriate for intelligent network routing algorithms, such as SWARM-based approaches like ant-based control (Schoonderwoerd et al., 1997) or AntNet (Di Caro & Dorigo, 1996).

Another important difference between communication strategies and the flow of bags in the BHS, is that a lost package always can be resubmitted in a package-switched network, that is

not an option in the design of a BHS. In contrast to traffic control systems the BHS is actual aware of the correct destination for a tote, as soon as the bags enters the BHS, which makes it more attractive to use more system-wide collaboration of the agents.
Inspiration from other approaches of applying multi-agent technologies to manufacturing systems and material handling systems, such as the Production 2000+ project at DaimlerChrysler (Bussmann & Schild, 2001), could also be relevant. The Production P2000+ project has a strong focus on flexibility in a more traditional job shop manufacturing environment, where high diversity in orders and production flow through operational stations is the main issue. The BHS could still be considered as a production system, as mentioned above, because we have the input facilities (toploaders), which receive baggage from arriving planes or check-in. There are a number of processing stations in the BHS as well, but primarily they fall into the category of diverters and mergers, which split or merge conveyor paths respectively. There exists special processing stations in the system, such as manual handling stations, which are used e.g. for bags that have lost their tracking id. Also elements such as lifts or temporary storage elements are some specials versions of elements that form the entire conveyor system of the BHS.
A number of research papers deal with agent-based manufacturing from a more general perspective, such as (Maionea & Naso, 1996; Giret & Botti, 2005). Primarily the research has focused on flexibility in scheduling and planning of resources in the productions environment under the constraint of the processing steps the different orders have to go through. Approaches for planning are more or less formalized, such as the Generalized Partial Global Planning (GPGP) applied e.g. for scheduling and resource optimization in (Decker, 1995; Decker & Li, 1998; Decker & Li, 2000). Others general strategies include the PACO planning, described in (Gufflet & Demazeau, 2004), and more deliberate agents using BDI-based architecture for local optimizations (Flake et al., 1999).
Besides the physical characteristics of the BHS a numbers of external factors influence the performance

- Arriving baggage are not sorted, but arrives mixed from different flights and with different destinations, as baggage for baggage-claim are usually separated and handled by other systems.
- Identity and destination of bags are unknown to the system until the bag is scanned at the input facilities, thus preplanning and traditional scheduling is not an option.
- Obviously the airport would try to distribute the load of not only baggage, but all air-traffic related issues over the entire airport, but changes in flight schedules happen all the time, due to both weather conditions and delayed flights.
- Most airports have a number of peak times during the day, and flight schedules may also differ on a weekly basis or the season of the year. Peak times may influence the strategy on routing empty totes back to the inputs, as they share the pathways of the full totes.

4.1 The performance criteria

Top priority for a BHS is that no bags are delayed, which postpone flights and the airport will be charged by airline companies. Therefore the BHS must comply with the maximum allowed transfer time, which is this case is between 8-11 minutes depending on the number of terminals to cross. Keeping the transfer time low is also a competitive factor among airports, as airline companies want to offer their customs short connections.

Besides securing that bags reach their destination in time the capacity of the BHS should also be maximized, and the control system should try to distributed the load and utilize the entire system, if it should be capable of handling peak times.
Robustness and reliability is also of top priority, as breakdowns and dead lock situations inevitable will lead to delayed baggage, and in worst case stop the airport for several hours.
To fully understand the importance of delayed bags, the concept of *rush bags* must be introduced. Dischargers are temporarily allocated to flights, which define a window where bags can be dropped for a given flight. Normally, the allocation starts 3 hours before departure time, and closes 20 minutes before departure. Bags arriving later than 20 minutes before departure will miss the characteristic small wagon trains of bags seen in the airport area. Thus the system must detect if the bag will be late, and redirect it to a special discharger, where all bags are handle individually and transported directly to the plane by airport officers. Obviously this number should be minimized, due to the high cost of manual handling.
Bags entering the system more than 3 hours before departure are not allowed to move around in the system waiting for a discharger to be allocated, they must be sent to temporary storage – *Early Baggage Storage* (EBS). Figure 4 illustrate the system life time of a bag with the mentioned phases.

Figure 4. States of a bag in the BHS

Given those criteria, the traditional approach for controlling a BHS uses a rather simplified policy of routing totes along static shortest paths. By the static shortest paths is meant the shortest paths of an empty system, but during operation minor queues are unavoidable, which lengthen the static shortest routes. In the traditional control all totes are sent along the static shortest routes irrespective of the time to their departure in order to keep the control simple and reliable. A more optimal solution would be to group urgent baggage and clear the route by detouring bags with a distant departure time along less loaded areas.
On top of the basic approach described above the control software are fine-tuned against a number of case-studies to avoid dead lock situations, but basically it limits the number of active totes in different areas of the system. The fine-tuning process is time-consuming and costly for developers; hence a more general and less system specific solution is one of the ambitions with an agent-based solution.
Naturally the control of the BHS should try to maximize through-put and capacity of the BHS, which is indirectly linked to the issues of rush-bags. Beside that a number of secondary performance parameters apply as well, such as minimising energy consumption of the motor and life time of the mechanics, e.g. by minimizing the number of start and stops of the elements and avoid quick accelerations.

4.2 Worst-case scenario

Apparently from the descriptions above there should be opportunities for improvement of the control logic in the BHS, and one might ask why it has not been tried before, but it has...
Still listed as one of the history's top ten worst software scandals are the BHS of Denver airport in Colorado, US. The Denver International Airport was scheduled to open in October 1993, but caused by a non-working BHS the opening of the airport was delayed in 16

months costing $1 million every day. When it finally opened in 1995 it only worked on outbound flights in one of the three terminals, and a backup-system and labour-intensive system was used in the other terminals (Donaldson, 2002).

The original plan for the BHS developed and built by BAE was also extremely challenging, even compared to many BHS built today. Instead of moving totes on conveyors the BHS in Denver is based on more than 4,000 autonomous DCV (Destination coded vehicles) running at impressive speeds of up to 32 kph on the 30 km long rail system. It was a kind of agent-based with many computers coordinating the tasks, but the first serious troubles was caused by the overloaded 10Mbit Ethernet. Also the optimistic plan of loading and unloading DCVs while running caused DCVs to collide, baggage to be damaged or thrown out of the DVCs. Even unloading a bag from one running DVC into another was part of the original plan, whereas many systems today still stops a tote or DCV before unloading, even at stationary discharging points.

5. System setup

Developing control software for material handling and manufacturing systems are generally a slow task, if all tests are carried out on the real hardware. In Munich two years was spend by engineers to test the system, and during the final tests on the real system more than 10,000 real bags were checked in to the BHS. Thus just the clean-up is time-consuming, and tests must be planned in details.

Luckily, recent year's advancement in computer and graphics performance has made it possible to do realistic real-time simulations of very complex environments, including material handling systems like a 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.

5.1 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 software package. AutoMod is a de-facto-standard for systems analysis of manufacturing and material handling systems.

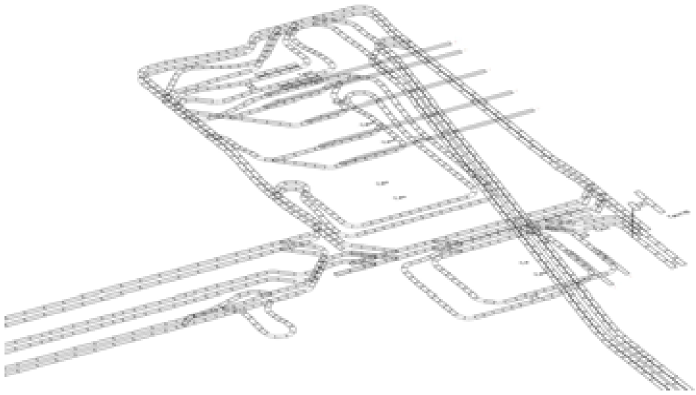


Figure 5. Snapshot of the emulation model of the BHS

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 figure 5. It shows the area with input facilities for terminal 3 of the airport.

5.2 Agent platform

As already mentioned at the end of section 3, JADE has been chosen as the agent platform for the researched BHS. Firstly because JADE is FIPA compliant, it is well documented, continuously updated, and among the most reference agent platforms. In our setup the agents are still virtual collaborating processes running in a single JADE container on a single computer due to performance reasons[1].

6. Agent design

In this section we will in details describe the tasks of the different elements in the BHS, which will form the final strategies we have applied to control the BHS. The elements are the building block of the BHS and from an intuitive point of view the potential candidates for agents in the system, as all actions of the system are performed by the elements. A classic approach of software engineering methods would lead to a functional decomposition, such as one agent for scheduling, another for resource management, etc. (Parunak, 1999), which is appropriate for centralized systems, but inspired from natural distributed systems Parunak also evinces that a physical decomposition of manufacturing systems into an agent model is both obvious and appropriate (Parunak, 1997).
Our approach to model the system concentrates on the reasoning part of agents and their interaction. Following the notion of holonic agents by Mařik & Pěchouček, the holon part of the agent is packed into logic of the emulation model, but no special attention was given to comply with the IEC standard, which would modify the existing hardware.
An alternative approach would be to consider the totes as “consumer” agents and the BHS as a collection of “producer” agents, as the BHS can solve the tasks that the totes want to have performed - bringing the tote to the destination. In principle a tote could then negotiate its way through the system, and if the bag was urgent it would be willing to pay a higher price than non-urgent bags.
This approach often leads to other complications, such as communication overhead and complex agent management (Brennan & Norrie, 2003). Because the BHS generally consists of pathways of FIFO queuing lanes with little and often no possibilities of overtaking it is more appropriate to design the agents around the flow of the BHS, which makes the elements the potential candidates for agents alone. The element agents should then coordinate their activities to optimize system performance and should therefore be considered as collaborative agents, rather than competitive agents.

---

[1] There is a huge overhead, when communicating across different agent containers or even worse when communicating across different JAVA JVMs.

6.1 Toploader

The input facilities of the BHS are called toploaders, as they drop bags into the totes from a conventional conveyor belt, see figure 6. Before the bag is *inducted* into the tote it is has passed a scanner, which reads its id and destination that are coupled with the tote, so the control system has exact tracking of the bag at all time.

Identity and destination of the bag are unknown until the bag passes the scanner at the toploader shortly before being inducted if a tote is ready. The scanning initializes routing of the tote, but the short time leaves no option for global optimized planning of all current totes, and replanning when the next arrives.

Figure 6. A toploader, where bags arrive on a traditional conveyor belt

Basically the task of the toploader could be decomposed into scanning of the bag, which happen automatically and have no direct impact on the control. Secondly it initiates the journey of the tote on the BHS. In order to start the routing of the tote, the end-point (discharger) must be set for the tote. In order to optimize the capacity several dischargers are often allocated to the same flight destination[2]. Therefore the toploader agents initiate a negotiation with the possible dischargers to find the best suited discharger, the evaluation of the proposals from the dischargers is not trivially chosen as the lowest offer, but weighted with the current route length to the dischargers, which the toploader requests from a route agent - a mediator agent with a global focus on the dynamic route lengths of the BHS.

The toplader can take two different approaches for routing the tote:

- **Routing by static shortest path:** After the toploader has decided on the discharger it could instruct all diverting elements along the route to direct that specific tote along the shortest path. Then the agent system would in principle work as the traditional control system by sending all totes along predefined static shortest routes[3].
- **Routing on the way:** Instead of planning the entire route through the BHS, the toploader could just send the tote to the next decision point along the shortest route. A more dynamical and flexible approach, as the tote can be rerouted at a decision point if the route conditions have change, perhaps another route have become the dynamical shortest one, or the preferred discharging point have changed.

[2] Due to the stopping of totes while unloading, the discharger has a lower line capacity than straight elements.

[3] In the researched BHS the decision between the alternative dischargers would also be predefined in the conventional control. The BHS is built in layers to minimize cost and maximize space utilization, and alternative dischargers are always split on different layers, and the control system would try to avoid switching layers.

6.2 Straight elements

Most of the elements of a BHS are naturally straight or curved elements that connect the nodes of the routing graph. Straight or curved elements are not considered as agents in our current design, because mechanically they will always forward a tote to the next element if it free, thus there are no decisions to be made. In principle the speed of each element could be adjusted to give a more smooth flow and avoid queuing, so one could argue that these decisions should be taken by the element itself. In the current setup it would generate an enormous communication overhead, because each element should be notified individually and the agents should be very responsive to change the speed in order to gain anything from speed adjustments.

6.3 Diverters

When straight elements are not considered as agents, divert elements becomes the first natural decision points on the routes. A diverter splits a conveyor lane into two, either a left or right turn and straight ahead. Lifts and so-called cross-transfers could be considered as special editions of the diverters. The cross-transfer allows the tote to be forwarded in all four directions.

Figure 7. A diverter element with an empty oversize tote

In respect to the strategies described above the diverter would either just forward the tote in the direction determined by the toploader, or it should reconsider alternative routes by restarting the negotiation process with dischargers and requesting updated information on dynamic route lengths. A diverter should be concerned about the relevancy of reconsidering the route for a tote, because in many cases there is only one possible direction at a given diverter for a given tote. We want to generalize the control logic of the diverter agents instead of customizing it according to the placement of the diverter in the BHS layout. Thus initially it adjusts itself to different destinations based on static route information. As mentioned for many diverters there are no alternative direction, for some the decision will have little impact, e.g. if the tote is close the discharger and there is little difference in dynamical route length between the two directions. For a few diverters the decision would have great impact on future decisions. That is mainly diverters placed at the points in the BHS, where it is possible to change layers[4]. Basically it is only important to reconsider the alternative dischargers at this point, because due to the layout of the BHS it is not possible to switch back to the other layer again, before the dischargers have been passed.

[4] BHS is constructed as two layers of conveyor to save both space and cost.

That leaves us with decision logic rather identical to the dynamic routing principle at toploaders, but diverters should fine-tune their decisions according to the local environment in which they are situated. In other words a strong influence on the decision logic of the diverter is based on its position in the routing graph.

6.4 Mergers

Mergers are the opposite of diverters, as they merge two lanes. Traditionally mergers are not controlled, as there are no alternatives to continuing on the single lane ahead, and the merger simply alters between taking one tote from either input lane, if both are occupied.
Obviously, more intelligent decisions could be considered than just switching between the input lanes, which is the argument for applying agents to the merger elements. The ratio between merging totes from the input lanes should be determined by the aggregated data of the totes in either of the two lanes. E.g. if the number of urgent totes waiting to be merged are higher in one lane that lane should be given higher priority. Also waiting totes in one lane could have greater impact on the overall system performance, if a queue of totes in one lane is more likely to block other routes behind that point.

6.5 Dischargers

Dischargers are responsible for unloading bags from the totes, when the tote reaches its destination. When bags are discharged they fall onto carrousels similar to those at baggage claim and are drown to the plane in small wagon trains.

Figure 8. A discharger element that can tilt the tote, so bags slide onto the conveyor belt

Besides being involved in the negotiation process described for the toploaders, the task of the discharger could seem rather simple - just tilting the tote, but a discharger also has to take care of the empty totes. Some BHSs have separate conveyor system for the empty totes, but many systems including the researched BHS use the same lanes for routing the empty totes back to the tote stackers at the toploaders.
The task of routing empty totes is similar to routing full totes at the toploaders, but actually much more complex, due to a number of considerations that must be taken into account.

- The number of destinations (tote stackers) is larger than alternative dischargers for full totes (typically 2), whereas the number of tote stackers is equal to the number of toploaders, which is 12 in our case.
- Specially in the input area, empty totes are mixed with full totes and the area could easily get overloaded and blocked.

- During peak times it should be considered sending some empty totes to temporary storage in the EBS area, which is far from the input area, and released again when the load on the system is lower.
- The status of empty tote stackers. If a stacker runs empty, no totes will be available at the toploaders for new bags.
- The distance to the stackers. It is more appropriate to return the empty tote to a stacker nearby than sending it half way through the system.

All factors should be considered and measured against a fuzzy set, which are weighted to an aggregated value - a proposal of a bid plan to go to each of the stackers.

6.6 EBS elements

Early baggage storage elements, or EBS for short, are temporary storage elements for totes with bags for which a discharger has not been allocated yet, as described above when defining the concept of rush-bags.

Figure 9. EBS elements, here storing a line of empty totes

It is a complete research area for itself to optimize the utilization of the EBS, as totes are stored in lanes, which are released into the system again, but planning and coordinating the totes in different lanes is not a simple task, but will not be given further attention in this chapter.

7. Agent interactions and ontology

As mentioned in section 3 FIPA has not yet approved specifications and guidelines for agent applications in the manufacturing domain. Thus we have developed the agent interactions based on the elements responsibility and participation in the function of the BHS, as described in the previous section.

Due to both time constraints and focus of the DECIDE project, it was never an issue to change the setup of the hardware. The layout of the BHS was determined in advance of the project, and research goals were to investigate if multi-agent based control software could replace traditional control structures in an efficient way.

7.1 Adapter Agent

Given the setup, where the agents resides as virtual representations of their corresponding elements in a container of the JADE platform, and the actions of the agents are visualized in the emulation software, we required a gateway agent to handle the communication protocol. This agent was named *AdapterAgent* and could be understood as a converter of stimuli and action messages between the agent community and the real world or hardware, which in our setup is represented by the AutoMod model. The interface between the AutoMod model and the AdapterAgent follows the same protocol and telegram structure as the interface to the real hardware.

The AdapterAgent constantly listen for new telegrams from the AutoMod model. Thus it encapsulates the entire perception of the whole agent community towards the environment, the notifications received from the model are converted to agent messages complying with the FIPA ACL specification and message contents are encoded with the ontology we have defined for the BHS domain.

The AdapterAgent is not thought to be a filter of the message exchange, thus all information from the model is encoded into the messages and forwarded to interested agents, and no notifications are omitted. Ideally all agents would be connected to the model, and communicate directly with their corresponding element, but that would require more than 300 concurrent threads listening on their own socket connection to the model, which is inappropriate for performance reasons. Thus perception has been extracted and isolated into the AdapterAgent, in order leave the element agents with a natural external interface they subscribe to interesting messages from the AdapterAgent. It is not the task of the AdapterAgent to figure out, where to send messages. During initialization all element agents initiate a subscription for each of the emulator notifications they are interested in. The negotiation of subscriptions follows the FIPA subscription specification, but subscriptions are always approved.

Because no handshaking is defined for communication with the model, all message sent from the AdapterAgent are *inform* messages with no expectation of *agree* or *accept* replies. That also has the positive side effect that the AdapterAgent is not required to understand the messages (except simple conversation) and couple a response to a notification. Element agents are fully responsible for replying if require. Exactly the same apply in the opposite direction as well (no coupling between commands or queries to the model and the replies), thus inform message are a natural choice.

7.2 Mediator agents

There is a balance between giving agents detailed information about the environment and maintaining an internal world model, or let them query the environment about information when required.

The FIPA Directory Facilitator is a strong tool when dynamically searching the environment for appropriate services. It is used when dischargers are allocated to a flight, they register a new service for that departure in the DF and agents requesting a flight can simply search the DF, instead holding information about all dischargers. That is a quite trivial use of the yellow page service of the DF.

In theory the service and discovery approach could be used to route totes around in the system to fully decouple agents from physical connections, but that would generate too much overhead and complicate the simple routing principles. Instead agents can be assisted by mediator agents, who collect aggregated information for the entire system. The *RouteAgent* is an example of such an agent. In the initialization process the RouteAgent generates all possible routes in the system by building up a graph for the BHS with nodes corresponding to the element agents. During operation it constantly monitors traffic on edges of the graph by subscribing to such information in the AdapterAgent and update the weights in the graph, so dynamic shortest paths can be calculated using classic Dijkstra for dynamic shortest path calculations (Dijkstra, 1959).

Following the FIPA *query-ref* communication act element agents can request routes to a given destination packed in a referential expression of the query message. The referential expression is composed using the ontology we have defined for the BHS domain, which extends and follows the structure of the FIPA-SL. The RouteAgent understand two concepts of the ontology, *RouteBetween* and *LineBetween*:

- **RouteBetween** is the concept used, when agents are interested in full or parts of a route, but only with a granularity of finding other element agents along the path – only information on nodes of the graph are returned.
- **LineBetween** is the fine-grained concept providing all details about a conveyor line of connected elements in the BHS – information about edges between two given connected nodes.

To give an example of the generality embedded in ontology-based messages, a query to the RouteAgent could contain the following abstract referential expressions:

```
(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
        )
)
```

Abstract because it contains the variable `x` that must be replaced by the responder in a response to the query. In this case a set of points (id's of element agents between the given origin and destination). The predicate `iota` is just one of three from the FIPA-SL specification, which means exactly one object that fulfils the expression, whereas the other predicates, *any* and *all*, would return any or all routes between the origin and destination, respectively.

The approach of the RouteAgent was not only taken to omit the world model in the element agents, but also to support a simple implementation of the basic approach of the traditional control software to compare against the agent-based solution.

In current and future experiments conducted on the BHS we try to exclude the RouteAgent by giving elements agents a dynamic profile of their local environment, which is further described in under future work, because it simplifies agent interactions.

7.2 Routing negotiations

The negotiation and collaboration interactions framing the simple routing principle in the BHS, have been selected to give an illustration of how the interoperability among agents are achieved using normative FIPA specifications.

The toploading and routing principle are described in general terms in the beginning of section 6. It is also mentioned that the inter-agent communication is almost identical at both toploaders and diverter, when we deal with dynamic routing – in case of routing along static shortest path, the toploader simply make all the decisions and informs all diverters along the route how to direct the tote.

The toploader receives a stimuli by means of an *inform* message from the AdapterAgent that a new bag has been scanned. Assuming no problems the toploader searches the DF for discharge agents that have been allocated to the destination of the bag revealed in the

scanned code of the bag. If no discharger are available the bag will be routed to the EBS, otherwise the toploader will initiate a FIPA *contract net* with the dischargers to collect proposals of accept for the new bag. Concurrently the toploader request route information to the dischargers from the RouteAgent, where route lengths are combined with the proposals from the dischargers to take the final decision of the best discharger for the bag.
Shortly after the first stimuli from the AdapterAgent the toploader will receive another stimuli from the model that the bag has been inducted into the tote and the tote is released from the toploader. This *inform* message purely serves the purpose of combing the scanned bag id (the IATA tag) with the tote id, so routing can be handled correctly for the rest of the journey.
In case of the static routing principle the toploader send FIPA *request* messages to all node agents along the route with an action request to direct the tote in the given direction, when it is seen at the node. This approach purely violates the autonomy of those agents, but as mentioned above it primarily serves the purpose of comparison. Figure 10 illustrates the interactions in a simplified sequence diagram.

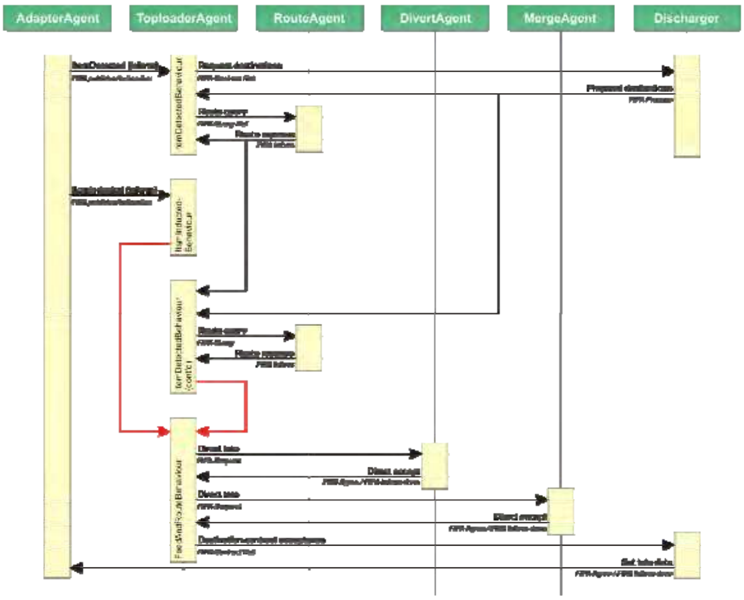


Figure 10. Example of agent interoperability

As outlined above all communication follows the FIPA specifications in order to generalize the interoperability among agents and increase the possibility of reuse. The content of the messages are encoded in abstract task descriptions or status reports from the individual element agents, again to minimize the influence of application domain.

8. Internal agent reasoning

The previous section dealt with agent interoperability and the communication design in respect to the FIPA specification. As stated in section 3 FIPA is only concerned with the external behaviour of the agents, but the internal behaviour of agents and the reasoning part is also of extreme importance.

In this section we will present internal agent reasoning principles to optimize the flow in the BHS in different ways to meet some of the performance parameters. Deep reasoning and long-term goals are not currently pursued in the strategies, due to the flow speed and high number of totes in the system. Instead the intensions behind the strategies are to optimize the situation for more than a single tote or forthcoming actions.
We will give examples of three different deliberate behaviours, which take part in both necessary routing and optimizing strategies for the BHS.

8.1 Returning empty totes

As explained in section 6.5 the task of dischargers is more complicated than just emptying the tote. The tote continues on the conveyors and should be routed back to tote stackers located at the input facilities. We currently omit the EBS logic from the control software (no arriving bags will have more than 3 hours to departure), so we do have to consider temporarily storing empty totes in the EBS area.
The most important factor that influences the decision of where the empty tote should be returned to, is the full status of the tote stackers, but also the distance to the tote stackers should be considered. There is no reason to send it to the other end of the system, if a stacker is nearby, unless the other is empty.
Each stacker monitors its full status as a simple ratio between the current and maximum number of totes in the stacker. By a standard indeed fuzzy hedge (Negnevitsky, 2005) the ratio is converted into a priority s_i for requesting extra totes.

$$s_i = \begin{cases} 2r_i^2 & 0 \le r_i < \frac{1}{2} \\ 1-2(1-r_i)^2 & \frac{1}{2} \le r_i \le 1 \end{cases} \tag{1}$$

where r_i is the full-ratio for the i 'th stacker. A plot of the function is shown in figure 11.

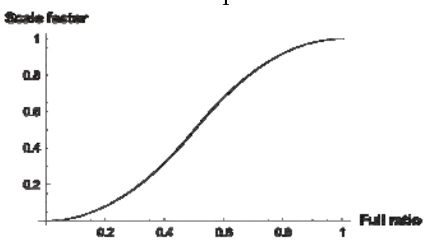


Figure 11. Plot of ETS priority function

The priority determined is used to scale the dynamic route length to each tote stacker, so a nearly empty stacker will have a very short route length or value in the decision, whereas a full stacker will have its full route length.

$$v_i = d_i \cdot s_i \tag{2}$$

where d_i is the dynamic distance (requested from the RouteAgent) to the stacker from the decision point.
This behaviour clearly serves our purpose of refilling the empty tote stackers at the input facilities, but the importance of correct routing will be far more interesting when the EBS area is included in the agent system. For the discharger the EBS could be considered as just another destination (stacker) that never runs full in practice, but in the long run it generates extra traffic to send the tote back via the EBS area.

8.2 Overtaking urgent bags

Consider a typical layout of a discharging area in figure 12. The bottom lane is a fast forward transport line, the middle a slower lane with the dischargers and the upper lane is the return path. A diverter (in the bottom lane) has the option to detour non-urgent to the middle lane to give way for urgent baggage in the transport line, but with no queues in the system all totes should follow the shortest path. When the routes merge again at the mergers in the middle lane, it will give higher priority to totes from the merging leg with the most urgent baggage.

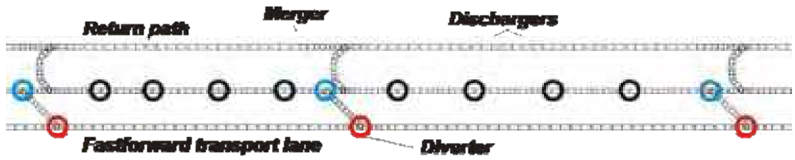


Figure 12. Area of the BHS layout with indication of diverters, mergers, and dischargers

Urgency is a constructed function, which gives high priority to urgent totes and negative priority to totes, where remaining time to departure exceed a threshold.

$$u_j = \begin{cases} \dfrac{1}{t_j^{\,2}} & t_j < U_T \\ \dfrac{1}{(U_{\max} - U_T)^2}(-t_j^{\,2} + 2U_T t_j - U_T^{\,2}) & t_j \geq U_T \end{cases} \tag{3}$$

where $U_{\max}$ is the full window size of the allocated discharger. If the tote's remaining time exceed this value it should go to EBS. U_T is the threshold value, which is set to 20 min, as no tote should be considered urgent, if it has more than 20 min left before the discharger closes[5]. t_j is the remaining time for the j 'th tote. The graph is plotted in figure 13.

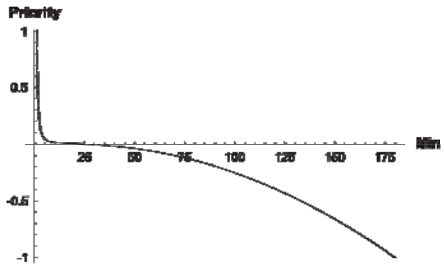


Figure 13. Urgency function for totes

The urgency factor is converted to a scale factor for the dynamic route lengths of alternatives routes. Then the principle of simple modification of the route lengths can be used here as well.

$$s_j = \begin{cases} (1-u_j)(1+v_{k+1}) & u_j < 0 \quad \text{(non} - \text{urgent tote)} \\ (1-u_j)(1-v_{k+1}) & u_j \geq 0 \quad \text{(urgent tote)} \end{cases} \tag{4}$$

where v_{k+1} is the aggregated urgency value for the next decision point along the route, which is requested in a communication act (FIPA *request-ref*) to the divert agent. The formula secures that

[5] When the discharger closes the tote becomes a rush-bag, but the threshold of 20 minutes is independent of the 20 minutes time limit for rush bags, described in section 4.1, so in total a tote is considered non-urgent if it has more than 40 minutes left to departure.

urgent totes will group along the shortest route (as v_{k+1} is close to 1), whereas non-urgent are punished along the detour. If there are no queues on the routes the v_{k+1} is 0, and the scale factor has no effect.

The mergers in the middle lane simply give higher priority to input lanes with more urgent totes. The ratio between the aggregated urgency factors of the input lanes becomes the ratio for merging totes from the input lanes.

8.3 Saturation management

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[6]. These matters result in the characteristics of the work in-progress against capacity curve (WIPAC), which is further described in (Kragh, 1990) that states the capacity of a system goes dramatically down, if the load on the system exceed a certain threshold value, as indicated in the figure 14.

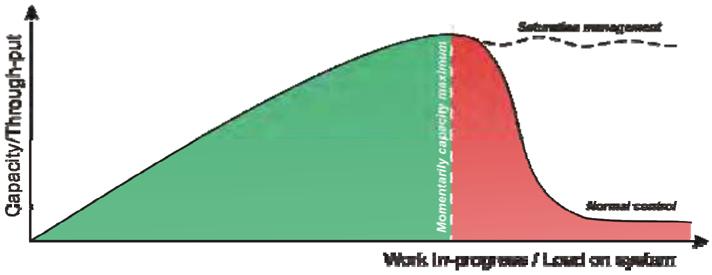


Figure 14. Theoretical WIPAC

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.

$$v_t = \frac{\sum_i w_i q_i}{\sum_i w_i} = \frac{\sum_i \frac{\alpha}{d_i} q_i}{\sum_i \frac{\alpha}{d_i}} \tag{5}$$

[6] 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.

where v_t is the full speed of the toploader, and w_i are weights of the queue statues, q_i, along the routes. The weight is given by a fitted coefficient, α, and the distance from the toploader d_i. Queue statuses, q_i, are always a number between 0 and 1, where 1 indicates no queue.

The effect of the saturation management strategy is clearly documented by the graph in figure 15. Thus the decision taken by the toploader agent is highly dependent on the current configuration of the environment around the toploader.

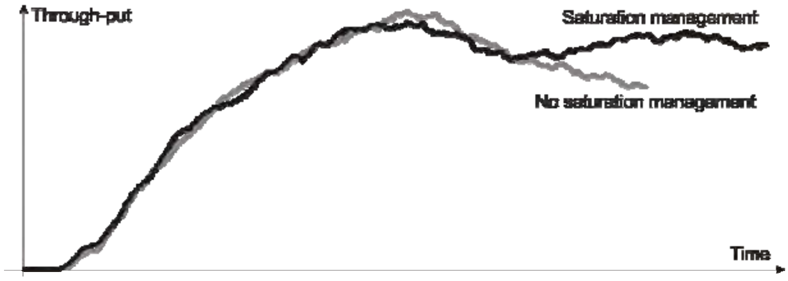


Figure 15. Result of a test scenario with and without the saturation management strategy

9. Conclusion

Is this chaper we have presented novel research contributions from an application project under the DECIDE project that deals with multi-agent based control in production systems. In this case a baggage handling system (BHS) in a major airport hub in Asia. Agents were intended to substitute existing control logic, but not change the layout of the BHS.

Interoperability of the agents have been secured through the use of FIPA specifications, which generalized the design of the agents to be applicable for other material handling systems.

We have succeeded in keeping the decision logic of the agents rather general in order to improve reusability and understandability for the agent based control.

Special attention has been given to the task of the different type of agents, and examples of implemented decision logic have proven successful compared to the traditional approach.

9.1 Future work

We continue our research on the BHS and will develop more new strategies for the local agents, and increase their mutual collaboration to maximize the utilization of the BHS during peak times. We will try to avoid the use of centralized mediator agents (the RouteAgent) and rely on roles and profiles for the agents. Ideally a swarm of local agents would provide the most general setup, which easily can be ported to other manufacturing and material handling systems. During the research we will pay special attention to develop abstract and general design methodologies for the topological domain of impact for agent collaborations.

9.2 Acknowledgements

We would like to thank all the participants in the DECIDE project, which is supported by the The Ministry of Science, Technology and Innovation in Denmark. A special thank to FKI Logistex (http://www.fkilogistex.com) and Simcon (http://www.simcon.dk) for their support, feedback, and creation of the AutoMod model of the major hub airport.

10. References

Brennan, R. W. & Norrie, D. H (2001). Agents, holons and function blocks: distributed intelligent control in manufacturing. *Journal of Applied Systems Studies Special Issues on Industrial Applications of Multi-Agent and Holonic Systems*, Vol. 2, No. 1, 2001, pp. 1-19.

Brennan, R. W. & Norrie, D. H (2003). From FMS to HMS, In: *Agent-Based Manufacturing – Advances in the Holonic Approach*, Deen, S. M., (Ed.), pp. (31-49), Springer, ISBN 3540440690, Berlin, Germany.

Brückner, S.; Wyns, J.; Peeters, P. & Kollingbaum, M. (1998). Designing Agents for Manufacturing Control. *Proceedings of the 2nd Artificial Intelligence and Manufacturing Research Planning Workshop*, pp. 40-46, Albuquerque, August 1998.

Bussmann, S. (1998). An agent-oriented architecture for holonic manufacturing control. *Proceedings of First Workshop on Intelligent Manufacturing Systems Europe*, pp. 1-12, Lausanna, Switzerland, 1998, EPFL.

Bussmann, S. & Schild, K. (2001). An Agent-based Approach to the Control of Flexible Production Systems, *Proceedings of 8th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA 2001)*, pp. 169-174, Antibes Juan-les-pins, France.

Christensen, J. H. (1994). HMS: initial architecture and standard directions, *Proceedings of the 1st European Conference on Holonic Manufacturing Systems*, pp. 1-20, HMS Consortium, Hannover, Germany.

Decker, K. (1995). *Environment Centered Analysis and Design og Coordination Mechanisms*, Ph.D. thesis, Department of Computer Science, University of Massachusetts, May 1995.

Decker, K. & Li, J. (1998). Coordinated Hospital Patient Scheduling, *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98)*, pp. 104-111, IEEE Computer Society, Washington, DC, USA.

Decker, K. & Li, J. (2000). Coordinating Mutually Exclusive Resources using GPGP. *Autonomous Agents and Multi-Agent Systems*, Vol. 3, No. 2, pp. 133-157, Springer.

Di Caro, G. & Dorigo, M. (1997). *AntNet: A Mobile Angents Approach to Adaptive Routing*. Université Libre de Bruxelles, IRIDIA/97-12, Belgium.

Dijkstra, E. W. (1959). A Note to Two Problems in Connexion with Graphs. *Numerische Mathematik*, Vol. 1, pp. 269-271.

Donaldson, A. J. M. (2002). *A Case Narrative of the Project Problems with the Denver Airport Baggage Handling System (DABHS)*. Middlesex University, School of Computing Science, TR 2002-01, January 2002.

FIPA (2002a). *FIPA Abstract Architecture Specification*. Foundation for Intelligent Physical Agents, SC00001L, December 2002.

FIPA (2002b). *FIPA Communicative Act Library Specification*. Foundation for Intelligent Physical Agents, SC00037J, December 2002.

Flake, C.; Geiger, C.; Lehrenfeld, G.; Mueller, W. & Paelke, V. (1999). Agent-Based Modeling for Holonic Manufacturing Systems with Fuzzy Control, *Proceedings of 18th International Conference of the North American Fuzzy Information Processing Society (NAFIPS'99)*, pp. 273-277, June 1999, New York, USA.

Giret, A. & Botti, V. (2005). Analysis and Design of Holonic Manufacturing Systems, *Proceedings of 18th International Conference on Production Research (ICPR2005)*.

Gufflet, Y. & Demazeau, Y. (2004). Applying the PACO paradigm to a three-dimensional artistic creation, *Proceedings of 5th International Workshop on Agent-Based Simulation (ABS'04)*, pp. 121-126, May 2005, Lisbon, Portugal.

Koestler, A. (1967). *The Ghost in the Machine*, Penguin, ISBN 0140191925, London, United Kingdom.

Kragh, A. (1990). *Kø-Netværksmodeller til Analyse af FMS Anlæg*. Ph.D. thesis, Informatics and Mathematical Modelling, Technical University of Denmark.

Maionea, G. & Naso, D. (1996). A soft computing approach for task contracting in multi-agent manufacturing control. *Computers in Industry*, Vol. 52, No. 3, pp. 199-219.

Mařik, V. & Pěchouček, M. (2001): Holons and agents. Recent developments and mutual impacts, *Proceedings of the Twelfth International Workshop on Database and Expert Systems Applications*, pp. 605-607, IEEE Computer Society.

Mařik, V.; Pěchouček, M.; Vrba, P. & Hrdonka, V. (2003). FIPA Standards and Holonic Manufacturing, In: *Agent-Based Manufacturing – Advances in the Holonic Approach*, Deen, S. M., (Ed.), pp. (31-49), Springer, ISBN 3540440690, Berlin, Germany.

Negnevitsky, M. (2005). Artificial Intelligence – A Guide to Intelligent Systems, 2. edition, Addison Wesley.

Parunak, H. V. D. (1987). Manufacturing Experience with the Contract Net, In: *Distributed Artificial Intelligence*, Huhns, M. N. (Ed.), pp. 285-310, Pitman Publishing, London.

Parunak, H. V. D. (1995). *Autonomous Agent Architectures: A Non-Technical Introduction.* Industrial Technology Institute, ERIM, 1995.

Parunak, H. V. D. (1996). Applications of Distributed Artificial Intelligence in Industry, In: *Foundations of Distributed Artificial Intelligence*, O'Hara, G. & Jennings, N. (Ed.), pp. 139-163, Wiley Interscience, ISBN 0471006750, New York, USA.

Parunak, H. V. D. (1997). "Go to the Ant": Engineering Principles from Natural Multi-Agent Systems. *Annals of Operation Research*, Vol. 75, pp. 69-101.

Parunak, H. V. D. (1999). Industrial and Practical Applications of DAI, In: *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*, Weiss, G. (Ed.), pp. 377-421, MIT Press, ISBN 0262232030, Cambridge, MA, USA.

Sandell, N.; Varaiya, P.; Athans, M. & Safonov, M. (1978). Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, Vol. AC-23, No. 2, April 1978, pp. 108-128, ISSN 0018-9286.

Schoonderwoerd, R.; Holland, O.; Bruten, J. & Rothkrantz, L. (1996). Ant-Based Load Balancing in Telecommunication Networks. *Adaptive Behaviour*, No. 5, 1996, pp. 169-207.

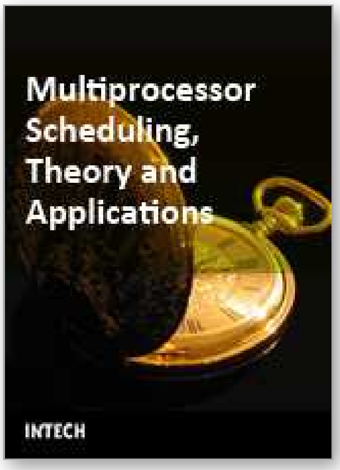
Searle, J. R. (1969). *Speech Acts*, Cambridge University Press, ISBN 052109626.

Sipper, D. & Bulfin, R. (1997). *Production: Planning, Control and Integration*, McGraw-Hill College, ISBN 0070576823.

Swamidass, P. M. (1988). *Manufacturing Flexibility*, Monograph No. 2, Operations Management Association, Texas, USA.

Taylor, F. W. (1911). *The principles of scientific management*. Harpers & Brothers Publishers, ISBN 048629980, New York, USA.

Willliamson, D. (1967). System 24 – a new concept of manufacture, *Proceedings of the 8th International Machine Tool and Design Conference*, pp. 327-376, University of Manchester, September 1967, Pergamon Press, Oxford.



Multiprocessor Scheduling, Theory and Applications

Edited by Eugene Levner

ISBN 978-3-902613-02-8
Hard cover, 436 pages
Publisher I-Tech Education and Publishing
Published online 01, December, 2007
Published in print edition December, 2007

A major goal of the book is to continue a good tradition - to bring together reputable researchers from different countries in order to provide a comprehensive coverage of advanced and modern topics in scheduling not yet reflected by other books. The virtual consortium of the authors has been created by using electronic exchanges; it comprises 50 authors from 18 different countries who have submitted 23 contributions to this collective product. In this sense, the volume can be added to a bookshelf with similar collective publications in scheduling, started by Coffman (1976) and successfully continued by Chretienne et al. (1995), Gutin and Punnen (2002), and Leung (2004). This volume contains four major parts that cover the following directions: the state of the art in theory and algorithms for classical and non-standard scheduling problems; new exact optimization algorithms, approximation algorithms with performance guarantees, heuristics and metaheuristics; novel models and approaches to scheduling; and, last but least, several real-life applications and case studies.