A Generic Control Architecture for Material Handling Systems Applied to a Baggage Handling System

S. W. A. Haneyah, J. M. J. Schutten, P. C. Schuur and W. H. M. Zijm
IEBIS, School of Management and Governance, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Keywords: Baggage Handling System (BHS), Real-time Scheduling, Control Architecture, Workload Control.

Abstract: This paper is part of research on generic planning and control of automated Material Handling Systems (MHSs) in different industrial sectors. We build upon previous work to provide a proof of concept for the applicability of a generic control architecture on a specific MHS. To this end, the baggage handling system (BHS) of a major European hub represents our business case. We present the control architecture and apply it to the BHS under study in a simulation environment. This application shows how the generic control architecture adapts to the specificities of this BHS and how it handles unconventional workstation types, i.e., robots. Finally, we highlight the lessons learned and make recommendations for future applications.

1 INTRODUCTION

We apply a generic control architecture, which was developed for Material Handling Systems (MHSs) by Haneyah et al. (2013a), to a baggage handling system (BHS). The aim is to provide a proof-of-concept for the applicability of the generic control architecture to a specific MHS, while preserving the generic structure. Moreover, the generic and standardized methods should handle system-specific elements in the BHS under study. We provide modeling and implementation aspects for the BHS under study, compare the generic control architecture to the current practice approaches, discuss the behavior of each approach, and comment on the architectural design and software complexity.

The paper is organized as follows: Section 2 presents key literature. Section 3 illustrates the BHS of our business case. Section 4 presents the control architecture and its application to this specific BHS. Section 5 provides the experimental setting. Section 6 provides general results and recommendations for future applications of the generic architecture. Finally, Section 7 ends with concluding remarks.

2 KEY LITERATURE

Haneyah et al. (2013b) present a comprehensive literature review on control architectures for MHSs. They conclude that only few studies attempt to build a generic control architecture for different industrial sectors. A common trend is that a control architecture normally targets a specific sector, or deals with material handling as part of a manufacturing environment. Babiceanu et al. (2004) use a holonic manufacturing systems approach as a basis to suggest a framework for material handling. However, this architecture has limitations and does not address a number of necessary aspects (Haneyah et al., 2013b). Tařau et al. (2009a) study route control in BHSs. They find centralized control approaches computationally expensive and not robust. Tařau et al. (2009b) pay attention to hierarchical control for route choice. They design a hierarchical control architecture and examine multi-agent systems in the same study, but find them disadvantageous due to the extensive communication required between agents. In general, Tařau et al. study routing problems and how to control divert switches within BHSs, but they do not consider other operations, e.g., the early bag storage. Johnstone et al. (2010) study status-based routing. In their approach, the status of the bag determines its processing requirements and triggers the computation of the route to be followed, depending on the states of required resources ahead. They contrast an algorithm that is based on learning agents with another algorithm using a graph representation of the network to find possible routes at switches via Dijkstra's algorithm. Hallenborg &

W. A. Haneyah S., M. J. Schutten J., C. Schuur P. and H. M. Zijm W..
A Generic Control Architecture for Material Handling Systems Applied to a Baggage Handling System.
DOI: 10.5220/0004427201210130
In *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2013), pages 121-130
ISBN: 978-989-8565-70-9
Copyright © 2013 SCITEPRESS (Science and Technology Publications, Lda.)

Demazeau (2006) find global planning and scheduling an impossible or inappropriate strategy for large and complex MHSs. They use multi-agent technology as a basis for generic software components to replace traditional system-specific centralized control software.

3 THE BHS

The core task of any BHS is to deliver each bag from some point A (related to its source) to some point B (related to its destination), within a specific time limit. However, the airport environment of a BHS is very dynamic and stochastic, which complicates the delivery job and raises many additional considerations to take care of. Moreover, the transport network can be rather complex.

The *build* for a flight starts a couple of hours before the scheduled time of departure. Then, at least one *lateral* (a type of outfeed conveyor used in BHSs to gather bags in preparation for loading on planes) is open to handle the baggage for this flight. The baggage arrives from the sorter system at one of the designated laterals. There, workers take the baggage off the lateral and transport it towards the airplane, either on ramp carts or inside special containers called *Unit Load Devices (ULDs)*. This paper focuses on a terminal in a major European hub as an application case for the generic control architecture of Haneyah et al. (2013a). Figure 1 presents a simplified material flow diagram that gives insight into the main components of this BHS, which are:

Baggage: bags that can be transported on the BHS are referred to as conveyables, which represent the main material flow. On the other hand, non-conveyables represent a very small portion of the total number of bags that cannot be loaded on the BHS because they are too small, too large, too light, too heavy, too unstable (e.g., a ball), etc. Non-conveyables are handled with dedicated equipment. Moreover, there is another flow that we do not consider as it is not a bottleneck resource for the performance of BHSs, which is the flow of empty totes that are used in the storage area to carry bags; other parts of the BHS use conveyors.

Diverts: diverts can direct the bag to one of two possible routes. The selected route should eventually lead to the bag's next process step (the main sorter or the bags storage). In this BHS, streams of bags from check-in desks or transfer belts mix and proceed to one of a group of 8 diverts (depending on the source point).

Screening Area: security control occurs in this area. In this BHS, there are four conveyor loops. Each two loops lead to one (out of two) cluster of screening machines. Each cluster has three screening machines. The design of the screening area with the upstream diverts is redundant, i.e., a bag can follow either one of the two route options at the diverts to go through one of the parallel loops, and then one of the parallel screening machines within a cluster. Suspicious bags go to a second screening level by other machines. However, we model only one screening level. After the screening machines, the bags proceed via conveyors to downstream system areas, e.g., storage and laterals build areas. In this BHS, the screening machines do not have the same connections to downstream resources. Each cluster has one machine giving access only to the storage area, a second giving access only to the main sorter, and the third giving access to both areas.

Automated Storage and Retrieval System (ASRS): the main function of the ASRS is to store early bags until laterals of the corresponding flights open. In this BHS, there is a racking system to store totes with bags or without (empty locations), cranes with a double load unit device, and pickup and delivery stations at the end of each storage aisle. The ASRS has 12 storage racks and 13 cranes, where each rack is accessible by two cranes.

Main Sorter: This is a conveyor based sorting system, which sorts bags to corresponding laterals.

Laterals Build Area: this is the main build area, including the sorter(s) and the laterals. In this BHS, there is one sorter with 18 build laterals and one exception handling lateral to collect bags that missed their flights.

Robots Build Area: in this BHS, robots can be used to build ULDs for different flights, where each flight may use robots for 45 minutes, starting an hour before lateral open time (for the flight). There are five robots, two of which are fully automated, while the other ones are semi-automated. The semi-automated robots can handle bags at a higher rate, because a number of operators work on them. Section 4 discusses the build operation further.

Conveyors: system areas and resources are connected by a network of conveyors, which we take into account in our control methods, but do not model explicitly as they are at a low level of control that mainly execute higher level control decisions.

The key performance indicator (KPI) for BHSs is the *irregularity rate (IR)*. For a certain plane, the IR is the number of bags *per 1000*, which are supposed to be on the plane but are not. From a practical point of view, minimizing the IR is most challenging

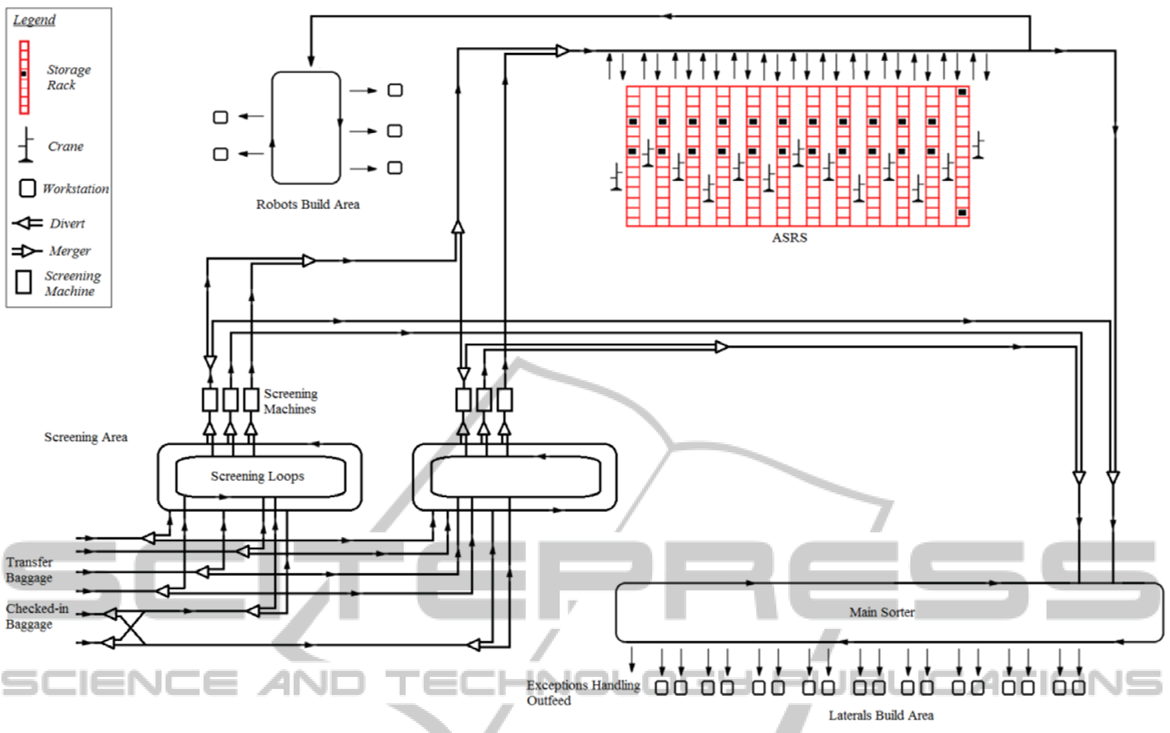

Figure 1: A basic Material flow diagram of the BHS.

when dealing with connecting flights. This is because several things can go wrong when trying to correctly deliver an arriving bag to the next connecting plane within a given (often short) time window.

In large BHSs it is common to have several resources that can do the same job, e.g., parallel screening machines and redundant transport systems. The logistic control must use these resources as efficiently as possible to minimize the bag's flow time in the system.

4 CONTROL ARCHITECTURE

This section presents the control architecture and the decision-making processes involved in this BHS.

4.1 The Architectural Design

Haneyah et al. (2013a) analyze MHSs in different industrial sectors and propose the building blocks of a generic control architecture. Their control architecture consists of three hierarchical levels: planning, scheduling, and local traffic control, where each level contains several generic controllers. Local traffic control problems are the easiest to deal with, as they do not affect the overall control structure or the communications between different controllers. Control methods for local traffic problems can be integrated in this control architecture with minimal difficulty. We focus on the higher levels of control, i.e., planning and scheduling, which are the levels of which the functionality highly depends on the control structure and communication interfaces.

Planning control units, i.e., *planners*, have an aggregate view of the system and are not directly connected to system resources. Scheduling control units, i.e., *schedulers*, are directly connected to system resources, where each workstation or transport resource has its own controller. Planners communicate with each other and assign tasks to subordinate schedulers. Schedulers also communicate with each other to schedule the assigned tasks and report to higher level planners. We now present the controllers involved in this BHS and their main duties (see Figure 2):

Build Planner: this controller is responsible for the build areas, i.e., it coordinates the build workstations being manned-laterals or robots. For a BHS this means planning the build operations for flights, i.e., it activates the build of certain baggage groups on workstations and communicates with the storage planner to request the release of relevant bags from the ASRS.

Storage Planner: this controller is responsible

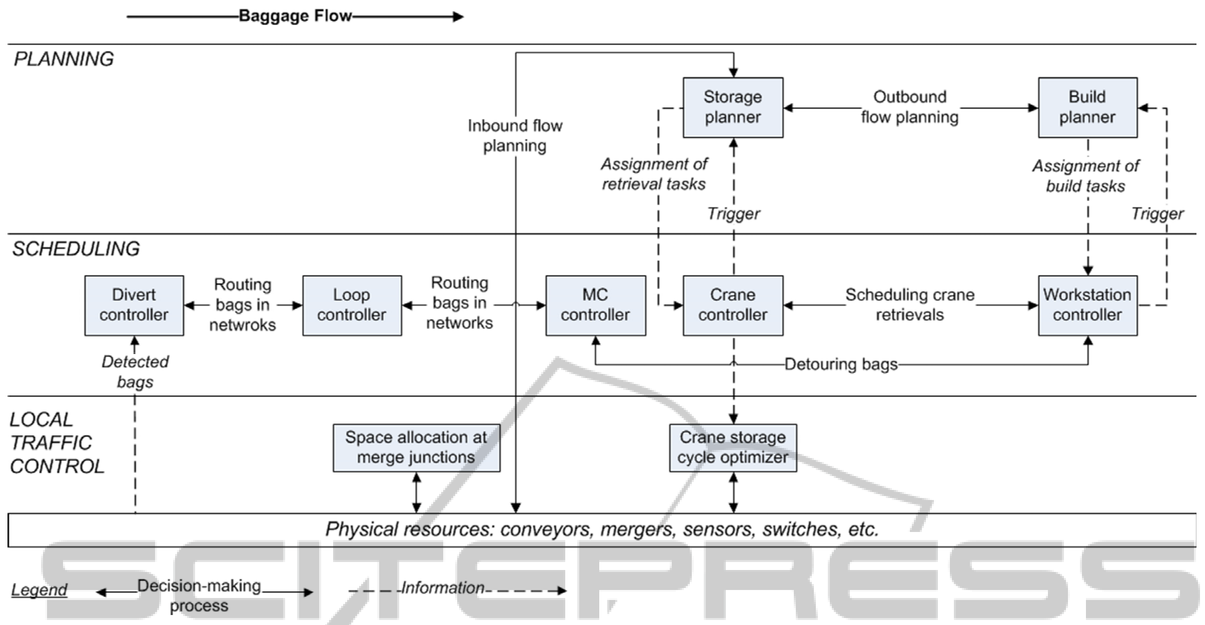

Figure 2: Control structure.

for the storage area, i.e., the ASRS consisting of cranes and storage racks. The storage planner assigns retrieval tasks to subordinate crane controllers based on information from the build planner. This controller also examines the possibility of releasing baggage groups to build ULDs at robot-workstations (see Section 4.2).

Workstation Controller: in the modeled BHS we have two types of workstations, i.e., laterals and robots. Flight build times are planned beforehand on laterals and so laterals are reserved for certain flights during some time frame. On the contrary, robots are flexible workstations that can be used to fill a ULD for any candidate flight. A candidate flight at a certain moment in time is a flight that is allowed to use robots at this moment in time and that has a sufficient number of bags in the ASRS to fill a ULD. A lateral workstation triggers the build planner when the planned time to build a flight commences, while a robot triggers the build planner when it finishes the build of a certain ULD and can start a new one.

Crane Controller: at the scheduling level, the main task of the crane controller is to schedule the retrieval tasks (timing and sequencing). This scheduling process considers the urgency of retrieval tasks and the pipeline of destined workstation(s).

Machine Cluster Controller: a machine cluster controller is responsible for monitoring a group of parallel machines that are connected to the same resources upstream and for posting information to upstream controllers about estimated throughput times. To this end, information about bags in the pipeline from upstream controllers is required.

Loop Controller: loop controllers participate in the routing process by transmitting information from downstream machine clusters to upstream diverts and the other way around. In addition, loop controllers post information about space utilization on the loop (see Section 4.2).

Divert Controller: using information transmitted from downstream controllers, the divert controller makes scheduling decisions on diverting bags to which downstream system in the screening area.

4.2 Decision-making Processes

Haneyah et al. (2013b) apply a generic control architecture to a material flow model entailing a sorter system and an ASRS. Their model can be configured either as a baggage handling system or as a distribution system. Haneyah et al. (2013b) report on interesting results with regard to achieving a high level of synergy in control over common decision-making processes in different industrial sectors. Moreover, they show how uncommon decision-making processes (which are specific to one industrial sector) are built as functional add-ons to the generic control architecture.

In this paper, we detail the generic architecture further on the BHS at hand, which adds the screening area and the robots build area (Section 3) to the material flow model of Haneyah et al. (2013b). In the following, we describe the decisions taken at each level of control and communications that are taking place between different controllers.

At the ***Planning level*** of control, decision-making processes are: planning the inflow of bags to the ASRS and planning the outflow from the ASRS towards build areas.

Inbound flow to the ASRS: When a bag requires storage, it is announced to the storage planner, which responds with a destination rack and crane to perform the storage operation. The storage planner selects the storage rack with the smallest number of bags, and with at least one active crane on it.

Outbound flow from the ASRS: In current practice, cranes trigger a higher level of control that they are ready to perform a retrieval cycle. At the higher level of control, two different approaches are employed to release baggage:

- Retrievals for robots are released on an individual basis considering the pipeline of the destined robot and the sequence of bags required in the ULD. Also, robots have priority over laterals.
- Retrievals for laterals are released in baggage groups considering a limit for the ASRS on the rate of retrievals for a certain baggage group. The baggage groups are classified at a high level of control in different priority classes according to the number of bags in the ASRS and to the planned retrievals' finish times of these bags. Priority classes of the baggage groups and the limits on release rates are dynamic and updated continuously as they depend on time and on the number of bags in the ASRS. The high level of control assigns the bags that the triggering crane has to retrieve in its next cycle. Prioritization rules within a certain priority class are also considered.

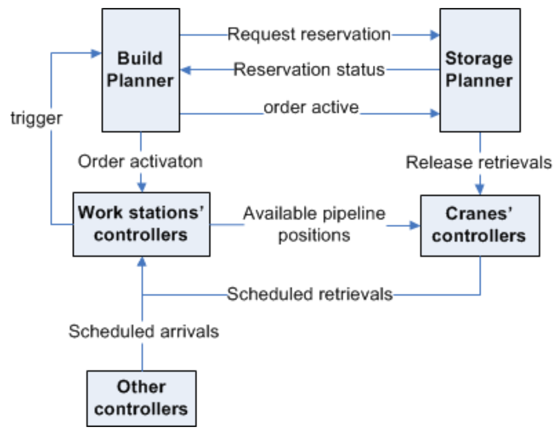

Figure 3: Communications in planning and scheduling.

In the proposed control approach, we have a generic release approach that is based on standardizing the two types of workstations (laterals and robots). This standardization requires: first, setting pipeline size limits for all workstations and second, imposing due times on all crane retrievals (see Figure 3). Baggage is released from the ASRS in groups, where a baggage group can be as large as all bags belonging to a certain flight or a sub-set of these bags defined by the storage planner. There are two main sub-processes in outbound flow planning:

- Stock reservation: this is a generic planning process that Haneyah et al. (2013b) use in the distribution sector to assign product totes to orders. In baggage handling, this process is often not needed because each bag entering the system via check-in desks or as transfer baggage is already assigned to a specific flight. However, in our model the extension to the robots build area requires the use of this process, because the storage planner has to make a selection of bags from a possibly larger set to be assigned to a certain ULD. In other words, when a robot-workstation announces its availability to build a ULD, the build planner inquires the storage planner about candidate flights to build a ULD for. Candidate flights should have enough content of bags in the ASRS to fill a ULD and should be within the time allowed to build ULDs on robots. The storage planner informs the build planner about the options. The build planner may then request to release a baggage group (ULD content of bags to be selected by the storage planner) for a certain candidate flight (e.g., earliest departing) and assign this group to the triggering robot.
- Order release: workstations trigger the build planner to activate the build of baggage groups, based on work progress for robot-workstations (baggage group is the content of a ULD), or according to planned build times for lateral-workstations (baggage group is all bags for the concerned flight). As soon as a baggage group is active on a workstation, bags belonging to this baggage group have to be released from the ASRS. Therefore, the build planner informs the storage planner that a certain baggage group is active. In turn, the storage planner dynamically assigns relevant bags to candidate cranes as retrieval tasks, i.e., if the reserved bag is accessible by more than one active crane, then the storage planner assigns the retrieval to the crane with the smallest workload. Moreover, the storage planner sets due times for retrieval tasks. The due time for bags going to robot-

workstations is a parameter that we use (see Section 5), whereas the due time for bags going to a lateral-workstation is the planned end time of the flight build.

At the ***Scheduling level***, decision-making processes are: scheduling crane retrievals, routing by diverts, and detouring bags.

Scheduling crane retrievals: Given a set of retrieval tasks, crane controllers schedule these tasks based on their due times and the pipelines of destination workstations. The pipeline size of a workstation is the number of bags that can be in transport to this destination at any point in time. We only send bags if the number of bags in the pipeline is less than the pipeline size, in order to prevent overloads and congestion. We use the pipeline size concept often in our control in the following manner: each workstation receives information about incoming bags (from cranes or diverts) and in turn updates the number of empty positions remaining in its pipeline. This includes scheduled retrievals that are not physically in the pipeline yet. Other controllers, e.g., crane controllers, observe the pipeline capacities and consider this information when scheduling. For robot-workstations, the flow has to be strictly controlled, because bags are to be handled according to a predetermined sequence. Hence, recirculation of bags due to blocked entry to a robot is prohibited, and so the pipeline size is typically equal to the number of locations in the inbound buffer of the robot-workstation. In this way, if any problem occurs in the workstation, then all bags in transport can be accommodated in the inbound buffer with a preserved sequence. For lateral-workstations, we set the pipeline sizes according to Formula 1 as developed and motivated by Haneyah et al. (2013b) who use a time allowance parameter (to account for traffic delay and blocking considerations) and the following variables:

PS_i = pipeline size of workstation *i* (bags).

TT_i = Average transport time from all cranes and the arrivals divert to workstation *i*.

C_i = capacity of workstation *i* (bags/minute).

$$PS_i = C_i * (TT_i + Allowance) \qquad (1)$$

When the pipeline to a certain destination is full, system controllers react by blocking further bags from being retrieved or routed to this destination, which implies a pull-concept for workload control.

Routing bags in networks: Haneyah et al. (2013a) outline an approach for routing items in networks with service points, where divert controllers make the routing decision based on the state of the system downstream. We apply this approach in the screening area, where screening machines (see Figure 1) are available at alternative systems. In such configurations, a divert controller has to decide which system to divert an incoming bag to. Contrary to current practice where static shortest path algorithms are often implemented, we propose a dynamic control logic.

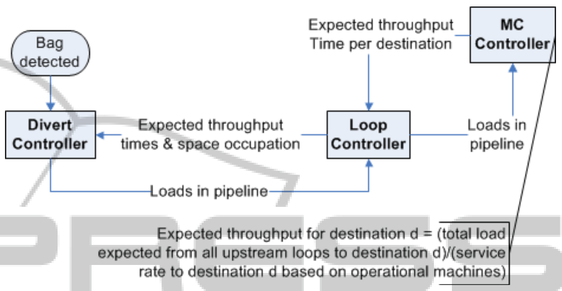
Bag detected
Divert Controller
Expected throughput times & space occupation
Loop Controller
Expected throughput Time per destination
MC Controller
Loads in pipeline
Loads in pipeline
Expected throughput for destination d = (total load expected from all upstream loops to destination d)/(service rate to destination d based on operational machines)

Figure 4: Routing bags in networks.

We aim to balance the load on parallel systems and react to machine failures, i.e., fewer bags should go to the system having a lower service rate. We focus on expected throughput time to pass through the system. As the bag can go either to the laterals build area or to the storage area, we calculate the expected throughput time per destination. To this end, machine cluster controllers post expected throughput time to pass through (see Figure 4). In turn, upstream controllers use this information to make routing decisions. In order to estimate throughput time, downstream controllers need information about bags in the pipeline from upstream controllers. The decision to divert a bag to system A or to system B is impractical to take for each bag separately due to the high rate of bags passing the divert at a high speed, and may cause excessive switching of diverts (which is undesirable for the equipment). Therefore, the divert is positioned to one of the downstream systems until the difference in throughput times between downstream systems exceeds a certain threshold. Then, the divert switches position to react to the imbalance. As long as throughput times are balanced, we check whether space occupation on downstream loops is balanced in the same manner. We use a dashboard logic to post status updates to upstream controllers. Each component (cluster, loop, or divert) has a dashboard that posts accessible destinations downstream, expected throughput times, and space occupation on loop(s) downstream. Upstream flow is blocked when the system has completely absorbed the allowed capacity.

We emphasize that the standardized controllers and interfaces allow the implementation of the same control logic on different system layouts by merely defining connected controllers upstream and downstream for each resource.

Detouring Bags: One screening machine per cluster has connections to both the main sorter and to the ASRS. Obviously, bags are routed to the ASRS when the build for the corresponding flight is not open yet. Also, if the build is open and the pipeline(s) of the destined workstation(s) is (are) not full, then the bags are routed to the main sorter. However, if the pipeline(s) is (are) full then, in order to maintain a controlled flow on the main sorter, we test routing bags to the ASRS and delegate the scheduling task to crane controllers there. The latter option is not used for urgent bags as it may cause them to miss their flight. In this case, bags are better off recirculating on the main sorter.

At the ***local traffic level***, we model two local traffic control processes in an aggregate manner as they do not affect the overall architecture and merely execute scheduling decisions. The first process is space allocation at merge junctions, e.g., the loop controller has to allocate free spaces on the loop to bags waiting to enter the loop from outbound buffers of cranes. For this merge problem, we dedicate another study (Haneyah et al., 2011). The second is the crane storage cycle and in-aisle travel optimizer, which concerns the determination of travel sequences for a crane within an aisle, e.g., to execute a storage cycle to store bags waiting on its inbound buffer. The latter is a simple problem for which a simple algorithm or control rule can be incorporated in the architecture. Note that schedulers are responsible for workload control, because they decide when to start planned tasks, e.g., retrievals. Moreover, pipeline size limitations reflect a pull system for material flow. In contrast, local traffic controllers have to deal with material physically moving as a result of scheduling decisions.

5 IMPLEMENTATION

In order to test the performance of the generic control architecture and compare it to current practice, we build a simulation model in *UGS-Tecnomatix Plant Simulation* that includes the main building blocks of the BHS under study and of the architecture. First, we need to tune the control parameters: threshold values for routing in parallel systems, the time allowance in Formula 1, and due times on retrievals to robots.

Experimental setup: The operational scenario covers a day of operation in each simulation run. We include common screening machines failures occurring during normal operation with exponential distribution for the time to failure (mean = 6 hours) and an exponential distribution for repair time (mean =10 minutes). Each simulation run includes: 61 flights where each flight is scheduled to build on two laterals for 75 minutes, ending 15 minutes before the scheduled time of departure. There are two operators working on each lateral at a handling capacity of 120 bags per hour per operator. Automated robots and semi-automated robots handle bags at a capacity of 200 and 350 bags per hour respectively. Finally, the number of bags modeled per simulation run is 25,198 including transit and check-in bags.

Model Parameterization: We test the generic routing logic in parallel screening systems and the detour option (Section 4.2). We test the detour option versus always sending bags to laterals when the build time of the flight is open (even if the pipelines are full). Routing parameters (see Section 4.2) are selected according to desirable values in practice. We use a throughput time threshold of one minute, which means that a divert switches position only if switching leads to at least one minute saving in throughput time downstream. Likewise, the space utilization threshold is set to 20%, which means that, as long as throughput times are balanced, a divert switches position only if switching leads an incoming bag to the loop which is at least 20% less occupied than the other accessible loop downstream.

Table 1: Setting due times for retrievals to robots.

Option	Due time criteria	IR	ULDs built
1	Robots build close time *Flight-based*	27.33	78
2	Capacity-based, with slack *ULD-based*	0.095	475
3	Capacity-based with allowance *ULD-based*	0.095	479
4	Experimental *ULD-based*	0.071	481

We conduct several experiments to configure the time allowance parameter in Formula (1). The best value for the time allowance is found to be 100 seconds for both options; with detour and without detour. We observe that, in this BHS, the detour option is seldom used (on average 0.4% of all bags were detoured) and does not have a big effect on the IR. The causes for this result are: first, the possibility of the detour option is limited to only two out of the six screening machines in the screening area due the design of this particular BHS (see Section 3). Second, in this BHS, a bag can be detoured if it has at least 30 minutes until departure.

In Section 4, we state that in order to achieve standardization among workstations, it is necessary to impose due times on all retrieval tasks assigned to cranes. Then, cranes can have a standard approach for retrievals and do not have to distinguish retrievals to robots from those to laterals. While due times on retrievals to laterals are straightforward (Section 4.2), we have to set due times on retrievals to robots. We study several options (see Table 1) to set due times and test the effect of each option on the IR. We also observe the number of ULDs built by all robots during a day of operation.

Option 1 builds upon the fact that each flight has 45 minutes allowed to use robots, starting one hour before laterals open, but gives unacceptable results. What happens is that a retrieval to a certain flight may have 45 minutes until its due time, when it is released early, but another retrieval to the same flight may have few minutes until its due time, when it is released towards the end of the time allowed for the flight to use robots. A resulting behavior is that crane controllers delay the retrievals to robots as they had initially long time until they are due. Thus, robots remain waiting and do not trigger for more work because the assigned ULD builds were not processed. Moreover, when retrievals for robots are due soon, cranes start working on them and delay retrievals to other flights building on laterals.

Consequently, a due time for each ULD is more logical. For Options 2 and 3 in Table 1, we propose a time interval allowed to build a ULD. We calculate this interval using the number of bags planned for the ULD divided by robot service rate. To take into account issues such as transport times, delays and blockings, we plan for less than 100% robot capacity (Option 2, best at 60% of capacity), or plan for 100% capacity and add a time allowance (Option 3). In fact, we found it best to use experimental build times for ULDs (Option 4). We achieve a 0.071 IR when 15 minutes is given to build a ULD for any robot. Although semi-automated robots have a higher handling capacity than automated robots, we achieve best results when both robot types are given the same time to build a ULD. Giving both the same time to build a ULD at some point in time means that the due times for all retrievals released at this point in time are the same. Thus, cranes work on retrieving bags for all robots in the system, but if due times for semi-automated robots were few minutes sooner, then the cranes would serve these robots and delay the automated robots. So, at the system level, bags are handled at a lower rate.

6 RESULTS AND DISCUSSION

We compare the performance of the generic control architecture to current practice for this BHS, and find that both approaches achieve zero IRs under normal operational conditions. Thus, we analyze other performance indicators (see Table 2). We observe that, due to the pull concept, generic control performs better in minimizing the percentage of bags that have to recirculate on the main sorter, which decreases traffic delays. However, current practice compensates for less output on the sorter by better utilization of robots (more ULDs are built). This occurs because retrievals for robots get priority in current practice, while in generic control there is no strict distinction between workstations. The percentage of bags receiving special handling due to missing the lateral close time is comparable.

Table 2: Generic control versus current practice.

Performance Indicators	Generic control (no detours)	Generic control (detours)	Current Practice
Average traffic delay	1.05 sec	0.90 sec	2.20 sec
Maximum traffic delay	79.58 sec	64.06 sec	175.43 sec
Percent recirculations	15.93%	15.44%	53.94%
Percent detours	0.0%	0.12%	0.0%
Number of ULDs built	480	481	490
Percent special handling	0.02%	0.02%	0.01%

Current practice approaches at the planning level are customized, complex, and computationally intensive. On the other hand, the generic control architecture identifies the decision-making processes at the right level of control, and handles layout-specific details by configurable parameters. Hence, the architecture is scalable and tunable to different system layouts and designs. Moreover, the architecture allows for a much faster implementation and is both flexible and more robust; still without compromising overall performance. Finally, we stress that the comparisons we made are based on normal operational conditions. When more severe and unexpected disturbances in the material flow occur, we expect generic control to outperform current practice as generic control reacts directly to problems in material flow and takes actions to avoid possible congestions and imbalances.

In the following, we list some of the lessons learned, which are useful when trying to implement the generic control architecture to other MHSs:

- Only use generic modules, e.g., we model any type of *build* workstations by the generic workstation module. Customized modules hamper the generic structure.
- Maintain standard interfaces between different controllers.
- System size and layout characteristics should not affect the implementation of generic scheduling processes. This is shown by our application of the generic dynamic routing in the screening area of the BHS.
- The planning level of control is generic, but may include system-specific business rules in making decisions, since it is the interface to the users' processes. This level can also include some algorithms to make decisions within certain controllers. For example, the storage controller may use an algorithm to make decisions on which bags in particular are to be assigned to a certain ULD.
- A distributed decision-making structure is necessary, as it supports the modularity, robustness, and generic nature of the control architecture. For example, if a central controller executes routing decisions, then it would not be able to easily handle different system layouts.
- Specific algorithms can be used for local traffic control, they can be easily integrated as add-ons to the control architecture and do not affect communication at the higher levels of control.

7 CONCLUSIONS

In this paper, we provided a proof-of-concept for the applicability of generic control for MHSs in different sectors. One of the advantages is generic modeling of workstations, being laterals or robots. This resulted in a simpler control software for the retrieval process. Moreover, we implemented a dynamic routing strategy that uses the dashboard logic to make routing decisions and to react to breakdowns and congestion. These control methods have a modular and generic structure, which allows them to be implemented in different BHSs and different MHSs in other industrial sectors.

In future research, we will return to the concept of generic control and apply the generic control architecture to a large MHS in the distribution sector. In this MHS, we will study a large ASRS and a large order picking area. The latter consists of four order picking stations and a network of conveyors. For this future implementation, we will test the applicability of the generic control architecture and analyze the extent to which we maintain the structure of standardized control procedures. In particular, we are interested in studying the applicability of the generic routing method to a MHS in the distribution sector, and in comparing the implementation in the distribution sector with the implementation in the baggage handling sector.

REFERENCES

Babiceanu, R. F., Chen, F. F., and R. H. Sturges (2004) Framework for the control of automated material-handling systems using the holonic manufacturing approach. *International Journal of Production Research*. 42(17): p. 3551-3564.

Hallenborg, K. and Demazeau Y. (2006) Dynamical Control in Large-Scale Material Handling Systems through Agent Technology, *in Intelligent Agent Technology, 2006. IAT '06. IEEE/WIC/ACM International Conference on*, pp. 637-645.

Haneyah, S. W. A., Schutten, J. M. J., Schuur, P. C., Zijm, W. H. M. (2013a) Generic Planning and Control of Automated Material Handling Systems. *Computers in Industry*, 64(3): p. 177-190

Haneyah, S.W.A., Schutten, J. M. J., Schuur, P. C., Zijm, W. H. M. (2013b) A generic material flow control model applied in two industrial sectors. *Computers in Industry,* in press (DOI: 10.1016/j.compind.2013.03.007)

Haneyah, S. W. A., Hurink, J. L., Schutten, J. M. J., Zijm, W. H. M., and Schuur, P. C. (2011) Planning and Control of Automated Material Handling Systems: The Merge Module. In: B. Hu, K. Morasch, S. Pickl, and M. Siegle, editors. *Operations Research Proceedings 2010*, Part 8, pages 281-286, Springer Heidelberg Dordrecht London New York, 2011. ISBN 978-3-642-20008-3.

Johnstone, M., Creighton, D. and Nahavandi, S. (2010) Status-based routing in baggage handling systems: Searching verses learning. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*. 40(2): p. 189-200.

Tařau, A., De Schutter B., and Hellendoorn H. (2009a) Centralized versus decentralized route choice control in DCV-based baggage handling systems. In: *Proceedings of the IEEE International Conference on*

Networking, Sensing and Control, Okayama, Japan, March 26-29, 2009.

Tařau, A. N., De Schutter B., and Hellendoorn H. (2009b) Hierarchical route choice control for baggage handling systems. In: *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems,* St. Louis, MO,USA, October 3-7, 2009.