# Term Thesis

Xujia Li

## Agent-based Control to Enhance the Reconfiguration of Material Flow Systems

Matrikelnummer: 03684894

Betreuer: Juliane Fischer, M.Sc.

Ausgabe: 01. November 2017

Abgabe: 30. April 2018

Lehrstuhl für Automatisierung und Informationssysteme
**Prof. Dr.-Ing. B. Vogel-Heuser**
Technische Universität München
Boltzmannstraße 15 Geb. 1
85748 Garching bei München

Telefon 089 / 289 – 164 00
Telefax 089 / 289 – 164 10

http://www.ais.mw.tum.de

## Statutory Declaration

I, Xujia Li, born on December 22, 1994 in Fujian, P.R. China, hereby declare, that I composed the enclosed thesis with the title “*Agent-based Control to Enhance the Reconfiguration of Material Flow Systems*” by myself, solely supported by my academic supervisors Juliane Fischer. I have not used any sources or means, but the ones declared. Any thoughts of others or literal quotations are marked as such.

This thesis has not been published elsewhere in the same or in a similar version.

Garching b. München, den 30. April 2017

Xujia Li

# Contents

# 1 Introduction

Industry 4.0 is a megatrend in automation and manufacturing field, which is strongly supported by modern information technology. One obvious challenge in this technical revolution is the current inefficient automatic control system of a production line, which is inflexible and unresponsive facing increasing product diversification. In other words, manufacturing companies need one universally adaptable production system to handle different kinds of products in a short term, which only requires a small change of whole system with a little engineering effort. A significant step to upgrade the flexibility of a production system is to design a corresponding material flow system (MFS) with high flexibility and reconfigurability, which is able to achieve required layout changes quickly without any interaction of human beings.

However, current control software of MFSs is normally constructed in a single node, which is only suitable for traditional static MFSs. And it is time-consuming to repetitively develop new control software and adapt it to a mutable MFS. Therefore, a powerful software system fitting for such frequently changeable MFSs is required, which can be realized with two main design principles, decentralized decision and interoperation ability [1]. Decentralized decision means there are various decision-making nodes supporting the controlling unit of MFS instead of a single central one during processing. Interoperability means MFS should be able to establish connections between machine tools, sensors, conveyors and people. Thus, the core problem to enhance flexibility and reconfigurability of MFS is replaced by how to involve these two concepts into control software.

An agent-based automation system, named multi-agent system (MAS), is gradually concerned by modern industry, because agent concept is suitable to be implemented considering the decentralized and flexible controlling strategy. The whole MFS is divided into several independent parts, namely modules. Each intelligent agent, who can communicate with each other, controls one local module to execute specific operation. And from view of system level, this MAS enables MFS with a completed database of whole plant and an intelligent data processor, who can coordinate the task of each module with help of a data-exchanging agent.

Due to this structure, each part of MFS obtains the plug-and-play characteristics, which make any arbitrary module function like a building block. Engineers can freely add or remove modules to satisfy new production line without reprogramming control software, which will greatly shorten the interval of production period. In conclusion, this agent system improves not only the utilization rate of data, but also the flexibility of MFS to achieve the requirements of Industry 4.0.

The existing version of MAS in Institute of Automation and Information Systems (AIS) is based on the works of Regulin [2], Aicher [3], Dengler [4], Fischer [5] under the research project named iSiKon[1]. The software is coded by language IEC 61131-3 and loaded on programmable logic controllers (PLCs) to accomplish evaluation task. And with the inserted communication method of Beckhoff PLC, the agents can communicate with each other and send commands to real operating plant. The concept of MAS has been developed over several years in AIS. After the enlargements of MAS done by Fischer [6], the performance of latest MAS facing some common use cases in logistics domain has been evaluated in simulation.

## 1.1 Aim of Term Thesis

After the simulating phase mentioned above, it should be a comprehensive evaluation of MAS carried out on a real model of industrial production line. Out of this purpose, AIS assigned this term thesis and planned to implant MAS on “Hybrid Process Model" (HBFG), which possesses higher complexity and offers possibility to be divided into independent modules for evaluation. Further introductions about HBFG is exhibited on AIS website [7].

Meanwhile, several unfinished topics should be solved, which are defined after simulation of the enlarged MAS. Firstly, de-registration of a module, or called plug-out, is unachievable because of the limited number of modules in previous simulation model. Secondly, one significant ability of MAS, automatically route determination facing transportation route alternatives, is not evaluated in simulation, because the complexity of MFS is insufficient. Thirdly, at the present stage, only two types of modules are considered, conveying and handling. This is far from enough contrasting to the real

---

[1] Increased flexibility for heterogeneously structured material flow systems enabled by intelligent software agents controlling self-configuring conveyors (iSiKon), research project funded by the German Research Foundation (for details see [34])

production systems. More real scenarios should be added to make MAS more universal.

## 1.2 Design Rationale

To subdivide the main targets assigned, an overview of design rationale is explained in this section.

***Step 1: Preparation***

Understanding the project background of iSiKon and basic concepts of MAS is helpful to get a deep view of modern automation. Furthermore, adaptation work requires to master IEC 61131-3 for PLC programming (Structure Text) in the environment of Beckhoff TwinCAT. Except for software, getting familiar with hardware equipment is also needed, which includes control parameters of motors, meaning of feedback signal from HBFG, network configuration, etc.

***Step 2: Adaptation of existing MAS to demonstrator HBFG***

This is the main task of this thesis. First, the demonstrator is divided into suitable modules according to four defined scenarios, which require enough complexity and practicality. Subsequently, the measurement about the location of each module should be carried out to complete module dividing plan, which keeps core position throughout adaptation.

According to this plan, loading current MAS program on each agent (PLC) and adjusting code to fit hardware is followed, which is to establish corresponding connection between HBFG and controlling program, for example, the software side takes signal from light sensor on the plant as input and then feedback a start signal to next module to execute next order. After fully adaptation, MAS should be evaluated based on four scenarios on debugged demonstrator.

***Step 3: Improving current drawbacks and continually enlarging MAS***

After implementation of the MAS concept at HBFG, drawbacks and potentials to improve the concept are identified, because HBFG has several first-appeared characteristics, which are quite different from any other model used before. In this step, several new concepts should be added into MAS, such as the route selection criteria when alternative routes exist.

***Step 4: Documentation***

In this step, all practical works, together with project background, preliminary knowledge and state of the art, are concluded into a thesis.

# 2 Preliminary Knowledge and Background

In this chapter, the preliminary knowledges about material flow system, multi-agent system and programmable logic controller, which are highly related to project iSiKon, are briefly explained.

## 2.1 Material Flow System

Entering the 21st Century, very large innovation of machining technology was hard to be achieve. Many companies dispelled the idea about improving production efficiency by speeding up mechanical processing, because the tremendous investment to update hardware equipment is not economic when compare to other methods, such as outsourcing, localization of production or software development. In 1980s, Taiichi Ohno, leader of Toyota, advanced a project named Toyota production system (TPS), which gradually developed into the theory of lean production today [8]. In this theory, value creation and waste reduction during production are key points to ramp up production. Five of seven main wastes [9] are highlighted here: Transporting unrequired products to processing sites; Inventory of finished products; Extra motion of people or equipment when perform the processing; Waiting for the next production step. In conclusion, all the mentioned problems are related to the allocation of manpower and material resources inside a production cycle, namely the management of MFS or called logistics system.

### 2.1.1 Modern Material Flow Management

Modern production logistics need to be more systematic and flexible because of the required adaptability to modern production requirements. Modernization of material flow management is mainly reflected in three aspects according to Stevenson [10]. The first is the modernization of equipment, such as the real production lines represented by model HBFG, which contain varieties of sensors and motors with high degree of freedom. The second modernization includes the use various software for management, such as the central theme in this thesis, multi-agent system. And the third is the concept of management ideology about systematization and integrated thinking.

Modern production logistics is specifically managed by four means: production information management, status monitoring, job management and material management [11]. MAS is designed to accomplish first two items with high independency. Among these four means, Information management is the core and foundation of modern material flow management. The production line is actually the combination of material flow processing and information flow processing. In the flow of materials, there are enormous parameters about quantity, physical location, and variety of materials, which are changing with actual processing needs. To manage them, the collection, disposal and transmission of information should be synchronously executed. In other words, controlling entities of physical flow is the purpose during the operation of MFS. And to achieve this goal, the management carried out is based on information. Due to this motivation, MAS are developed to upgrade the ability of data processing, which will be introduced detailed in later chapter.

### 2.1.2 Hybrid Process Model

The Hybrid Process-Model (HBFG) is aimed to implement and evaluate the concepts and approaches investigated at the chair AIS. With the maturity of multi-agent system, the project is stepping into the phase to test MAS on a model with high degree of similarity compared to a real plant.
The configuration of HBFG is shown in Figure 1.

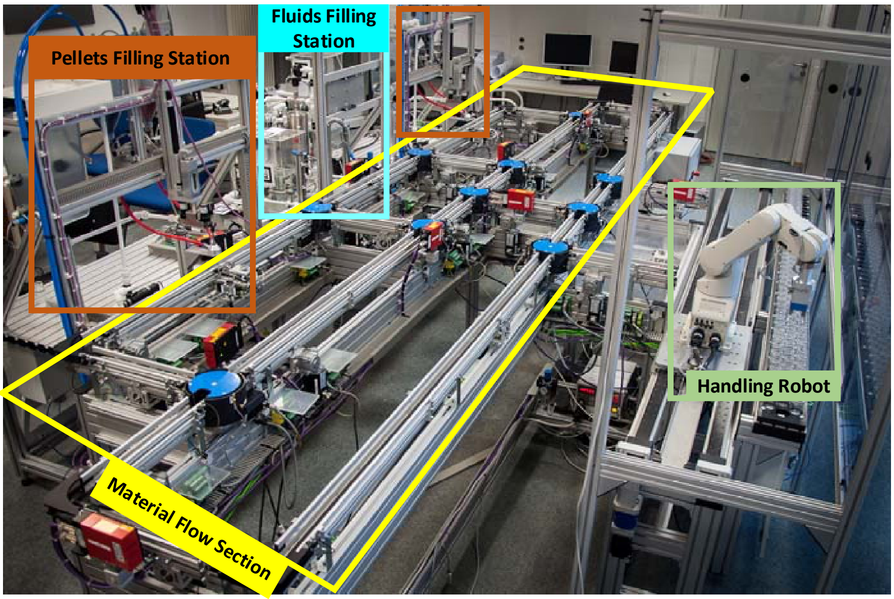



Figure 1 Consisting Sections of HBFG Demonstrator [7]

The HBFG consists of mainly four sections. Handling Robot is responsible for placing bottles in and taking them out from material flow section or called logistics section. Two designated positions on conveyors for the robot arm to place bottles are treated as global input and output points of the MFS in this thesis. In Material Flow Section, bottles will be transported to planned manufacturing position by conveyors and switches, which are equipped with light sensor on each terminal. With the signal from sensor, the controlling system can follow the status of bottles, and do corresponding operation, such as starting motors of switches to change the orientation of transportation. Two type of filling stations are regarded as manufacturing machine in real production line. According to different orders, filling blue pellets or red, MFS will transport the bottle to appointed location. And after “manufacturing”, the bottle will be sent back to output position by MFS and handed over to robot arm. In this thesis, there will be a comprehensive explanation about using MAS to control Material Flow Section and achieve all its functions.

Two main operating components to execute transport task are conveyor and switch, which is shown below in Figure 2. When a bottle passes by the light barrier and comes into the groove of switch, the control software will receive a signal from HBFG and send back a driver signal to motor, which turns the orientation of switch to the desired exit. And then the subsequent conveyor takes over the bottle to continue the transportation.

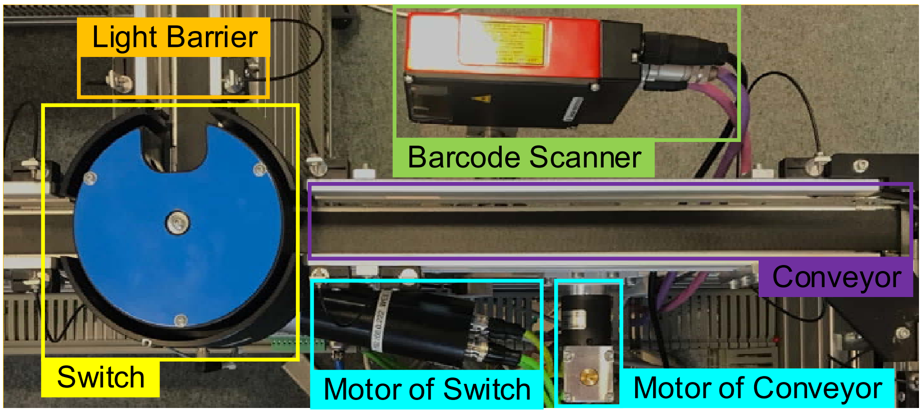



Figure 2 Devices on Logistics Part of HBFG

There are two input parameters of conveyors, transport direction and velocity. As for switches, the inputs are initial direction and output direction. In case of different products, user side should define these inputs accordingly. And then PLCs deals with this information based on imbedded software, to translate it into electrical signal, which are acceptable by actuators on the plant.

Since HBFG is a compositive plant with equipment from several vendors, there are three kinds of filed bus serving the communication system, EtherCAT, PROFIBUS and PROFINET. Due to the benefits of EtherCAT, faster communication and better motion control [12], the channels of data exchange between PLCs and HBFG or among different PLCs are built by EtherCAT.

### 2.1.3 Material Flow Cost Accounting (MFCA)

In current stage, MAS has only been evaluated with simple MFSs, which only possess one route of transportation. But in real industrial production, there is always a complex network of transportation lines. This indicates that, for the same source of raw material and the same sink of products, several potential transportation alternatives exist simultaneously. But there is usually an optimal one decided by different strategies. This situation is also displayed on the MFS of HBFG. In order to add decision-making ability into MAS when facing the choice of optimal route. The theory of MFCA is introduced in this section.

MFCA is an environmental management method, was proposed by the German Institute of Business Environment (IMU) at the beginning of this century, and received good results facing the requirements of modern MFS. It helps engineers better understand the potential environmental and financial consequences of potential material transportation logistics, which is supported by energy flow in real-world, and seeks to improve them via finding optimal solution in practices [13]. It does so by assessing the physical material flows in a company or a supply chain and assign adequate associated costs to them.

The MFCA requires the material center to trace the manufacturing processes, measure and analyze the information from both in-kind and monetary aspects, so as to obtain the gains and losses of various production processes.

The MFCA classifies the material cost and logistics cost involved in manufacturing processes into three categories: raw material cost, system cost, and delivery and handling cost. These negative costs align with several wastes inside production line from energy domain, shown in Figure 3.

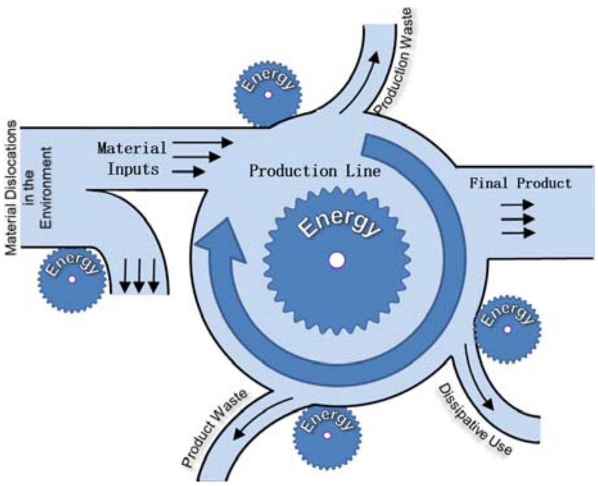



Figure 3 Negative Costs in Production Line (edited based on [14])

The cost of raw materials mainly includes three categories of materials, power, and direct purchase cost. The system cost refers to the cost incurred for the maintenance and support of production during the internal circulation of the company's materials, including labor costs, depreciation, and other related administrative expenses. The transportation of a material between two points on MFS are also counted in system cost. Delivery and handling cost refers to external shipment of a company and includes the cost of transportation and waste disposal fee.

From the view of HBFG, most energy is consumed by supplying the motors on the plant. In order to minimize energy waste during running of the plant, finding the optimal route automatically, which possesses shortest distance or minimal duration between source and sink, is another pivotal task of MAS.

### 2.1.4 Inventory Control System in MFS

Inventory control system is a method, which focuses on management of the status of inventory in a supply chain. Inventory is also a significant part in MFS domain, because the number of products being processed on a production line should also be monitored by inventory control system, so that the material flow and manipulating machines can cooperate with each other to ensure the production moving forward smoothly.

According to David et al. [15], inventory control system helps to master the dynamic state of inventory for making appropriate orders timely to avoid overstocking or out of stock. Reducing inventory space usage and total inventory costs is also included in the control system. Similarly, Costanza [16] extended idea of inventory control into level of a single MFS. On the production line, it is important for the material to reach the point of manipulation at or before the use of the material. In addition, in order to reduce material inventories and storage space in manufacturing plant, companies hope to keep the smallest amount of material. For this purpose, Costanza also invents a scheduling method, to arrange and replenish raw materials reasonably in time and space, which is quite fit for the case of HBFG, because HBFG is capable of processing several transportation orders at the same time.

Except for the mentioned effects, another important meaning of inventory is about buffering of overflow to avoid logistic conflict. For example, if one bottle is transported in clockwise direction on HBFG, and another one is being

processed in the opposite direction. The way to avoid the “impact” of them is calling the buffer inventory to store one of them temporarily.
In conclusion, inventory system is indispensable for most of MFSs, because of its core position to allocate resource and the ability to solve special cases. Thus, in this thesis, a new kind of module, named Buffer Module, is developed into MAS for emulating a real warehouse on MFS.

## 2.2 Intelligent Agent and Multi-Agent System

In this chapter, multi-Agent System and some relative concepts, which construct the foundation of software domain in this thesis, are detailed introduced. Include the definition of “Agent”, the development history and characteristic of MAS, the standard in this branch of artificial intelligence founded by Foundation for Intelligent Physical Agents (FIPA).

### 2.2.1 Agent

The research and application of agent technology in the computer field originated from the research on Distributed Artificial Intelligence by researchers at the Massachusetts Institute of Technology in the 1970s [17]. When they analyzed the capabilities of information systems to deal with complex problems, they discovered that the system’s ability can be significantly improved by coordinating some simple information systems into a large one. And by defining a reasonable collaboration mechanism, the intelligence of the entire system can be improved. This result evolved into the concept of Agent, which can passively response to the request of information processing. In addition to completing the tasks specified in advance, it can also predict and adapt to various environment and even actively seeks the optimal solution to support the user in completing the task.
Basically, the term agent is a virtual concept, which usually refers to an intelligent subject, who can independently finish designed target in a preset environment without any interaction of humans.
Software Agent is to study "Agent" from the perspective of software design. Shoham [18] and others defined it as follows: "Agent is a software entity that can run continuously and autonomously under certain circumstances and usually jointly solve problems with other Agents." The demand for continuity and autonomy comes from the fact that we need the Agent to complete its activities in a flexible and intelligent manner without human guidance and

interference in response to changes in the environment. The ideal situation is that an agent can operate continuously in an environment for a long period of time and learn from its own experience. Besides that, an Agent can coexist in an environment with other Agents, they should communicate and collaborate with each other. But at present, most of the software agents that have been developed are very fragile and designed for specific purposes, and no one can accomplish these functions in the general form.

Desired agents should complete some of the specific tasks assigned to it as humans, and that it can derive the desired results from what we tell it. The Agent can only do this when it "knows" about the background knowledge of the request. Therefore, the best agent not only needs a specific form of expert knowledge, but also needs to consider the unique nature of the user and the current state. Although there has not yet been a unified definition, there is a weak definition admitted by most scholars. A hardware system or software-based system with the following characteristics is an Agent [19]:

1) Autonomy: Operation without the direct intervention of humans or other systems and the ability of controlling its behavior and internal state;
2) Social ability: Through ACL, agents can interact with others;
3) Reactivity: Perceive the environment in which it is located and respond to changes in the environment in real time;
4) Pro-activeness: Not only simply respond to the environment, but also actively demonstrate goal-driven behavior.

According to the characteristics of the Agent, the Agent's architecture consists of various basic modules, shown in Figure 4 below [20]:

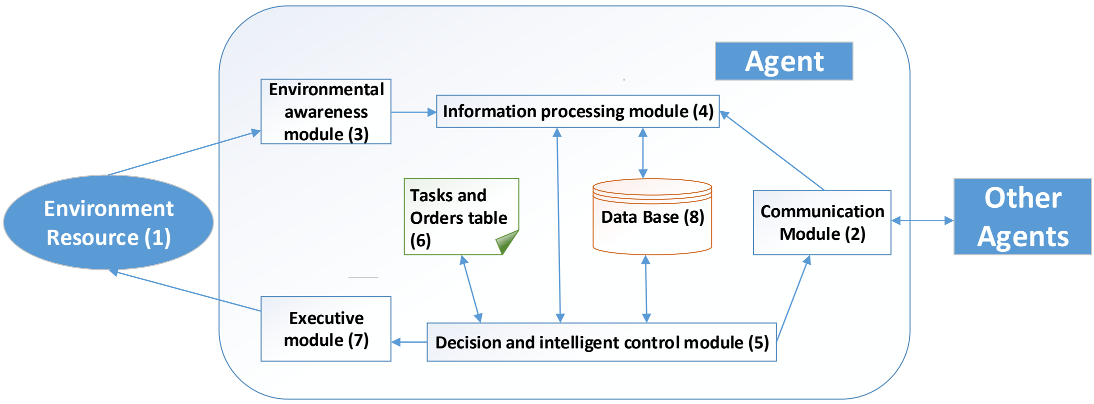



Figure 4 Structure of Single Agent

(1) Environment and Resource generate input information to Agent and wait for the output execution from Agent;

(2) Communication Module is used to communicate with other agents or applications;
(3) Environment Awareness Module filters and classifies input information;
(4) Information Processing Module makes inference about the input data based on the Agent's own data base and knowledge;
(5) Decision Module evaluates and decides the corresponding reaction;
(6) Orders Table formulates an action plan based on the decision;
(7) Execution Module performs the actions as planned;
(8) Data Base provides supports for reasoning, decision making and planning.

### 2.2.2 Multi-Agent System

With the deepening of research, people have found that in human society, most of human activities involve social groups composed of multiple individuals. Solving large-scale and complex problems also requires the collaboration of multiple professionals or organizations. The Agent, as a person's counterpart, should be able to form a similar large system. These thoughts gave birth to multi-Agent System. In the 1980s, Smith and Davis proposed the Contract Net Method, which gave an effective solution of problem distribution and decomposition [21]. In real life, problems are not naturally distributed in the nodes of the system. It is necessary to decompose the problem from a node to a sub-problem. Then, according to the solution capabilities of other nodes in the system, the sub-problems can be assigned to other nodes [22]. This is said to be the origin of multi-agent system research.
Later, MAS is verified that it owns outstanding ability to solve distributed problems. It is proven that MAS can improve the efficiency and robustness of a system. Therefore, when it is necessary to use Agents to solve problems, individual agents with different abilities should combined into a whole system that cooperates to complete a certain task, so as to achieve the purpose that the overall utility is greater than the sum of all parts. In this system, each Agent has its own independent local knowledge base, goals and capabilities. Any two Agents can communicate with each other, coordinate their behaviors, and achieve common goals through cooperation. In this process, the MAS detects and coordinates conflicts between Agent members and divides, assigns, and manages tasks and resources. Most current research in the field of Agents is focused on multi-agent systems.

A mature MAS should own these four characteristics below [22]:

1) Distribution: each agent only has partial, incomplete information, or only has the ability to solve part of the problem, so the role of a single agent is limited. Meanwhile, there is no one global control system in the multi-agent system, data is distributed and stored in different places. The multi-agent system is not only distributed in structure, but also logically distributed. The calculation is also performed asynchronously. Therefore, the multi-agent system is very suitable for parallel operations.
2) Adaptability: for environmental changes and uncertainties, agents can adapt to new conditions outside through interaction and self-learning under the coordination mechanism.
3) Openness: The agent is a package model both conceptually and practically. Its internal structure and algorithm can be implemented by different people at different time and place in different ways. Any new agent can be added through a standard message interface into MAS.
4) Robustness: against external interference, MAS can adjust the parameters through the interaction of agents to maintain system's performance level.

In conclusion, MAS has good robustness and flexibility, and it is very suitable for dealing with distributed problems of data, knowledge, and control.

However, not all system problems are suitable to be solved by MAS. If each part of a system is better to own the ability of independent operation and there is a demand of building interaction with remote program over the Internet and the application must perceive the status of operating environment, MAS is valuable to be chosen.

### 2.2.3 FIPA and Framework of MAS

FIPA is established to develop the standard of agent system in 1995. The first step is about the Agent Communication Language (ACL), aiming at improving the interactivity between heterogeneous agent and agent systems. And a series of technical specifications were developed for this purpose, ranging from architecture, communication language, content language to interactive protocols. The main idea of FIPA is to provide a standard way to understand the communication between agents through the fusion of Speech Act Theory, Predicate Logic and Public Ontology. The standards drafted by FIPA can be used publicly. It is not a specific application technology, but a general technology for different application areas. In other words, it is not an independent technology, but a set of basic technologies that makes divergent development possible. It also provides convenience to develop a complex system with high degree of synergy [23].

The FIPA Agent must run within the FIPA Agent Platform (AP). The platform provides basic service support. Therefore, the platform can also be called as the middleware of the Agent to some extent. The agent management reference model of the FIPA platform is shown in Figure 5 [24].

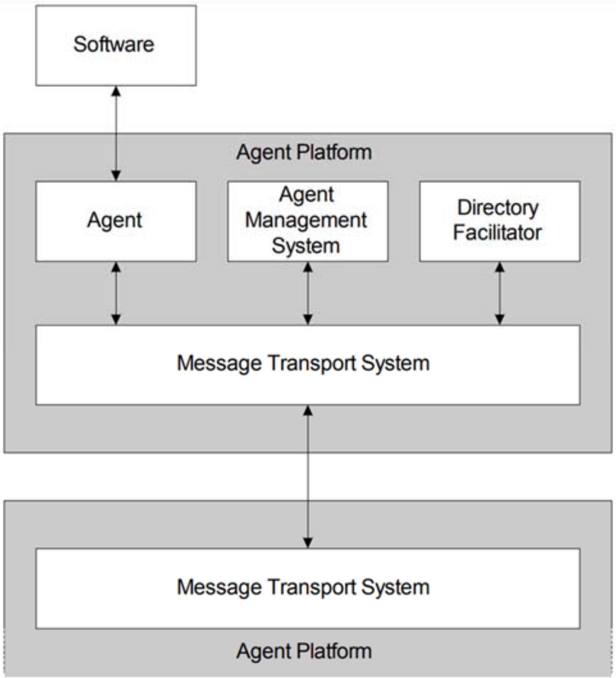



Figure 5 Agent Management Reference Model

The main function of the FIPA platform is to manage the life cycle of the agent, enable the agent to communicate with the agent on the platform or outside the platform, and enable it to access the services provided by the platform.

The FIPA management model mainly has the following functional modules:

1) AMS (Agent Management System) is an essential part of the FIPA management model and is equivalent to the administrator of the platform. It monitors the Agent's access to the platform. It holds the IDs of the active agents on the platform. Each agent must first register with the AMS to obtain a valid ID.

2) Message Transport Service (MTS) provides ACL message exchange mechanism between different agents. The message transmission model is shown in Figure 6 [25]. Message Transport Protocol (MTP) is a physical message exchange protocol between different Agent Communication Channels (ACCs), which support the transmission of ACL messages between two agents on APs. ACC is an entity that provides message interaction for the Agents on an AP, it can also access information from AMS and DF.

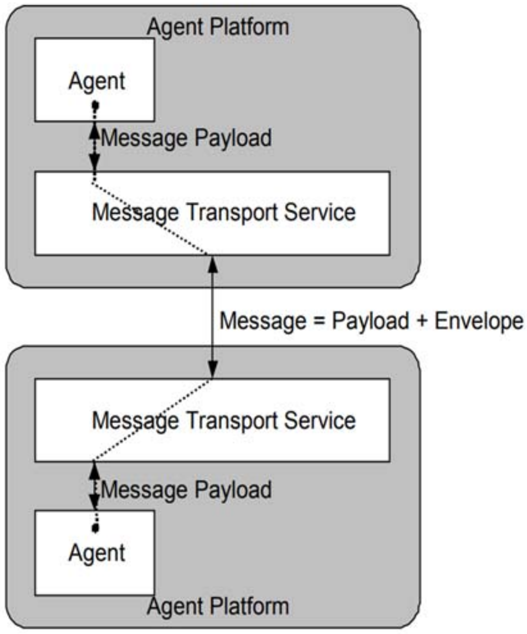



Figure 6 Message Transport Model

3) DF (Directory Facilitator) is also a necessary part of platform. It provides yellow pages within the platform. The Agent can register its own capabilities or service information with the DF and can also query DF for the information about other Agents and theirs services.
4) AP (Agent Platform) is a physical entity that can be deployed and operated by an agent. It includes not only the preceding AMS, DF, MTS but also the operating system and underlying hardware, such as HBFG.

## 2.3 Programmable Logic Controller and IEC 61131-3

Programmable logic controller is the main programming object in this thesis, which completes the tasks to control HBFG with MAS. In this section the basic knowledge of PLC and its programming language IEC 61131-3 are presented. PLC is a digital computing operating electronic system designed specifically for industrial environments. Because of its advantage of high reliability, easy programming, flexible configuration and quick installation, PLC is widely used in industry. PLC can control various mechanical equipment or a completed production line through digital or analog input and output. Due to the advantages mentioned, project iSiKon attempts to create an agent system running on PLCs.

PLC consists of the following four major components: Central Processor Unit (CPU) is to process and run user programs (MAS). PLC Storage is to store system programs, user programs, logic variables. Input Unit is the input interface between the PLC and the controlled device (HBFG). Its task is to receive the signal from the master side (user computer) and the detection equipment (light barriers). Output Unit is to convert the weak electric signal from CPU into a level signal and transmit it to drive devices (Motors).

Working process of PLCis generally divided into three stages, namely input sampling, user program execution and output refresh. The completion of the above three phases is called a scan cycle. During the entire operation, the CPU of PLC repeatedly performs the above three phases at a certain scanning speed. In the input sampling phase, PLC reads all input states sequentially in scan mode and stores them in the corresponding cells and keep them unchanged until a completed scan cycle. During the user program execution phase, PLC always scans the user program in order from top to bottom. Then, according to the result of the logic operation, the state of the corresponding bit in the I/O mapping area is refreshed. When the user program is scanned, PLC enters the output refresh phase. During this period, the CPU refreshes all the output latch circuits according to the corresponding states and data in the I/O mapping area and drives corresponding peripherals through the output circuit.

IEC 61131-3 is the third part of a PLC standard developed by the International Electrotechnical Commission (IEC) in 1993 [26]. The advantages of IEC 61131-3 can be concluded in two aspects: First, IEC 61131-3 supports modularization, divides used program functions into several units, and encapsulates them to form the basis of programming. During modularization, user sets as few necessary input and output parameters as possible to minimize internal data exchange. The second aspect is that IEC 61131-3 supports top-down and bottom-up program development methods. The user can first perform the overall design, divide the control application into several parts, define the application variables, and then program the various parts: This is top-down. The user can also start programming from the bottom, for example export function and function block first, and then program according to control requirements: This is bottom-up. The bottom-up method is used for adaptation of MAS in this thesis, from the bottom part of a single module to the whole agent network.

# 3 State of the Art

In this chapter, some previous researches using the agent concept to enable MFS to possess high flexibility and capability of reconfiguration are briefly introduced and compared with each other to make a clear view of development direction in this area.

## Design of MFSs with the Support of MAS

Beyer et al. [27] propose a method to speed up the efficiency of redesigning a MFS facing various production conditions, which can highly improve the transparency of a material handling system based on agent system. They simplify the design of material handling systems by developing a decentralized approach. In this approach, the material handling system is modelled as a complex adaptive system, which contains a number of agent systems. Each agent can describe in detail the functions and related parameters of a certain system part. Then based on the inserted information exchange program, the data of each part can be read and used by each other. The design approach consists of composition, configuration and adaptation. In composition step, agent detects the neighbouring agents by matching parameters of interface and checks whether such connections fulfil the production requirement or not. And then stepping into configuration, agents collect all necessary information from connected modules and calculate their capacity. In the end the system integrates and adapts these partial solutions and provides evaluation of each potential solutions to planer. This bottom-up planning process and some of the previously defined rules provide the engineers with strong support, when choosing a new MFS layout.

Beyer et al. state a feasible method to design but not control MFS during runtime. There is no consideration about calculating transportation or manipulation sequences and also no decision-making ability facing multi potential alternatives of transportation routes.

## Decentral Control and Communication of Dynamics MFSs

According to Wilke [28], there is a large cost of the entire logistics and controlling system nowadays, because the number of batches is decreased, and meanwhile the number of transportation orders is increased. In other words, small-batch production is an inevitable trend of development but based on high production and logistics costs. Therefore, in the dynamic production scenario, the material handling system is mostly adopted manually, because it is too complicated to be controlled by the existing automated logistics system. Aiming at this problem, Wilke [28] develops an automated MFS containing communication ability of each sub-part, which fulfil requirements of current dynamic production structure.

For the technical design of logistics system, a function-oriented modularization method is also proposed by Wilke. Using this method, the logistics system installs mechatronic modules based on system boundaries and unit's function. Based on mentioned instructions, the concept of vehicle control and communication is followed. Wilke refers to the scenarios happening in real traffic system. The car drivers are regarded analogous to mobile modules in MFSs, and the function of traffic sign is transformed into a matrix, namely waypoint matrix, which contains all input information for a mobile module. Based on this structure, each mobile is able to decide the transportation direction and terminal independently, which achieve decentral control. Every module possess authority to read and write data of waypoint matrix, which means waypoint matrix also functions as a communication bridges between modules. After a route is chosen by the first mobile module, some parameters of waypoint matrix will be revised. This revision will influence the later decision of another module. This indirect connection endowed all module of MFS the ability of communication.

## MFSs Control based on Internet of Things (IoT)

Chisu [29] develops a new type of decentralized MFS and inserted the concept of “Internet of Things” to support a distributed data network. Three kinds of intelligent agents, which are loaded on each transportation units, are suggested to achieve route choice, communication and order assignment. This means, there is no hierarchical command unit functioning as the central decision-maker.

The system architecture consists of standardized and extensible components that can be implemented iteratively based on each other. Chisu sets three classifications of hardware, namely serve platform, transportation platform and module platform. And in the domain of software, MAS constructs the fundamental framework and IoT is responsible for the communication and order distribution. With these function-oriented modularization, the difficulty of controlling a complex MFS are highly reduced.

Beside introduction of IoT into MFSs, Chisu also enriches conveying and storing ability on each module. This setting should be supported by a method, which is able to calculate routes for each order. Chisu refers to the waypoint matrix presented by Wilke [28]. At the beginning, when an order is set, all potential mobile module, who is able to handle this order, are ranked by the distances between its current position and the target. Obviously, the module with shortest path wins the order. And then, same negotiation process repeats to elect the order successor until the transport object reaching final termini. The procedures to realize order assignment is based on the blackboard approach in computer field, which is an artificial intelligence approach mainly to solve the complex, ill-defined problem, where the solution is the sum of its subparts [30].

After simulation of this system, Chisu concludes that although the intervention of IoT cannot simplify the complexity, but actually increase the decentralization of control logic, which benefits from the high frequency of data exchanging and the distributed information storage. The importance of interconnection of a system is verified, because it makes the blackboard approach more feasible in the domain of MFSs control.

### MDE for Generating Reconfigurable Automation Systems

Priego et al. develop an agent-based approach to support reconfiguration procedures from the view of control hardware [31]. The control system is divided into several mechatronic components (MCs). Every controller owns the potentiality to control all MCs in the system. But at runtime, only the part of software controlling a certain MC is activated in each controller. When one of controllers fails, its belonging MC can be reallocated into remaining controllers to ensure the availability of whole system during runtime. To achieve reallocation, the knowledge base of each MC contains information about its relocation to a different controller. Priego et al. focus on the failure of control hardware (PLCs), rather than the failure of devices on the plant (MCs), which is defined as an important scenario in the scope of this thesis.

## Dynamic Configuration of Communication Interfaces of PLCs

Under the structure of iSiKon, a routing tool, which can monitor the existence of all PLCs in a MFS and feedback this information to each operating module, is developed by Dengler [4]. MAS should own the advantage of plug-and-play property. Following this guide line, each time a new module is added to the system or removed, there is no massive calculation to rearrange the system. The routing tool replaces the manually reset during run time to achieve this goal. Via ADS protocol, PLCs can communicate with computer through Ethernet in the environment of TwinCAT. With the support of this technology, the routing tool recognizes every PLC with their typical IPv4. Moreover, this address also takes the role as postal address for every single communication between any two modules. For example, in order to notice the successor module to execute the following transportation order, the communication agent belonging to current module should search through the data base about the address of the successor, and then, send the order to continue the transportation task until the end. And routing tool inside each module keep updating the list of address in database to ensure every module agent is aware of their neighborhood and operating environment. The routing tool is written in C# and for evaluation purpose is exclusively designed for the network of Extended Pick and Place Unit (xPPU)[2] in AIS.

Dengler also set an electoral mechanism to select the most suitable module to be central processing unit. Every module is ranked using Bully-Algorithm according to its centricity, availability period, size together with the computing power and work load of connected PLC. The specific definitions of these items are introduced in [4].

After supplement these two algorithms, Dengler evaluates communication ability and reconfigurability of xPPU under agent-based control and concludes that the plug-and-play characteristic is fulfilled the set objective and the initialization of MAS are faster without any manual operation.

## Enlarged functionalities of MAS for general use case in MFS

Fischer et al. [5] enlarge the functionality of modules in MFSs from pure transportation to various operating possibilities with high degree of freedom,

[2] The xPPU is used at the AIS as a demonstrator for new technologies and as an evolution. It is used for processing (stamping) and sorting 10 cm high, cylindrical workpieces.

namely manipulation ability, which can fulfill the requirements of many common use cases in real production. Manipulation ability have a wide definition. In the scope of Fischer's work, manipulation mainly refers to the operation to change orientation and position of a transport object in three-dimension space.

Regarding to the "increase throughput" scenario in MFS, Fischer chooses the grouping station for simulation purpose, which is usually the site to receiving boxes at the end of MFSs and sorting them into a reasonable combination for subsequent container transport. The packaged boxes are transported into grouping station in arbitrary orientation and incoming position. Coordinator will compare the current state of box with the preset layer pattern at accumulation point and calculate the routes of transportation and manipulation order. Fischer designs two Portal Modules in series to deal with increasing steam of box. The model of grouping station under the control of the enlarged MAS are shown in Figure 7.

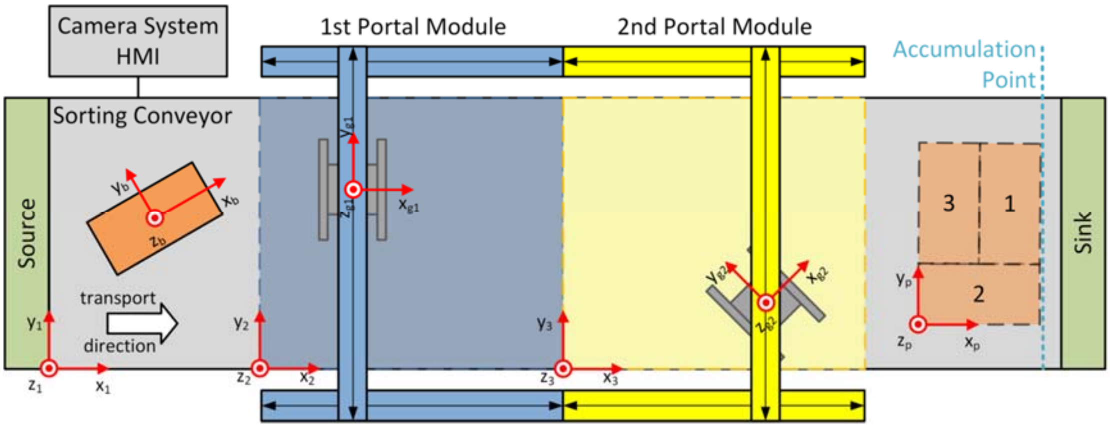



Figure 7 Grouping Station Controlled by enlarged MAS [6]

However, Fischer uses this simulation for testing the enlargements, but there is no practical application of current MAS to a real MFS with high number of modules in the same system.

# 4 Details about Module Dividing Plan

Dividing a completed MFS into suitable modules is the first step to adapt MAS. The plan, which describes all necessary parameters of each module, called module dividing plan. This plan functions as the fundament for any further adaption work.

In this chapter, the structure details about dividing the completed logistic part of HBFG are explained from the view of hardware, which includes the boundary of each module, the affiliation of each motor or light sensor, the global position of modules in the system and so on. The first part is about the strategies of division, which should fulfill the requirements requested by four evaluating scenarios according to real use cases in production. The second part presents the dividing map, showing 4 modules and accessory conveyors and switches. The last part is the network connection plan from user computer, via PLCs and finally to the motors on HBFG, which is also taken apart to fit the decentral requirement of agent system.

## 4.1 Four Scenarios in Real Use Case and Requirements of Module Dividing Plan

Each time when new series of product is stepping into stage of batch production, the corresponding new configuration of MFS should be modified based on the existing system. From the domain of MAS, the reconfiguration of MFS is equal to the addition or removal of modules. In order to evaluate current version of software agent system on HBFG, the four scenarios which occur with a high probability during reconfiguration of a MFS are stated below.

Table 1 Four Scenarios for Evaluating MAS on HBFG

| **Scenarios for Evaluation** |
| --- |
| 1. Initializing the system. |
| 2. Addition of a new module to system during run time. |
| 3. Removal of a module from system during run time. |
| 4. Removal of the module with active coordinator from system during run time. |

Initialization approach of current MAS (Scenario 1) can be divided into two steps. Firstly, with the server tool developed by Dengler [4], each activated module can realize other existing module, which is represented by NetID of each PLC, in the system. And then based on the present electoral mechanism,

the module with the highest score is initialed as coordinator, as the central process unit for the whole MAS.
One of main tasks of active coordinator is to keep checking the existence of all functioning modules during the run time of MFS. Coordinator should be aware any change of the whole system. When adding or removing modules (Scenario 2 and 3), coordinate will handover this information to AMS. AMS recalculates and decides new transportation sequences to ensure the MFS operating smoothly with frequently changed MFS.
Coordinator takes core position in current version of MAS. But in some specific situations, the module with activated coordinator could be removed (Scenario 4). If this special case happens during run time, all module will resume negotiation to select a new coordinator with the support of the communication Agent. In order to prevent the crush down of all system, this coping mechanism against the removal of coordinator should also be evaluated in this thesis.
Considering four scenarios above, there are some necessary requirements of module dividing plan. The complexity of the system should as similar to real production line as possible. Only with enough number of modules, the removal and addition scenarios can be realized and build a changeable operating condition to evaluate the stability and decision-making ability of MAS. The dividing plan should also add diversity into the system, which help to distinguish the importance of each parts on HBFG. Only with variety of modules, the method inside MAS to choose coordinator can be evaluated.
The dividing plan should also consider the characteristics of each MFS. Only two import and export terminals are available on HBFG, so no matter how to adjust the existence of modules, this two global I/O point should be always connected. In this thesis, only the logistic part of HBFG is evaluated. But in further step, to realize full capabilities of this plant, filling stations and robot arms should also be controlled by MAS in this foremost plan. For the future implementation of manipulation modules, agents on filling station should communicate with each other to decide which one to execute filling order of the coming bottle. On the other hand, the scenarios, that manipulating multi orders on the plant at the same time, also request that two filling station should be assigned to two modules. As the importance of inventory on MFS discussed in 2.1.4, one section in Logistic part should be defined as Buffer Module, to make this model more likely to real plant.

## 4.2 Module Dividing Plan

According to the scenarios and corresponding requirements mentioned above, the divided plant should own as more modules as possible, which can provide various combinations of modules. Meanwhile, because the number of PLCs connected with HBFG are limited currently in the demonstrator room, the final number of modules is set to four considering all objective conditions.

The Module Dividing Plan on HBFG are present in Figure 8. Four modules are highlighted in different colors. Each module contains inner conveyors, switches, light barrier and related substructure like motors and cables. Since only Logistic part of HBFG is under the framework of this thesis, robot arm, barcode scanner, filling stations don't belong to any modules in Module Dividing Plan. Two global input and output terminals are respectively located on module 1 and module 3, which indicated that during the whole evaluation process, module 1 and module 3 aren't removable, because the possibility of breakthrough between source and sink should be ensured. For scenarios 1, 2 and 3, module 3 will be activated as coordinator based on the electoral mechanism defined by Dengler [4] introduced in Chapter 3. The reason can be stated mainly in two aspects. Firstly, compared to Module 1, Module 3 is connected with all other three modules, thus the factor of centricity is larger. Another reason is Module 3 owns one of global I/O interfaces and one of filling stations, which indicates a higher value of available period than Module 2 and Module Buffer. The evaluation of scenarios 1 to 3 can be carried out by addition or removal of Module 2, to check the real-time decision-making ability and reconfigurability of MAS. About specific scenarios 4, Module 2 is manually set to undertake the role of active coordinator. Theoretically, based on electoral mechanism, Module 3 will become as new coordinator after Module 2 is removed.

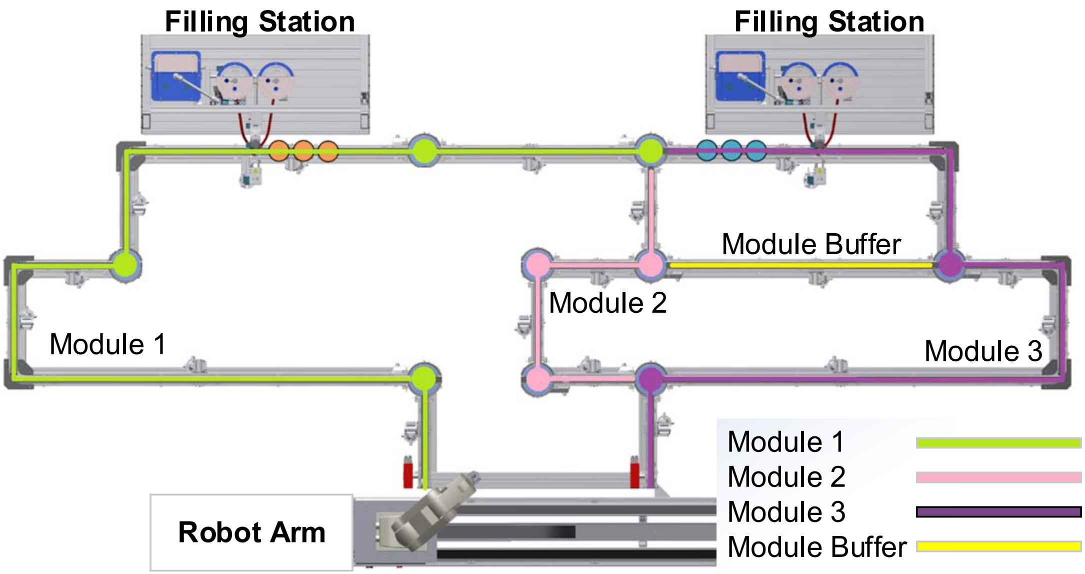



Figure 8 Module Dividing Plan of HBFG

There are many important parameters for module description and motor controlling details recorded also in Module Dividing Plan. The completed Plan is shown in Appendix A. Since the dividing plan is highly related to a very specific use case (HBFG) and also need to hold general concepts in MAS, two major types of information contained in module dividing plan are divided as follow.

General parameters requested by control software (MAS):

- *Interface ID*: Each terminal of a module is regarded as an interface, no matter it is a global I/O interface or a connection interface to another module. The Interface ID is combined with three numbers, such as 322. The first number shows the Module it belongs to; the second one means the neighboring module of this interface, if it is a global I/O interface, the second number will be the same as the first number; the third number indicates the transport direction through the interface. If the bottle is transported from global I/O terminal of Module 1 to global I/O terminal of Module 3, the third number will be 1, also named as positive direction. And for the opposite direction, the third number is 2. Under this rule, Interface 322 is the Interface belonging to Module 3 and connected with Module 2. And the bottle passes through this interface in the opposite direction. The Interface ID supports the further definition of Ability ID, Link ID in the domain of agent system. The interface IDs of all three modules are respectively enclosed in Appendix A (Table 6, Table 9 and Table 12).

Specific parameters requested by characteristics of different MFS (HBFG), which are enclosed in the Device Description Table in Appendix A (Table 8, Table 11 and Table 14):

- *Device ID and Conversion Table of Initial Box Number of conveyors and switches*: The device ID is written as C-18 or S-02. The first capital means Conveyor or Switch. The following number docks with corresponding global variables in Logistic Function, which is a function block at bottom level of MAS to generate signal and drive the motor directly. But when initializing devices into software program, PLC scan all the motors through EtherCAT in a specific sequence. Each motor will be shown as a “Box No.” in TwinCAT Device List. Thus, the conversion table helps to align these two kinds of Motor ID, so that engineers can precisely locate the variable from software domain to remote hardware.
- *Interface coordinates*: Right hand coordinate system is established on horizontal operating plane, z-axis is upward in vertical direction. Based on size measurement, each interface is presented by two coordinates. These coordinates are used to check the overlap between two interfaces, which helps agent system to recognize its neighborhood and build global mapping. The interface coordinates of all three modules are respectively enclosed in Appendix A together with interface ID (Table 6, Table 9 and Table 12).
- *Orientation Value Table of Switches*: Each Switch have two or three orientations to collect or export bottles. Each orientation of a switch is a typical value of corresponding input variable in Logistic Function. By changing value of these variable, the Switch can turn to the desired orientation and execute order.
- *Default direction of Conveyors*: Each conveyor has two transportation directions. Normally, the value of default direction of a conveyor, positive or negative, is according to the sign of actual speedy when transporting a bottle from module 1 to module 3.
- *The length of conveyors* is a determinant in Cost Function to choose optimal route.

In conclusion, when adapt MAS to a MFS, drafting Module Dividing Plan is the first and most significant step, which directly decides the performance of MFS in flexibility and reconfigurability.

## 4.3 **Network Connection Map**

According the module dividing plan, the cables to transmit signal between PLC and MFS should also be taken apart from a radial central structure to a decentral network. Following requirements of plug-and-play principle, each module together with one connected PLC forms an independent block. Only after these blocks access to the same network, the communication bridge between them can be built by MAS.

The decentral Network Connection Map and corresponding addresses of HBFG are shown in Figure 9.

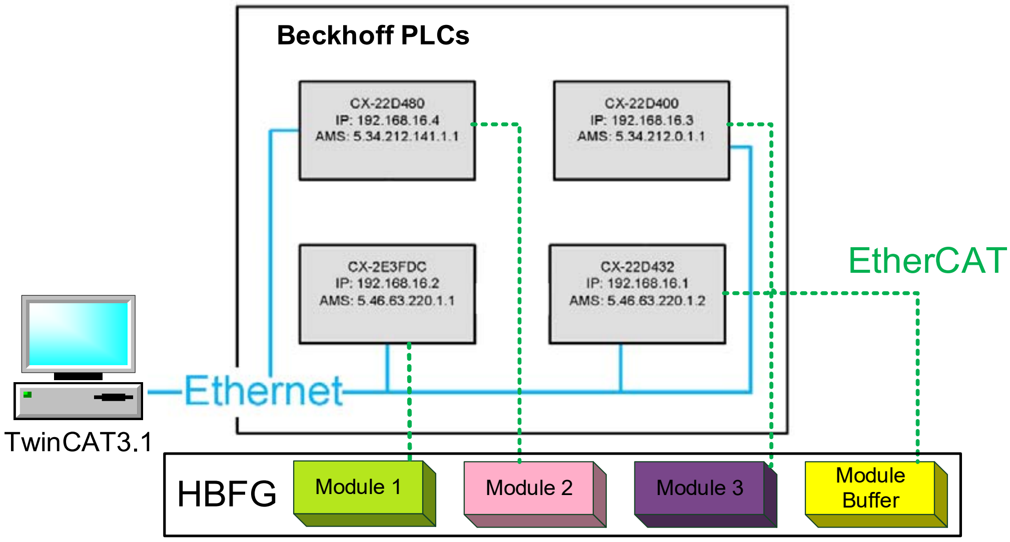



Figure 9 Decentral Network Layout

# 5 Implementation of MAS on Demonstrator HBFG

In this chapter, the currently adapted MAS fitting for HBFG will be divided and introduced in three sections, the framework, local control system on the level of each module and the Logistic Function, which is the driver function to directly control the motors on HBFG.

The structure of MAS after adapting to HBFG is shown in Figure 10. When agent system controls the plant in real time, two primary information streams circulate between modules or inside module to accomplish responding tasks. The first one (pink, dash line) is about collecting, sorting and calculating system data base, which can describe the current status of the whole MFS. This information normally flows from bottom level to central coordinator. The other stream (black line) is to transmit information of order assignment and execution, mainly originates from activated coordinator and flows through related modules following an established transportation sequences.

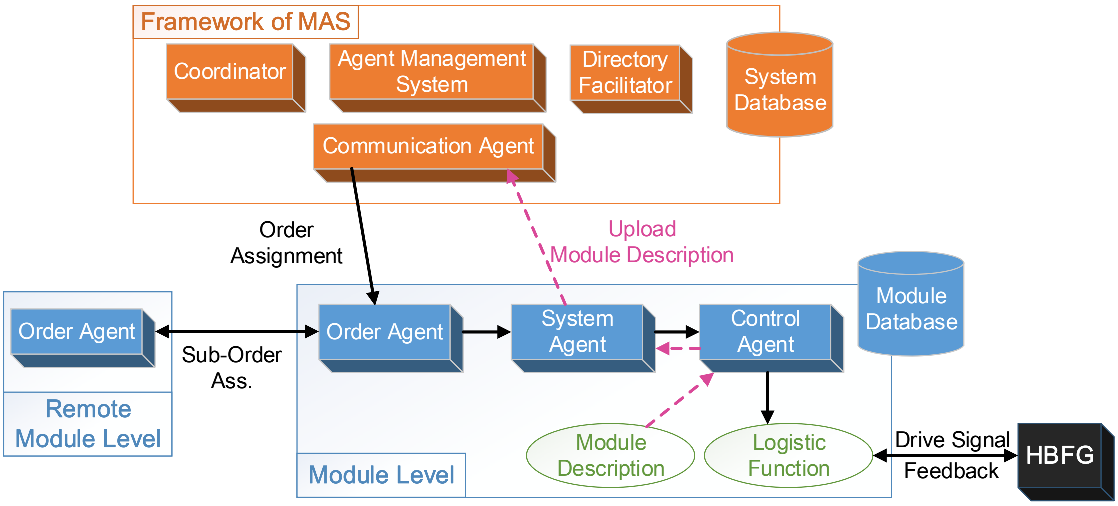



Figure 10 Structure of MAS adapted to HBFG (edited based on Fischer [6])

The following sections give a comprehensive introduction about the definitions of each component in Figure 10 and the program flowcharts to present the processes to control HBFG with MAS.

## 5.1 Framework of MAS

According to FIPA standard [24], there is a part of the MAS loaded on each module responsible for constructing a framework to make every divided module cooperate with others and form a completed MFS. This part of program is identical for all modules, which consists of four main function blocks and belonging methods. The definitions of each function block and their main tasks are stated in Table 2.

Table 2 Definition and Task of four FBs in Framework

| **Name of FB.** | **Main Tasks** |
|---|---|
| Agent Management System (AMS) | Register or deregister modules when the configuration is changed; |
| | Collect raw module description from all running modules; |
| | Sort inflowing information and calculate completed system database; |
| | Calculate transportation sequences for each new order; |
| Communication Agent (ComA) | Support active coordinator and other modules circularly checking the existence of each other; |
| | Support active coordinator gathering data from remote modules; |
| | Handover order assignments from active coordinator to the first executing module, or from previous module to the successor module; |
| Coordinator | Select most suitable module to be coordinator; |
| | Generate new order; |
| | Call AMS and DF to execute related commands and calculation; |
| Directory Facilitator (DF) | Store and manage all NetID of running modules in MFS; |
| | Order assignment; |

After setting up all line connections based on Network Connection Map and activating configuration of MAS program, all modules are initialized with function block Coordinator and then wait for a new order. The flowchart, which demonstrates the working principles of Coordinator, are shown in Figure 11.

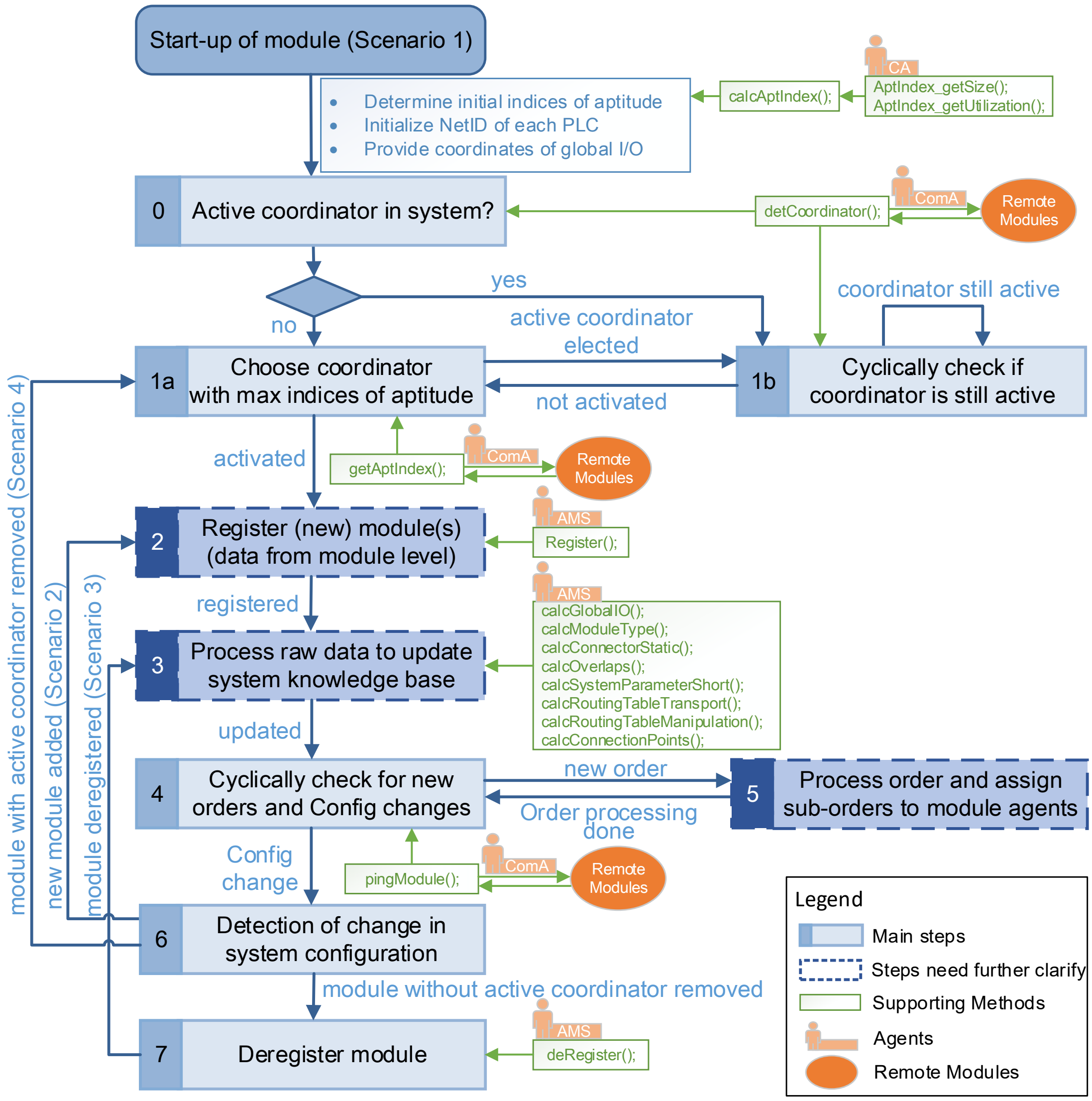



Figure 11 Flowchart of FB. Coordinator (enlarged based on Fischer [6])

For initialization step, the NetID list of all modules in network and the global I/O location should be preset manually in current version. According to Dengler [4], the indices of aptitude is the input parameters of the method, calAptIndex(), which scores each module and elects the most suitable module to be active coordinator. And following the guide of activated central coordinator, the control system on the one hand monitors the configuration of whole running MFS corresponding to step 2 and 3. And on the other hand, coordinator waits for the signal of new order to start-up relative functions in step 5.

When the structure of MFS is changed as defined scenarios, coordinator will activate reconfiguration steps to register or deregister modules, recalculate and update system data base. The flowchart of registering new added module is depicted in Figure 12.

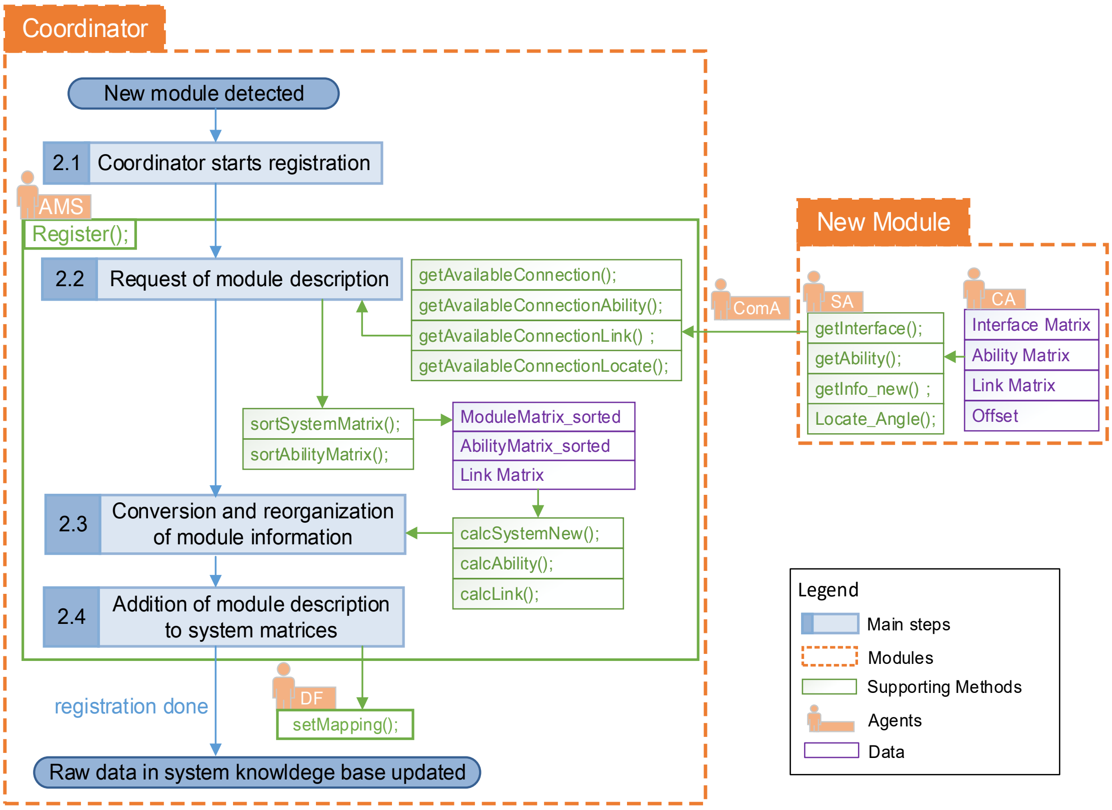



Figure 12 Flowchart of New Module Registration (enlarged based on Fischer [6])

For deregister a module, AMS will simply erase all matrixes belonging to this module and recalculate system knowledge base.

For scenario 4, the inactivated coordinator function blocks in all running module should trace back one more step and make joint efforts to elect a new Coordinator. Then MAS starts all over again to configure the system knowledge database.

In conclusion, the registration is performed by coordinator, highly supported by the collecting methods of AMS, which are able to access data from remote database of new added module (depicted in purple square), including Interface Matrix, Ability Matrix and Link Matrix, the components of mentioned matrixes are stated in Table 3.

Table 3 Matrixes of Module Descriptions

| **Data Type** | **Definition** |
|---|---|
| Interface Matrix | Contains coordinates, Interface ID and type of interface (input of output) for all interfaces in a module. |
| Ability Matrix | Contains type of ability (transportation or manipulation), Ability ID and Moving Directions of a transported object referring to local coordinate system. |

| | |
|---|---|
| Link Matrix | Builds links between each ability and related interfaces. And defines the cost of each ability. |
| Offset | Records the offset between local coordinate systems and global coordinate system. |

Fischer distinguishes the operations of object in MFSs into two different type, namely transportation and manipulation [6]. Transportation is a pure linear transportation on a conveyor, only position of an object can be changed. Manipulation is a general term including other possible operations, which can be done to the objects in MFSs, such as changing the orientation of a box or filling pellets into a bottle like the filling stations on HBFG. In this thesis, only transportation is taken into consideration based on the Logistic part of HBFG.

No matter a transportation or a manipulation ability, its operating range is limited by certain boundaries, normally interfaces of a module. The Link Matrix gives the operating boundaries to every transportation ability. Meanwhile, the cost of each transportation is also defined, which is a new element inside the iSiKon MAS concept in the scope of this thesis. After that the description of a transportation ability is completed and ready to be exported for the use of constructing system database in Step 2.2 of registration.

In step 2.3, the newly collected information will be sorted and added into existing system matrixes, where all raw module descriptions are stored. Finally, after the storage of NetID into DF, the registration is finished. Coordinator has gathered all necessary information to reconfigure system description. Then it steps into the Step 3 in Figure 11 to integrate these raw data into system routing table.

The central data base provided raw data to AMS. With the inserted algorithms, the new configuration can be established step by step, shown in Figure 13. In the scope of this thesis, the manipulation ability of agents is neglected because only logistics part of HBFG are considered. Therefore, the step 3.4 and 3.6 are depicted for the completeness of current MAS but commented out from program when adapting MAS to HBFG.

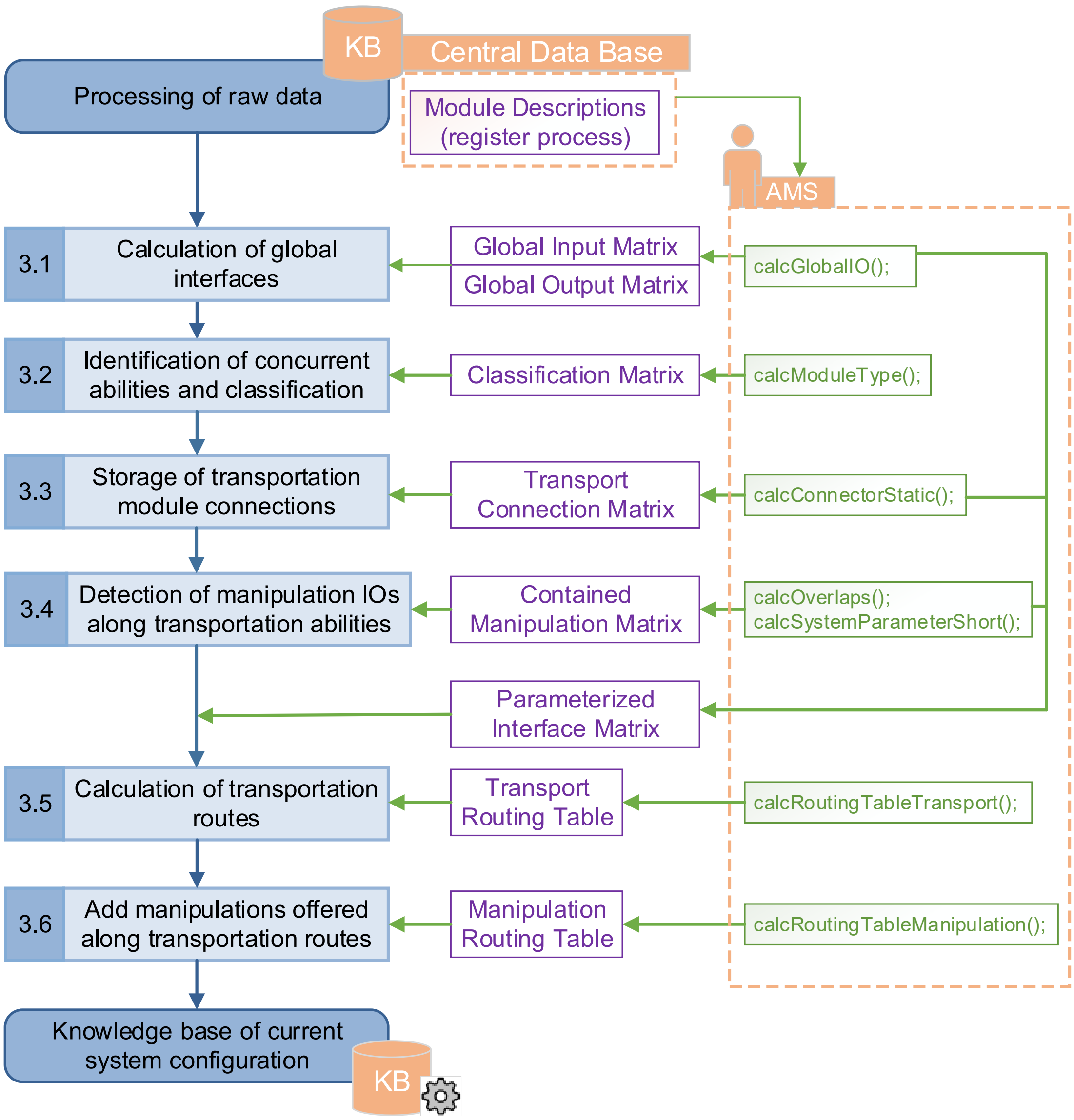



Figure 13 Processing of Raw Data

Transport Routing Table is an ultimate summary of a steadily running MFS from domain of this thesis, which concludes all information about connections between modules. Based on this table, MAS is able to search potential routes and calculate corresponding total cost for each new order. The Routing Table of HBFG after initialization is recorded in Appendix B.

After all steps in Figure 13 are accomplished, the configuration work of MAS is completed, which also means the first information stream about uploading module description is used up. From now on, the system will keep in steady state and ready to execute new order assignment, Step 5 in Figure 11.

In current version, new transportation order should be set manually from user computer, which will be replaced by electrical signals from the light barrier belonging to global input interfaces in subsequent implementation. Once the coordinator receives an order signal, the steps in Figure 14 are carried out.

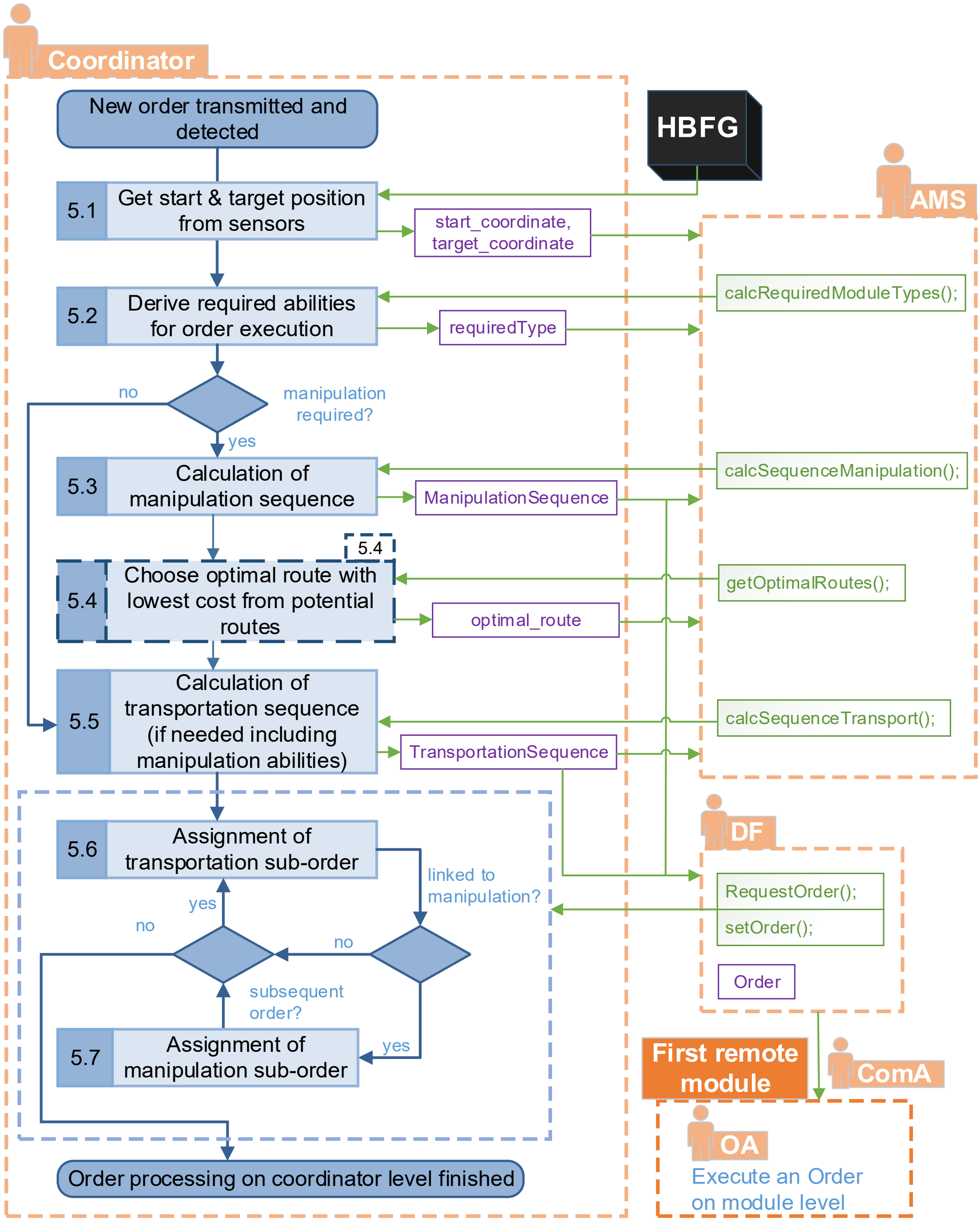



Figure 14 Order Assignment

The active coordinator, cooperating with whole central framework, firstly reads simple order command, only contains information about global I/O. And then scans through Transportation Routing Table to find suitable connections to build several potential routes for this order. Subsequently, the optimal routes can be decided with related cost of each ability. At the end, the transportation sequences of optimal route will be encapsulated in several sub-orders and sent to Order Agent on module level. The measurement standard of cost and the selection criteria of optimal route are clarified in section 5.4.1

Up to now, all procedures on Framework level are depicted and explained. The rest tasks are performed by local agent system on module level.

## 5.2 Agent System on Module Level

After unified order assignment done by coordinator, each module follows the received sub-order to drive related conveyors and switches within its compass and monitors order executing state by collecting feedback signals from HBFG. According to Figure 10, there are five main function blocks establishing the agent system on module level, which are defined in Table 4.

Table 4 Components of Agent System on Module level

| **Name of FB.** | **Main Tasks** |
|---|---|
| Order Agent (OA) | Receive or generate sub-orders following Transportation Sequences via ADS-communication. |
| System Agent (SA) | Tabulate matrixes of local module description and build connection to transmit order information between OA and CA. |
| Control Agent (CA) | Interact with Logistics Function to control MFS directly;<br>Transform signals from plant into readable order information and upload them to higher level of MAS. |
| Logistics Info | Provide a platform for MFS designer to input original module description. (Input description of each module of HBFG are listed in Appendix.) |
| Logistics Function | Control motors to perform assigned transportation works;<br>Receive signals from sensors and hand over to MAS. |

The information stream to upload module description originates from FB. Logistic Info, and flows upstream successively via CA, SA to the top of module level. After sorted by SA, the mature local module description is ready to be accessed by the framework with the ADS-communication provided by ComA. Functioning together with docking methods in framework part mentioned in previous chapter, Agent system successfully transforms module description from a human readable style into a wrapped database for MAS.

The other main operation happening on module level is about order execution. Order Agent takes over the sub-orders generated by framework. The flowchart of executing an Order on module level is demonstrated in Figure 15.

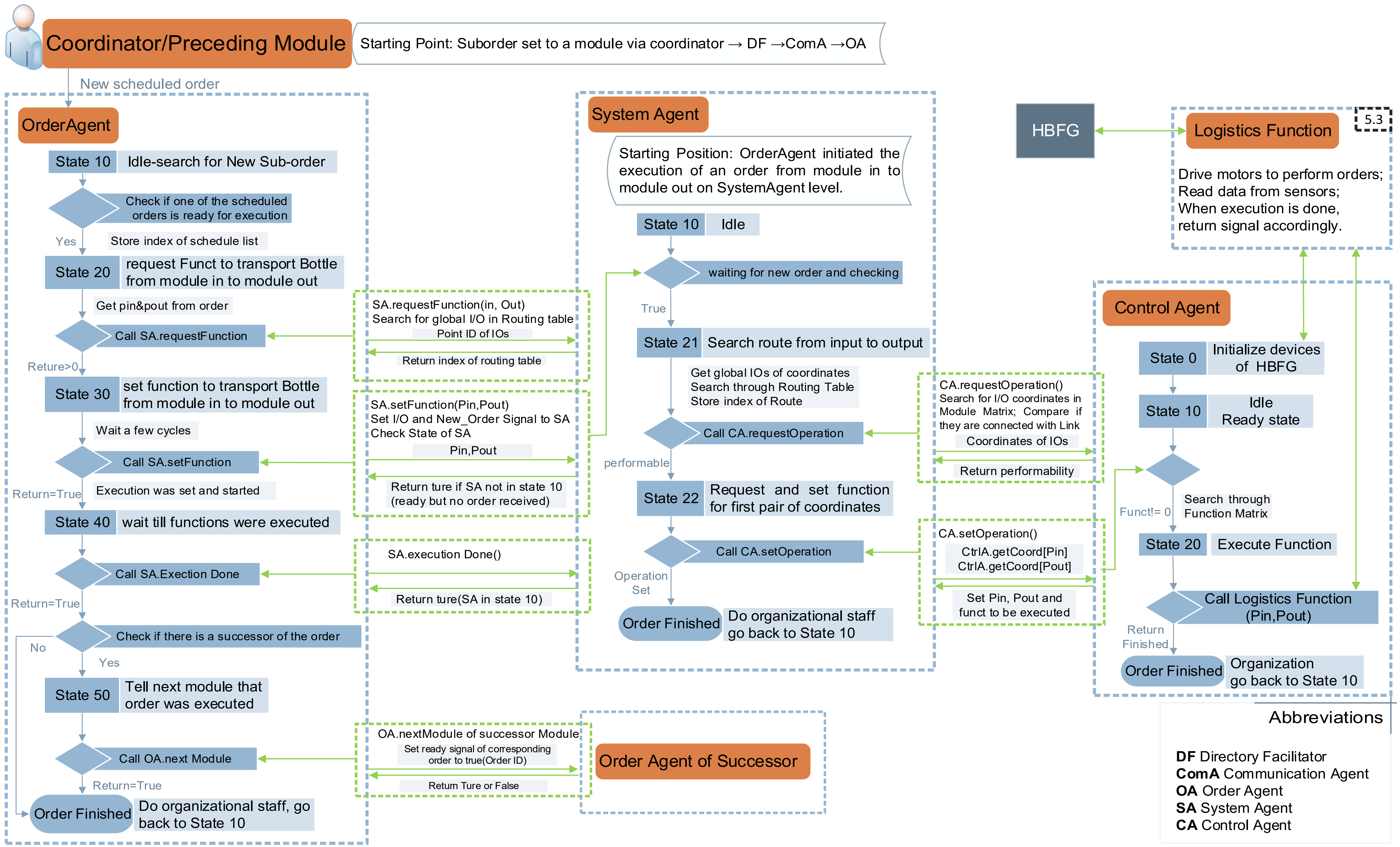


Figure 15 Order Execution on Module Level (enlarged and summarized the concepts in Regulin [2], Dengler [4] and Fischer [6])

Once the light barrier of output interface detects that the bottle is exported, this signal will be reported to agent system via Logistics Function. And accordingly, this sub-order will be sent to successor module and removed from the local order list in Order Agent. And all agents and functions in module level step back to ready state waiting for new assignment from central framework. From the view of global MAS, only until a bottle is picked up by robot arm at global output position, this overall order can be deleted from global order list.

By the end of this section, all tasks of MAS for controlling purpose is introduced in detail by tables and flowcharts. This structure is universally adaptable for most logistics part of MFSs, which is responsible for transporting object from source to sink.

## 5.3 Logistic Function to Control HBFG

The controlling program in hardware orientation, namely Logistic Function, is presented in this section. Logistic Function should be specifically designed and programmed for each module before it is added into MFS, because each module has its own characteristics, such as geometry and transportation direction, to carry out different designed tasks. The other reason is that, various devices from different suppliers are assembled in a typical configuration, which should be controlled by identical MAS in the same programming environment.

The Logistic Function used for implementing MAS is revised and developed based on the existing central program to control HBFG in AIS project “MyJoghurt” [32], where the whole HBFG is controlled via a single PLC by an integrated program, which is similar to traditional MFSs lack of flexibility and reconfigurability.

There are four independent modules defined according to module dividing plan. Thus, there are four corresponding Logistic Function named after each module. For a clearer explanation, all flowcharts below are based on the Logistics Function of module 1.

When the whole plant is powered on and the MAS is successfully configured on PLC, the initialization algorithm in Logistic Function will be activated to reset variables of all motors. Especially for switches, they should calibrate itself to align with conveyors, which prevent bottles getting stuck at joint point between switch and import conveyor.

Logistics Function only need to get the information of I/O position, which is also the final result calculated by MAS. For example, if the transportation direction is positive, the ID of input interface for module 1 is always 111. There are two output interfaces 121 and 131 available in this case. With the decision-making ability, MAS will choose a better route which contains one of the interfaces above. And Control Agent, which locates at the bottom of agent system, writes relative variables in Logistic Function to drive the plant. The flowchart of Logistic Function is depicted below.

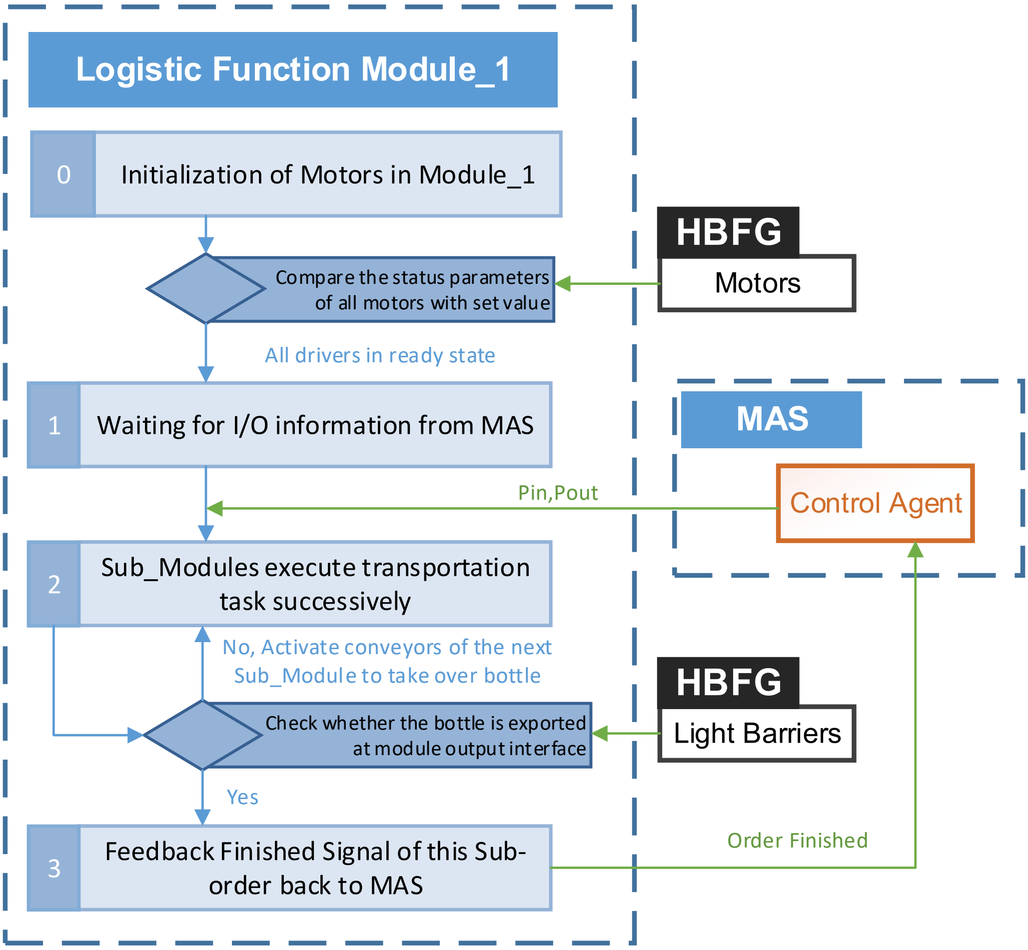



Figure 16 Logistic Function

For controlling HBFG in a more flexible way or providing more possibilities to continually development, each module is deeply divided into several sub-modules. Every switch or all conveyors between two adjacent switches is regarded as one sub-module. Motors on the same sub-module should run and stop simultaneously. According to the position of bottle on module level, sub-modules are activated one by one following the command of Logistic Function. The sub-modules of module 1 is shown below in Figure 17.

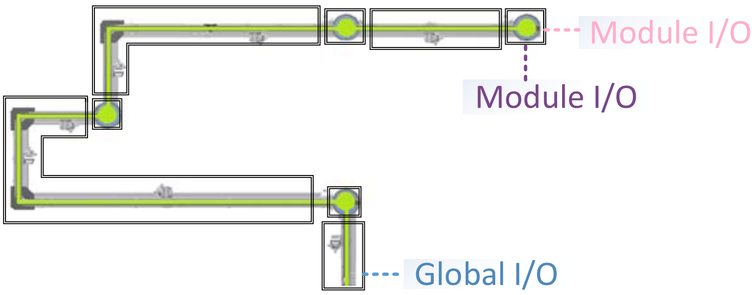



Figure 17 Sub-modules of Module 1

## 5.4 Enlargements of MAS based on Adaption to HBFG

In the scope of this thesis, current MAS of iSiKon is enlarged with some new concept to fit the characteristics of HBFG and make it more universally adaptable. The enlargements are introduced mainly in three following sections, cost of routes, bi-directional transportation and buffer module.

### 5.4.1 Decision-making Ability Aiming at Route Alternatives

In order to deal with increasingly complex MFSs with various routes between the same resource of transportation object and target position, MAS should possess the ability to decide which route is optimal under certain criterions, which is introduced in section 0. Normally, the cost of a typical route in MFS is a comprehensive assessment considering many influence factors, such as transportation distance, time consumption, energy dissipation and incidental labor cost. According to different MFS, there are different decision preferences of MAS, which should be designed by production engineers in the early stage.

The complexity of HBFG provides the possibility of route alternatives, which makes it more likely to a real production line. Thus, the corresponding decision-making section is added into agent knowledge base and evaluated in this term thesis. On the other hand, considering the limitation of HBFG, relatively small scale compared to real industrial plant, only total length of a module is treated as influence factor. However, for the purpose of inserting concept MFCA into MAS, this prerequisite is enough. Because only little effort is required for adapting to other MFSs with similar approach by increasing number of factors and updating the algorithm to calculate the cost.

The only rule to choose optimal route on HBFG is that, the shorter transportation distance a route has, the lower cost is accounted, and also the higher preference MAS has for this route. Every transportation ability of a module represents a potential component to build up a route connecting Global I/O. Therefore, a structure variable about the distance of each ability is defined and stored in Link Matrix as a part of module description, which will also be uploaded to central database in framework of MAS during registration process. A concise flowchart about the procedures of cost accounting and route deciding in current version of MAS is shown in Figure 18, which is the sub-processes of Step 5.4 in Figure 14. The final cost of each potential routes on HBFG are listed in Transportation Description in Appendix A.

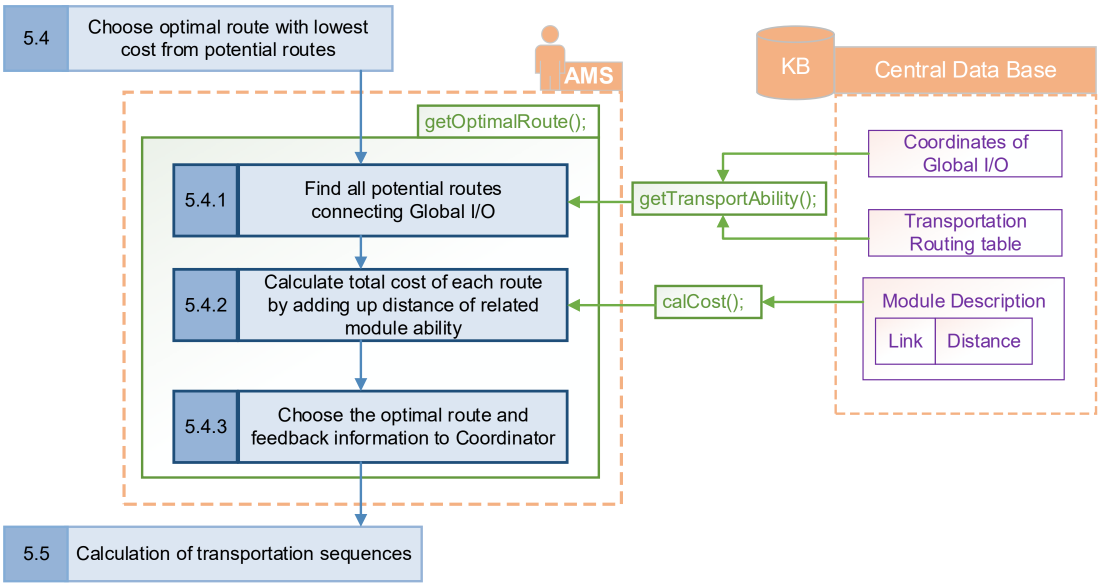



Figure 18 Decision of Optimal Route on Coordinator Level

### 5.4.2 Enlargement of Transportation Ability in Bi-direction

Increasing the utilization rate of single conveyor can highly improve the transporting efficiency of MFSs. Therefore, a common solution is that, the control system assigns bi-directional transportation order to one conveyor subsequently. This scenario requires conveyors toggle moving directions quickly. And from the view of MAS, Bi-directional transportation should be supported by a method to find available route between any two interfaces.

In order to achieve this target, one identical module interface in space are teared apart into two different interfaces, which share the same global coordinates, but own different value of direction, in or out. For example, if there is an order requiring transporting the bottle from module 2 to module 1, agent system will firstly check whether an overlap is existing between them. If yes, that means there are totally four interfaces with same coordinates at this joint, because each module owns two of them. And the interface of module 2 with direction value “out” and the interface of module 1 with direction value “in” will be activated in pairs and then send “OK” signal back, which functions as an admission to allow MAS to use this connection in route searching.

Although this bi-direction setting has the advantage in further strengthening the flexibility, some conflicts may arise, such as two bottles want to step into same module simultaneously but in opposite directions, which could be solved after the import of buffer module in further development. Another problem arisen is that the change of transport direction consumes energy and time, because the conveyor driver must slow down and restart in another direction. Although this kind of waste is not critical at HBFG, it is obvious especially to large scale of conveyors in real production.

### 5.4.3 Enlargement of Occupancy

In this section, a new concept named occupancy of a module is introduced to solve a shortcoming in current version of MAS.

In the real production, numerous objects are transported on MFSs at the same time, which indicates that each module in the system should own the capability to deal with multiple orders simultaneously, rather than execute order one by one. According to Figure 15, execution order of a module in a transportation route is activated by coordinator or by the previous module. Once a new sub-order comes in, the Control Agent will switch and keep in the busy state (state 20) until the finished signal is received. This program structure limits the number of order to only one.

The concept of occupancy is the first change to release this kind of limitation. There should be a parameter, which is able to record number of orders currently executed on the module and influence the policy made by MAS, namely occupancy. Cooperating with signals from light barriers on plant, occupancy can offer the central coordinator a clear scope of all running modules in MFS. Every time a new sub-order assigned to Order Agent with import of a new bottle, occupancy of this module plus one, if exporting, minus one. The central database reads occupancy from Control Agent of each module in real time. If the occupancy of a module is overflowing the limit value, coordinator will stop allocating new order to this module, by adding an extra influence factor into cost accounting algorithm when choosing optimal route for a new order. Considering a specific case that all modules are overflowing, the necessity of buffer module is now reflected. How to modify occupancy of a module during runtime in a reasonable way, how much does the occupancy influence cost accounting and how to deal with overflow situation. The processing mechanism aiming to above questions should be specifically designed for different types of module in real MFS and encoded into module knowledge base in advance.

The information stream of occupancy originates from Logistic Function, counting bottles between first and last light barriers. Then flows through Control Agent via ADS-communication and finally get to active coordinator to influence the next decision to allocate order. This kind of close-loop control method provides MFS with a higher stability and robustness. Considering the characteristics of HBFG, there is a limitation of multi-order assignment, the transportation on one module of more than one transportation object at the same time is only possible, if they share the same transportation direction, which is illustrated in Figure 19.

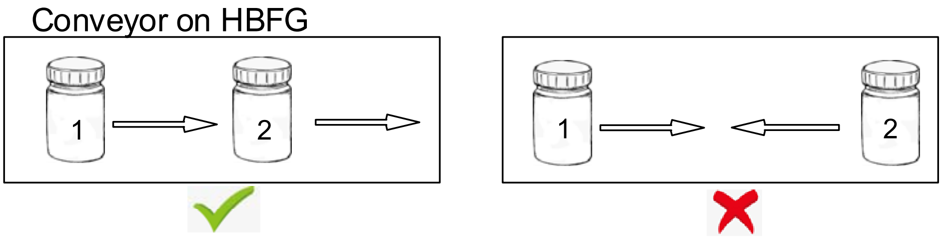



Figure 19 Limitation of Transportation Ability on HBFG

In the scope of this thesis, three enlargements are added into MAS during adaptation, respectively focusing on three common situations in real MFS, namely route alternatives, bi-directional transportation and synchronous execution of multi-orders.

# 6 Evaluation of MAS adapted to HBFG

In this chapter, a comprehensive evaluation about the performance and strain capacity of MAS facing four scenarios in section 4.1 is presented. Subsequently, the fulfillment of requirements and drawbacks of current MAS is also stated.

## 6.1 Evaluation of Different Scenarios

**Initialization of system（Scenario 1）**

After uploading corresponding agent program to each PLC and connecting the network between PLCs and HBFG, the whole system is ready to run. The coordinator section in each module will scan through the NetID list which is manually preset in the program. This work should be replaced by the Serve Tool [4] in future development to circularly check the existence of all PLCs in network and continually update NetID list. According to the list, every module starts to get indices of aptitude from the others and the electoral mechanism is activated. Module 3 is chosen as coordinator after negotiation and steps further to execute registration of whole plant. And other two modules keep pinging active coordinator to check its existence, upload module descriptions to Module 3 and wait for the sub-order assignment from central coordinator. After the initialization, a new order is set manually, coordinator will calculate the optimal route, which is transporting the bottle from Module 1, via Module 2, and finally exported at global output point in Module 3, because this route has shortest distance between global I/O under current configuration. Each agent system on module level follows the order assignment to control the motors of HBFG to transport bottle subsequently.

**Addition of a new module into system （Scenario 2）**

For this scenarios, Module 1 and Module 3 are initialized firstly, which is already enough to connect Global I/O with a route, although this route is not optimal with a longer distance without the participance of Module 2.

Then Module 2 is plugged into system and at the same time the NetID of Module 2 should be added into list manually, which also could be done by Server Tool later. Then the active coordinator will do registration for Module 2 by merging module description into central data base. From now on, the optimal route will be rescheduled, where the transportation abilities of module 2 are taken into consideration. That means the bottles imported after the addition of Module 2 will be transported from Module 1, via Module 2 and exported from system by the last conveyor of Module 3.

**Removal of a module from system（Scenario 3）**

As the opposite scenario to addition of new module, the procedures of evaluation are also inversed. Initialize the system with all three modules, and then remove Module 2 to check whether MAS could succeed in choosing other route to make up the absence of module 2.

**Removal of the module with active coordinator from system（Scenario 4）**

For this special scenario, Module 2 should be assigned as active coordinator to ensure when it is removed, the breakthrough between Global I/O is still existing. In this scenario, the negotiation between remained agents to selected new coordinator is key point to be evaluated. Module 1 and Module 3 aware the removal of Module 2 by the pinging method, once Module 2 is detached from network. And via Communication Agent between Module1 and Module 3, they exchange the indices of aptitude, which is regard as negotiation process, and then Module 3 becomes the coordinator with the highest suitability index. Under the direct of new coordinator, new system data base can be generated without any interaction with removed Module 2, and MAS go back into ready state for new order.

## 6.2 Fulfillment of Requirement and Discussion of Drawbacks

The real performances of HBFG during evaluation tests of all four scenarios is the same as expected situation, which is all foreseeable theoretically based on the software program described above. The evaluation verifies that current MAS possesses enough flexibility when controlling relative complex MFS like HBFG. The conformances to each requirement mentioned in 4.1 is listed in Table 5 below.

Table 5 Fulfillment of defined Requirements

| ***Requirements*** | ***Implemented in MAS*** | ***Successfully adapted to HBFG*** |
|---|---|---|
| Steadily controlling of MFS when system configuration is changed | + | + |
| Decision-making ability to choose optimal route and following order assignment | + | + |
| Negotiation mechanism to elect coordinator | + | + |
| Bi-direction transportation | + | - |
| Execute multi orders simultaneously | + | - |

( + fulfilled, - not fulfilled )

In following part, different drawbacks of the presented concept and its enlargements such as the lack of influence factors in cost accounting, manually resetting list of NetID, unaware transportation done by HBFG, etc.

In the scope of this thesis, only length of conveyors is taken into consideration when doing cost accounting. However, in real MFS, the cost of a specific module is normally influenced by manifold factors, such as operating duration, energy consumption, labor allocation. Meanwhile, how to define the weight of each influence factor is other core point in early management stage.

Some human interactions are still needed during run time, which violates the original design principle of independency. Firstly, without the implementation of server tool, three modules should be started almost simultaneously to ensure they can aware the existence of each other for the first time. And the same interaction is necessary to update NetID list when a new module is added into system. Secondly, the signal to start route-calculating steps of coordinator, namely New Order Signal, is also manually imported.

In current control software of HBFG, Logistics Function does not fully follow the instructions from MAS. It has own ability to deal with the feedback signal from plant independently. That means the plant could execute some operation, which the control system does not request and is not aware of. This drawback requires reprogramming of Logistics Function by removing its knowledge to react to coming signals.

# 7 Conclusion and Future Work

In this term thesis, a multi-agent system, developed under framework of project iSiKon, is adapted to a model of real MFS, namely the logistics part of HBFG. The adaptation work is divided into three aspects broadly. The first step is to divide HBFG into reasonable modules, which meet requirements proposed by agent system itself and four set scenarios for evaluation purpose. And then based on existing program, several revisions and three enlargements are supplemented in software domain to make MAS more suitable for the characteristics of HBFG as well as real industrial MFS. The last step is to combine hardware devices with agent system via a controlling function block, namely Logistic Function, to achieve real time control.

The MAS is programmed in IEC 61131-3 in the environment of TwinCAT, and configurated on Beckhoff PLCs to control HBFG. The final version of MAS in the scope of this thesis can be taken apart into three main parts, central framework of MAS, local agents on module level and controlling functions connected with hardware equipment, all of them are depicted step by step in flowcharts for a clear presentation.

After fully implementation of MAS on HBFG and following evaluation process, current MAS is capable to deal with four preset scenarios and reflects advantages of flexibility and reconfigurability, which achieves the main target of this thesis.

In further research, more manipulation abilities should be added into MAS and evaluated with the help of rotor arm and filling stations on HBFG to make agent system more universally applicable. Moreover, a more decentralized agent system should be developed to fully exploit its benefit in solving large-scale distributed problem in MFS. Another withdraw in current MAS is that, the coordinator is not able to update the list including all NetIDs of existing modules in the system in real time, which can be solved by implementing the Server Tool developed by Dengler [4] into MAS. To make full use of the functionalities of HBFG, buffer module can be taken into consideration to achieve more practical scenarios in real production.

# Reference

[1] M.Hermann, T.Pentek, B.Otto, "Design Principles for Industrie 4.0 Scenarios," in *49th Hawaii International Conference on System Sciences (HICSS)*, Koloa, HI, USA, May 2016.

[2] D.Regulin, D.Schütz, T.Aicher, B.Vogel-Heuser, "Model based design of knowledge bases in multi agent systems for enabling automatic reconguration capabilities of material flow modules," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2016.

[3] T.Aicher, D.Regulin, D.Schütz, C.Lieberoth-Leden, M.Spindler, W.Günthner and B.Vogel-Heuser, "Increasing flexibility of modular automated material flow systems: A meta model architecture," *IFAC-PapersOnLine,* Vols. vol. 49, no. 12, p. 1543 1548, 2016.

[4] S.Dengler, "Entwicklung eines Konzepts zur dynamischen Konfiguration der Kommunikationsschnittstellen von Speicherprogrammierbaren Steuerungen," Term Paper, Technical University of Munich, 2017.

[5] J.Fischer, M.Marcos, T.Aicher and B.Vogel-Heuser, "Agentenbasierte Steuerung von Logistiksystemen, Erhöhung der Rekonfigurierbarkeit in Materialflusssystemen," *atp-edition,* vol. 59, no. 09, pp. 16-26, 2017.

[6] J.Fischer, Agent-based Automation Systems for Material Flow Systems That Enable an Easy Re-configuration of Material Handling Modules, Master Thesis, Technical University of Munich, 2017.

[7] Institute of Automation and Information Systems, "Hybrid Process Model - Processes from Manufacturing and Process Technology," TUM Department of Mechanical Engineering, [Online]. Available: http://www.ais.mw.tum.de/en/research/equipment/hybrid-process-model/. [Accessed 25 Apirl 2018].

[8] Toyota Motor Corporation, The Toyota Production System – Leaner manufacturing for a greener planet, Tokyo: TMC, Public Affairs Division, 1998.

[9] P.James, M.Jones, T.Daniel, "Lean Thinking," *Free Press,* p. 352, 2003.

[10] W.Stevenson, Operations Management (Operations and Decision Sciences), 2000.

[11] R.Binder, F.Claudia, From material flow analysis to material flow management,

2006.

[12] EtherCat Technology Group, "Features and benefits of EtherCAT," 16 August 2011. [Online]. Available: http://www.controlengeurope.com/article/44201/Features-and-benefits-of-EtherCAT.aspx. [Accessed 27 Apirl 2018].

[13] International Organization for Standardization , "ISO 14051: Environmental management-Material flow cost accounting-General framework," *Standards catalogue,* p. 38, 2011.

[14] M.Schmidt, M. Nakajima, "Figure 1: Material Flow Cost Accounting as an Approach to Improve Resource Efficiency in Manufacturing Companies," 3 9 2013. [Online]. Available: http://www.mdpi.com/2079-9276/2/3/358. [Accessed 27 Apirl 2018].

[15] K.Ong, David M. Teller, M.Richard, Inventory control system, Viac Inc, 1989.

[16] R.John, "Material and inventory control system for a demand flow process". United Stated Patent US6594535B1, 15 07 2003.

[17] J.Ferber, Multi-Agent System: An Introduction to Distributed Artificial Intelligence, Addison Wesley Longman, 1999.

[18] Y.Shoham, "Agent oriented programming," *Artificial Intelligence,* vol. 60(1), pp. 51-92, 1993.

[19] M. Wooldridge, N. R. Jennings, "Intelligent Agents: Theory and Practice," *The Knowledge Engineering Review,* vol. 10, no. 02, pp. 115-152, 1995.

[20] G.Weiss, Multiagent Systems(2nd ed.), Cambridge: The MIT Press, 2013.

[21] R.Davis, R.G.Smith, "Negotiation as a Metaphor for Distributed Problem Solving," in *Readings in Distributed Artificial Intelligence*, San Mateo, Calif., Morgan Kaufmann, 1988, pp. 333-356.

[22] N.R.Jermings, M.Wooldridge, "Applications of Intelligent Agents," *Agent Technology: Foundations, Applications, and Markets,* pp. 13-28, 1988.

[23] Object Management Group, "XML Metadata Specification," [Online]. Available: http://www.omg.org/docs/formal/05-05-01.pdf. [Accessed 27 Apirl 2018].

[24] FIPA, "FIPA Agent Management Specification," 18 03 2004. [Online]. Available: http://www.fipa.org/specs/fipa00023/SC00023K.pdf. [Accessed 03 Apirl 2018].

[25] FIPA, "FIPA Agent Message Transport Service Specification," 6 12 2002. [Online]. Available: http://www.fipa.org/specs/fipa00067/index.html. [Accessed 03

Apirl 2018].

[26] Standards International Electrotechnical Commission, *IEC 61131-3:2013 Programmable controllers - Part 3: Programming languages,* 1993.

[27] International Conference on Automation Science and Engineering, "Flexible Agent-based Planning and Adaptation of Material Handling Systems," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015.

[28] M.Wilke, "Wandelbare automatisierte Materialflusssysteme für dynamische Produktionsstrukturen," PhD thesis, Technical University of Munich, 2006.

[29] R.Chisu, Kommunikations-und Steuerungsstrategien für das Internet der Dinge, Technical University of Munich: PhD Thesis, 2010.

[30] D.Daniel, "Blackboard Systems," *AI Expert.,* vol. 9, pp. 40-47, 1991.

[31] R. Priego, A. Armentia, E. Estévez and M. Marcos, "On Applying MDE for Generating Reconfigurable Automation Systems," in *IEEE International Conference on Industrial Informatics (INDIN)*, 2015.

[32] Institute of Automation and Information Systems, "MyJoghurt – involved in the Platform Industrie 4.0's Germany-wide online roadmap „Industrie 4.0“," Department of Mechanical Engineering of Technical University of Munich, [Online]. Available: http://i40d.ais.mw.tum.de/index/index/l/en_US. [Accessed 17 04 2018].

[33] German Reserach Foundation, "Gepris-iSiKon.," [Online]. Available: http:// gepris.dfg.de/ gepris/projekt/ 251665026? contrast=0. [Accessed 09 March 2017].

# List of Abbreviations

| | |
|---|---|
| **ACC** | Agent Communication Channel. |
| **ACL** | Agent Communication Language. |
| **ADS** | Automation Device Specification. |
| **AIS** | Institute of Automation and Information Systems. |
| **AMS** | Agent Management System. |
| **AP** | Agent Platform. |
| **DF** | Directory Facilitator. |
| **FB** | Function Block |
| **FIPA** | Foundation for Intelligent Physical Agents. |
| **HBFG** | Hybrid Process Model. |
| **IEC** | International Electrotechnical Commission. |
| **IEEE** | Institute of Electrical and Electronics Engineers. |
| **IMU** | the German Institute of Business Environment. |
| **IoT** | Internet of Things. |
| **MAS** | Multi-Agent System. |
| **MFCA** | Material flow cost accounting. |
| **MFS** | Material Flow System. |
| **MTP** | Message Transport Protocol. |
| **MTS** | Message Transport Service. |
| **PLC** | Programmable Logic Controller. |
| **TPS** | Toyota production system. |
| **TwinCAT** | The Windows Control and Automation Technology. |

# List of Figures

# List of Tables

# Appendix A Description of three modules of HBFG

This appendix includes detailed descriptions of all three modules based on module dividing plan. According to 4.2, each module description includes Interface Description, Transportation Description and Device Description.

## A1: Description of Module 1

Module 1 is controlled by Beckhoff PLC type CX2040.

### Interface Description

As depicted in Figure 20, module 1 has three interfaces and each interface owns two global ID to distinguish the transportation directions. And the origin of global coordinate system is highlighted, which is identical for all modules of HBFG.

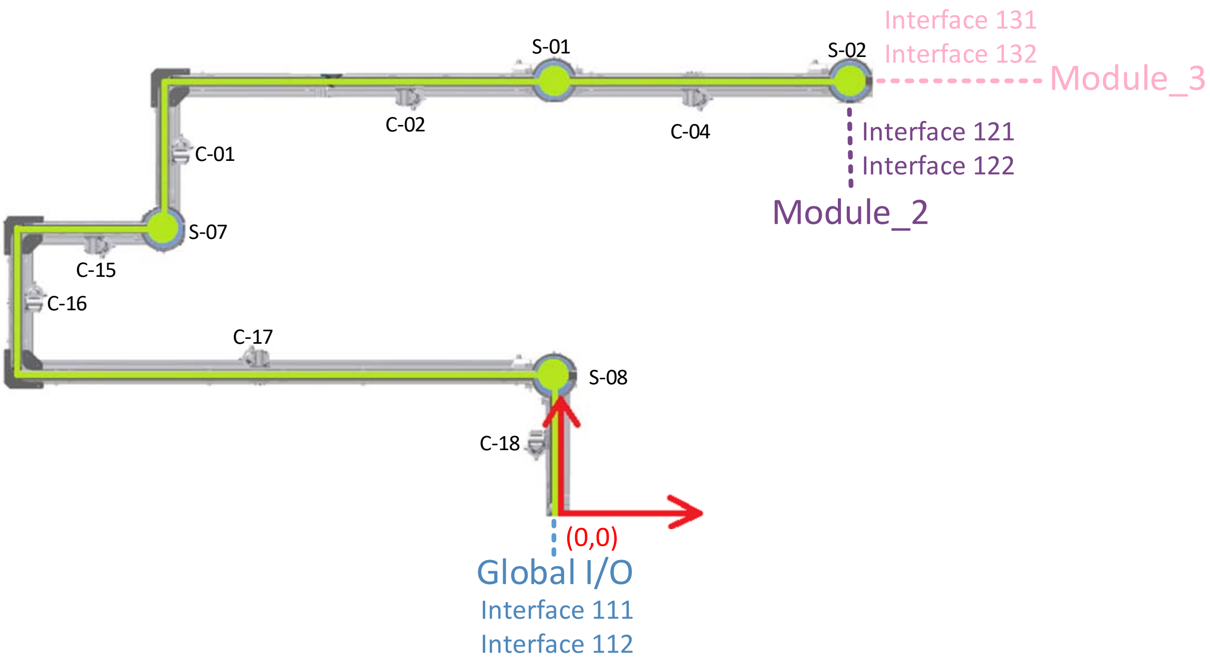



Figure 20 Interfaces of Module 1

Based on measurement, parameters to describe each interface are listed in Table 6.

Table 6 Interfaces of module 1 and global coordinates

| **Global ID** | 111 | 112 | 121 | 122 | 131 | 132 |
|---|---|---|---|---|---|---|
| **position.x1 /cm** | 0 | 0 | 107 | 107 | 121.5 | 121.5 |
| **position.y1 /cm** | 0 | 0 | 171.5 | 171.5 | 180.25 | 180.25 |
| **position.z1 /cm** | 0 | 0 | 0 | 0 | 0 | 0 |
| **position.x2 /cm** | -3 | -3 | 110 | 110 | 121.5 | 121.5 |
| **position.y2 /cm** | 0 | 0 | 171.5 | 171.5 | 177.25 | 177.25 |
| **position.z2 /cm** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Input** | True | False | False | True | False | True |
| **activated** | False | False | False | False | False | False |

### Transportation Description

Ability vector and Link vector jointly describe transportation abilities for each module. There are totally four transportation abilities for module 1, which are listed in Table 7.

Table 7 Transportation Ability and related Link Matrix (grey shaded) of Module 1

| **Global ID of Ability** | 1121 | 1131 | 1212 | 1312 |
|---|---|---|---|---|
| **ID of Interface_In** | 111 | 111 | 122 | 132 |
| **ID of Interface_Out** | 121 | 131 | 112 | 112 |
| **Ability_Cost** | 12 | 12 | 12 | 12 |
| **Direction_x_fwd** | 1 | 1 | 0 | 0 |
| **Direction_y_fwd** | 1 | 1 | 0 | 0 |
| **Direction_z_fwd** | 0 | 0 | 0 | 0 |
| **Direction_x_bwd** | 0 | 0 | 1 | 1 |
| **Direction_y_bwd** | 0 | 0 | 1 | 1 |
| **Direction_z_bwd** | 0 | 0 | 0 | 0 |

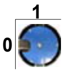
Link vector connects each ability with related interfaces and also contain the cost of this transportation ability. One unit of cost is about 44 cm, which is the length of a shortest conveyor, such as Conveyor-18. Therefore, Ability 1121 has a cost 12, which means the distance between output and input point is approximately 528 cm. And the direction parameters indicate in which direction the bottle is being transported.

### Device Description

Device description table is important in scope of Logistic Function. Only by changing the value according to this table, the motors of HBFG can execute correct action to transport bottles. The meaning of each term in this table is detailed introduced in chapter 4.2.

The width of every conveyor on HBGF is equal to 3 cm, and the diameter of switches is 14.5 cm. And other individual control parameters of each device defined in Logistic Function are list in Table 8 below.

Table 8 Device Discerption of Module 1

| **Device ID** | C-18 | S-08 | C-17 | C-16 | C-15 | S-07 | C-01 | C-02 | S-01 | C-04 | S-02 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Box ID** | 24 | 23 | 22 | Klemme 16 | | 15 | 6 | 7 | 8 | 10 | 11 |
| **Length of Conveyor/cm** | 44 | / | 190 | 52 | 46 | / | 46 | 135 | / | 95.5 | / |
| **Default direction** | + | / | - | - | - | / | - | - | / | - | / |
| **Orientation of Switch** | / | 2  1 | / | / | / | 1 0  | / | / | 2  0 | / | 2  0 1 |

### *A2: Description of Module 2*

Module 2 is controlled by Beckhoff PLC type CX9020.

#### Interface Description

As depicted in Figure 21, module 2 has two interfaces and each interface owns two global ID to distinguish the transportation directions.

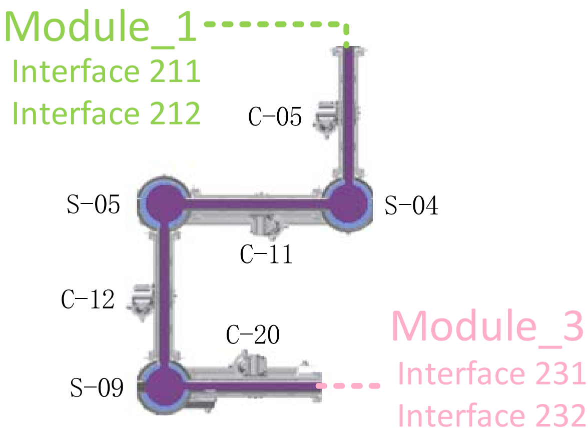



Figure 21 Interfaces of Module 2

Based on measurement, all parameters to describe each interface of module 2 are listed in Table 9.

Table 9 Interfaces of module 2 and global coordinates

| **Global ID** | 211 | 212 | 231 | 232 |
|---|---|---|---|---|
| **position.x1 /cm** | 107 | 107 | 101.25 | 101.25 |
| **position.y1 /cm** | 171.5 | 171.5 | 52.75 | 52.75 |
| **position.z1 /cm** | 0 | 0 | 0 | 0 |
| **position.x2 /cm** | 110 | 110 | 101.25 | 101.25 |
| **position.y2 /cm** | 171.5 | 171.5 | 49.75 | 49.75 |
| **position.z2 /cm** | 0 | 0 | 0 | 0 |
| **Input** | True | False | False | True |
| **activated** | False | False | False | False |

**Transportation Description**

Ability vector and Link vector jointly describe transportation abilities for each module. There are two transportation abilities for module 2, which are listed in Table 10.

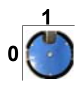
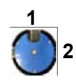
Table 10 Transportation Ability and related Link Matrix (grey shaded) of Module 2

| Global ID of Ability | 2131 | 2312 |
|---|---|---|
| ID of Interface_In | 211 | 232 |
| ID of Interface_Out | 231 | 212 |
| Ability_Cost | 4 | 4 |
| Direction_x_fwd | 1 | 0 |
| Direction_y_fwd | 0 | 1 |
| Direction_z_fwd | 0 | 0 |
| Direction_x_bwd | 0 | 1 |
| Direction_y_bwd | 1 | 0 |
| Direction_z_bwd | 0 | 0 |

**Device Description**

Control parameters of devices belonging to module 2, which are defined in Logistic Function, are listed in Table 11 below.

Table 11 Device Discerption of Module 2

| Device ID | C-05 | S-04 | C-11 | S-05 | C-12 | S-09 | C-20 |
|---|---|---|---|---|---|---|---|
| Box ID | 4 | 5 | 6 | 8 | 7 | 9 | 10 |
| Length of Conveyor/cm | 46.5 | / | 40.5 | / | 52 | / | 40.5 |
| Default direction | + | / | + | / | + | / | - |
| Orientation of Switch | / |  | / |  | / |  | / |

### *A3: Description of Module 3*

Module 3 is controlled by Beckhoff PLC type CX9020.

**Interface Description**

As depicted in Figure 22, module 3 has three interfaces and each interface owns two global ID to distinguish the transportation directions.

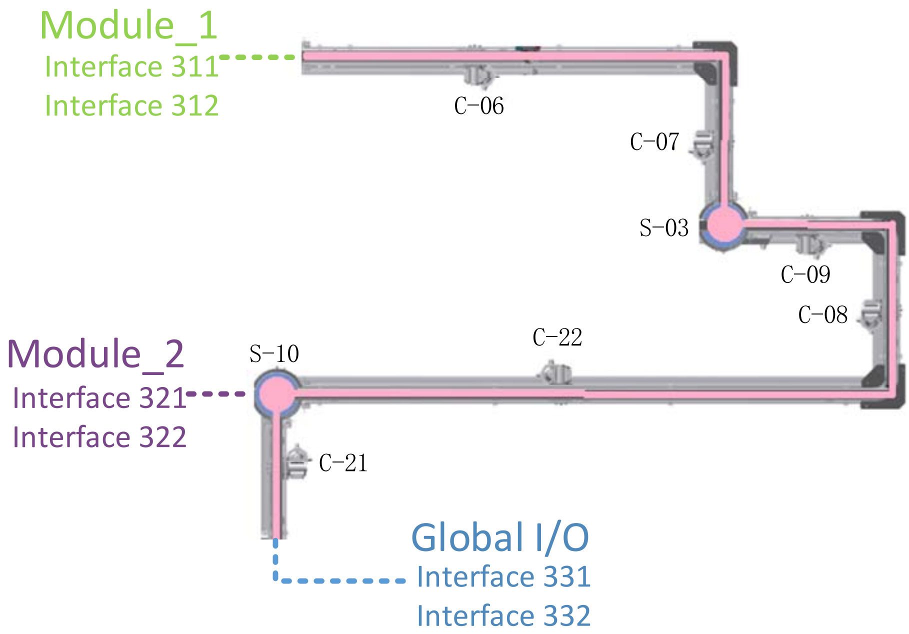



Figure 22 Interfaces of Module 3

Based on measurement, all parameters to describe each interface of module 3 are listed in Table 12.

Table 12 Interfaces of module 3 and global coordinates

| **Global ID** | 311 | 312 | 321 | 322 | 331 | 332 |
|---|---|---|---|---|---|---|
| **position.x1 /cm** | 121.5 | 121.5 | 101.25 | 101.25 | 107 | 107 |
| **position.y1 /cm** | 180.25 | 180.25 | 52.75 | 52.75 | 0 | 0 |
| **position.z1 /cm** | 0 | 0 | 0 | 0 | 0 | 0 |
| **position.x2 /cm** | 121.5 | 121.5 | 101.25 | 101.25 | 110 | 110 |
| **position.y2 /cm** | 177.25 | 177.25 | 49.75 | 49.75 | 0 | 0 |
| **position.z2 /cm** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Input** | True | False | True | False | False | True |
| **activated** | False | False | False | False | False | False |

**Transportation Description**

Ability vector and Link vector jointly describe transportation abilities for each module. There are two transportation abilities for module 3, which are listed in Table 13.

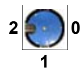
Table 13 Transportation Ability and related Link Matrix (grey shaded) of Module 3

| **Global ID of Ability** | 3131 | 3231 | 3312 | 3322 |
|---|---|---|---|---|
| **ID of Interface_In** | 311 | 321 | 332 | 332 |
| **ID of Interface_Out** | 331 | 331 | 312 | 322 |
| **Ability_Cost** | 10 | 1 | 10 | 1 |
| **Direction_x_fwd** | 0 | 1 | 1 | 0 |
| **Direction_y_fwd** | 0 | 0 | 1 | 1 |
| **Direction_z_fwd** | 0 | 0 | 0 | 0 |
| **Direction_x_bwd** | 1 | 0 | 0 | 1 |
| **Direction_y_bwd** | 1 | 1 | 0 | 0 |
| **Direction_z_bwd** | 0 | 0 | 0 | 0 |

**Device Description**

Control parameters of devices belonging to module 3, which are defined in Logistic Function, are listed in Table 14 below.

Table 14 Device Discerption of Module 3

| **Device ID** | **C-06** | **C-07** | **S-03** | **C-09** | **C-08** | **C-22** | **S-10** | **C-21** |
|---|---|---|---|---|---|---|---|---|
| **Box ID** | 4 | 5 | 12 | Klemme 10 | | 15 | 13 | 14 |
| **Length of Conveyor/cm** | 135 | 46.5 | / | 46 | 52 | 190.5 | / | 44 |
| **Default direction** | - | - | / | - | - | - | / | + |
| **Orientation of Switch** | / | / |   | / | / | / |   | / |

# Appendix B Transportation Routing Table

In this appendix, the transportation routing table of HBFG is presented, which is calculated by AMS after coordinator finishes registration for all three modules.
MAS finds optimal transportation route for each new order based on this table.

Table 15 Transportation Routing Table of HBFG under Control of MAS

| Route No. | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Interface ID of Global In | 111 | 111 | 332 | 332 |
| Interface ID of Global out | 331 | 331 | 112 | 112 |
| Transport Direction | 1 | 1 | 2 | 2 |
| Ability ID of Module 1 | 1121 | 1131 | 3322 | 3312 |
| Ability ID of Module 2 | 2131 | / | 2312 | / |
| Ability ID of Module 3 | 3231 | 3131 | 1212 | 1312 |
| Total Cost | 17 | 22 | 17 | 22 |

Transport direction 1 means transport bottle from module 1 to module 3. And transport direction 2 is opposite.

# Appendix C Network Connection Table

This section contains a list of all line connections and their sockets to fulfill the structural requirements of module dividing plan on HBFG. In this table, “standard mode” is designed for a central controlling program, and “Agent mode” supports the decentralized MAS.

This table is edited by E.Trunzer and M.Menara.

Table 16 Network Connection Table

| **Standard Mode** | | | **Agent Mode** | | |
|---|---|---|---|---|---|
| **Input** | **Socket** | **Output** | **Input** | **Socket** | **Output** |
| P1A_1S | P1A | P1A_2S | **Module 1** | | |
| P1B_1S | P2A | P1B_2S | P1A_1S | P1A | P1A_2A |
| P1C_1S | P1C | P1C_2S | P1B_1A | P2A | P1B_2S |
| P2A_1S | P2A | P2A_2S | **Module 2** | | |
| P2B_1S | P2B | P2B_2S | P1A_1A | P2X_1 | P1A_2S |
| P3A_1S | P3A | P3A_2S | P2A_1S | P2A | P2A_2A |
| P3B_1S | P3B | P3B_2S | P2B_1A | P2B | P2B_2S |
| | | | P1B_1S | P2X_2 | P1B_2A |
| | | | P1C_1A | P1C | P1C_2S |
| | | | **Module 3** | | |
| | | | P2A_1A | P3X | P2A_2S |
| | | | P3A_1S | P3A | P3A_2A |
| | | | P3B_1A | P3B | P3B_2S |
| | | | **Module 4** | | |
| | | | P3A_1A | P4X | P3A_2S |

# Appendix D List of Files on CD

The files on the CD are as follows:

- pdf-file of this document
- Microsoft Word sources of this document
- folder *visio files of figures* containing editable Microsoft Visio files of the figures used within this term thesis
- folder *references* containing the most related documents of this work
- folder *centralized control of HBFG* containing the centralized control program of HBFG, which functions as a basis of Logistics Function in this thesis
- folder *multi agent system TwinCAT3* containing multi-agent system program (TwinCAT version 3.1.4020) of the three modules and a short tutorial for running them.