Dissertation

WORKING SCENARIOS OF HYBRID SELF-ORGANIZING ASSEMBLY SYSTEM

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften (Dr. techn.), eingereicht an der TU Wien, Fakultät für Maschinenwesen und Betriebswissenschaften von

Mag. ing. mech. Damir HASKOVIC

Mat.Nr: 1229551

Arnethgasse 31/28, 1160, Wien, Österreich

unter der Leitung von

Univ.Prof.Dipl.-Ing.Dr.techn.Dr.mult.h.c. Branko Katalinic

Institut für Fertigungstechnik und Hochlaserleitungstechnik, E311

begutachtet von

Valentin Pryanichnikov
Keldysh Inst. of Appl. Mathematics
Russian Academy of Sciences, KIAM
Int.lab."Sensorika" Miusskay sq.4,
Moscow 125047, Russia

Dragan Covic
Faculty of Mechanical Engineering
and Computing, University of Mostar
Matice hrvatske bb, Mostar 88000
Bosnia & Herzegovina

I confirm, that going to press of this thesis needs the confirmation of the examination committee.

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume. If text passages from sources are used literally, they are marked as such.

I confirm that this work is original and has not been submitted elsewhere for any examination, nor is it currently under consideration for a thesis elsewhere.

Vienna, September, 2018

Signature

Kurzfassung

Die vorliegende Arbeit beschäftigt sich mit der Untersuchung von Arbeitsszenarien und der Effizienz moderner Montagesysteme der nächsten Generation. Diese Systeme sind als *Bionic Assembly Systems* (BAS) bekannt. Sie basieren auf biologisch inspirierten Prinzipien der Selbstorganisation mit reduzierter Bedeutung der zentralen Steuerung, mit starker Vernetzung von Maschinen und Robotern und paralleler Verteilung von Prozessen.

BAS-Steuerungssysteme kombinieren zwei Steuerungsprinzipien: zentralisiert und selbstorganisiert. Das BAS-System ist der Fabriksteuerung untergeordnet. Maschinen und Roboter innerhalb des BAS-Systems agieren selbstorganisierend. In dieser Arbeit wird dieses Konzept hybride Kontrollstruktur genannt.

Im Gegensatz zu den klassischen, hochautomatisierten Systemen ermöglicht BAS dank Selbstorganisation die Integration von Arbeitern in den Arbeitsprozess.

Am Leitstand übernimmt der Systemoperator die Verantwortung für das Funktionieren des gesamten BAS-Systems. Er trifft die endgültigen Entscheidungen. Die BAS-Effizienz hängt stark von seiner Fähigkeit ab, qualitativ hochwertige Entscheidungen in der zur Verfügung stehenden Zeit zu treffen. Die Qualität der Entscheidungen und die Zeit, in der diese getroffen werden, sind wichtige Parameter.

Während des normalen Arbeitsmodus ist die Notwendigkeit von Systemoperatorentscheidungen relativ gering. Während Störungen und Störungen des Übergangsmodus ist die Notwendigkeit von Entscheidungen des Systemoperators hoch und entscheidend. Er muss Entscheidungen unter Stress und folgenden Einschränkungen treffen: große Datenmengen, unvollständige Informationen und zeitliche Begrenzung.

Um den Systemoperator bei seinen Entscheidungen zu unterstützen, wird ein Tool namens *Intelligent Adviser Module* (IAM) eingeführt. Die IAM-Vorschläge antworten auf die Frage: Was ist richtig hier und jetzt zu tun? Das IAM aktualisiert seine Vorschläge kontinuierlich. Sie sind für den Systemoperator beratend, aber nicht zwingend zu befolgen. IAM ist ein integraler Bestandteil des BAS-Steuerungssystems. Als Ergebnis dieser Integration sollte IAM in der Lage sein, zu lernen und die Qualität seiner Vorschläge im Laufe der Zeit zu verbessern. Solche Vorschläge sind ein Ergebnis der Fähigkeit des IAM, tatsächliche Systemzustände, vordefiniertes bereichsspezifisches Wissen, menschliches Fachwissen und systemspezifische Erfahrungen zu integrieren.

Die Ergebnisse zeigen, dass IAM insbesondere bei der Lösungssuche bei Konfliktsituationen, komplexen Situationen, Nichtstandard-Situationen und dem Ausfall von Maschinen hilfreich ist. Einen besonderen Beitrag leistet das IAM-Konzept bei der Unterstützung von weniger erfahrenen Systemoperatoren.

Die Verifizierung des IAM-Konzepts in einer industriellen Anwendung wurde realisiert. Die Ergebnisse zeigen, dass es mit hoher Effizienz funktioniert.

In dieser Arbeit wurde der Beitrag der IAM-Vorschläge zur Qualität der Entscheidungen und der benötigten Zeit zur Entscheidungsfindung vom Gesichtspunkt der Effizienz des Gesamtsystems untersucht.

Die Untersuchungen zeigen, dass das IAM-Konzept breiter anwendbar ist, vor allem bei Produktionssystemen mit hoher technischer Ähnlichkeit mit BAS. Es hat sich gezeigt, dass das IAM-Konzept eine vielversprechende Entwicklungsrichtung zur Effizienzsteigerung zukünftiger, moderner Montagesysteme ist.

Abstract

The research presented in this thesis focuses on the investigation of working scenarios and efficiency of next generation of modern assembly systems. These systems are known as Bionic Assembly System (BAS). It is based on biologically inspired principles of self-organisation, reduced centralized control, networking between units and natural parallel distribution of processes. BAS control system combines two principles: subordination from factory level to BAS control structure, and self-organization at the shop floor level. This concept is here called hybrid control structure.

BAS is a human centric system which promotes the integration of workers in the working process. Human tasks on the shop floor are performed by the shop floor operators. Human tasks in the control system are performed by the system operator. He makes the final decisions. BAS efficiency strongly depends on his ability to make high quality decisions and / or in shorter time.

During normal working mode, the need for system operator decisions is relatively low. During disturbances and transition mode, the need for system operator decisions is high and crucial. He must make decisions under stress and following restrictions: large amounts of data, incomplete information and time limitation.

To support the system operator during his decision making, a tool named Intelligent Adviser Module (IAM) is introduced. The IAM proposals answer to the question: What to do here and now? The IAM is continuously updating its proposals. They are not mandatory for the system operator. IAM is an integral part of the BAS control system. As a result of this integration, IAM should be able to learn and to improve the accuracy of its proposals over time. Such proposals are a result of the IAM's ability to incorporate actual system states, predefined domain specific knowledge, human expertise and system specific experiences.

The investigation of contribution of IAM proposals on the quality and time of decisions is made. The results show that the quality of decisions is higher, and / or time is shorter, especially during: solving of conflict situations, solving of complex situations, solving of non-standard situations, support of less experienced system operators.

Verification of IAM concept in an industrial application is realized. The results show that this is operational and functioning efficiently. Production systems with high technical similarity with BAS can increase their efficiency using the IAM concept. This represents a promising direction of development of future modern assembly systems.

Dedication

This dissertation is dedicated to *grandpa* Mate and *grandma* Marija – thank you for all the great memories and moments which I spent with you by the sea. I am very grateful to my family, especially my parents, because without their moral and financial support this dissertation would not have been possible. Furthermore, I wish to thank Emil and Maja including the rest of the family as well as all my friends who helped and supported me. Additionally, a big thanks goes to Dr. Dragan Manojlović and Predrag Popović for all the valuable advices and limitless help.

A very special thanks goes to my wife Anja for all of her support, love, care and attention. During the writing of this dissertation, the best thing in my life happened – I have meet you.

Posveta

Ova disertacija je posvećena *djedu* Mati i *baki* Mariji - hvala vam na svim lijepim uspomenama i trenutcima koje sam proveo s vama na moru. Veliko hvala upućujem svojoj obitelji, posebno mojim roditeljima, bez čije moralne i financijske podrške ova disertacija ne bi bila moguća. Nadalje, želim se zahvaliti Emilu i Maji te ostatku obitelji kao i svim mojim prijateljima koji su mi pomogli i podržali me. Također veliko hvala kumovima Dr. Draganu Manojloviću i Predragu Popoviću za sve dragocjene savjete i izuzetnu pomoć.

Posebnu zahvalnost upućujem svojoj supruzi Anji, za svu njenu podršku, ljubav, brigu i pažnju. Tokom pisanja ove disertacije, dogodilo mi se nešto najbolje u životu - upoznao sam tebe.

Acknowledgment

I would like to take this opportunity to acknowledge the invaluable support of a number of people, who helped me to complete this thesis.

First of all, this dissertation would not have been possible without the mentorship and encouragement from Univ. Prof. Dr.sc. Dr.mult.h.c. Prof.h.c. Branko Katalinic. I would like to express my deepest gratitude for his outstanding supervision, motivation and support. It was a pleasure and an honour to meet a world class scientist and an overall happy and positive person such as himself.

I would also like to acknowledge the help and encouragement from Professor Zoran Mrsa from the Technical University of Rijeka, for changing my course of life by recommending me to Prof. Katalinic.

Likewise, I would like to extend my gratitude to Professor Valentin Pryanichnikov and Professor Dragan Covic for their helpful and insightful reviews and comments regarding this dissertation.

Special thanks to my friends and colleagues Ilija Zec, Roberta Cvetkovska and Ilya Kukushkin for their pleasant company, encouragement and a lot of time spent working together, discussing and having fun in the IMS office as well as during DAAAM conferences.

Damir Haskovic

Contents

Chapter 1

Introduction

The growing impact of science and technology causes our world to constantly change. A perfect example of this is globalization and its impact on the production industry. It has introduced big changes through high competitiveness and a dynamic environment. There is an increased worldwide demand for new products which means that the companies need to expand to new markets.

As a result, the current product development has been affected with new challenges. These include shorter product lifetimes, shorter adoption times for new technology, increased number of product variations as well as increased product complexity. Additionally, assembly can be the most complex and expensive phase, because it heavily depends on the product.

To stay competitive, modern assembly systems need to respond to these challenges through adaptability, efficiency and robustness. They need to be able to quickly adjust their functionality and capacity according to market changes. Additionally, they need to be capable to assemble a large variety of products with high complexity within defined customer deadlines while, at the same time, reduce their production costs and increase their productivity by minimising standstills.

To realise such a system, there are various directions of development. One such direction is flexibility. It allows the system to quickly adjust its capacity and functionality in response to any changes. Another direction is self-organization. It allows to reduce the role of a centralised control system. It is based on networking and parallel distribution of tasks among the executing units on the shop floor. Another direction includes "*Intelligent systems*" which should improve their performance over time. These development directions are a result of Industry 4.0 where it is possible to observe the convergence of computer integration, networking, interconnectivity, sensors, technology etc.

However, at the same time, these modern assembly systems need to be human centric. That means they need to promote the integration of humans within the assembly process where the main decision maker is the system operator. The system efficiency depends on his limited ability to reach repetitive, high quality decisions in good time.

Good time means that the decision is made while the machines are working. This is not always possible due to time shortages which can cause stress for the system operator as well as his limited physical and cognitive capabilities.

Therefore, there is a need to develop a support system which will help the system operator to reach decisions with higher quality and / or in shorter time. Such a system needs to be able to learn, to propose decisions and to be user friendly. This is becoming possible thanks to already mentioned growing impact of science and technology or more precisely, information technologies and its field of artificial intelligence. In the frame of this work, the focus is a new type of modern assembly system – Bionic Assembly System.

1.1 Aim and objectives

Bionic Assembly System (BAS) represents a next generation of hybrid assembly systems. It is a part of a natural development within Industry 4.0. BAS is a complex production system. Here, the role of the system operator is crucial. He must be able to make decisions with high quality and / or in shorter time with fragmented and incomplete information about the actual system states and its components. BAS functionality and high work efficiency depend on this ability.

The main sources of information for the system operator are coming from: human communication, control system feedback, shop floor feedback. The main data stream is between the control system and the shop floor. Data stream from the control system to the shop floor are commands. Data stream from the shop floor to the control system are responses. Only a small part of this data stream is presented to the system operator. It gives data about actual system states at the system operator's disposal.

Quality of decisions and time needed to reach them by the system operator are variable, due to his human nature (concertation, stress, fatigue and other). Because of this, the quality of decisions can be lower and the time needed to reach them can be longer than objectively possible. The focus of this research is to improve that, by developing a decision support tool for the system operator. This tool here is named as the Intelligent Adviser Module (IAM). As an integral part of BAS control system, the IAM should take into consideration actual system states, past system states, external data from manuals and other documentation, human experts and past system behaviour.

Work of the IAM should be based on:

- Actual system state data from the interface between the control and the controlled system.
- Digitally recorded data from a significant period of past working time.
- Extraction of expert knowledge and expertise from humans directly involved with the system.
- Forecast of the execution of working scenarios for a short time horizon.
- Accumulated "situation-decision-results" cases from the past.

- Constant generation of IAM proposals according to the situation. IAM proposals should always be available. The final decision is made exclusively by the system operator. He decides if he will accept, partially accept or ignore the proposals.

1.2 Thesis structure

Title: Working scenarios of hybrid self-organizing assembly system

Thesis: This thesis represents research results derived during the author's doctoral program at the Technical University in Vienna which was conducted under the direct supervision of Professor Dr.sc. Dr. mult. h.c. Prof. h.c. Branko Katalinic.

The dissertation is presented in the following 8 chapters:

1) First chapter: introduction, main goals, structure and a glossary of abbreviations used in the thesis.
2) Second chapter: four main phases in the production process of a complex product. This includes design, process planning, machining and assembly. Each phase is analysed based on complexity, level of automation, disturbances and costs. Key phases in the development of assembly systems are described. Types of assembly systems according to range, volume and investment characteristics are introduced. Analysis of Flexible Assembly System (FAS).
3) Third chapter: Description of Bionic Assembly System (BAS) concept which is based on biologically inspired principles of self-organisation, reduced centralized control, networking between units and natural parallel distribution of processes. BAS has a hybrid control structure, which combines subordinating and self-organizing control principles. BAS main layout, key components, working scenarios, reconfiguration abilities and characteristics are described.
4) Fourth chapter: Description of BAS as a human centric system which promotes the integration of workers in the working process. Definition of human tasks on the shop floor which are performed by the shop floor operators. Definition of human tasks in the control system which are performed by the system operator as the main decision maker. Challenges and limitations of the system operator: stress, large amounts of data, incomplete information and time limitation.
5) Fifth chapter: Introduction of a decision support tool for the system operator named Intelligent Adviser Module (IAM). It is an integral part of the BAS control system. As a result of this integration, IAM should be able to learn and to improve the accuracy of its proposals over time. IAM classification, structure, functions, characteristics, realization prerequisites and working modes are described.

6) Sixth chapter: BAS and IAM are in their concept stage. For this reason, real-world performance data is unavailable. Experiment has been set-up for the investigation of contribution of IAM proposals on the quality and time of decisions. The IAM is simulated and is assumed to be fully functional. First run includes 10 human subjects and the second run includes additional 15 human subjects performing the duty of a system operator as a single decision maker.
7) Seventh chapter: analysis of experiment data. Random data was generated after two runs of experiments for 6 different case studies. The results show that the quality of decisions is higher and / or time is shorter, especially during: solving of conflict situations, solving of non-standard situations, support of less experienced system operators.
8) Eight chapter: practical IAM implementation in “Smart production” project successfully realized through the cooperation between Intelligent Manufacturing Systems (IMS) group from Vienna University of Technology, company Festo, International laboratory "Sensorika", together with specialists and graduate students from MSTU “Stankin”, INET RSUH, KIAM Russian Academy of sciences and JSC “TechInvest”. The results show that the IAM concept is functioning efficiently. Production systems with high technical similarity with BAS can increase their efficiency using the IAM concept.

The conclusion: key points of the research, results analysis and future research.

1.3 Glossary of abbreviations

AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AS	Assembly Station
BAS	Bionic Assembly System
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CIM	Computer Integrated Manufacturing
CNC	Computer Numerical Control
DMS	Decision Making System
DSS	Decision Support System
ES	Expert System
FAS	Flexible Assembly System
FEM	Final Elements Method
GUI	Graphical User Interface
IAM	Intelligent Adviser Module
IMS	Intelligent Manufacturing System
IoT	Internet of Things
IT	Information Technology
KB	Knowledge Base
KBS	Knowledge Based System
KD	Knowledge Discovery
KDD	Knowledge Discovery in Databases
KE	Knowledge Extraction
KR	Knowledge Representation
M	Match
MR	Mobile Robot
OP	Operation
R	Rank
RS	Repair Station
SUP	Supplier
SDR	Situation – Decision – Result

Chapter 2

Assembly Systems

Modern production systems produce the vast majority of global products today (Dicken, 1998). These products can be simple or complex. The level of complexity increases with the number of the product's integral components (Rodriguez-Toro, 2004).

A production overview is shown in Fig. 2.1. Complexity of each component depends on the manufacturing conditions such as choice of materials (hard, brittle, ductile, heavy, strong…), specific geometric shapes (symmetrical, asymmetrical, round, cylindrical, flat…), tolerances (high, low) and required special working environments (vacuum, pressure…).

Complexity of the assembly process depends on the number of input components, their individual complexity as well as the required sequence of operations. Based on these factors, the following types of products can be defined:

1) Simple single component products (nails, bolts,…)
2) Complex single component products (vases, engine blocks…)
3) Simple multi component products (tables, chairs…)
4) Complex multi component products (smartphones, electric motors…)

It will be much more difficult to manufacture a complex shape with a sensitive material than a simple symmetrical shape made out of a rouged material (glass vase vs steel nail). Likewise, it is much simpler to assemble flat symmetrical surfaces in an arbitrary sequence than it is to connect a large number of components, in a specific sequence of operations, where each part has a different shape, size and complexity (table vs smartphone).

As the name suggests, simple products are going to be produced faster and more economically. However, manufacturing and assembly are not the only phases a product goes through during its development. In order to improve the production of complex products, the entire development process needs to be analysed.

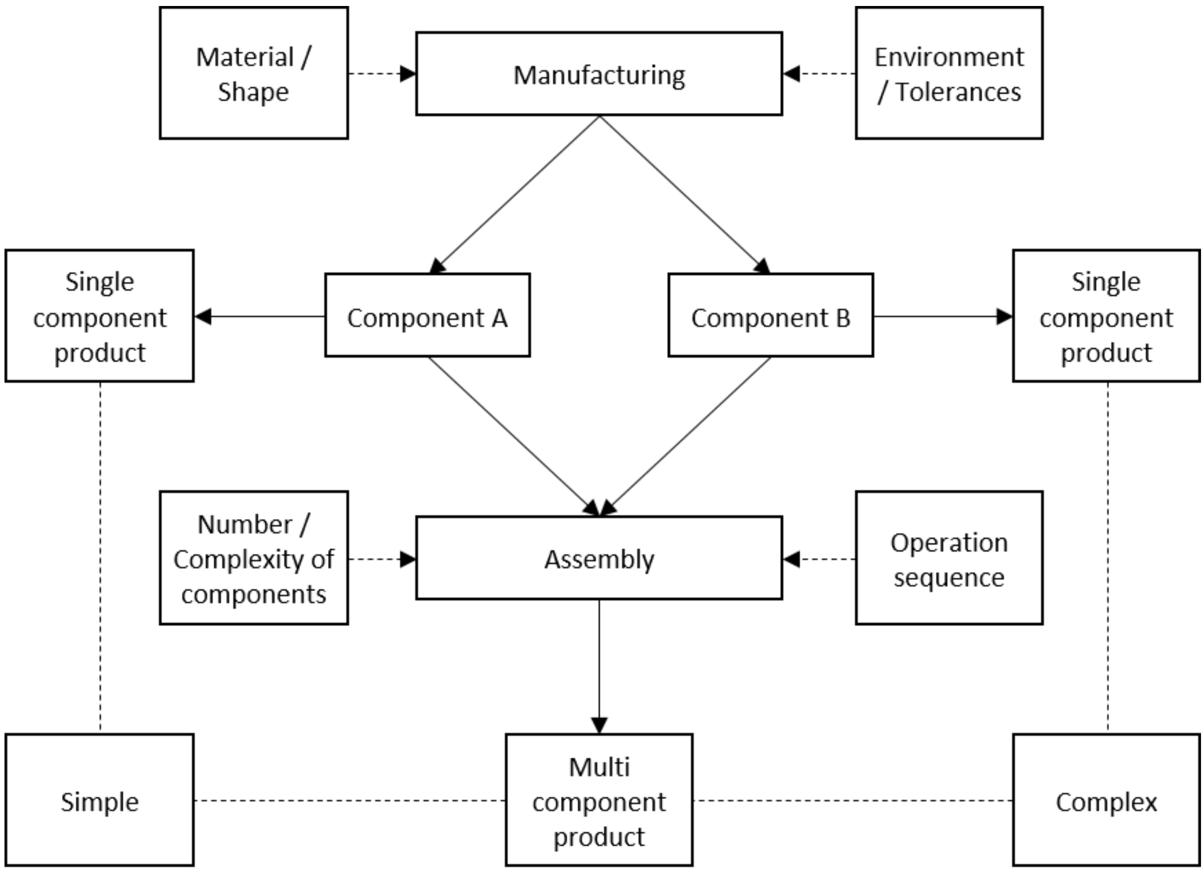


Fig. 2.1. Production overview

2.1 Production process of a complex product

A production process of a complex product overview is shown in Fig. 2.2. Each phase of this process is defined by the following characteristics:

- Disturbances represent unpredictable occurrences which impact the planned execution of a particular phase. They can be internal (hardware, software, human resources) or external (energy, material, information). A certain amount of disturbances is introduced into every phase. Each flaw, error or mistake from the previous phase is automatically transferred to the next phase.
- Complexity in the context of a production process can be defined as an inverse factor of repeatability. The higher the repeatability, the lower the complexity is. That means, wherever the human factor is essential for problem solving, it can be a complex task and therefore, can be very difficult or expensive to program or to automate.
- Level of automation is therefore inverse to complexity. It represents "how easy or economical something is to repeat in a sequence of operations based on certain conditions."

- Costs are an integral part of any production process. Each phase introduces costs from different aspects. Costs are directly connected with the resources which can be, just like disturbances, internal (hardware, software, human) and external (energy, material information).

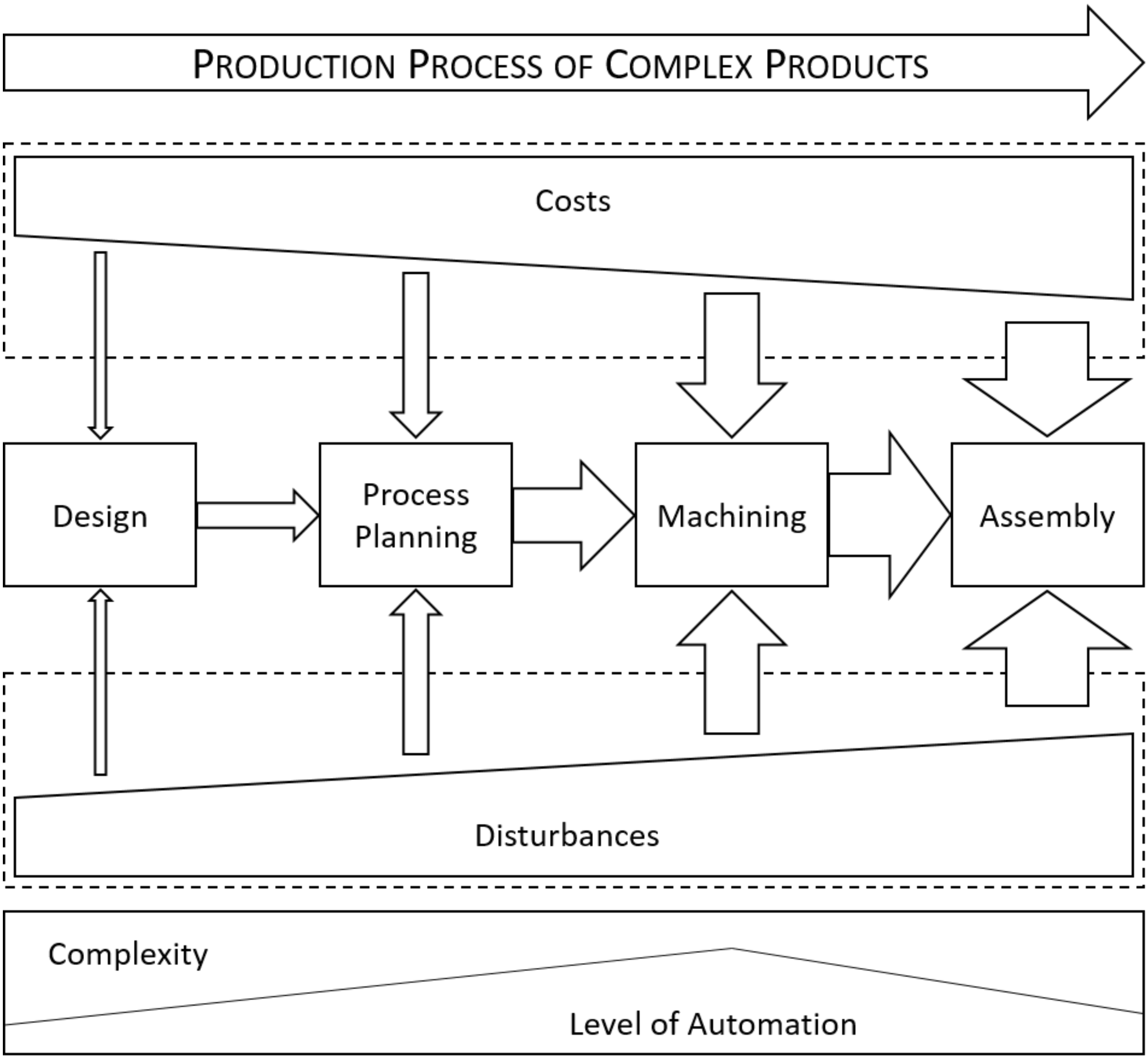


Fig. 2.2. Production process of complex products (Katalinic, 1990)

There are four main phases in the production process:

- Design
- Process planning
- Machining
- Assembly

Each phase has its own set of parameters, specifications and attributes that will be analysed in order to determine where further improvements could be applied.

2.1.1 Design

All products start as an idea or a concept. It is necessary to transfer this idea to a technical drawing or some other form of technical design specification. This allows to analyse the tolerances between the components, identify stress levels in critical areas and to revise the design in iterations in order to improve the original idea or concept.

Traditional tools, such as clay, wood or other deformable and cheap materials allow to physically manifest an original idea. Modern tools such as computer aided design (CAD) software increase the speed and accuracy of creating technical specifications (Koren et al., 1999). Final elements methods FEM is a numerical approach for solving complex problems. It is used to analyse stress levels, thermal expansions and other physical properties of various materials even before the actual production has started (Dhatt et al., 2012). Another very important tool during the design phase is the use of rapid prototyping (3D printing). It allows to transfer a computer model to the real world. This way, cheap, iterative modifications are performed until the desired functionality or aesthetic has been reached (Lipson & Kurman, 2013).

This early phase of the production process has a low level of disturbances. Although there have been some attempts at automating this step (Bentley, 1999), the level of automation is still low. The complexities of creative thinking and human problem solving skills are still beyond today's algorithms. Some design automation techniques exist in a form of analysis and re use of existing technical solutions (bolts, drawings, modular products…). It can be said, that if a new product is very similar to an already existing product, it is possible to automate the design process to a certain extent.

Once the design is verified it is necessary to plan a process for its production. This does not mean, of course, that the design is final. It is always possible and quite often, that some design flaws are detected in the later stages of the product's lifecycle (Saaksvuori & Immonen, 2008).

2.1.2 Process planning

Next phase is to plan a manufacturing process according to the design of the product. This phase determines the sequence of manufacturing operations and the selection of appropriate tools (Feng & Zhang, 1999).

The main criteria for process planning is the most efficient use of internal (hardware, software, humans) and external (energy, material, information) resources. In other words, how to produce something as quickly and accurately as possible with the minimum use of resources.

Tools such as CAPP (Computer aided process planning) are used in combination with the experience of specialized professionals who have great knowledge about machines, tools, materials and manufacturing technology.

Planning phase can discover if there are any design flaws (wrong element position/orientation, tolerances…). This is often due to technological limitations not taken into account in the earlier stages (tools size, material properties…)

Level of disturbances is medium low as we are introducing design errors as well as new disturbances from the process planning. Level of complexity is medium because the use of technology is standardized and as such is easier to automate with the help of CAPP (Kamrani et al., 1995).

Once the documentation is completed and the sequence of operations is determined, the next phase is to physically produce the components of the final product – machining.

2.1.3 Machining

Various methods (additive or subtractive machining, forming, joining, moulding …) and tools (lathes, mills, drills, moulds, 3D printers…) are used to transform input raw material to a specified size and shape (Boothroyd, 1988). The choice of method and tool depends on multiple factors such as material properties, desired tolerances, surface finishes, operation speeds, investment costs etc.

In the entire production process, machining has the highest level of automation (Liang et al., 2002). This implies that the complexity is the lowest. One of the reasons is that once the machining parameters have been set, a computer software controls the process. This sequence is highly repetitive and as such is perfect for automation. Machining as such, produces parts which are independent from the final product. In other words, a single machine can produce parts for essentially different final products. This makes machining more universal and easier to automate. CAM Computer aided Manufacturing and computer numerical control (CNC) are great examples of computer integration with machining (Lee, 1999).

Level of disturbances is medium high. Flaws from the design and process planning phases are directly combined with the unpredictable occurrences happening during the machining (machine shut downs, tools breaking…).

Once the parts have been machined, they are ready to be assembled in the case of a multi component product.

2.1.4 Assembly

The last phase in the production process of a complex product is assembly. During assembly, all the individual components are connected in a sequence of operations. For some products, the sequence is arbitrary and for some it is specified and very strict.

The level of complexity is very high. There are many variables that need to be taken into account. In order to assemble a complete and functional product, the right parts, in the right condition, in the right quantity, at the right time, in the right place are necessary. Because there are a lot of complex steps that need to be completed, assembly automation can be very difficult, expensive and time consuming (Hu et al., 2008).

Depending on the complexity of the final product (Lotter & Wiendahl, 2006), assembly process expenses and production time can account for 30 % (consumer grade products), 50 % (automotive industry) and 80% (advanced electronics) of the entire production.

The level of disturbances is highest in this phase. A large number of operations needs to be performed. All the errors from previous phases are directly influencing the outcome during the assembly process. In addition to that, each component, every machine and human performance are a source of potential error or disruption during assembly.

The type of assembly systems depends heavily on the final product. It is not possible or at very least, it is not economical to use assembly systems for a large number of different families of products. Therefore, assembly can be the most time consuming, complex and expensive phase in the production process. Improving the assembly process in the context of time, efficiency and throughput of final products will results with increased performance of the entire system. This in turn ensures that a higher level of product quality can be reached at lower production costs (Amen, 2000).

The rest of this chapter will focus on the development of assembly systems through key technological and organizational breakthroughs, main types of assembly systems based on the volume, investments and range of products and at the end, introduce modern hybrid assembly concepts.

2.2 Development of assembly systems

The development of assembly systems has been defined with various key technological and organizational breakthroughs. These breakthroughs have increased quality, efficiency and assembly times for complex products. All throughout the history of development, a paradigm shift from mass production to mass customization was taking place (Pine, 1991). Production of goods started from a very low volume of unique products, transitioned through high volumes of low range of products towards modern variable range of products and medium volumes.

The key development points are:

2.2.1 Division of labour

It was becoming apparent that one highly skilled craftsman producing an entire final product one at a time was inefficient and expensive. This worker had to be very skilled which made him very hard to replace. As introduced in “The Wealth of Nations”, by Adam Smith, the term “d*ivision of labour”* was introduced (Smith, 1776) as a method to increase production output. The main principle is to divide a complex production process into individual, simpler assembly tasks which could be completed by less skilful workers. This was the first crucial step in the development of assembly systems.

2.2.2 Interchangeable parts

Additional problem with craftsman was that each part or product they made was in a sense, unique. Each component was tailor-made to fit into the final product. This introduced difficulties during repairs. As a necessity for war efforts, (Woodbury, 1960) Eli Whitney created machine jigs that created same musket parts repetitively. Now, it was possible to repair or replace a single part of a product with ease.

Another additional breakthrough came with the use of descriptive geometry (Monge, 1811) introduced by Gaspard Monge. Each product was described by its components using precise measures and distances between them. This represents the introduction of technical drawings. With the use of technical drawings it was simpler to produce products with common parts which increased efficiency and range of products. Industrial revolution introduced increased volume production with the use of steam powered machines and in combination with precise technically specified, interchangeable part, mass production was ready to begin.

2.2.3 Conveyor line

Next step which lead to the increase of product throughput was the introduction of a system that connects individual stations in an assembly sequence. The main principle was that a product from start to finish moves along a carrying line. The conveyer belt system was introduced by Henry Ford (Ford, 1926). Each worker or machine performs a single task in a repetitive manner. Doing this, it was possible to reduce the production costs by a huge margin and increase the volume of production.

2.2.4 Information technology

The second half of the 20th century saw a rapid development of computers and information technology. The industrial potential of computers was apparent in a sense that they are easily programmed to execute a series of commands under specific conditions. This lead to the development of computer integrated manufacturing (CIM) (Polckofl, 1990).

In combination with computers, the continued development of the assembly line was defined with the use of robots as a substitute for human workers. Robots are great at hard physical labour, dangerous working conditions and repetitive work where high precision is needed.

In the 21st century all the devices started to be interconnected. This lead to the definition of Industry 4.0 (Post, 2014) where the emphasis was on automation, data exchange and modular manufacturing. Each device, machine, robot and sensor is connected to an informational exchange server. As a result, big amounts of data are used in order to analyse performance and predict future trends.

One possible future development directions is the use of artificial intelligence in modern production systems. With computer systems and their algorithms becoming ever more sophisticated, it is very possible to expect such implementations (Kusiak, 1987).

2.3 Types of assembly systems

There are three main characteristics that serve as a basis for comparison between different types of assembly systems (Spath et al., 2007):

1) Range of products
    a. Custom
    b. Stock
    c. Modular
2) Investment
3) Volume

The definition of assembly system type starts with the choice of products. Products can be custom, stock or modular (Kumar, 2013). Custom products are assembled after the customer order has been received. If there are many options and variations, it is difficult for the manufacturer to predict what exact combination the customer wants. These types of products are assembled in low volumes but with high variation. An example of a custom product is a ship, specialized heavy goods vehicle etc.

Stock products on the other hand, are assembled before the customer orders it. They are produced in large volumes with small variations such as home appliances. Modular products are a combination of custom and stock products. Standardized modules are produced before the customer order is received. The final product is assembled according to customer wishes using the standardized modules. An example of modular products are kitchen or furniture assemblies.

Production volume and investments are proportional. This means that in order to assemble a larger number of products, higher investments are necessary. These investments include custom machines, energy, material, as well as operational and maintenance costs. However, this also implies that the range of products will be determined by the specialized hardware needed to assemble them. It will be more difficult to assemble a large number of products that are not from the same family and do not share the same components.

Therefore, volume of production and range of products are inversely proportional. The higher the specialization of the system is, the lower the range of products become. Fig. 2.3. shows the overview of types of assembly systems in correlation with the characteristics that define them.

Based on the types of products, initial investments and the desired volume of production, assembly systems can be classified as (Groover, 1987) manual, automated and hybrid. Table 2.1 presents a comparison of each of the assembly systems characteristics.

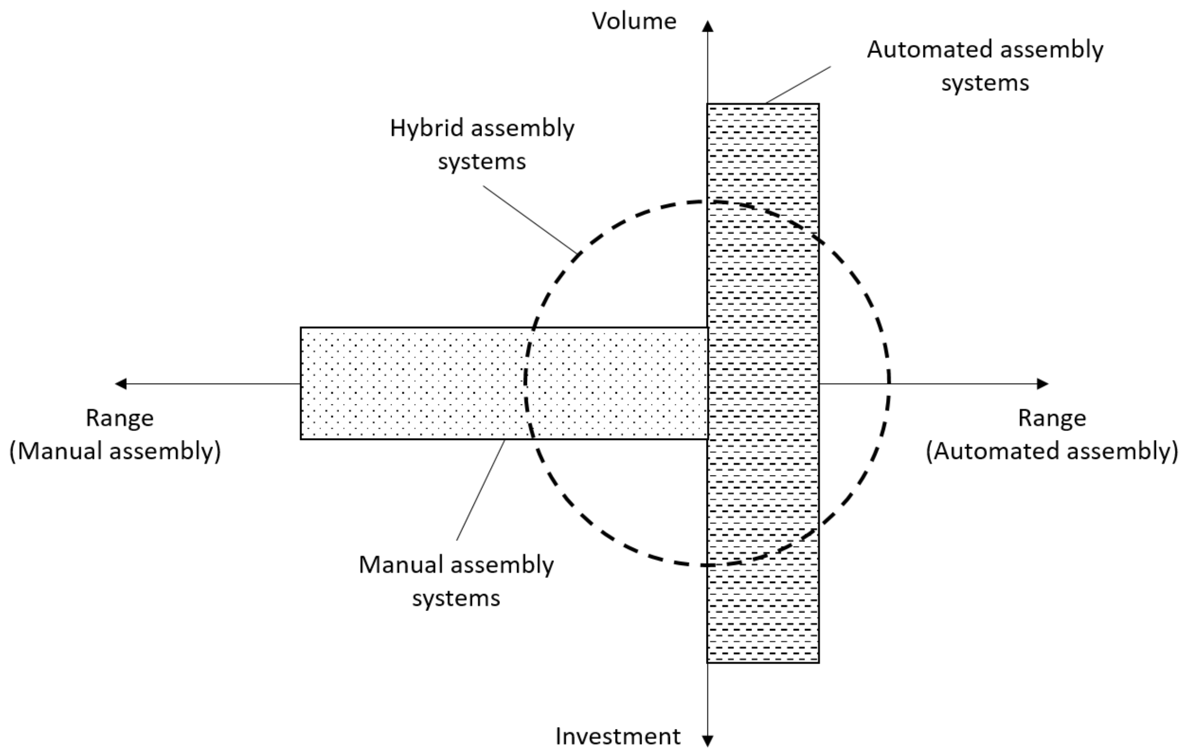


Fig. 2.3. Types of assembly systems

- Manual assembly system is used to assemble a large range of custom products in low volumes. The production is very flexible because more general purpose equipment is used in combination with non-determined routes of assembly. Multiple paths of assembly are possible for the same type of product. The use of standard machines and tools keeps the investments low. The volume of production is low because the wait times between operations are high (tool change, machine preparation…). Example of a manual assembly system is a job shop (metal fabrication shops, woodworking…).
- Automated assembly system is used to assemble a large volume of stock products with low variety. The path of assembly is set and the equipment is very often specialized for a specific type of product. This involves high investments that need to be amortized through high volume of production. Wait times between operations are low and the productivity of each station is high. Example of an automated assembly system is an assembly line (cars, electronics...).
- Hybrid assembly system is a synergy between manual and automated systems. It includes automated machines and human operators working simultaneously on the same shop floor. The main goal is to combine strengths of each system. Some operations are too dangerous, too heavy or require high repetitive precision. In such cases automated specialized machines or robots are the most suitable option. On the other hand, some operations are too difficult or expensive to automate. In other words, human operators are best utilized for tasks where their intellect, dexterity, problem solving, creativity and relative low costs are primary requirements (Owen, 1984).

Table 2.1. Overview of assembly characteristics

Characteristic	Manual	Automated	Hybrid
Product type	Custom	Stock	Modular / Stock
Break point	No	Yes	No
Production volume	Low	High	Medium
Investment	Low	High	Medium / High
Product range	High	Low	Medium
Human compatibility	Yes	No	Yes
Wait times	High	Low	Medium

2.4 Hybrid assembly systems

Current manufacturing trends are defined with shorter product life cycles, high demand for complex products, variable volume and range of production. The existing manual or automated assembly systems are not able to answer to these challenges. As described, a new hybrid assembly concept was introduced. One such hybrid assembly system is Flexible Assembly System (FAS) which combines the high flexibility of a job shop with the efficiency of an assembly line.

2.4.1 Flexible assembly system

A Flexible Assembly System is an integrated, computer controlled complex of automated material handling devices and numerically controlled (NC) assembly stations that can simultaneously process medium sized volumes of a variety of part types. (Stecke, 1983).

All Flexible assembly systems (FAS) have the following components that define them (Heilala & Voho, 2001):

- Machine tools (universal or specialized)
- Material handling system (conveyors, carts, AGV, carousel, manual)
- Storage area (buffers, warehouses)
- Centralized computer control (distributed or centralised)

Based on these characteristics the following types of FAS are defined (Browne et al., 1984):

- Flexible Assembly Cell is the simplest one. It consists out of a single general purpose CNC assembly machine where a material handling system (pallet changer, robot arm) loads the work piece from input buffer and unloads the finished product to the output buffers.

- Flexible Assembly System consists from multiple different types of flexible assembly cells. Multiple operation routes are possible using various material handling systems as shown in Fig. 2.4.
- Flexible transfer Line is a system where each operation is performed on a single machine following a fixed route on a carousel or a conveyor. It can combine general purpose and specialized machines.
- Flexible Transfer multi line – consist from multiple flexible transfer lines which increases the routing flexibility in case of a breakdown.

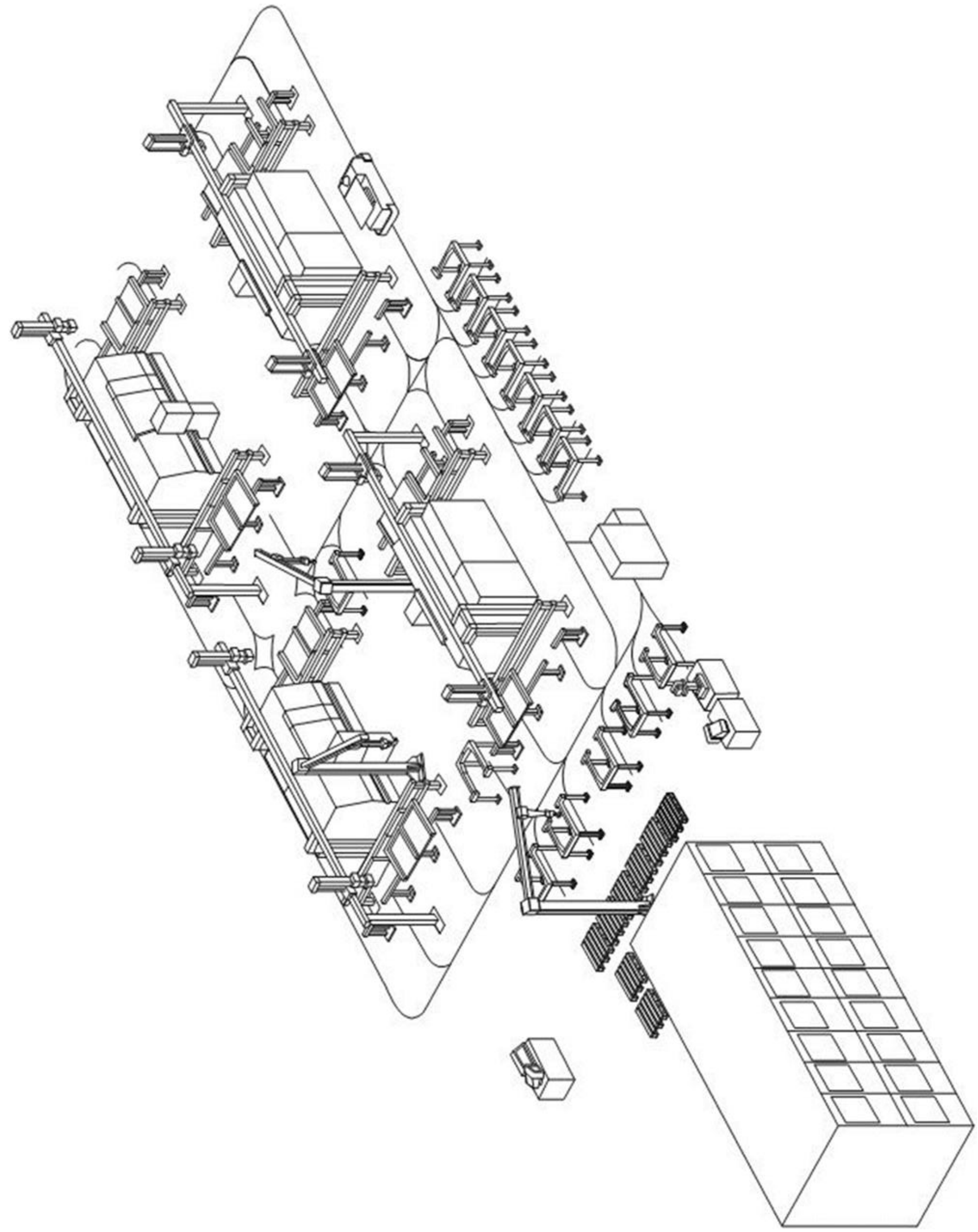


Fig. 2.4. Flexible Assembly System for rotational parts (Katalinic, 1990)

The flexibility of a system can be defined according to the specific characteristic (Browne et al., 1984):

- Machine flexibility – how fast and simple a machine can prepare to assemble a required product type (tool change and positioning, NC program load…).
- Process flexibility – capability of the system to perform multiple operations and to process multiple part types simultaneously without batches.
- Product flexibility – capability to perform a changeover in order to assemble a new family of products.
- Routing flexibility or error recovery flexibility is the capability to continue the assembly using alternative routes in case of a machine failure or other form of disturbance.
- Volume flexibility is a capability to profitably process variable production volumes.
- Expansion flexibility is a capability of the system to be easily expanded if there is a need.
- Operation flexibility is the capability of the system to keep the ordering of operations free in cases where the operation sequence is arbitrary and the machine is available
- Production flexibility represents the versatility of the entire system through the list of all products that can be processed. In other words it is the range of products that the system can handle

2.4.2 Current FAS disadvantages

However, according to numerous researches and analyses, (Whitney, 2004), (Katalinic, 2004), (Fiorentino, 2014), the following results have indicated the disadvantages of using flexible assembly systems:

- Complexity – these systems can be comprised of large number of components and subsystems. It ca be very difficult to design, control and maintain them.
- Cost of equipment – the equipment used is often very specialized. The use of nonstandard equipment also implies that maintenance and repair costs will be higher. Most of the equipment is automated, which requires operational and setting expenses.
- Adaptation issues – it is not always possible to adapt to changes in a product mix due to limited machine capacity and tools.
- Equipment utilization – utilization is often not as high as expected due to scheduling and possible longer waiting times.
- Product price determination – difficult to determine the amount of work performed on a specific machine for a specific product. Lot of additional operators, technicians and engineers needed for maintenance.
- Substantial preplanning and scheduling – as it can be a very complex system, a lot of careful and resource consuming prescheduling is necessary in order to utilize the equipment.
- Human worker resistance – due to being scared of being replaced by a machine. Also a lot of additional training necessary for humans to operate in FAS.

2.4.3 Desired Characteristics of modern assembly systems

Based on these limitation, a new hybrid system approach was needed. The general concept had to revolve around modularity, extended flexibility, reduced complexity and worker friendly environment. As proposed by Katalinic, (Katalinic, 2002) the following characteristics should define further development trends:

- Realization of working scenarios with high:
    - Efficiency
    - Adaptability
    - Robustness
- User suitable and friendly for:
    - Planning
    - Human / machine coexistence
    - Controlling
    - Monitoring
- Ability to learn from past working cycles
- Control structure focused on the use of:
    - Computer integration
    - Intelligence
    - Self-Organization

2.5 Emerging concepts

Today's highly competitive global environment has created a demand for systems to be dynamic and respond efficiently to any changes. As a result, fractal, holonic and bionic concepts have been proposed. The main inspiration is taken from nature in regard to the organisation of the smallest units. The main principles of these emerging concepts are distribution, autonomy and adaptation.

2.5.1 Fractal Factory

The main concept of a fractal factory is that it is composed of small components or fractal entities (Warnecke 1993). Fractals are independent units which have similar characteristics. This modularity allows to react quickly to the production environment and to adapt to dynamic conditions. The fractals need to operate as a coherent system. This is achieved through participation and coordination among the fractals (Tharumarajah et al., 1998). There is teamwork among the fractals which enables distribution of power and ability. A fractal object is defined by the following features (Leitao & Restivo, 2003):

- self-organisation, the objects do not need external impulse to organise themselves
- self-similarity, all objects in fractal factory have similar characteristics and goals
- self-optimisation, the system is constantly increasing its performance

2.5.2 Holonic Manufacturing

The term “holonic is derived from the word “holon”. Holon is a combination of a Greek word “holos” meaning “whole” with the suffix “on” which, as in proton or neutron, suggests a particle or part. It was introduced in (Koestler, 1967).

Holons are autonomous and cooperative units which have independence and perform duties without asking higher authorities. A holon can be a robot, a machine, order or a human. The important feature is that holon can be part of another holon, or it can be broken into many other holons. It has the information about itself and the environment.

Holonic manufacturing originated in the framework of the Intelligent Manufacturing Systems (IMS) programme (Bongaerts, 1998).

2.6 Summary

This chapter presented the main 4 phases during the production process of a complex product. These include design, process planning, machining and assembly. Out of these 4 phases, assembly has been identified as the one which can be the most expensive. This is because, unlike machining, assembly heavily depends on the final product.

A new, hybrid assembly concept was introduced as an answer to current manufacturing trends which are defined with shorter product life cycles, high demand for complex products, variable volume and range of production. One such system is the Flexible Assembly System (FAS) which combines the high flexibility of a job shop with the efficiency of an assembly line.

However, FAS can be very complex and expensive. A new approach was needed which would be based on modularity, dynamic reconfiguration and reduced complexity. One such concept draws direct inspiration from nature in form of biological systems. This new type of a hybrid assembly system concept was introduced by Katalinic (Katalinic 1997, 2002). Bionic Assembly System (BAS) was developed for real industrial purposes where there was a demand to significantly reduce production costs of electrical motors in mass production. BAS main concept, system overview, key components and features are presented in the next chapter.

Chapter 3

Bionic Assembly Systems

As described in the previous chapter, FAS can be very complex due to their design, control and maintenance. Additionally, they use expensive, specialized, automated equipment which can be difficult to utilize due to substantial preplanning, scheduling and longer wait times. FAS represented the most optimal solution for the time it was introduced. New breakthroughs in science and technology have defined new development trends for modern assembly systems. One such example is the IT sector which includes improved hardware to price ratios, communication technology, software, AI, data technology etc.

Bionic Assembly System (BAS) represents a next generation of modern, hybrid assembly systems. The initial concept and layout was proposed, described and developed by Katalinic (Katalinic, 2001). The main concept is based on a biologically inspired principle of self-organisation. A system that is capable of self-organization is able to deal with a highly variable environment through adaptation, evolution and learning (Leitao, 2009). The main idea is to divide a complete system into smaller subsystems that interoperate between each other (Botti & Boggino, 2008). The principle advantage of such a system is reconfiguration. In an assembly context, it allows the system to quickly adjust its capacity and functionality in response to sudden unpredictable changes as well as during the introduction of new products or production technology. Overall, system needs to be modular, less rigid and able to distribute tasks. As a result, it would be possible to reduce the complexity of such a system (Colombo et al., 2004).

This chapter presents a research overview conducted by the Intelligent Manufacturing Systems (IMS) group from Vienna University of Technology. The research results were presented in the following published articles:

- BAS initial layout concept, system elements and description (Katalinic, 2010)
- System modularity and adaptability capabilities (Kukushkin et al., 2011)
- BAS hybrid control structure (Katalinic et al., 2013)
- Role of the Adviser Module in the Hybrid Assembly Subordinating Control Structure (Haskovic et al., 2014)
- Intelligent Adviser Module functions within the BAS control layout (Haskovic et al., 2015)
- Structure and Working Modes of the Intelligent Adviser Module (Haskovic et al., 2016)

3.1 Self organization

As stated in the introduction of this chapter, the main concept of BAS is a biologically inspired principle of self – organization. It is therefore, important to define the key characteristics and the main behavioural structure of such a system in order to apply it in a technological assembly context.

Self-organization is taking place regardless of the environment, scale or physical properties of the entities. It is present in inanimate or biological systems. Likewise, it can occur from a subatomic scale to the magnitude of the entire universe (Glendening, 2013). Technological systems are normally organized through external commands and instructions. But in contrast, often times natural systems are organized by their own internal processes. These have been identified as self-organizing systems, where simple actions and interactions produce emergence of more complex systems (Yates, 2012).

3.1.1 Examples from nature and technology

The phenomenon of emergence and pattern formations has intrigued scientists from many fields. As a result, there are many examples and descriptions from all aspects of the natural world, as well as a high variety of technological implementations. Science fields such as biology (living organisms, sand dunes, skin patterns…), chemistry (crystallization, molecular self-assembly…) and physics (magnetism, fluid dynamics…) help to observe, describe and reach conclusions about various self-organizing mechanisms (Haken, & Jumarie, 2006). On the other hand, these mechanisms are being adapted for use in computer sciences (distribution of processing tasks, artificial intelligence, swarm theories, multi agent systems, traffic behaviours…), economics (self-organizing aspects of the free market), anthropology (human society, crowd behaviour etc.

Biology is the primary source of inspiration for BAS. Self-organization in living organism structures is represented as an interaction between units of a specific species. Classical examples are school of fish, swarm of bees, herd of sheep and so on. Mechanisms such as natural distribution of tasks, hierarchical roles of individual units, collision avoiding and community decision making are observed. In order to apply such mechanisms to BAS, it is necessary to define key points in self-organization and to create a comparison line between biological and assembly systems and the units involved.

3.1.2 Introducing self-organization concept to assembly systems

Self–organization is a very complex natural occurrence with multiple applicable definitions and explanations. Although it seems simple and basic in its function, it has been very difficult to define it mathematically or formally. It consists out of many different phases, each with its own characteristics, forms and affects.

The following definitions of self–organization can be cited:

"Self-organization is a process whereby pattern at the global level of a system emerges solely from interactions among the lower-level components of the system. The rules specifying the interactions among the system's components are executed using only local information, without reference to the global pattern." (Camazine, 2003).

"Self-organization is a set of dynamical mechanisms, whereby structures appear at the global level of a system from inter-actions of its lower-level components. Self-organization relies on four basic ingredients (Bonabeau et al., 1999):

1) *Positive feedback, or amplification*
2) *Negative feedback that counterbalances positive feedback*
3) *The amplification of fluctuations*
4) *Multiple interactions between units"*

Based on these definitions, several key points can be observed on what characteristics should be applied to technological assembly systems. Simple units should interact with each other and perform tasks based rules. This interaction presents a lower level of the system, where goal oriented actions define the global level. This in turn defines emergence where complexity is derived from simplicity of individual interactions. The changes in the system can be spontaneous or controlled by an external force from outside of the system.

Based on these factors, BAS concept can be defined with some analogies form the nature. Table 3.1. shows the correlation between BAS and biological structures. The following terms are defined as (Katalinic & Lazinica, 2003):

- Unit – building block of a system. It is the most basic component which performs tasks
- Task – specific action or a performance that is being completed
- Source – supplies the unit with the necessities required for completing a task
- Performance – the unit's movement towards the source in order to complete the task

Table 3.1. Correlation between Bionic Assembly System and biological structure (Katalinic & Lazinica, 2003)

Term	**BAS environment**	**Biological similarity**
Unit	Robot, AGV	Ant, bee
Task	Transport	Defend, supply, mate
Source	Pool of pallets, assembly station	Flower, food
Performance	Connect the assembly process	Ensure food supply to the colony

After setting the correlation between the biological and technological structures, main characteristics of BAS and the operating conditions are defined.

3.2 Characteristics of Bionic Assembly System

BAS represents a continuation of development of modern assembly systems. The main purpose is to solve several key issues with existing solutions which are in use today. Modern assembly parameters include products with high complexity, high variety and shorter lifecycle. Current highly automated assembly systems such as Flexible assembly systems (FAS) have a very specific use for assembling a group of very similar products.

Based on the range size and type of products, such a system is very rigid and predetermined in its function and operability. It is very complex for scheduling and inflexible towards introducing changes both in products and in assembling processes. Each addition or change can cause high costs, introduce additional complexities and difficulties.

It is therefore necessary, for this new, modern Bionic Assembly System to be defined with several founding characteristics that are drawn from nature and self-organization structures. The system needs to be:

- Modular – based on product types, demand and sequence of operations, the system can be expanded, reduced or recombined. It allows to accommodate different assembly stations, people, robot trajectories etc. There is no need for expensive shutdowns in order to rearrange or change a system part. Everything is self-contained and allows for easy restructure.
- Reconfigurable – this property is closely connected to modularity. It allows the system to accommodate different assembly techniques. For a certain family of products, a better quality is achieved with manual or semi-automatic assembly. For other types, it should be fully automatized. No matter what the requirements are, the system is able to quickly reconfigure and complete the order.
- Decentralized – traditionally, assembly systems are very centralised. That means that every process, assembly station and robot need to be directly controlled. This makes the entire control system very complex. By decentralizing the system, each component is a self-sufficient unit that follows a simple set of instructions and interact with other units.
- Flexible – the system is adaptable to variable demands. This is a result of modularity and reconfigurability.
- Robust to disturbances – because there is no centralised control, a natural parallel distribution of tasks among the units is taking place. If one robot or an assembly station becomes non-operational, it does not cause a system wide failure. The system automatically reorganizes itself in a way that non-operational units are not a part of the overall process.
- Able to learn – the system stores all the operational data, disturbances and conditions under which they occurred. Using data analysis, certain patterns are discoverable. This way, the system can adapt or avoid a potential future disturbance.

Based on these key characteristics, BAS has to operate under the following conditions, rules and limitations (Katalinic, Visekruna, Kordic, 2002):

- Product variety and complexity – the system is dynamically able to complete assembly of wide range of products with variable degrees of complexities.
- Variable assembly run size – this system is designed for a small to medium assembly run size. As such, the system is suitable for assembly runs varying from one to thousand.
- High quality of the assembled final product – the final product needs to be up to standards of the original design Depending on the product, price and deadlines, a sufficient number of quality checks during and after completion of the assembly are required.
- Quick repairs – in case of an error during assembly, or a sub quality component, quick and simple repair stations are available during the assembly process. The system needs to recognize and organize necessary repairs in order to satisfy the number of ordered products.
- System performance analysis – the system has a real time performance feedback. Each station, operator or robot have a statistic which shows if there are underperforming or malfunctioning system components.
- Large amounts of data – during the execution of working scenarios, all of BAS components are producing data which is recorded. This data can be used to extract new knowledge.
- Failure predictions – based on data analysis of past working cycles, hardware malfunctions can be predicted and avoided.
- Alternative working scenarios – in case of a disturbance during the execution of working scenarios, the system offers alternative solutions so that the customer orders are satisfied within the deadline.
- Self-optimization – during the execution of working scenarios, the system is by design, choosing the most appropriate route of assembly.
- Order needs to be finished and ready for customer delivery according to the order specification. The order is defined with deadline, type and number of pieces.
- Worker friendly – the system easily incorporates human workers within the assembly process. Each worker can perform either a quality check, repair, particular assembly operation, or he can perform supplementary tasks (maintenance, supply, set ups…).
- Working shifts – based on demands, the operation of the system can be continuous or divided into shifts.
- System internal reserves – the system needs to have accumulated reserves for reducing negative influences from external disturbances.
- Product lifetime – Product lifetime is limited and shorter than the lifetime of the assembly system.

3.3 BAS Layout and elements

Bionic Assembly System is divided into a control and an execution level. Tasks performed on the control level include scheduling, resource monitoring, data analysis and serve as an overall overview of the system. The execution level represents the physical synergy of information, material and energy. In other words, it contains all the hardware necessary to complete a task. The principle layout is divided into two subsystems as shown in Fig. 3.1. and 3.2. These are the core subsystem and the supplementary subsystem (Katalinic, 2001). The main elements of BAS subsystems are shown in Table 3.2.

Table 3.2. Elements in the Core and Supplementary subsystems

Core Subsystem	**Supplementary Subsystem / Storage**
Shop operator	(lavatories, breakroom, operational rooms)
Mobile robots	(service parts, batteries, replacements)
Assembly stations	(operational fluids, tools, parts, service)
Quality control station	(replacement tools, special measurement devices)
Repair station	(surplus parts, defective bin, recycling)
Loading / Unloading station	(product components, assembled products)
Packing station	(packing pallets, wrapping material)

Main activities in the core subsystem include assembly operations, quality control, repair and packaging of assembled products. The supplementary subsystem's main function is storage of parts, components, replacement tools, operational fluids, auxiliary or special equipment. The entire system is designed to be flexible, adaptable and reconfigurable. It is important that the operational disruptions are minimised. For this reason, the hardware and material exchange between the two subsystems needs to be as efficient and fast as possible. Therefore, the supplementary subsystem surrounds the core subsystem. The exchange itself is flexible and can be completed by an automated transport system (mobile robots, AGV) or by a human worker. The outputs from the supplementary subsystem are components and parts. The output from the core subsystem is a final, assembled product with a satisfactory level of quality. The following paragraph describes the functions of the core subsystem elements.

Shop floor operators – human workers that perform their duties on the shop floor, next to the mobile robots and assembly stations. Some tasks are more suitable or economic for human operators to complete. Their problem solving skills, dexterity and intellect are a valuable advantage. BAS allows and promotes the integration of humans in the work process. The specific activities which they perform are described in the following chapter 4.

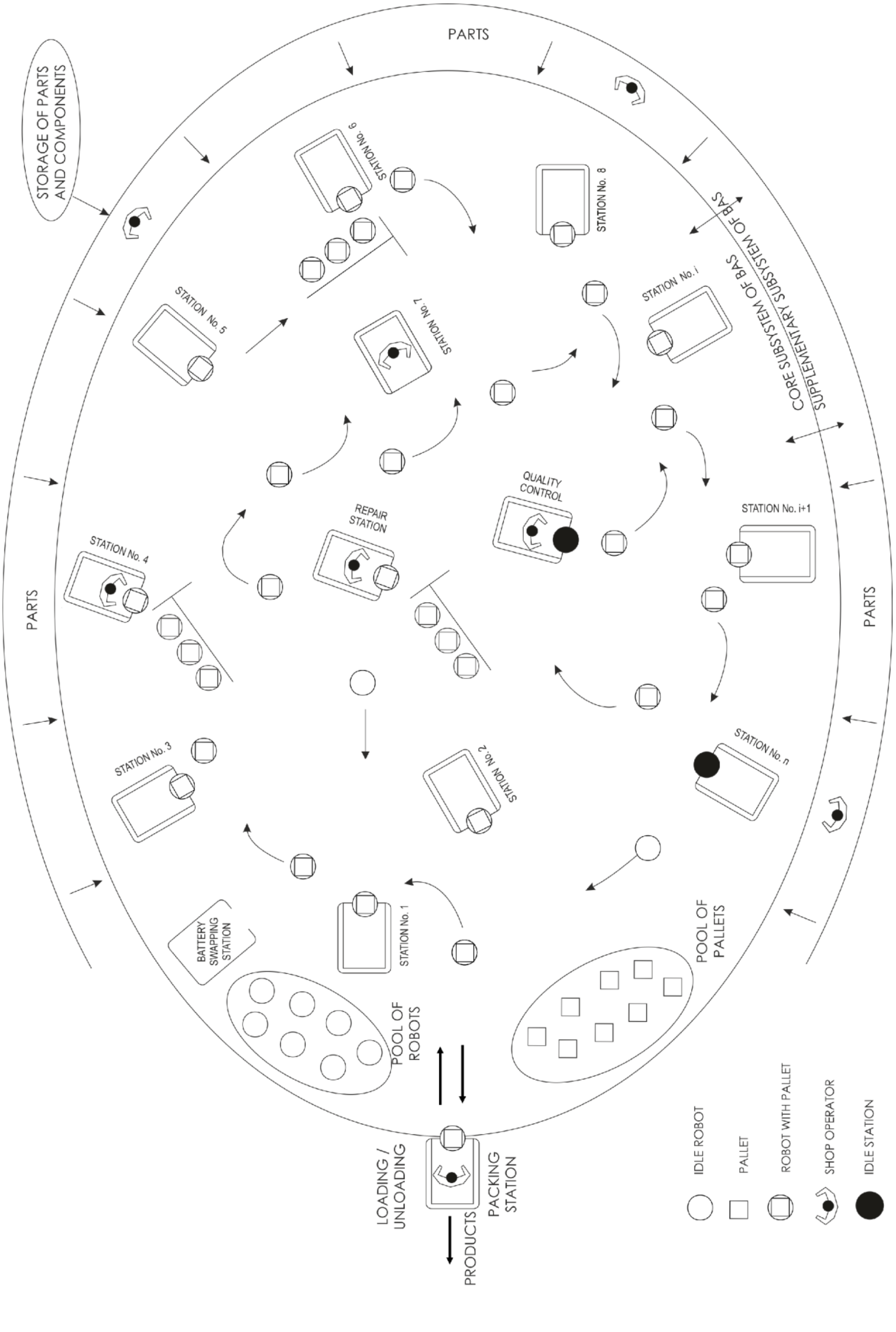


Fig. 3.1. Layout of a Bionic Assembly System (Katalinic, 2001)

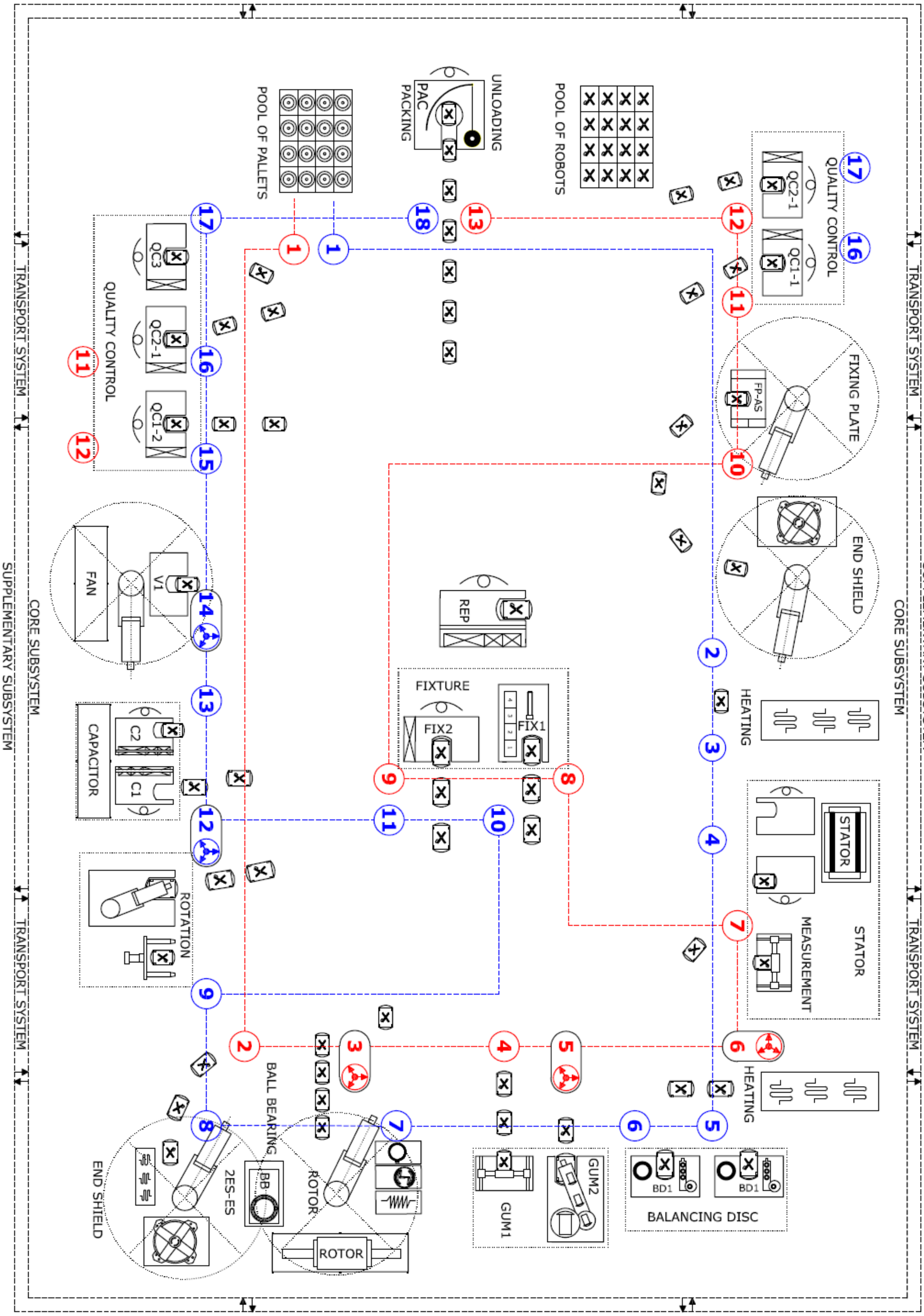


Fig. 3.2. BAS assembly flow organisation for one (red) and double sided (blue) motor families (Kukushkin, 2014)

Mobile robots – autonomous and automated transport units. Primary function is to connect the entire system and to ensure the execution of BAS working scenarios. They are designed to carry assembly pallets from station to station until the product has been assembled with satisfactory level of quality. Their trajectories are not predetermined but rather depend on the overall state of the system. Their basic behaviour is representing self-organizing qualities of BAS. In order to replicate a biological worker unit (bee, ant), they have to be equipped with a CPU, memory, ID scanners, visual / positional trackers and information transfer / receiver technology.

CPU and memory represents their brain. It has to be capable to perform simple algorithmic calculation such as status check, decisions for the next order and self-preservation (battery level recharging). ID scanners (RFID, magnetic, laser) and GPS serve for component identification and to determine the current position. Visual capabilities and trackers serve for space awareness and collision avoidance. Using transmitting technology (radio, wireless area connection, Bluetooth), they are able to communicate with other units in their surroundings.

Mobile robots can have the following states:

- Turned off – mobile robot is not actively participating in any assembly operations due to an error or it is simply not needed based on the current work load. Turned off robots are located in the supplementary subsystem storage.
- Idle – mobile robot is in a stand by mode. It is waiting for the next assembly order and is located in the pool of robots.
- Active – mobile robot is moving to complete an active assembly order.
- Error / repair – if there was an error during the execution of working scenarios or during standby mode, robot will report error state.

Assembly stations – In this dissertation, all machines on the shop floor, which are performing assembly operations will be called assembly stations. Some stations are designed to complete multiple operations and some are specifically designed for one type of assembly operation. Alternative stations are capable to perform the same set of operations. By implementing alternative stations, self-optimization is possible.

Assembly stations can be:

- Manual – the assembly operation is performed by a shop operator.
- Semi-automatic – the assembly operation is performed by a shop operator using a specific tool or an assembly station.
- Automatic – the assembly operation is performed by a fully automated assembly station.

Quality control station – based on the desired quality control level, the product can go through multiple control checkpoints. These stations can be automated, semi-automated or manually operated by shop workers. If the quality is positive, the mobile robot transports the product to further stations or in case of a finished product to the unloading station. In case that the quality is not satisfactory, the robot transports it to the repair station.

Repair station – repair is conducted in case of an error during assembly or if a certain component was defective. A shop floor operator examines the situation and decides if the product is suitable for repair or for recycling.

Loading / Unloading station – the assembly procedure starts with the loading station. A first component is placed on an assembly pallet which will be transported through the system by a mobile robot until all assembly operations have been completed. If the assembled product satisfied all the quality control checks, it is transported to the unloading station as a finished product. If the quality level is not satisfactory and the repair procedure was not successful, the product is scraped or recycled for parts if possible.

Packing station – assembled product is unloaded, packed and sent for customer delivery.

3.4 BAS Hybrid control structure

As already stated, main BAS layout is organized through a control and an execution layer. Each of those layers uses a specific control strategy. The control layer is based on subordination and the execution layer is based on self-organization, as shown in Fig. 3.3.

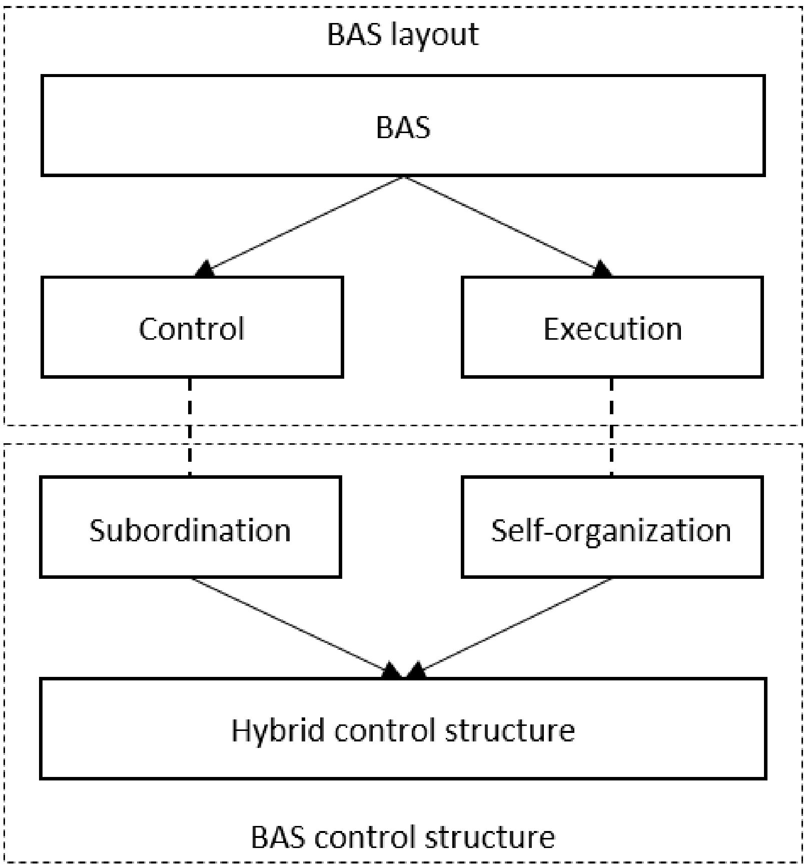


Fig. 3.3. BAS layout and control

BAS has a hybrid control structure because it combines two opposite principles: hierarchy and heterarchy. (Katalinic et al., 2012). An overview of hierarchy, heterarchy and hybrid control approaches are shown in Fig. 3.4.

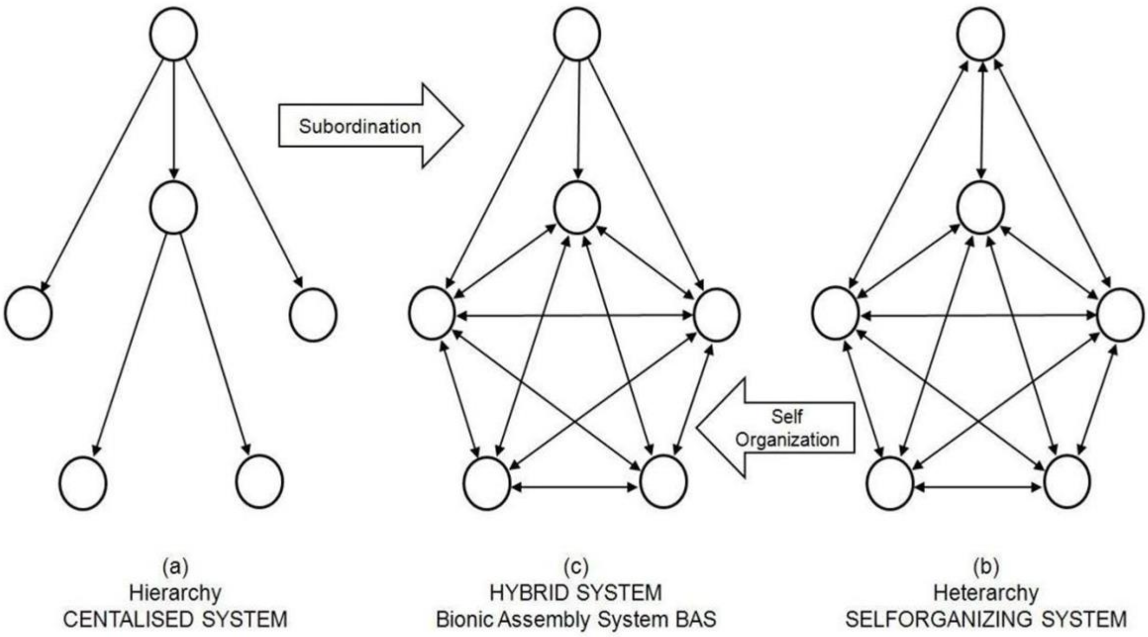


Fig. 3.4. Hybrid control system (Katalinic et al., 2012)

Hierarchy is defined through subordination. It uses the top-down control structure. There is one source of commands which represents the top level. This approach is commonly used in highly automated production systems, where each step is precisely controlled by a central computer.

Heterarchy is defined through self-organization. It uses the everyone – to – everyone principle. All units are equal and the responsibility is decentralised. There is no apparent source of commands but nevertheless, the common goals are completed (instinct, deployment of basic rules). This approach, by itself is currently not used in production systems as it is very difficult to achieve global factory goals.

Hybrid system aims to combine the decentralised control simplicity and robustness of self-organization with the goal oriented subordination. The main problem of introducing self-organization in the context of assembly systems is the conflict between non-compatible top – down concepts of orders at the factory level and self-organizing nature of the execution level.

3.4.1 BAS hybrid control system elements

The complete overview of the BAS hybrid control system is shown in Fig. 3.5. It shows the individual elements and their interactions within the entire system as well as the combination of the subordinating and self-organizing control principles. BAS hybrid control system elements are:

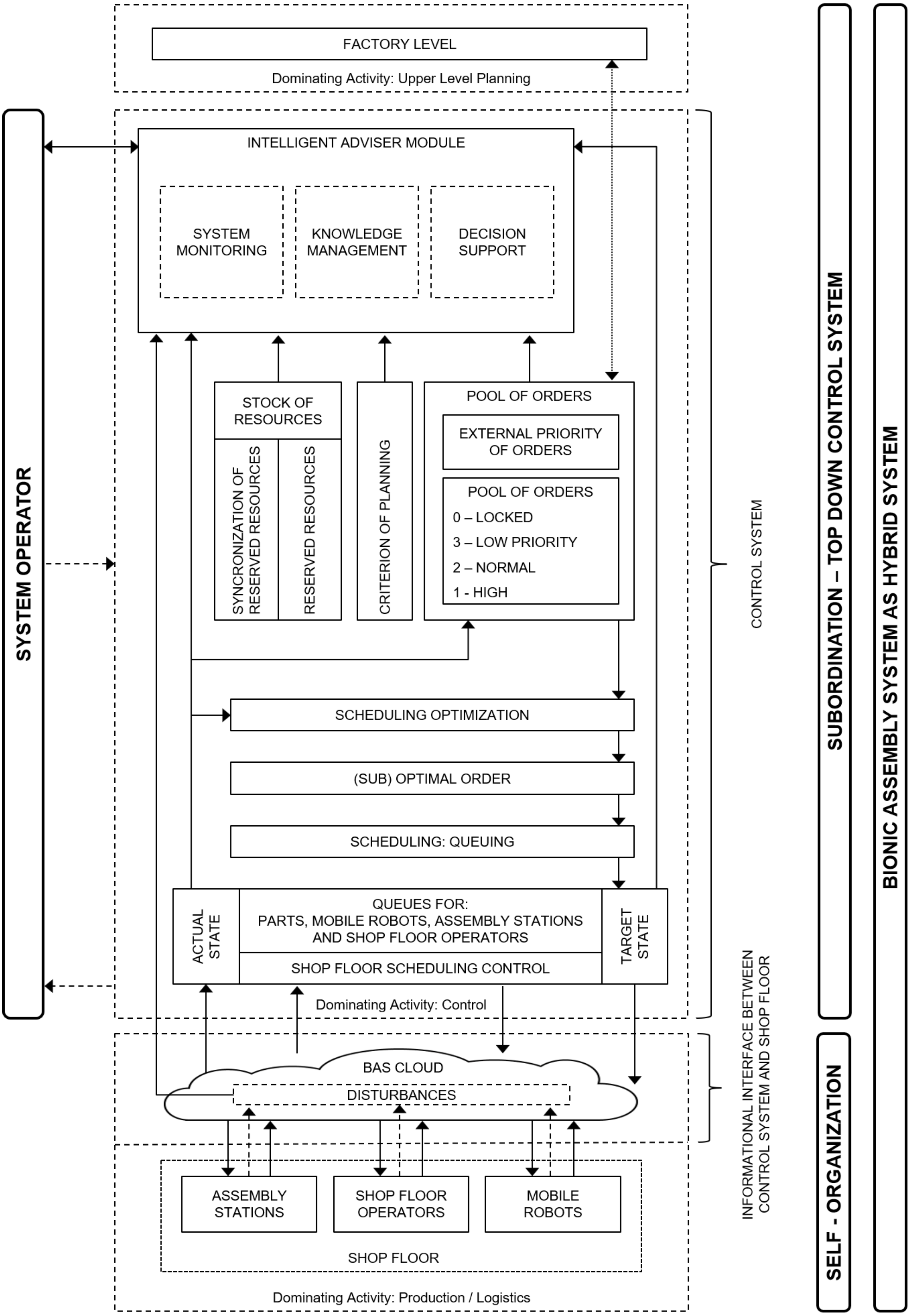


Fig. 3.5. BAS hybrid control structure

Factory level – Represents the highest level of planning for the entire assembly system. It is used to determine all BAS activities which should be completed. A long-term production strategy for the entire system is set at this level of planning. It defines what products in which quantity and by when need to be assembled. These production plans can be defined in yearly, monthly, weekly or daily timeframes. Planning goals are set according to the conditions of the system, desired results and methods of task completion. Very often, these goals can be in conflict. In such a case, a compromise solution is needed. There is a large number of goals that can be set (Katalinic, 1990). Some of them can include:

- Maximum workload during production
- Minimal storage period
- Minimal number of uncompleted products
- Deadline compliance
- Maximum production
- Maximum economy
- Shortest duration of working cycles…

There are two main tasks during planning in BAS:

1. Production planning determines all production activities which should be completed during a set timeframe (Year, month, week…).
2. These production activities within that period have to be set in an optimal sequence, were the main goal is to achieve the highest possible efficiency according to system states and priority orders.

Pool of orders –orders are coming from the factory level. One order is defined with the customer name, delivery deadline, type and number of products. All the orders are stored in pool of orders with their levels of relative importance to each other. This level corresponds with the urgency to complete an order. A priority system is introduced as a method to expel the finished product from the system. There are several levels of priority:

- Priority level 0 – products with the priority level 0 are locked.
- Priority level 1 – products with the priority level 1 have the highest priority. They have an advantage in relation to other products with lower priority.
- Priority level 2 – products with the priority level 2 have an advantage in relation to other products with lower priority, but not over products with higher priority.
- Priority level 3 – products with the priority level 3 have the lowest priority. All other products with higher priority have an advantage.

In BAS, the execution level (shop floor) is based on self-organization. In such an environment, there needs to be a mechanism which ensures that the products with the highest urgency will be completed first and will be processed out of the system. Instead of controlling every unit on the shop floor, the priorities ensure that the "*collective*" is aware of the factory goals and completes the tasks according to them.

Stock of resources –the primary function of this module is to track the status of all system resources which are necessary for the execution of working scenarios. These can be hardware (assembly stations, mobile robots, pallets, components, quality control equipment, tools, operational materials…), software (memory, CPU) or human resources (operators).

During the formation of the assembly order, the stock of resources is checked if everything required is available. If yes, the resources can be reserved. There are two main types of resources: consumable and non-consumable. Consumable resources (components, fluids, various materials…) are used only once to complete an operation and their stock is depleting over time. Non-consumable resources (robots, stations…) are reserved with start / end times. This gives an overview of occupancy over a period of time which is used to complete the planning of future assembly scenarios. If the level of a certain resource is getting low or the current stock of resources is not enough to complete an order, the module reports it.

Disturbances are always present during the execution of working scenarios. These disturbances cause a difference between the planned and executed operations. As a result, the resources will be available later or earlier than planned. The synchronisation of resources submodule has to compensate for these differences and reflect the actual state of system resources availability.

Criterion of Planning – criterion of planning module is used to determine a most suitable strategy for completing working scenarios (Katalinic, 1997). It operates in combination with the pool of orders and stock of resources modules as well as receiving feedback information about the actual status of the system as shown in Fig. 3.6. Based on the input information, a system order is formed as well as appropriate strategies for completing them.

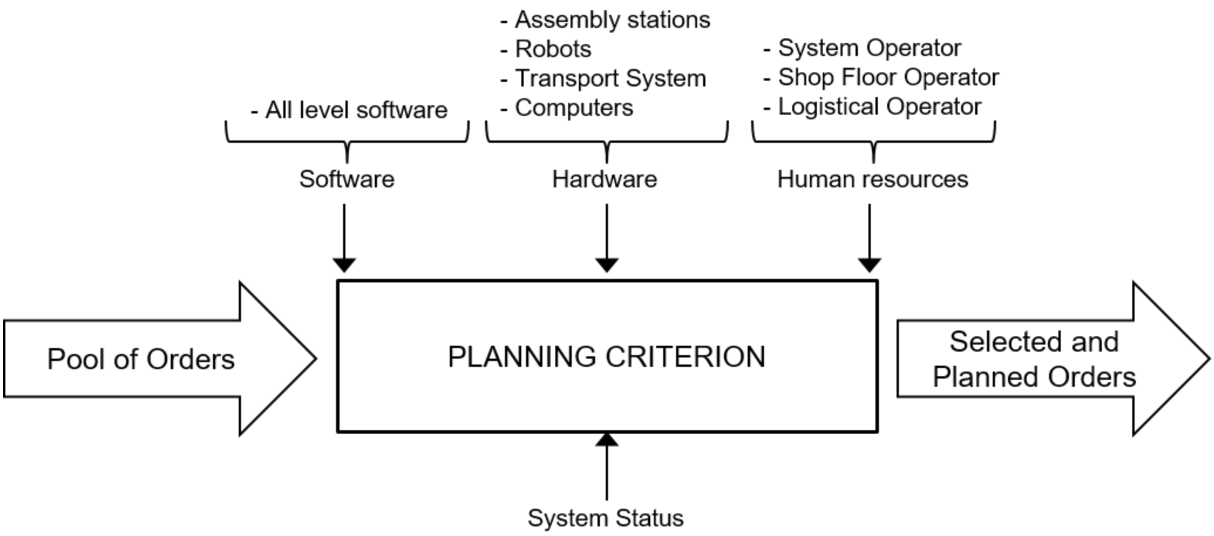


Fig. 3.6. Criterion of planning

Scheduling and system orders – The collection of customer orders starts at the factory level. All customer orders are combined in order to define the most optimal method of assembly. The most optimal method is achieved by producing the highest number of products within a set period of time with respect to customer deadlines, system resources and abilities. The result of this planning is called a system order (Katalinic et al., 2012). The system order contains information regarding the products (product type and their volume) and the urgency of their completion within a specified period of time (priorities). A system order is composed from all unlocked orders from the pool of orders.

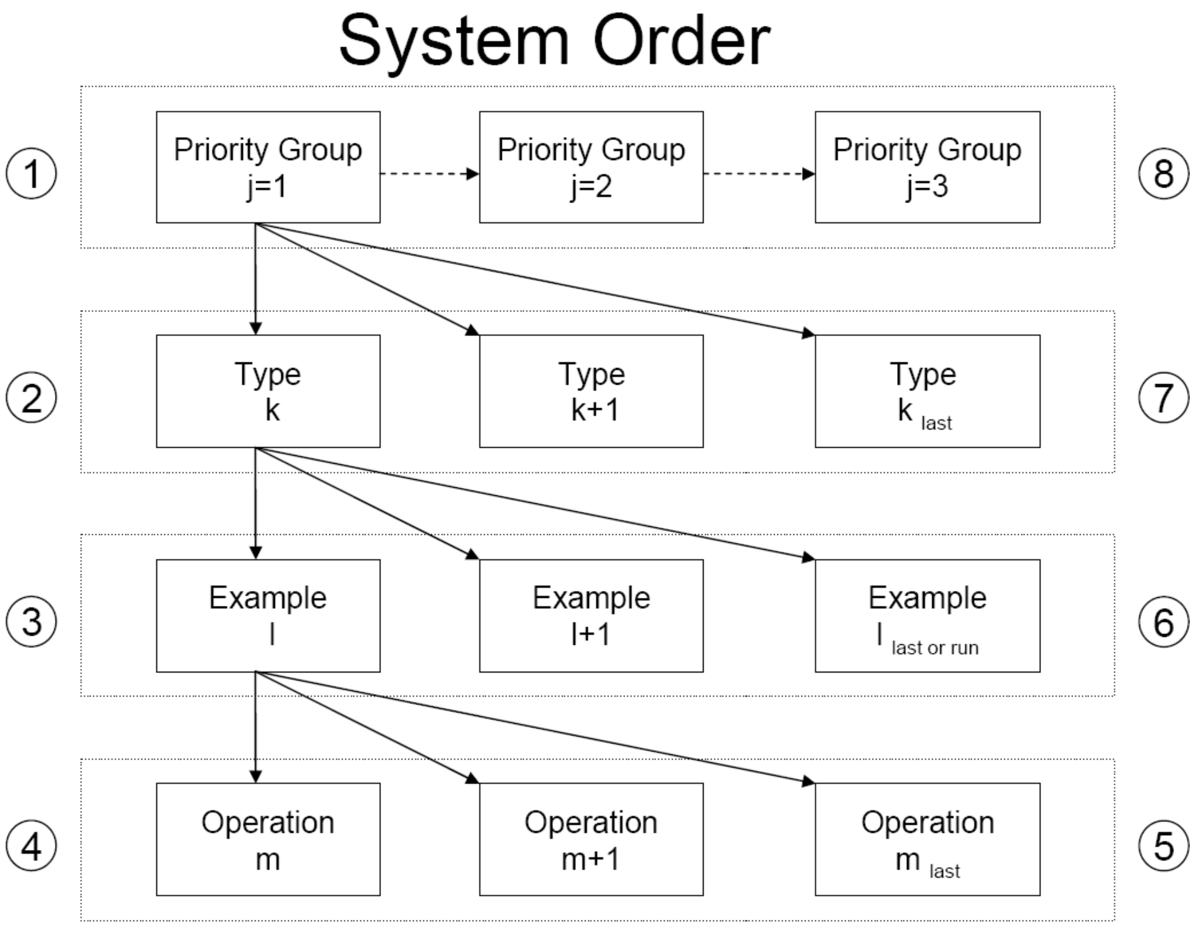


Fig. 3.7. Execution of a system order

Fig. 3.7. shows the execution of the system orders with the following points (Katalinic et al., 2012):

- 1 – the execution of BAS system orders starts with the highest priority group.
- 2 – the first product type from the highest priority group is selected. One assembly order means to assemble one run of product.
- 3 – the first product piece from the first product type will be assembled.

- 4 – the first operation of that product is being completed on the shop floor, where the main elements (mobile robots, assembly stations and operators) are executing tasks under self-organization principles.
- 5 – procedures 3 and 4 are repeating until the last piece of the run is assembled.
- 6 – the procedure is repeating for the next product type in the priority group.
- 7 – when the last product type from a priority group is assembled, the whole procedure from step 2 to 6 is repeated for the next priority group.
- 8 – system order is completed when the last piece in the run of the last product type in the lowest priority group is assembled.

The main task of the scheduling optimization module is to identify the most suitable order from the pool of BAS orders, taking into account the target scenario, criterion of planning, actual state of BAS and the state of system resources within the specified time of execution. The result of the scheduling optimization is a (sub)optimal order. This order can be used for simulation or for scheduled planning.

In case of a simulation, the order is a basis for a virtual scenario. A simulation model helps to compare different execution strategies through assumed assembly conditions for a defined period of time. Real assembly conditions cannot be predicted, described and defined in advance. For this reason, the simulation has a limited validity. The advantage of using a simulation is the possibility to perform tests and investigations in the development phase when the real system still does not exist.

In case of scheduled planning, a working scenario of BAS is created. Scheduled planning provides data which forms the queues. Queues determine the order and sequence of pieces, in which different products will be assembled.

Actual / Target state – In the real, unpredictable and random world it is normal that disturbances occur during the assembly process. These disturbances (assembly station failures, breaking of tools, robot shut downs…) cause deviations from the planned schedule. The difference between planned and realized activities is measured through the following two modules:

- Target state is the result of the subordinating control system planning activities. It represents what should happen under normal conditions during the execution of working scenarios.
- Actual state is the result of the self-organizing shop floor execution activities. It represents what has happened under real and unpredictable conditions during the realization of the planned activities.

As shown in Fig. 3.8., the system needs to achieve balance between the actual and the target state. There needs to be a harmony between subordination and self-organization. As soon as the difference starts increasing, it implies that there is a problem either during planning (resources are not synchronised, inadequate planning strategies…) or during the execution activities (technical errors, wrong tools, wrong processes…).

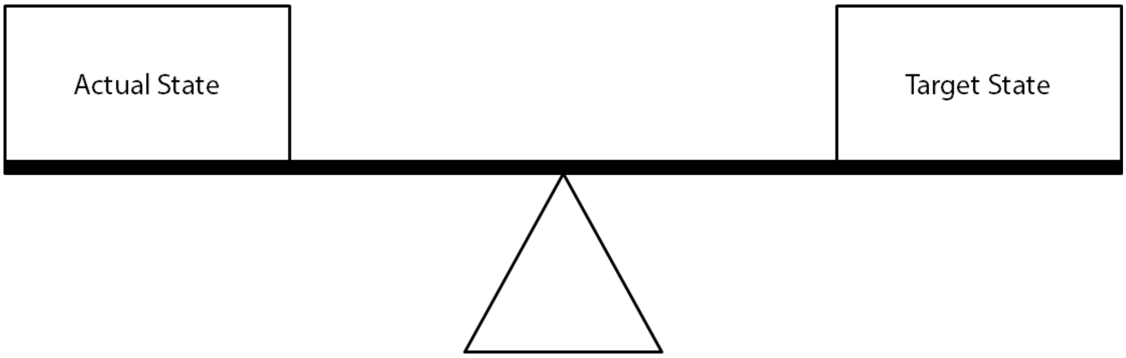


Fig. 3.8. Balance between the actual and target state

BAS Cloud – shop floor elements (assembly stations, operators, mobile robots) distribute and perform tasks. There is no central source of commands for each individual unit. They interoperate and function as a self-organizing system. In order to achieve this level of task distribution, the shop floor elements need to communicate with each other. There can be a large number of active units at the same time. If the communication is performed through "everyone to everyone" principle, the complexity of information exchange increases progressively. Additionally, there needs to be a simple and effective connection between the subordinating and self-organizing parts of the system.

The implementation of Bionic Assembly System Cloud (BAS Cloud) into the BAS hybrid control structure introduces a standardized communication protocol (Zharova, Elin, & Panfilov, 2017). BAS Cloud is an informational interface between the control system and the shop floor as shown in Fig. 3.9.

There are two main communication channels in BAS: vertical and horizontal. Vertical communication takes place between the subordinate elements of the control system and is completed with the BAS cloud interface. The information flow from the subordinating control system to the cloud is defined as a vertical upload and from the cloud back to the subordinating control system as a vertical download. Horizontal communication takes place between the self-organizing elements of the shop floor and is completed with the BAS cloud interface. The information flow from the shop floor elements to the cloud is defined as a horizontal upload and from the cloud to the shop floor elements as a horizontal download.

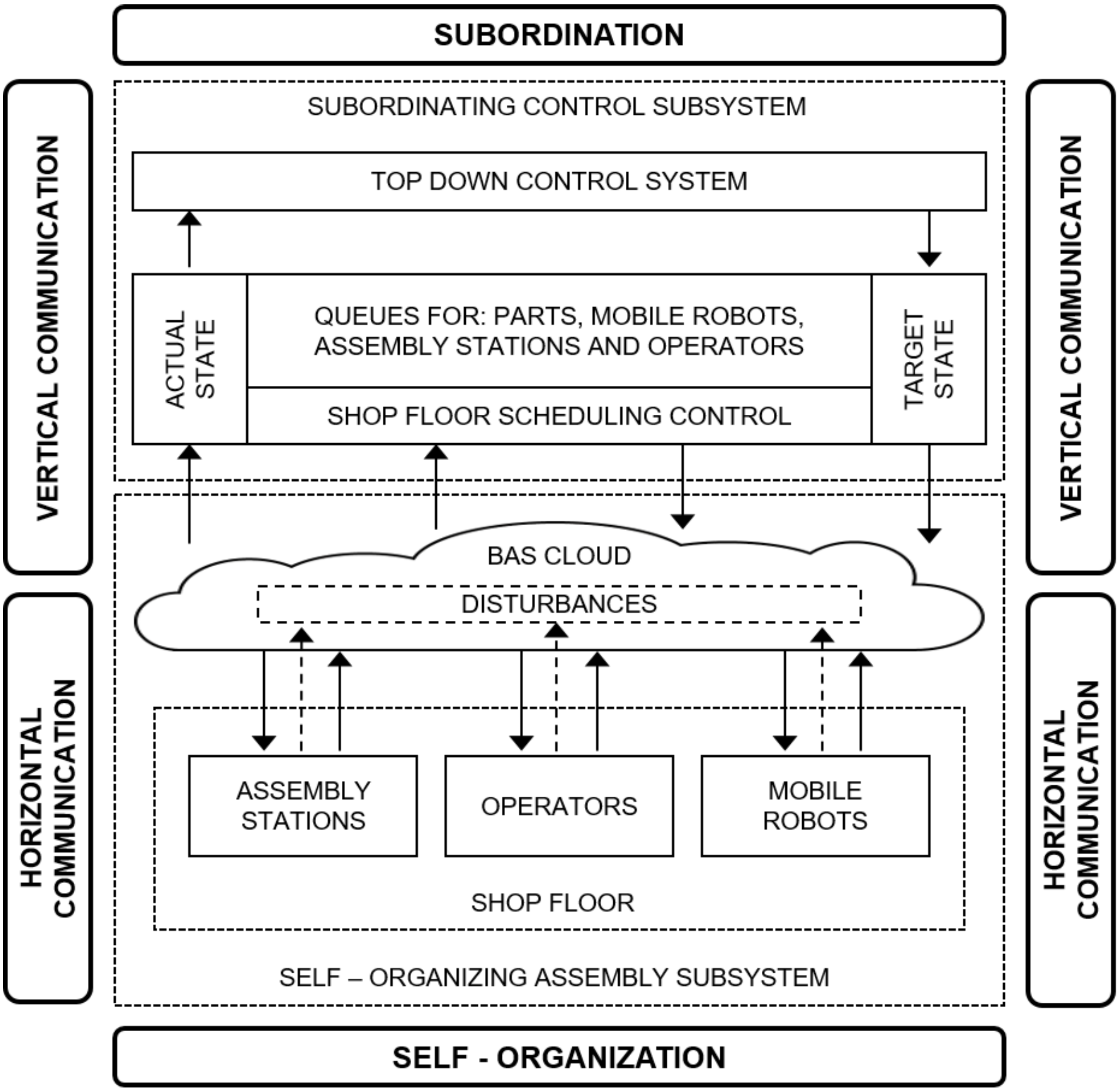


Fig. 3.9. BAS Cloud – Informational interface

BAS Cloud has the following functions:

- Connection of self-organizing and subordinating subsystems – the BAS Cloud is an informational interface between self-organization and subordination where horizontal and vertical communications are combined. It transfers the information between the planning level (target state or what needs to be completed) and the execution level (actual state or what is completed). Each of the shop floor elements uploads the status and time of a completed task. This allows to keep track of the system performance. When a disturbance (shutdowns, bad quality, errors…) occurs it is recorded in the "Disturbances" module.
- Connection of shop floor elements – BAS Cloud eliminates the need for a "everyone to everyone" communication. Using a direct "element to cloud" principle, a more robust, simple and efficient horizontal communication is possible. Each unit horizontally uploads or downloads the data from or to the cloud which is essential for task distribution.

- Data storage – BAS Cloud is used to store two kinds of information: predefined and recorded. Predefined information includes technological data and specifications of all elements that are involved in the assembly process. This includes station operating data, dimensions, NC programs, tool data, operating materials, product assembly instructions etc. Recorded information includes all data which is generated during the execution of working scenarios. This data is used for analysis and optimization.

Intelligent adviser module and the system operator – the interoperability between the system operator and the intelligent adviser module as well as their specific functions are described in more detail in further chapters.

3.5 BAS reconfigurations

One of the defining BAS characteristics is its ability to reconfigure and adapt to internal and external disturbances. The number of active shop floor elements (assembly stations, mobile robots, operators) constantly changes. It can increase (automatic mobile robot activation, new stations introduced in the system, additional operators…), or decrease (station malfunction, mobile robot low battery, operators missing…). Whatever the situation is, the system needs to be flexible and robust. If one unit fails, the execution of working scenarios has to continue. If a new unit becomes available, the workload needs to be distributed. In addition, there is a possibility for assembly station grouping and mobile robot trajectory optimization. All these characteristics help BAS to be a more efficient assembly system.

BAS has the ability to perform:

- Shop floor layout reconfiguration (physical repositioning of stations on the shop floor)
- Queue rearrangement (mobile robots' queue rearrangement according to available assembly stations)

3.5.1 Shop floor layout reconfiguration

Operators and mobile robots are movable as they can change their position according to the current task. Various stations can be movable, semi – movable and non-movable. Movable stations have the ability to change their position and rotation on the shop floor. Semi-movable stations cannot change their position but can rotate. Non-movable stations are fixed in their place and cannot rotate due to their dimensions, weight and operating demands.

In certain situations, during the normal execution of working scenarios and when there is a small number of active mobile robots, the system can complete layout reconfigurations on the shop floor, as shown in Fig. 3.10.

This reconfiguration is defined with the following characteristics:

- Movable assembly stations / change of position and orientation
- Optimization of mobile robot trajectories / distance reduction between stations / grouping

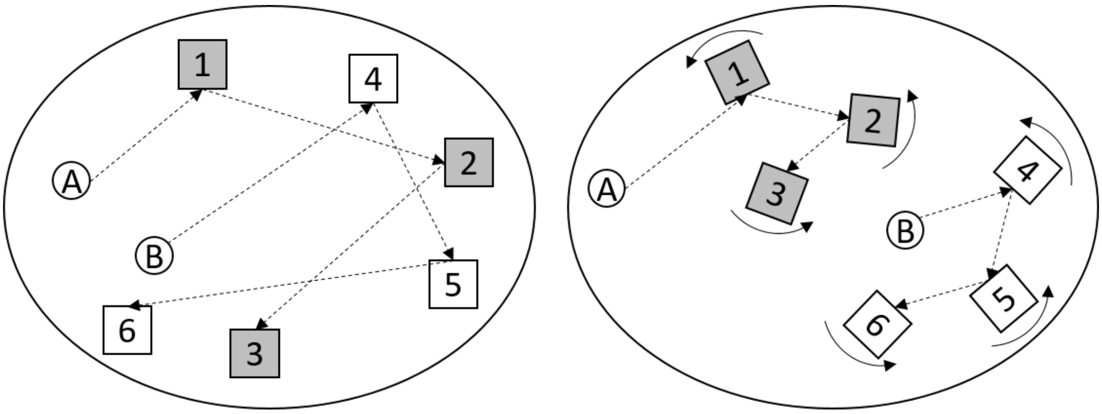


Fig. 3.10. Shop floor layout reconfiguration

Mobile robot carrying product A executes assembly orders on stations 1,2 and 3. Mobile robot carrying product B executes assembly orders on stations 4, 5 and 6. In this case, the mobile robots are traveling greater distances between stations and their trajectories are intersecting. After repositioning and rotating the stations, each mobile robot travels in its own group. The travel times have been reduced due to shorter paths and reduced number of collision avoidances.

3.5.2 Queue rearrangement

During assembly, it is normal that a queue of robots is forming in front of an assembly station S_x as shown in Fig. 3.11. In this example station S_x has the capability to perform i-th operation (O_i) on the m-th product (P_m), j-th operation (O_j) on the n-th product (P_n) and k-th operation (O_k) on the l-th product (P_l). A queue of mobile robots is formed according to the order priority for the product they are carrying. It can be: priority group 1 (product P_m), priority group 2 (product P_n) and priority group 3 (product P_l). Mobile robots from the priority group 1 are the most urgent and have the advantage. Priority group 2 has advantage in comparison with group 3 and group 3 is least urgent. Each priority group can have multiple mobile robots (1,2…middle…last).

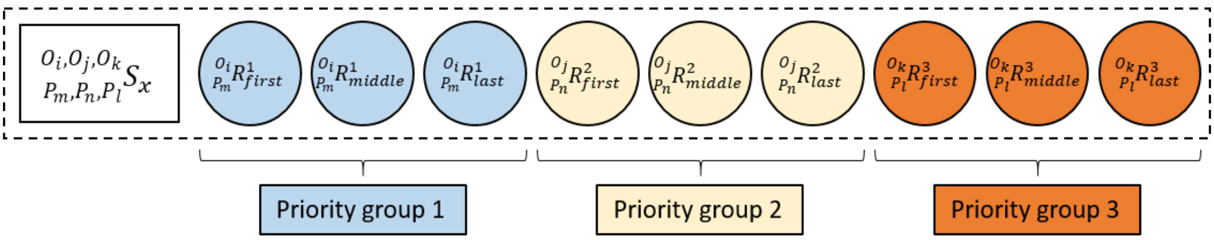


Fig. 3.11. Mobile robots queuing

Change of number of working stations can happen when:

- A current station becomes unavailable
- A new station becomes available

3.5.2.1 Queue rearrangement after a station becomes unavailable

Fig. 3.12. shows the process of queue rearrangement in case of a station failure. Section A shows two assembly stations S_1 and S_2, each with a formed queue of robots grouped according to the priority. Section B shows the failure of station S_1 and the regrouping paths of mobile robots. Each robot moves to the appropriate priority group in front of the functional station. Section C shows the completed rearranged queue of mobile robots in front of station S_2. This ability demonstrates BAS robustness towards disturbances. It allows to continue with the assembly through flexibility and task redistribution.

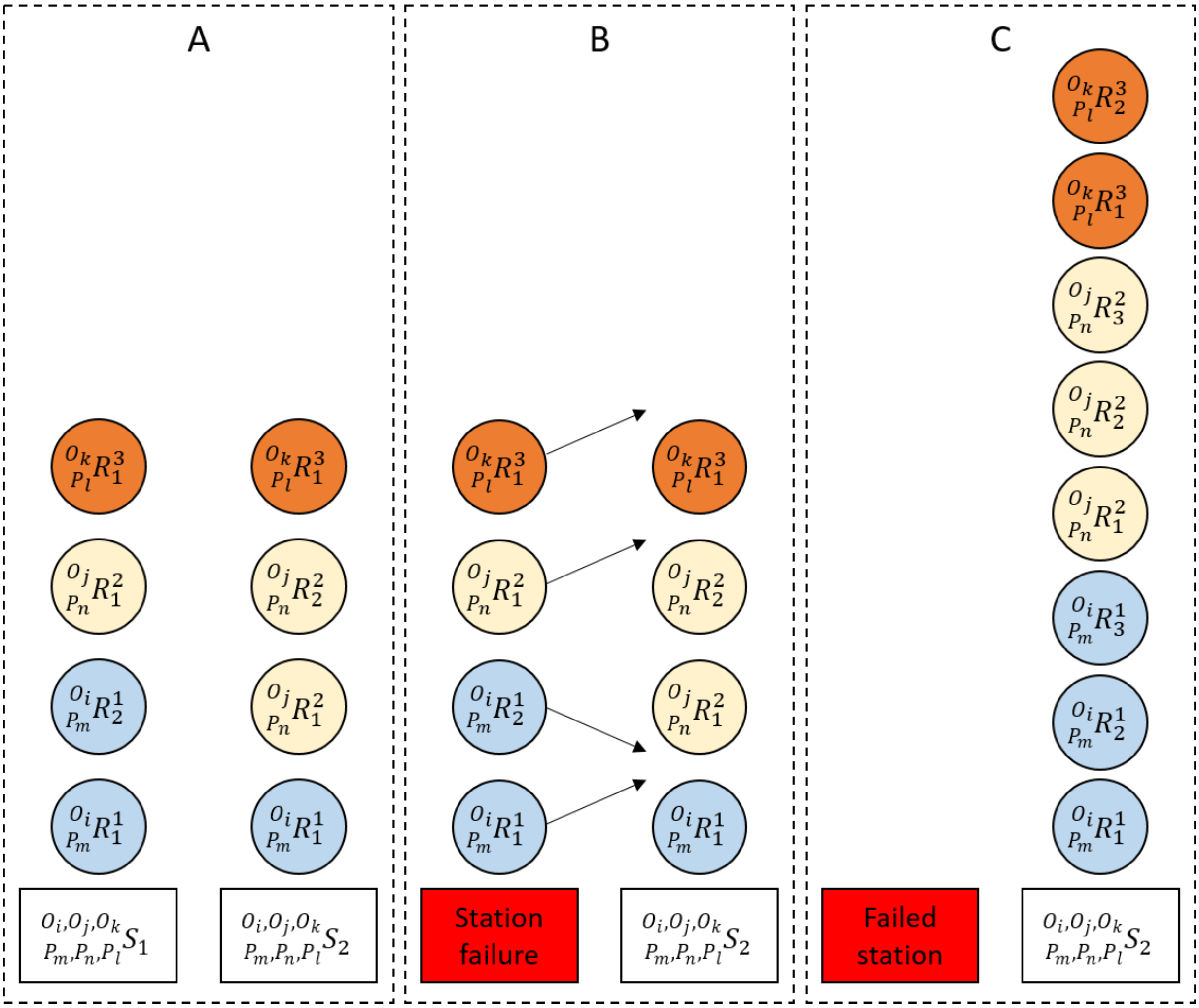


Fig. 3.12. Queue rearrangement– Failed station

3.5.2.2 Queue rearrangement after a new station becomes available

Fig. 3.13. shows the process of rearrangement in case when a new station becomes available. Section A shows a long queue of mobile robots waiting in front of a single operating station S_1. Section B shows the activation of a new assembly station S_2 and the regrouping paths of mobile robots.

As stated, each priority group can have multiple mobile robots (1,2…middle…last). Each priority group is divided to: S_1 group which is remaining in its original position (R_1, R_2,…R_{middle}) and the new S_2 group which is moving ($R_{middle+1}$, $R_{middle+2}$,…R_{last}). Section C shows the 2 new rearranged queues of mobile robots in front of stations S_1. and S_2.

The described queue rearrangements demonstrated BAS ability to dynamically adapt to different working scenarios. This includes different workloads and variable operability of the equipment.

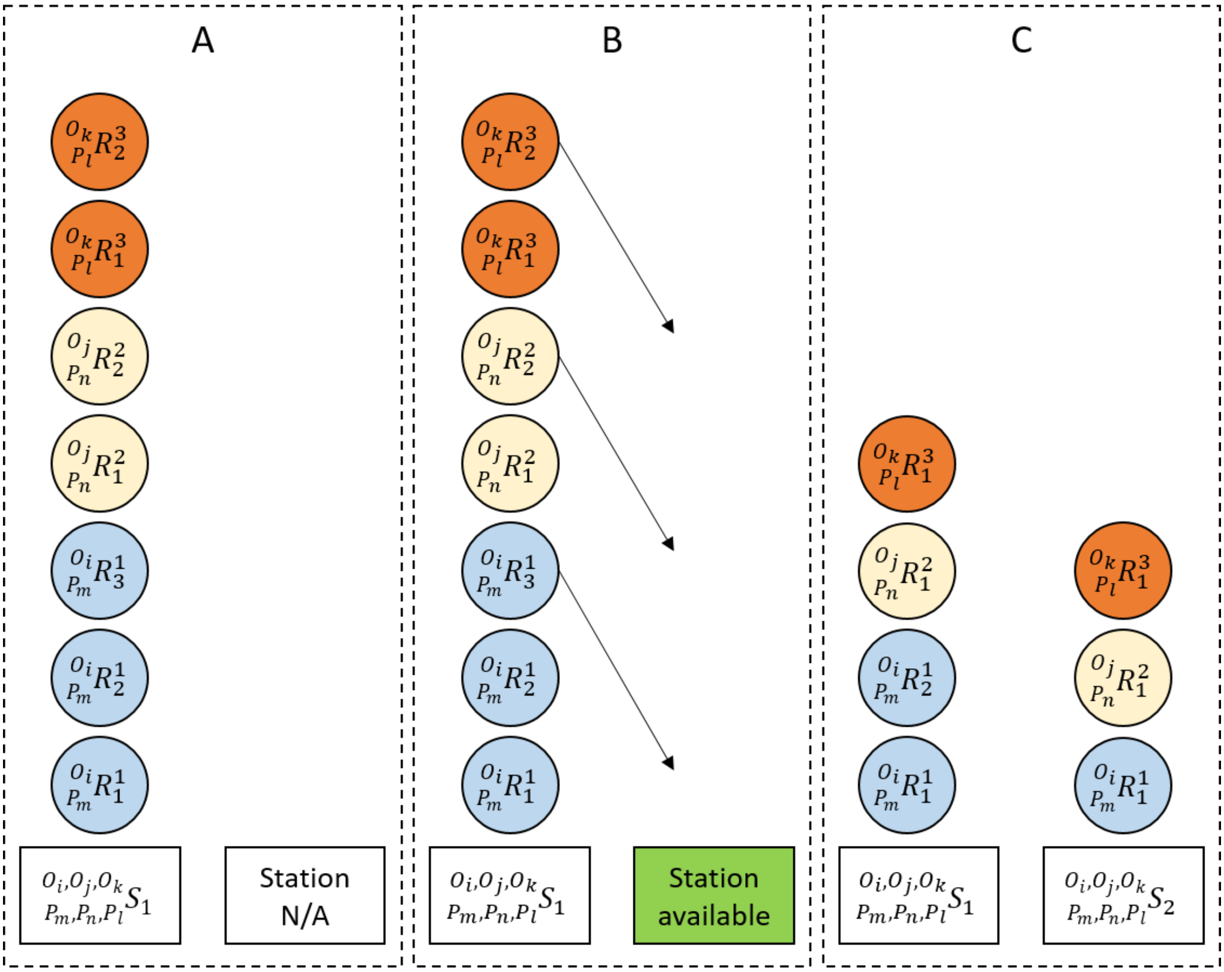


Fig. 3.13. Queue rearrangement – New station available

3.6 BAS working scenarios

3.6.1 BAS normal working scenario

BAS normal working scenarios are realized with the uninterrupted execution of all activities which are needed to assemble a continuous stream of products. Continuous stream of products is made with assembly orders which are formed by the subordinating control subsystem. These orders are vertically uploaded to BAS cloud. All standby mobile robots check if there are any available orders on the cloud.

When the robot finds and horizontally downloads an order, it takes the pallet from pool of pallets and is ready to start the assembly. During the assembly procedure, the robot can have alternative routes. This happens when one assembly operation can be completed by different assembly stations or shop floor operators.

During the selection of the most suitable station for the next assembly operation, the robot follows the smallest time resistance criteria in order to complete such operation in the shortest time of the next assembly station. That means that from all suitable stations, robot choses the one with the smallest assembly time which represents the sum of the transport time, waiting time and the operation time.

Therefore, the entire assembly process is taking place on the shop floor and follows the basic principles of self-organization where the main participants are the mobile robots, assembly stations and shop floor operators.

However, BAS working scenarios are not always realized with the uninterrupted execution of all activities which are needed to assemble a continuous stream of products. BAS represent the next generation of modern, hybrid assembly systems. It combines the self-organizing and subordinating control structures. As a result, specific BAS scenarios can occur. The following 2 described scenarios represent just some of them.

3.6.2 BAS specific working scenario 1

Restricting mobile robot movement

Let us suppose there are two stations which can perform the same types of operations. Here, stations S_1 and S_2 have the capability to perform i-th operation (O_i) on the m-th product (P_m), j-th operation (O_j) on the n-th product (P_n) and k-th operation (O_k) on the l-th product (P_l).

All mobile robots constantly search for the most suitable assembly station for the next assembly operation. However, each station has different operation times. As stated, robot choses an assembly station with the smallest assembly time which represents the sum of the transport time, waiting time and the operation time.

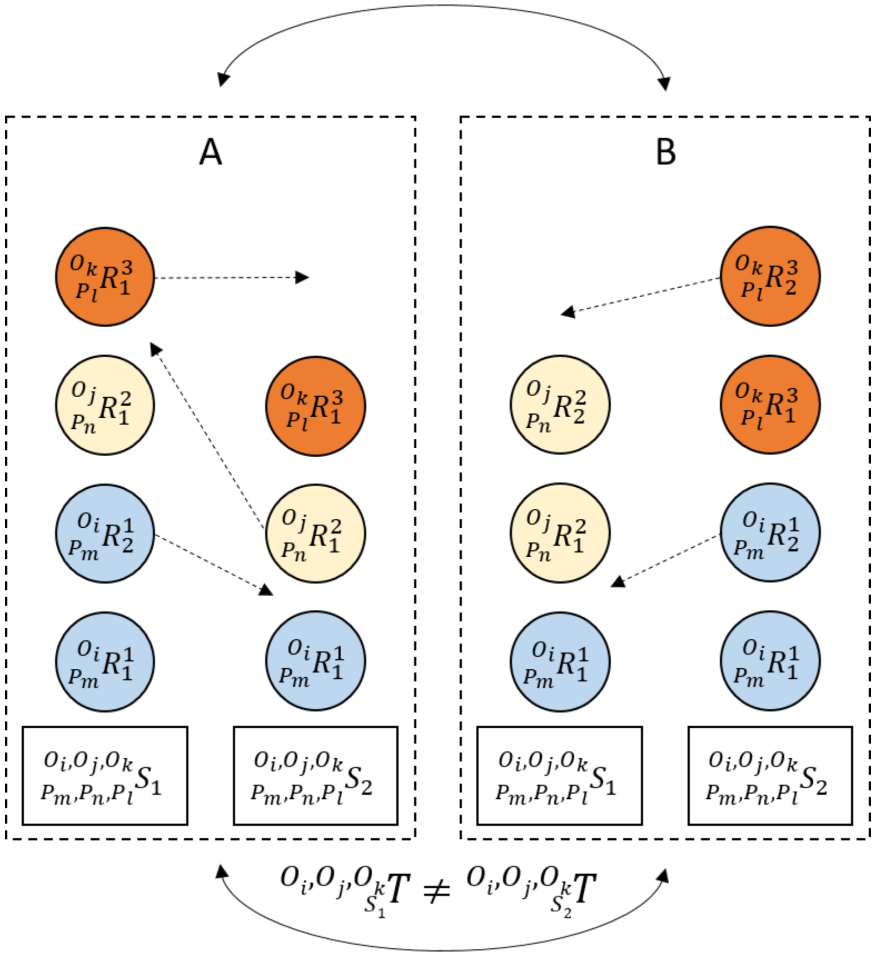


Fig. 3.14. BAS specific scenario 1 – Mobile robot movement

As a result, it can happen that the mobile robots constantly move their position from one queue to another, back and forth, based on the queue rearrangement logic as shown in Fig. 3.14. Every robot makes and uploads its decisions to the BAS cloud. This way a decision from one mobile robot affects the decisions from all other active mobile robot. To avoid confusion and to prevent more robots making a decision in the same time, decisions are made in periodic cycles according to the prespecified schedule. This schedule is defined according to the priority group of the robot and the time that this robot reserved its assembly order / picked up the pallet.

Every mobile robot makes a decision where to go next based on their own criteria. This includes number of active mobile robots, number of available stations, priority, workload, assembly times, status of product in assembly etc. This behaviour can emerge when there are multiple available assembly stations for a specific operation and the technological possibility to perform this operation in the required stage of assembly (assembly preconditions are satisfied).

If such a behaviour starts to be contra productive, it is necessary to reduce it or eliminate it altogether. There needs to be a mechanism which restricts the mobile robots from moving back and forth between stations for the same operation.

This can be achieved by changing the waiting time threshold value. This value represents the minimum time each mobile robot has to wait before it is allowed to change its position. In other words, changing the threshold value can "calm down" the collective behaviour of mobile robots. However, this value needs to be balanced. On one side, it is set too low, the mobile robots will constantly change their positions, and on the other side, if it is set too high, any free assembly station will not be utilized. This means that the self-organization capabilities are removed. Both extremes need to be avoided.

3.6.3 BAS specific working scenario 2

Last product in the run is damaged

In this specific working scenario an assembly of two types of product is taking place as shown in Fig. 3.15. The two product types P_m and P_n are assembled on stations S_1 till S_{last}. Here each station S has the capability to perform i-th operation (O_i) on the m-th product (P_m) and the j-th operation (O_j) on the n-th product (P_n).

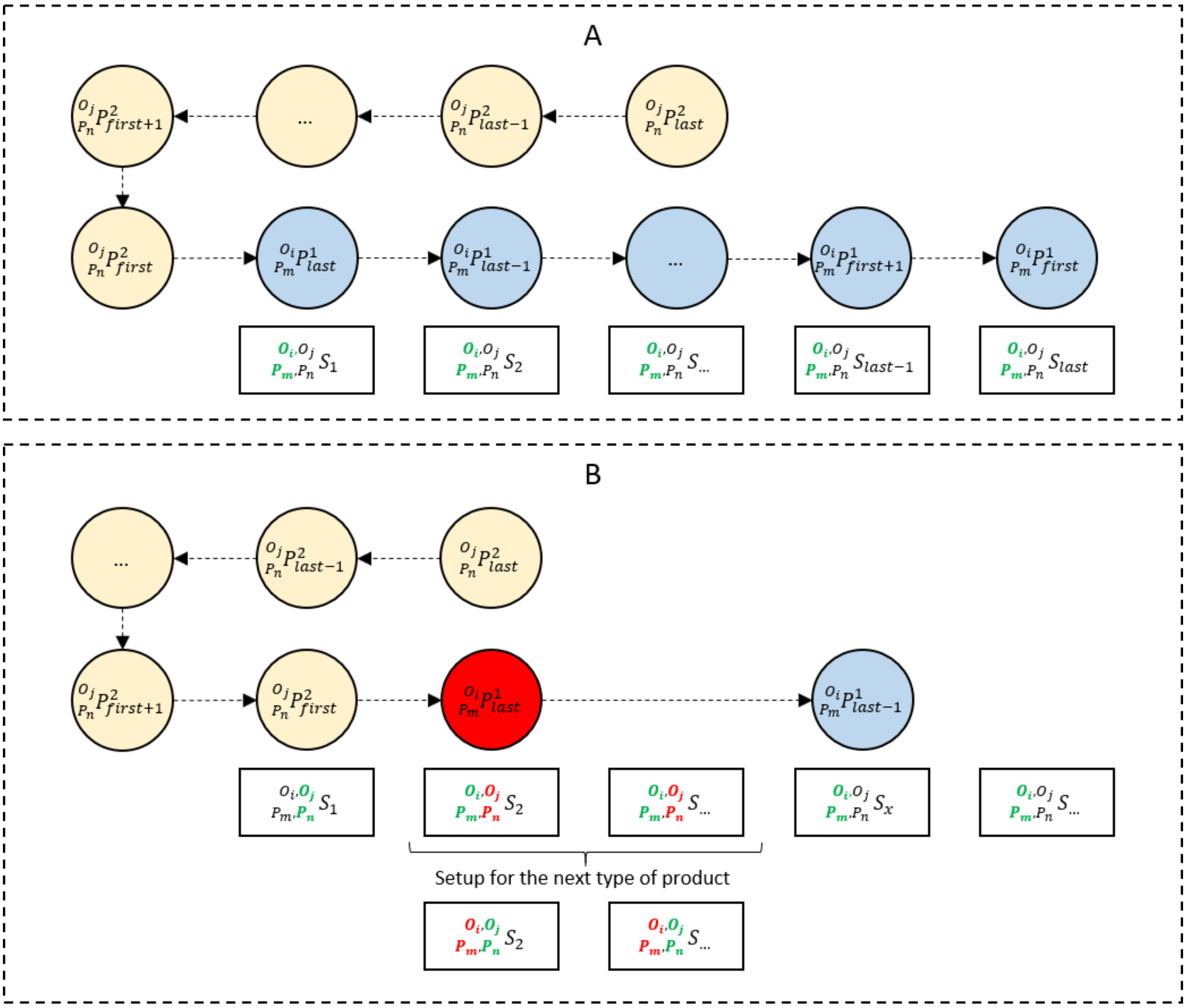


Fig. 3.15. BAS specific scenario 2 – Last product in the run is damaged

Product P_m has a higher priority (1) than the product P_n (2). That means that the last piece of P_m has to pass the first station S_1 before the first piece of P_n can start. This way, the system will clean the higher priority products from the shop floor first.

Let us suppose that the last piece of the product P_m run has been damaged at the second assembly station S_2. If this happens, the first question is: can the damaged piece be repaired before the deadline? If yes, it is sent to a repair station until it passes the control. After that it continues to visit the next required assembly stations.

However, if the repair will take too long, that means that the damaged last piece needs to be cancelled and the mobile robot takes it out of the system (scrap or recycling).

The entire delay has created a gap between the last and the second to last piece. If this happens, it is necessary to set up the assembly stations for the next type of product P_n. The set up includes all the stations from the station where the last piece has been damaged, (in this case $S_{2)}$, up to the station before the one where the second to last piece of product P_m is. Doing this ensures that in case of disturbances, or damaged pieces, lost assembly time is minimized.

3.7 Summary

This chapter presented BAS and its control structure, layout, key elements, reconfiguration abilities working scenarios and characteristics. BAS as such represents one development direction of next generation of modern, hybrid assembly system. It is based on a biologically inspired principle of self-organisation. Such a system is able to deal with a highly variable environment through adaptation, evolution and learning. As a result, a new evolution trend in assembly systems took place. It is focused around the integration of humans and technology.

The following chapter will focus around BAS as a human centric assembly system, which includes human operator tasks within BAS as well as challenges that are involved.

Chapter 4

System Operator

Modern product development trends are defined with higher product complexity, increased variety and decreased lifetime (Hobday, 1998). These trends introduce new challenges for traditional assembly systems. BAS represents a general evolution of modern assembly systems which answer to these new challenges.

The roles of human workers have changed as well during the development of assembly systems as described in chapter 2. Their roles shifted from highly skilled craftsman to low skilled labour which performed a single task in a repetitive manner. Doing this, workers were equated to a part of the system which was easily replaced either by another worker or with an automated assembly station. This gradually led to fully automated and centrally controlled systems. However, according to research and analysis (Whitney, 2004) and (Katalinic, 2004), this introduced additional complexity, costs, substantial preplanning and socio-economic problems (i.e. human unemployment). Because of these limitations a new evolution trend in assembly systems took place. It focused around the integration of humans and technology.

This chapter describes BAS as a human centric assembly system, roles of operators within BAS and challenges that are involved, limitations and advantages of humans as well as introducing concepts of artificial intelligence, knowledge, decision support and learning.

4.1 Human centric assembly systems

Production systems went through several key industrial revolutions. First industrial revolution (end of 18th century) was defined through mechanization, the second one (beginning of 20th century) through industrialization and the third one (from 1970's) through automation.

The fourth industrial revolution, Industry 4.0 or also referred as "*Internet of Things*" (IoT) is defined as an integration of social, ecological and economical systems. The result of this integration is a sustainable system based on computer integration, networking, adaptation, self-organization, autonomy, self-teaching and knowledge (Post, 2014).

These qualities ensure that the system is maintenance friendly, robust to disturbances, energy efficient and adaptable in real time.

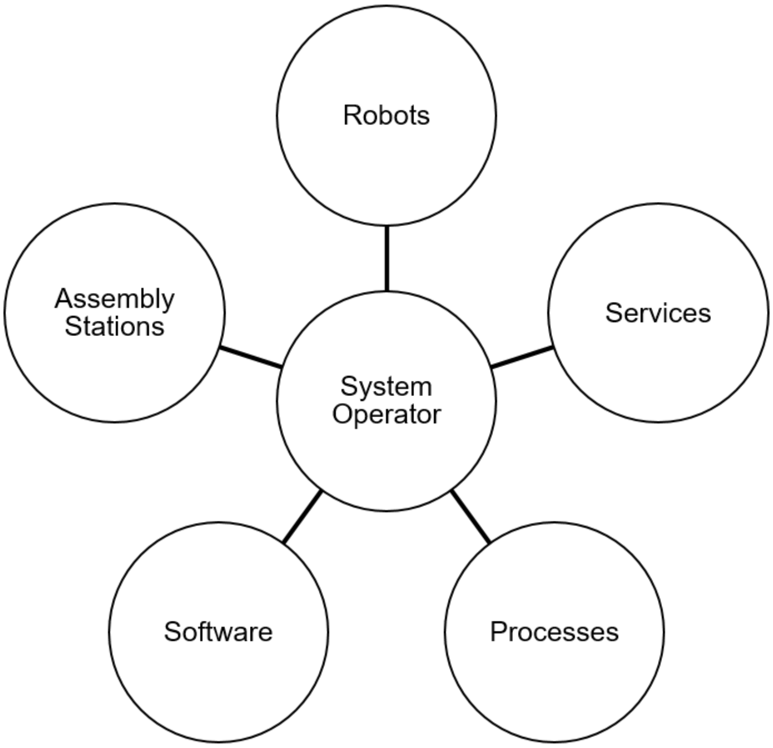


Fig. 4.1. Human centric assembly system (Post, 2014)

This new generation of assembly systems is grounded on the cooperation and interaction between man and assembly station and is known as a human centric assembly system. BAS is one example of such a system as shown in Fig. 4.1., where the role of the worker is redefined. People are occupying the same space, cooperating, interacting and supervising other BAS elements: robots, assembly stations, control software as well as supplementary services and processes. The main purpose of this new man-machine paradigm is to combine human intelligence and problem solving skills with the precision and repetitive capabilities of technology.

4.2 Activities of workers in BAS

Self-organizing assembly environment presents different requirements from the workers when compared to the traditional assembly systems such as an assembly line. In BAS shop floor, every worker, mobile robot or a station is an equal unit that achieves common goals and distributes tasks. Here, a shop floor worker has a subordinate position when compared to the upper control system supervised by the human, system operator.

Based on these tasks, worker activities within BAS are divided into two categories. They can be activities on the shop floor performed by the shop floor operators and activities in the control system performed by the system operator.

4.2.1 Shop floor operator activities

As previously described, the shop floor operators can perform a large variety of tasks. They are integrated in the execution of working scenarios with mobile robots and assembly stations. They represent very flexible “units” thanks to their problem solving skills, dexterity and intellect. For some types of operations, it is more economical or simpler that they are completed by a shop floor operator. Each station on the shop floor can be deployed as a fully automated, semi-automated or a completely manual work unit.

The shop floor operator activities are:

- Assembly – a shop floor operator can perform certain operations either manually, with the use of tools or by operating a specific assembly station.
- Quality control – inspection of an assembled product during or at the end of the assembly process. An operator can use a combination of tools (measuring devices, scanners…) and methods (visual, tactile, sound) in order to perform an inspection.
- Repair – repair activities can be performed on products as well as on the shop floor hardware (stations, mobile robots, transport systems…). If a product does not pass quality control, a worker inspects it and decides if the piece can be repaired or it should be recycled for parts. Repair tasks are most suitable for workers, because of their intellect, analysis and problem solving skills.
- Set ups – sometimes it is necessary to manually clean an assembly station, replace fasteners, replace jigs, replace work pallets, set up additional tools etc.
- Maintenance – regular maintenance of work pallets, ensuring the material flow, hardware (operating fluids, tools, moving parts…) and software (error status, bugs, drivers…).
- Supplementary activities - warehouse and inventory organization, logistics, cleaning…

During assembly, assembly station malfunctions or similar disturbances can happen. When such a case occurs, the shop floor operator has to eliminate any interruptions as soon as possible. He can accomplish this either by repairing the assembly station or in some cases by taking over the interrupted operation (if possible). It is very important that the shop floor operators are synchronised with other units and with the goals of the entire system. The shop floor operator’s behaviour in emergency situations is defined through instructions and guidelines set by the upper control system based on the proposal from the Intelligent Adviser Module.

Therefore, in case of a station malfunction the following rules are defined as (Nanasi, 1996):

1) The first free shop floor operator should eliminate the interruption without any delay.
2) In case that there are not any free operators available, the first operator who is not already eliminating an interruption, should stop his current activity and eliminate the new interruption.
3) In case that all operators are eliminating interruptions, the operator who was doing it the longest, should react.

4) This operator has to stop his current activity and eliminate the new interruption at the newly broken assembly station. If the station is damaged beyond repair and the operator cannot take over its operation, the operator stays at his current position. The system needs to be set up to avoid such a scenario.
5) If the operator stops with his current activity, he has to choose a malfunctioning assembly station which can be repaired in the shortest amount of time.
6) If there are no more awaiting or interrupted activities, the operator has to continue with his earlier stopped task. If overall, there are not any other activities, the operator is free until the next assembly operation.

4.2.2 System operator activities

The system operator is the main decision maker. He is a human expert who makes final decisions and his tasks include overview, control and planning. He needs to make sure that the assembly system completes the customer orders on time and with satisfactory quality.

The list of system operator activities includes:

- Verification of orders – each new order has to be checked if the necessary system resources and technological data are available. Sufficient resources need to be organised if they are missing or will not be enough to complete an order. The order is verified if everything will be available.
- Planning of working scenarios – allocation of order priorities based on the combination of different selection criteria. This represents the overall system strategy for order completion and is a very complex task. Long experience, detailed knowledge of the system and its components have influence on the choice and subsequently on the system performance.
- Technological data maintenance – in order for BAS system to properly operate, the technological data (assembly station and tools specifications, product-specific sequence of assembly, etc.) needs to be kept up to date.
- Perform system analysis – the system performance (component availability, productivity, adherence to customer deadlines…) is analysed through appropriate statistics, functions, parameters and methods of the analysis software. The operator has to interpret the results and make decisions.
- System status monitoring – the system operator keeps track of all system resources with the help of a specialized software that presents data using graphical methods. This way of presenting data speeds up and simplifies the recognition of a possible problem.
- Computer maintenance and supervision – includes all activities related to hardware and software: start, shutdown, data backup, software and hardware updates…

During the execution of working scenarios, the system operator has limited time to make real time decisions with an optimum balance between quality of decisions and time needed to reach them. His experience, knowledge and intuition should help him to reach quality decisions in good time. This ability directly influences the efficiency of the assembly system.

The activities of the system operator can be divided into several different phases: planning, realization of working scenarios and disturbances. The realization of working scenarios can either run according to plan (actual state is similar and within tolerances of the target state of the system) or not according to plan (the difference between the target state and the actual state is increasing which implies that there is something wrong).

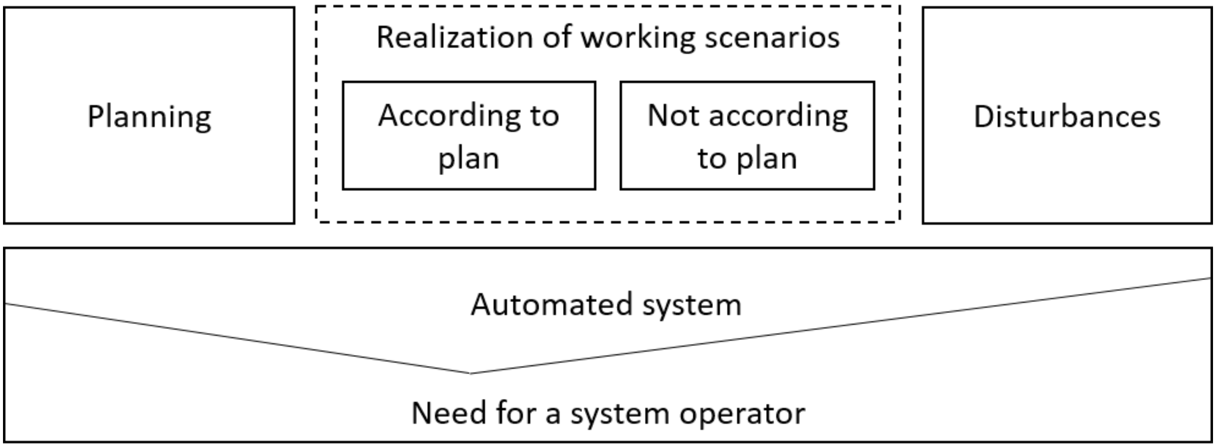


Fig. 4.2. Need for a system operator

System operator is not expected to constantly make all the decisions. The need for a system operator varies depending on the phase and is inverse to capabilities of the automated system as shown in Fig. 4.2. When the realization of working scenarios runs according to plan, the need for a system operator is very low. The system is able to automatically execute standard orders. However, in all other phases the need for a system operator is high:

- During the planning of working scenarios
- When the realization of working scenarios is not running according to plan
- When disturbances occur

4.3 System operator challenges

As the main decision maker, the system operator must have an overview of the entire system over a longer period of time. To reach a decision on how to proceed with the assembly, the operator has to know each individual state of different subsystems. However, there are several main challenges that the system operator faces during his working shift in BAS.

The main challenges for the system operator are:

- **System complexity** - BAS is a modern assembly system capable of assembling a large variety of products with high complexity within defined customer deadlines. As a result, the working scenarios are becoming more complex and the sophistication of the control structure is proportionally increasing (Goldratt, 1988). The system is modular and has a variable number of active assembly stations, mobile robots and shop floor operators. All subsystems need to be connected and synchronised. Such a system is very complex and presents a challenge for the system operator to run it efficiently.
- **Large amounts of information** - every assembly station, sensor and performance record create large amounts of data during their operation. This is used for analysis and optimization of the system performance. Other sources include energy and material consumption, specification sheets, customer and employee data, technological data etc. System operator can easily get confused and overwhelmed (Getty et al., 1995).
- **Distractions** - assembly systems are a highly dynamic environment where loud noises, audio visual notifications and lack of visual overview can cause distractions which influence the decision-making process.
- **Disturbances** - the execution of working scenarios starts with an order and finishes with an assembled product as shown on Fig. 4.3. During this execution, internal (hardware, software and human resources) and external (energy, material and information) resources are used. These resources represent a potential source of disturbances in form of hardware failures, software crashes, shop floor operators missing, energy outage, interrupted flow of material or incorrect information. As mentioned, the need for a system operator is very high when disturbances occur. He needs to react and make decisions which will minimize the difference between the target and actual state of the system. When disturbances occur, the main task is to develop alternative strategies as soon as possible.

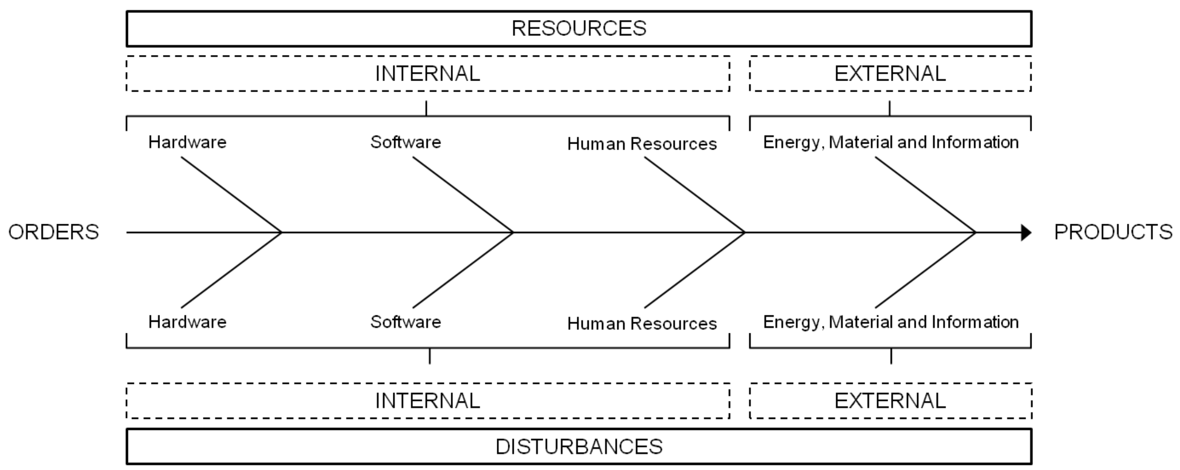


Fig. 4.3. Resources and disturbances in BAS

4.4 Artificial Intelligence

The system complexity, large amounts of data, distractions, disturbances and chaotic situations have a high impact on the capabilities of the system operator to run the entire system and perform his tasks. In addition to that, he as a human has limited physical and cognitive capacity to reach repetitive quality decisions over a long period of time.

The system operator needs a support system which will assist him in performing his tasks. Such a system needs to be capable of replicating human intelligence and thought processes. This is becoming possible thanks to the rapid development of information technology (IT) and the introduction of advanced computer methods and algorithms (NRC, 1999). Artificial Intelligence (AI) is the main characteristic of such support systems.

4.4.1 AI system characteristics

The main goal of Artificial Intelligence is to formalise human thought process and transfer it to a computer system. This system needs to be able to achieve similar performance as humans. In order to describe the AI characteristics, it is necessary to define the concept of "*Intelligence"*.

The following definitions of intelligence can be cited:

"Intelligence is the ability for an information processing system to adapt to its environment with insufficient knowledge and resources" (Troy, 1991).

"Intelligence is the ability or a capacity to recognize the correlation between a large number of factors in the shortest time." (Hartmann et al., 1990)

"Intelligence usually means the ability to solve hard problems." (Minsky, 1985)

"Intelligence means getting better over time." (Schank, 1991)

Human intelligence is the main inspiration in the development of AI. In order to replicate it within a computer system, the main characteristics need to be stated. Based on the extensive literature and research in cognitive sciences as well as cited definitions, the main attributes of human intelligence are:

- Ability to recognize situations based on ambiguous or contradictory information.
- Spotting similarities between multiple different situations.
- Ability to make decisions according to the situation. Decisions are made by comparing the relative importance of different elements in that situation. It is an ability to recognize and use favourable conditions.
- Ability to acquire new knowledge and learn new concepts.
- Ability to connect multiple concepts and knowledges.

Therefore, in order to have artificial intelligence, a computer system needs to be able to:

- Make conclusions
- Recognize items, patterns and signs
- Count and calculate
- Perform abstraction and classification of items
- Make decisions regardless to the state of knowledge
- Solve problems
- Learn

These abilities are realised through various AI methods as shown in Fig. 4.4. The system analyses information through recognition of patterns, images and speech. This represents input of facts, learning and forming of new knowledge. Once acquired, knowledge is processed for specific information about a subject and the output of facts is realized through knowledge representation methods. Classification, searching and planning represent memory methods for accessing, retrieving and differentiation of facts. The intellect is realized through deduction, reasoning and problem solving methods. It defines the AI's ability to reach conclusions based on facts.

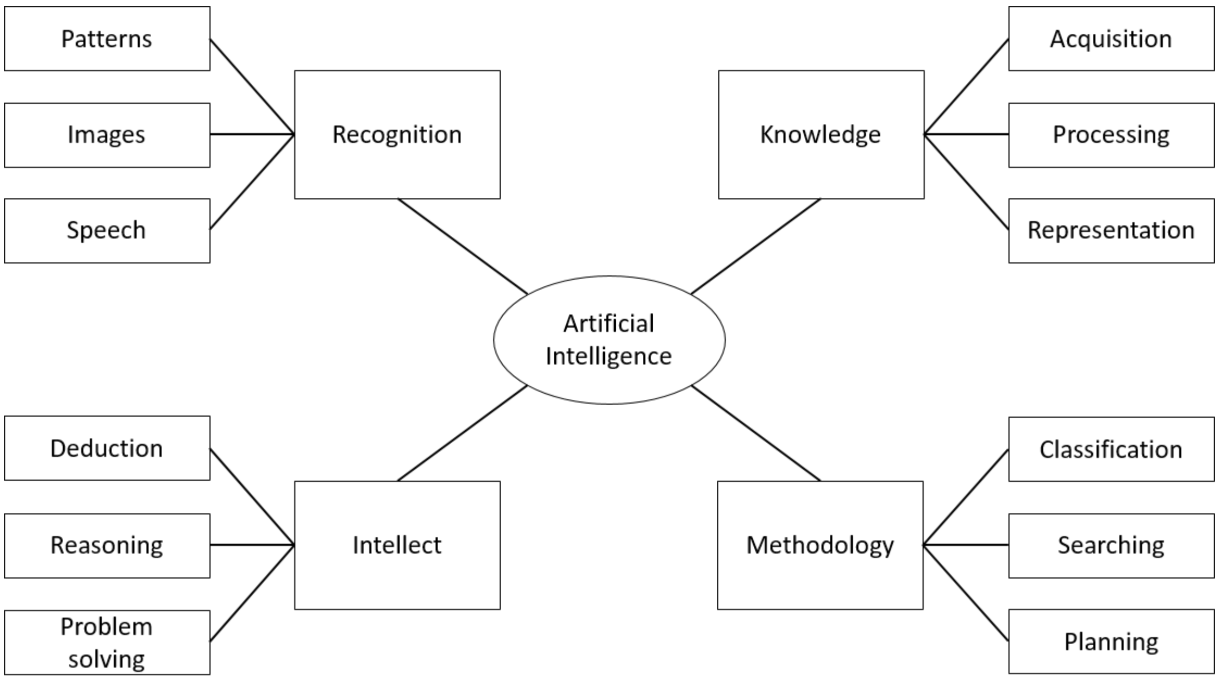


Fig. 4.4. Artificial Intelligence methods

Artificial Intelligence represents a very complex area of research and as such is comprised out of many sciences and methods.

It is based on computer science (use of algorithms in order to simplify complex calculation problems such as decision making), logic (formalisation of human thought processes), mathematics (analysis, calculations, statistics, probability…), linguistics (methods of knowledge exchange), psychology (methods of perception, cognition and deduction), neuroscience (replication of neuron function and biologically inspired computing) and many more.

4.4.2 AI system types

There are many different forms of Artificial Intelligence. Table 4.1. shows the types of AI systems and their utilization. These systems are based on the described methods and as such are specialized for different use case scenarios. AI can be applied in game playing (Chess, AI enemies…), speech recognition and synthesis (automated announcements), understanding of natural language (personal assistant in smartphones), computer vision (robot control, obstacle avoidance), simulators (aviation, space exploration…) and others.

Table 4.1. Utilization of specific AI systems

AI system	Utilization
Evolutionary computation	Optimisation, black box (unknown system) problem
Expert systems	Diagnostic, simulation, planning, knowledge processing
Fuzzy logic	Control, navigation, obstacle avoidance, transportation
Genetic algorithms	Optimization
Artificial life	Reproduction of artificial units
Neural networks	Pattern recognition, classification, learning
Data mining	Pattern recognition, prediction, analysis
Hybrid	Combination of 2 or more AI systems

Evolutionary computation – Evolutionary computation is based on the Darwinian principles of evolution. The development started separately in the 1960s by Lawrence J. Fogel and John Henry Holland in the US and by Ingo Rechenberg and Hans-Paul Schwefel in Germany. It can be described as a two-step iterative process consisting from a random variation followed by a selection The main concept starts with an initial population of possible solutions for a specific problem. These “parent” solutions generate “offspring” through defined variations. The new set of solutions are evaluated for their effectiveness or “fitness”. The most suitable ones are chosen and the cycle repeats (Fogel, 2000).

Expert systems – Expert systems are computer programs designed to emulate or support decision making process of a human expert (Jackson, 1998).

One example of an expert system is the implementation of the Intelligent Adviser Module in the control structure of the Bionic Assembly System.

Fuzzy logic – Fuzzy logic is a method for representing some form of uncertainty. The concept of "fuzzy sets" was developed by Zadech Lotfi in the 1960s. It is used to describe imprecise values (old, young, tall…) where the values can be any real number from 0 (false) till 1 (true).

Unlike Boolean logic, a statement can be both true and false and neither true or false. The main use of fuzzy logic is for especially complex problems, which cannot be described in a formal mathematical way (Klir & Yuan, 1995).

Genetic algorithms – As stated, genetic algorithms have been developed since the 1960s in the US by Lawrence J. Fogel and John Henry Holland independently to Ingo Rechenberg and Hans-Paul Schwefel in Germany.

The main goal was to implement evolutionary mechanisms to computational problem solving. Problems are solved by involving inheritance, mutation, selection and crossover from generation to generation of solutions, in hopes to keep the good trades and discard the bad ones. Doing this, a better, more optimal solution is derived (Mitchell, 1996).

Artificial life – Artificial life represents a field of study started by Cristopher Langton in the late 1980s. The main focus of research is applying naturally inspired concepts, such as living and reproduction to artificial systems.

Three different kinds of artificial life are defined: soft (software), hard (hardware) and wet (biochemistry). Doing this, biological phenomena is studied and recreated in order to gain a deeper understanding of information processing within a living system (Langton, 1997).

Neural networks –Neural networks emulate the interaction between nerve cells within the human nervous system. Information is processed through a large number of relatively simple processors which represent neurons. These processors operate independently and communicate with other processors using signals. Input line to a neuron is called a dendrite and the output line from the neuron is called an axon.

Each neuron is activated if the sum of inputs from other connected neurons exceeds a defined value. Neural networks are used when there is a large number of input data and rule –based programming would be very hard (Müller et al., 2012).

Data mining – Data mining is a synthesis of multiple techniques and methods used to analyse and predict system behaviour based on a large set of data. This data is used to extract useful knowledge (Manyika et al., 2011). This presents a great potential to improve the decision making process for the system operator in BAS.

Hybrid AI system – Hybrid systems are based on a combination of two or more types of artificial intelligence. They are used to utilize the most appropriate methods and techniques in order to achieve (as similar as possible) human like decision making qualities. These qualities are defined by intuition, uncertain reasoning, expert knowledge and adaptability to variable environments (Abraham et al., 2009).

BAS can be a highly dynamic and stressful working environment for the system operator. A support system needs to be implemented in order to assist him to make quality decisions in good time. Such a system needs to be contextually aware, have the ability to learn and to give proposals based on facts and rules. In other words, it needs to be able to acquire, formalise, process and represent knowledge. This intelligent behaviour is the main characteristic of such systems. These abilities are emerging from the previously described AI development trends.

For this reason, main concepts on which the decision support platform is based on need to be analysed. These concepts include knowledge, types of knowledge as well as decision, decision context, types of decision makers and different needs for decision support.

4.5 Knowledge

One of the defining features of our civilization is to combine knowledge, resources and energy in order to produce goods. Doing this, we are satisfying our needs and thus improving quality of life (Hays & Wheelwright, 1984).

Our intelligent behaviour is realized through reasoning, thinking and the ability to successfully acquire, organize, change and transfer knowledge. The main goal is to replicate such behaviour. In order to achieve this goal, knowledge needs to be represented symbolically and manipulated in an automated way by reasoning programs (Brachman et al.,1992).

Although it is difficult to precisely describe what knowledge is, the basic characteristics can be extracted from a few definitions in the literature:

"Knowledge is a familiarity, awareness or understanding of someone or something, such as facts, information, descriptions, or skills, which is acquired through experience or education by perceiving, discovering, or learning" (*, 2016).

"Knowledge is the information about a domain that can be used to solve problems in that domain. To solve many problems requires much knowledge, and this knowledge must be represented in the computer. As part of designing a program to solve problems, we must define how the knowledge will be represented" (Poole & Mackworth, 2010).

4.5.1 Types of knowledge

Table 4.2. shows the six types of knowledge (Holsapple & Whinston, 2013). They are divided into two main groups (primary and secondary). Primary group contains descriptive, procedural and reasoning knowledge which define the thinking and cognitive abilities of a system.

Secondary group contains linguistic, assimilative and presentation knowledge which are used for knowledge exchange (input, filter, output).

Table 4.2. Types of knowledge

Primary	**Secondary**
Descriptive ("What?")	Linguistic ("Request")
Procedural ("How?")	Assimilative ("Choose")
Reasoning ("Why?")	Presentation ("Response")

Descriptive knowledge - Also known in literature as conceptual or declarative knowledge (De Jong & Ferguson, 1996). It is static, predefined knowledge about facts, concepts, principles, states and definitions from a specific domain. It answers on the question "what".

Descriptive knowledge can include (Tseng et al., 1992):

- descriptive definitions of specific terms within a domain (assembly station, mobile robot, shop floor operator…).
- description of the relationship between domain specific objects to other objects from a different domain (shop floor operator is a human; mobile robots have wheels…).
- description of actions and events (mobile robot MR1 completed the operation OP4 on product PR123 at the assembly station AS3…).
- description of the decision-making rules or actions to be taken (if <quality of product PR123 is "NOT OK"> then <complete repair at the repair station RS6>).
- description of metaknowledge ("knowledge about knowledge") as a method for defining AI intuition. An example is bibliographic data. It gives an overview about all types of knowledge contained within a system. BAS analogy would be an overview of all technological data or rules.

Procedural knowledge - Also known in literature as imperative knowledge. It defines a step by step procedure and valid actions on how to complete a task within a specific domain. It answers on the question "how". This knowledge is formed during the execution of a task. Procedural knowledge includes three levels of procedures (McCormick, 1997):

- First level of procedures is directed to known goals and is automatic and structured
- Second level of procedures is directed to unfamiliar goals and includes problem solving
- Third level is a controlling function for switching between the first and second level

Reasoning knowledge - It defines what conclusions can be drawn after a decision has been made within certain situations. It answers on the question "why". Reasoning knowledge can include (Dunn, 2013):

- Deduction: starts from a general theory based on wide range of facts, rules and principles and narrows it down to a specific conclusion.
- Induction: starts from specific observations (measurements), identifies the existence of patterns or regularities and at the end defines a conclusion.
- Classification: grouping concepts, ideas and objects into related categories.
- Analogy: reaching conclusions after recognizing that two or more things have specific characteristics in common. By analogy, if they have one, they can have more characteristics in common.

Linguistic knowledge - It serves to interpret requests. It is the basis for a successful communication. Bidirectional exchange of information is realized in combination with the presentation knowledge. It is used to acquire new knowledge.

Assimilative knowledge – It is used as a filter for choosing which new knowledge is acceptable. It rejects low quality or unusable knowledge. Main goal is to keep useful knowledge which helps to solve problems.

Presentation knowledge - Its function is inverse to the linguistic knowledge. It is used for giving out pieces of information as a response to a request.

4.5.2 Knowledge representation methods

All of the described types of knowledge are used to define and capture information about the world. Knowledge representation (KR) methods incorporate these different types of knowledge within a computer system.

The main goal of KBS is to develop a computer representation which is able to solve a large variety of complex problems. Knowledge representation is much more suitable for defining and solving complex problem in comparison with procedural programming.

Representation methods should (Poole & Mackworth, 2010):

- Define the task and the solution.
- Represent the problem within a computer system.
- The computer system finds a solution to the problem in form of an answer or sequence of actions.
- Contain enough knowledge needed to solve the specified problem.
- Be maintainable. This is realized through clear and understandable representation. Small change in the problem results with a small change in the representation of the problem.
- Expandable by people, external data and past experiences.
- Efficient in comparison with the use of computational time.

The choice of an adequate knowledge representation method depends on the characteristics of the problem. There is a large number of different KR methods. Here is an overview of the more commonly used ones.

Rules

A rule consists out of condition and action pairs (Newell & Simon, 1972). They are also known as production rules or if-then rules. The condition is a Boolean expression which evaluates a situation in which the rule should be applied. Because the experts are formulating their knowledge in form of rules anyway, they are the most popular method of knowledge representation. A production rule system is a computer program which models human cognitive processes and is used to solve problems.

A production rule system consists out of:

- Set of production rules – IF this condition(s) occurs, THEN do these action(s).
- Context – detailed description of a given situation or a problem. It can be defined with a simple list or an array.
- Inference machine or an interpreter – program whose main task is to resolve conflicts during rule evaluation conditions and to decide which production rule to fire next. Rules are evaluated by matching the conditions of the rules to the context. When there is a match between the context and the conditions, a rule is fired.

Advantages:

- Production rules are easy to understand, add, delete or modify without affecting other rules (good for rapid development).
- Language and syntax provides a natural way of expressing knowledge.
- Easy to maintain and development.

Disadvantages:

- Inefficient use of computing power (matching) - it does not take advantage of efficient responsiveness or predetermined reasoning.
- Undisciplined order and structure of rules construction - hard to differentiate between rules that perform different functions.
- Knowledge base fragmentation makes it hard to maintain the integrity of the empirical knowledge base.

Semantic Networks

Semantic networks define hierarchical relationships between concepts from different domains (Russell & Norvig, 2002). They can be graphically represented by nodes (objects, concepts, situations in a domain) and arcs (relationships between the nodes). They are used to represent knowledge about the properties of an object. Semantic networks are widely used in natural language processing to represent languages. The nodes lower in the net can inherit properties from the higher nodes without having to represent these properties explicitly in the net. This logic is shown in Fig. 4.5.

Fig. 4.5. Semantic Network using the IS-A and HAS-A relations

Most commonly used arcs are “IS-A” and “HAS-A”. As shown in the example, IS-A represents class relationship where the object “Unit MR-123” belongs to a larger category of “Mobile robot”. HAS-A represents the characteristics or attributes of an object. There is no need to explicitly specify that the Unit MR-123 has a battery. That property is already described with the HAS-A statement, where “Mobile robot” has “Battery.

Frames

A frame is a grouped data structure organized very similarly to a semantic network (Minsky, 1975). It is composed from values that describe a specific object. It is hierarchically organized where the frames lower in the network can inherit properties form frames higher up in the network. The knowledge within a frame is organized into slots.

Each slot can hold descriptive or procedural knowledge. Using this method, frames can represent complex objects, situations or problems. They are used to document information about a domain model such as assembly station and their associated attributes as shown in Table 4.3.

Table 4.3. A frame describing an assembly station

Assembly Station Frame	**Heating Station Frame**
Superclass: BAS hardware	Model: HS-123
Subclass: Measurement, Heating, Rotating	Member of: Heating
Status: shut down, ready, malfunction	Status: ready
Operation time: short, medium, long	Operation time: long

Specific procedures can be associated with a particular slot which enables reasoning and defines the problem-solving behaviour of a system. This happens when the information in the slot changes. Such procedures can include when new information is added to a slot (*If added* procedure), when new information is deleted from a slot (*If removed* procedure) or when information is needed but the slot is empty (*If needed* procedure).

Computational logic

Computational logic is a method of converting statements and reasoning processes into a form suitable for computer manipulation. There are two basic types of computational logic: propositional and predicate logic.

Propositional logic (calculus) uses symbols to represent premises or conclusions. To express real world problems two or more premises can be combined using logical operators (and, or, not, implies, equivalent). However, this method is limited to represent complex problems as it can only deal statements that are true or false. For this reason, predicate logic has been introduced.

Predicate logic or first order predicate calculus describes real world objects, statuses logic relationships and their reasoning through calculus expressions. These expressions are called premises, axioms, facts or assumptions. They are used by a logical process to produce new facts and conclusions. Predicate logic uses variables and functions which break a statement down to its component parts.

Decision tables and trees

Decision tables and trees are used to represent knowledge of relations. Decision tables are organized into rows and columns and are divided into two parts. First part contains a list of attributes (conditions) where for each attribute all possible values (rules) are listed. The second part contains a list of defined conclusions (actions). To define a solution, different combinations of attributes (conditions) are matched against the conclusions (rules). Decision trees are composed of nodes representing goals and links representing decisions. They are simple to understand and are a natural way that experts define rules.

Other forms of KR methods

Such forms include ontology and constraints. They serve to represent types and properties of objects as well as to define relations between them. Doing this, information is organized and complexity is reduced.

Probabilistic logic is based on logic programming, but using probabilities instead of true or false values. The evaluation of statements with probability represents the degree of uncertainty.

Non-monotonic logic is very important for defining conclusions in the presence of incomplete information. In formal logic, a theorem stays unchanged over a period of time. Non-monotonic logic allows the introduction of changed assertions based on new observations which can invalidate old theorems.

Temporal logic is a KR method for specifying properties and behaviour of objects within reactive systems in terms of time. Temporal operators (always, eventually, never, whenever…) enable expressing statements in which their validity can vary over a period of time.

4.6 Decision Support

Once new knowledge is defined using the knowledge extraction methods, it is implemented within the Knowledge Base. Systems which are designed to support decision making based on facts rather than intuition are called Decision Support Systems (DSS). They support the decision maker to make higher quality decisions in good time. In order to develop such systems, the concepts of decision and decision support need to be defined.

Based on psychology and management literature overview, a decision can be described as:

"Decision-making is a process resulting in the selection of a belief, strategy or a course of action among several alternative possibilities. Every decision-making process produces a final choice that may or may not prompt action." (Janis & Mann, 1977).

When facing with a new problem, a system starts with a prior state of knowledge. During the decision-making process, the problem is analysed, and a course of action is selected. By choosing one among many possible actions, new knowledge is made. Regardless if the choice was effective or not, more is known after a decision has been made than before.

New knowledge contains the decision for a specific context as well as clarifications and justifications for why this decision has been made and other choices were avoided. The decision-making process is influenced by the context, by who makes the decisions and by the limitation of the decision maker.

4.6.1 Decision context

The context of decision making is defined by:

Organizational position:

- Shop floor operator (lower position) – More precise and detailed knowledge regarding a specific operation. Needs less creativity for making decisions and ensuring that specific tasks are efficiently completed.
- System operator (top position) – Wider knowledge and creativity used for making decisions regarding the overall execution of working scenarios, use of resources and resolving global organization problems in critical situations.

Type of situation:

- Repeated – It is much easier to make a decision in repeated, well known situations. Previously made decisions are used again if the situations are similar.
- New – New situations demand new decisions. It can be hard to choose a high-quality strategy in good time.

Degree of concurrence:

- Consecutive – It is easier to make decisions one after another. This implies that the decision maker can concentrate completely on reaching one quality decision in good time.
- Simultaneous – This is the hardest situation in which the decision maker reaches decisions. Neither decision will be reached with full concentration and quality.

4.6.2 Decision maker

Decision making can be performed by an individual or by multiple participants as shown in Fig. 4.6. Individual decision maker can be either a human or a computer. If the computer is making the final decision it is called a Decision-Making System (DMS). Such systems could be applied in well-known, repetitive and organized situations where structured or programmable decisions can be defined using procedural knowledge and if / then rules.

However, it is very difficult to define every rule for every situation which will cover all case scenarios. Real life situations are defined with uncertainty, novelty and unpredictability where unstructured or un-programmable decisions need to be made by the human decision maker. In such conditions, there is a need for ingenuity, creativity, imagination, intuition and exploration.

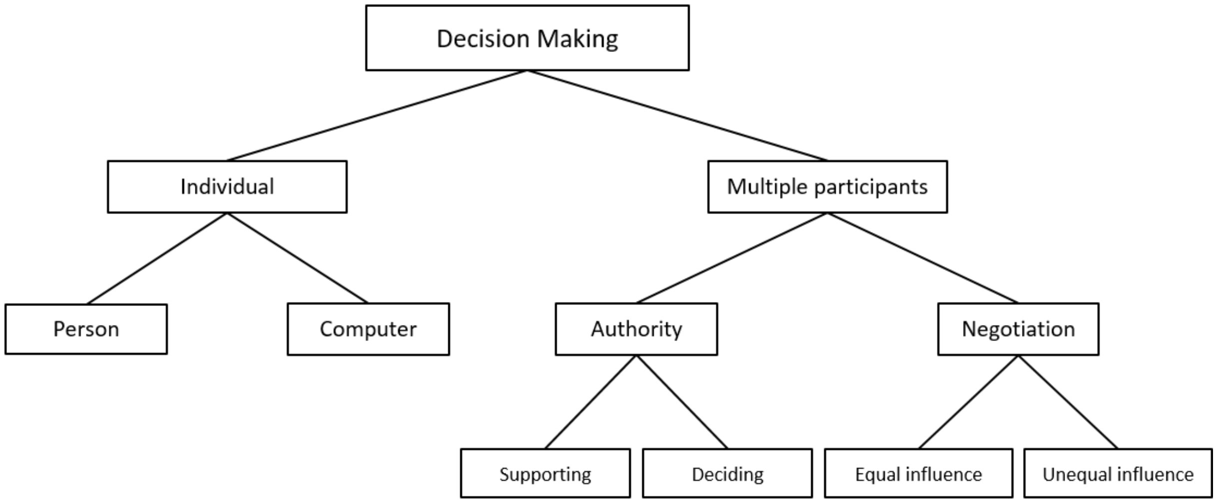


Fig. 4.6. Types of Decision Makers (Holsapple & Whinston, 1996)

Multiple participants make decisions either by negotiation or authority. A team for example, reaches authority based decisions derived from the divided roles of its supporting and deciding members. It is comprised out of multiple participants but only one, deciding person makes the decision. He can be influenced by the other supporting participants to a varying degree.

Negotiated decision can be made by participants who can have equal (group) or unequal (organization) influence on the final decision. A group reaches a decision when all individuals agree on a single mutual choice in a varying degree of satisfaction.

The advantage of this approach is that there is a bigger pool of knowledge within the group. However, at the same time, the disadvantage can be that decisions can be delayed if all members do not agree with the same choice. An organization reaches decisions based on the varied influences of its submodules. It can be viewed as a unity of multiple teams or groups.

4.6.3 Need for support

A decision-making environment is stressful, dynamic and requires the use of relevant knowledge. It can be very difficult for the decision maker to reach a quality decision in good time. There are different types of support methods used to help him. Each of them allows to extend or augment his natural abilities.

This can be achieved either by alerting him that there is a conflict situation and a decision is needed, by recognizing and solving a problem, by supporting his creativity and imagination, by offering advice, analysis, facts or by extending his knowledge.

The need for decision support is derived from the following limitations:

- Cognitive limitations – Humans have limited physical and mental capacity to store and process different types of knowledge. In addition to that, stress, errors, oversights, overload with facts or insufficient knowledge directly influence the quality of decisions.
- Economic limitations – To minimize the cognitive limits, a support team or a DSS system can be introduced. Both options present costs. A team of people needs to be educated, trained and payed. On the other hand, a DSS system needs to be developed, tested, maintained and the user needs to be educated on how to use it. In either way, the benefits of extending the decision makers abilities need to outweigh the costs of the chosen support method.
- Temporal limitations – The decision oftentimes needs to be made quickly. This introduces stress and increases the chance of an error. It is connected to cognitive limitations of the decision maker. In order to make a quality decision, he needs to analyse the facts and process his knowledge. Additionally, he gets tired and the concentration declines over time.

4.7 Learning

Next to the capability to represent knowledge, these decision support systems should be able to acquire new knowledge through learning. This makes it possible to improve their performance over a period of time by expanding the range of operations, improving the accuracy or increasing the speed of task completion. Learning is defined through a specific action, steps to perform that action and measure of improvement.

Already known actions can be improved by changing or adapting the operating steps. Measure of improvement represents how does the system perform in regard to accuracy or speed after including or changing a specific action.

Based on literature overview, learning could be described as:

"*Learning is constructing or modifying representations of what is being experienced*." (Michalski et al., 2013).

"*Learning is the acquisition of knowledge or skills through study, experience, or being taught*." (**, 2016)

There are several types of learning methods that a system can use. These include:

- Trial and error - A specific problem is approached using multiple varied strategies. More accurate and faster strategies are replacing the slower and less precise ones. The process is repeated until a solution is found or if no more approach variations exist.

- Rote learning - New information is stored in its original form. It represents facts from a specific domain. The emphasis is not on comprehension but rather on information indexing for quick retrieval.
- Learning from instructions - New information is not simply stored. First, it is compared to the already stored knowledge. If there is a correlation, the new information is reformulated and integrated. This way the existing knowledge is simply expanded and multiple inputs from same data are avoided.
- Learning by observing - It is also known as unsupervised learning by induction. It is used to explain observations by finding patterns and regularities in large sets of data. A system can learn by observing examples as well. It needs to recognize the key factors which have a direct influence on the results. If patterns or key parameters are identified, future behaviours and trends can be predicted.
- Learning by analogy - A system can have knowledge on how to solve a problem from a specific domain. When facing with a different problem from a similar domain, the existing knowledge can be used by analogy to solve it.

4.8 Summary

Bionic Assembly System is a human centric system and as such promotes integration of workers on the shop floor and in the control system. The system operator is the main decision maker. He faces challenges which include system complexity, large amounts of data, distractions and disturbances. Additionally, he has a limited physical and mental capacity to make repetitive decision with good quality.

As a solution, a support system is introduced. It helps the system operator to make quality decisions in short time. Such a system is based on artificial intelligence where the main goal is to replicate human cognitive and thought processes. To achieve this, many types of AI systems were developed to address specific intelligence functions. These include knowledge representation methods as well as decision support and learning capabilities. Further AI development is directed towards systems that integrate multiple AI types and is represented through hybrid AI systems.

One such example of a hybrid artificial intelligence system is the Intelligent Adviser Module, developed by the Intelligent Manufacturing Systems (IMS) group from Vienna University of Technology. Its classification, structure, functions, characteristics, realization prerequisites and working modes are defined in the next chapter.

Chapter 5

Intelligent Adviser Module

The efficiency of modern complex assembly systems depends on many parameters and also on the quality of control decisions and time needed to make them. The main decision maker in BAS is the system operator. His human ability to make repetitive high-quality decisions in good time is limited. This ability can be supported by a knowledge based, decision support software system represented by the Intelligent Adviser Module (IAM).

Outputs from the Adviser Module are proposals. These proposals are a result of analysis and processing of Bionic Assembly System data. This data is representing states of the assembly system during the defined period of assembly process. The main advantage of implementing the IAM within BAS is that it is able to combine existing, domain knowledge and human expertise with system specific related knowledge which is generated during the operation of the entire assembly system.

Such system specific knowledge is a result of actual system states and digitally recorded data from significant period of past working time. This includes knowledge about events which have not occurred ever before and as such are very difficult or impossible to predict. These events represent the past BAS system states. By combining all these different types of knowledge, the IAM is able to improve the quality of its proposals which help the system operator to reach a decision. The implementation of the IAM represents the continuation in further development of the human centred, hybrid self-organizing assembly systems.

This chapter defines the IAM classification, its structure, functions, characteristics, realization prerequisites, working modes and the man-machine interface problem.

5.1 Classification

Intelligent Adviser Module is a hybrid artificial intelligence system. It combines multiple types of AI in order to replicate the human ability to process knowledge and integrate it within the control structure of BAS. The IAM was introduced because traditional software methods (procedural programming, binary logic) did not offer satisfactory solutions in such complex systems.

The difficulty to pre-program and define variables, methods and functions for every event, disruption, or operation introduces a demand for new approaches in knowledge processing.

The IAM does not utilize traditional software methods because:

1) The number of all possible working scenarios which occur during BAS lifetime and which are executed either according to plan (normal scenarios), not according to plan or in conflict situations (disturbances) are impossible to plan or predict.
2) The frequency and distribution of disturbances is impossible to predict or avoid. A solution (exit scenario), has to be realised even if it is occurring for the first time. This capability defines an adaptive system.
3) Software based on binary logic and pre-programmed procedures is an unreliable and insufficient tool for defining exit scenarios and solutions in crisis moments, i.e. when disturbances occur.

The IAM has to utilize non predetermined software methods. It has to be able to produce proposals for exit strategies based on the current system status. At its core, the IAM is a decision support system. Its decision support capabilities are realized through a special kind of a Decision Support System (DSS), which relies on appropriate knowledge representation methods (i.e. production rules).

Such a system is called an Expert System (ES). It is a computer application used for solving complex problems within a particular domain by simulating human ability to reach conclusions based on expert knowledge (Jackson, 1998).

It takes a long time for human experts to develop such an ability. In order to provide a quick solution, human experts need to recognize and describe the problem. Once a satisfactory solution has been found, they need to be able to explain and justify how and why they reached their conclusions.

This allows them to learn from their experiences, expand their knowledge and use it in the future to solve new and unfamiliar problems (Waterman, 1986). Therefore, an ES needs to be able to replicate this behaviour.

Expert systems are designed to facilitate the development of software to model human knowledge or expertise (Giarratano, 2002). On one side, the expert system platform of the IAM enables to constantly update its expertise database. On the other side, the IAM is integrated within the control system of BAS.

This makes data collection, analysis and status monitoring data to be incorporated and used for knowledge extraction. Therefore, the IAM knowledge is based on actual and past BAS states, external data, predefined facts, experiences of human experts and from system operators directly involved with BAS.

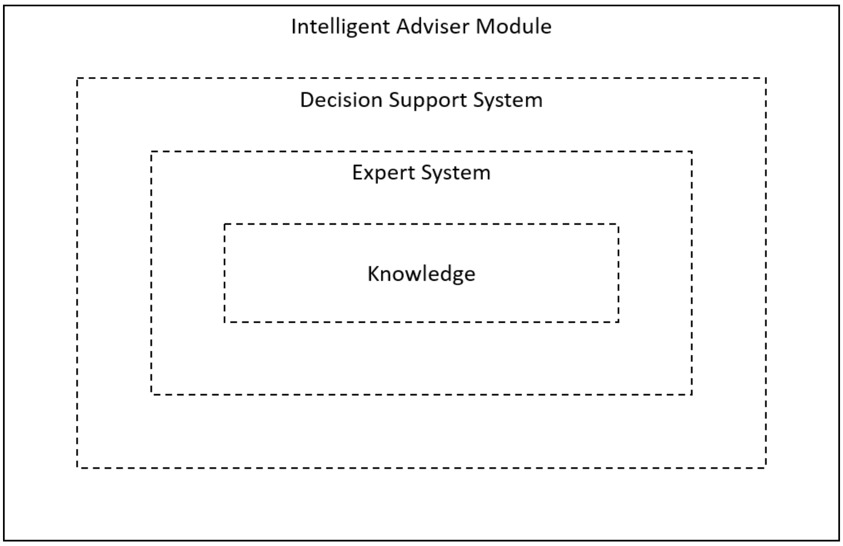


Fig. 5.1. Classification of the Intelligent Adviser Module

Fig. 5.1. shows the classification of the Intelligent Adviser Module. Based on it, the IAM can be defined as: "Intelligent Adviser Module is a knowledge based expert system which is a type of a decision support software used to help the system operator to reach quality decisions in good time. Its expert functions are realized with the ability to solve complex problems through representing, acquiring and processing knowledge".

5.2 Structure

The final decisions have to be made exclusively by the system operator. The IAM needs to support him during the decision making process. The primary rule is that the IAM must not interfere with the operation of the BAS control system or have any direct influence on it. Therefore, the IAM needs to be implemented exclusively as a data collection system. That means that the data stream has a single direction (read-only) from the BAS control system to the IAM as shown in Fig. 5.2.

Otherwise, if the IAM would have the ability to send data directly to the control system, it would stop being a decision support system. and become a decision-making system. This is not the goal of IAM in the context of BAS as a human centric system. The main concept is that the IAM learns and improves its accuracy over time. This is realized through the IAM structure and its integration within BAS.

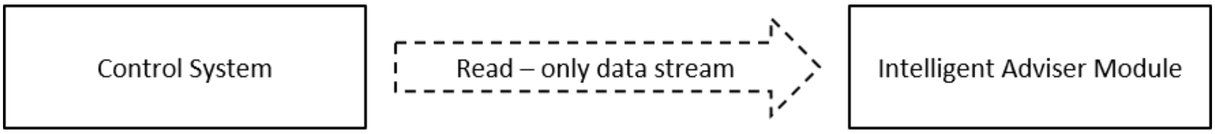


Fig. 5.2. IAM - Read only access to BAS control system

On the other side, bidirectional exchange of information (data) takes place between the system operator and the IAM. Here, the system operator can receive feedback information in form of graphs, tables, data and of course, proposals. He chooses if he will use this feedback information when he needs to make a decision.

As shown in Fig. 5.3., the main IAM structure is composed from the following submodules:

- System Monitoring
- Knowledge Management
- Decision Support

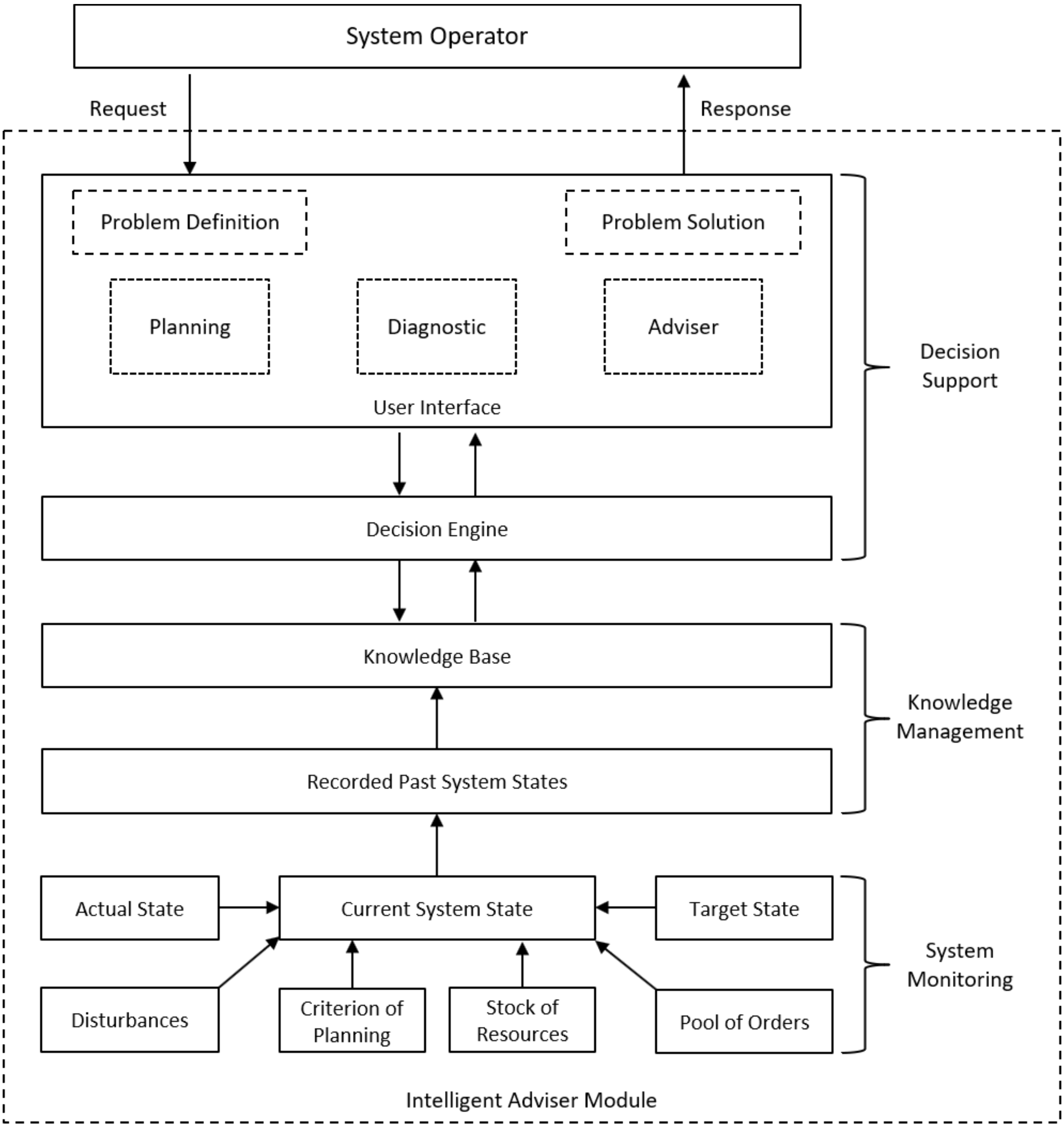


Fig. 5.3. Intelligent Adviser Module Structure

5.2.1 System monitoring

The realization of planned scenarios is tracked through real time system monitoring of actual system states. The progress of this realization is determined from the analysis of the target and actual state of the system. If the difference starts to be critical, the IAM notifies the operator.

IAM is an integral part of the BAS control system and is connected to all other system components (as described in chapter 3.). It serves as an information hub and as such it presents current system information to the system operator. This includes performance of the shop floor elements as well as the disturbances that can occur, resource levels and status of assembly order completion. If there is a critical error or a malfunction, an audio-visual notification is sent to the operator. Notifications can be triggered via:

- Threshold values (minimum, average, maximum) – This kind of notifications are based on measurements. They can be sent if the stock of resources is getting too low, if the duration of an operation is above a maximum scheduled time, if a robot's battery is almost depleted, etc.
- Boolean logic (yes / no, on / off) – These notifications can be sent according to a specific hardware status. This can include if a station is shut down or turned on, if a robot is moving, if the shop floor operator is present at a specific position, quality control is positive or negative, etc.
- Trend predictions (incline, stagnation, decline) – Trend prediction notifications are based on measurements, data analysis and correlation functions. For example, these methods are being successfully implemented in the prediction of tool breakage (Hsueh & Yang, C., 2008). This enables the IAM to proactively warn the system operator before a failure occurs. As a result, the system operator can make decisions with which disturbances can be avoided and consequently increase the efficiency of the assembly system.

5.2.2 Knowledge management

The output from the IAM are proposals. The quality of these proposals has a direct influence on the system operator during his decision-making process. He has to be able to reach quality decisions in good time.

This can be achieved by identifying previously unknown correlations in-between all the recorded past system data (suppliers, hardware age or performance, process duration, end product quality, breakdowns…). In other words, the system needs to be able to acquire new knowledge based on data derived from the execution of past working scenarios.

Knowledge Management in the IAM is based on:

- Recorded Past System States
- Knowledge Base

5.2.2.1 Recorded Past System States

System monitoring produces large amounts of data which is constantly recorded. This recorded data represents past system states and is the basis for the Knowledge Base (KB). The main information flow channel is between the control system and the controlled system (hardware, shop floor components, etc). A real-time exchange of information in both directions is taking place. Control system sends instructions, commands and data to operators, assembly stations and robots. They send feedback information about the realization and hardware status. The IAM receives all data from this communication channel and stores it within a database. This data consists out of measurements, quality reports, disturbance occurrences, robot travel times, operation times, external and internal resource tracking, status of orders, selection of orders based on specific criteria, past system operator decisions, etc.

5.2.2.2 Knowledge Base

A knowledge base (KB) contains information from a specific area of knowledge and is the foundation of the IAM. It is a collection of different knowledge types which need to be defined, stored and represented in a format which is understandable to a system (Fikes & Kehler, 1985).

The main function of KB is based on Knowledge Extraction (KE) and Knowledge Representation (KR) methods. KB contains complex structured and unstructured data. Structured data has defined relationships between elements (relational databases) and unstructured data is a set of elements (text, documents, images, numbers etc.) which are not organized in a pre-defined manner. Different types of knowledge can be derived either from undocumented or documented sources as shown in Fig. 5.4.

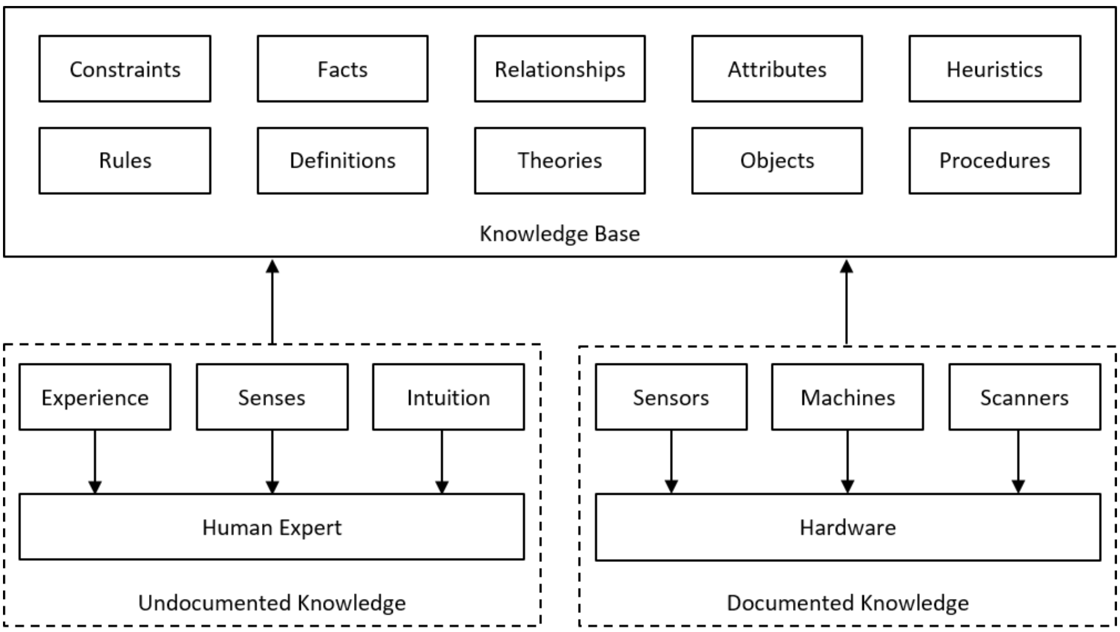


Fig. 5.4. Knowledge Base

The key components which define a KB are information, rules and heuristics. Information is a set of observations, facts, symbols, objects, procedures, theories, definitions, constraints, attributes objects etc. Rules define conditions and actions used to represent knowledge. Heuristics is used to solve problems using strategies which are not guaranteed to be optimal (when information is incomplete). In other words, it represents an ability to evaluate, judge and guess based on "intuition".

The human expert has undocumented knowledge which contains problem solving strategies based on his experiences, intuition and senses. This knowledge is dynamic (constantly changing), it can be difficult to articulate or to describe, extracting methods (interviews, process tracking, protocol analysis, observations, etc.) can be inefficient and it can be unstable or unreliable (nature of the human brain). These characteristics make the transfer of such knowledge complex.

Hardware (sensors, assembly stations, scanners etc.) on the other hand, create large amounts of data. This represents a basis for documented knowledge which is consistent (no further alterations once saved), theoretically infinite and permanent (depending on the storage technology) and easy to transfer from one system to another.

All this digitally recorded data allows the IAM to extract new knowledge. It is used to describe past events or to predict future system behaviour if a correlation between parameters is found. The quality of the KB improves over time as more and more data is accumulated. The success and efficiency of the IAM heavily depends on the quality, precision and completeness of acquired knowledge within its KB.

It is possible to identify the correlation between particular sets of data. Additionally, this recorded data contains the past system operator decisions made in contrast to the past system states. This allows to extract his expert knowledge and to define expert decision rules which can be used in future. If there is a similar situation where a decision needs to be made, the IAM can quickly recognize it and make a proposal according to the past results. If the past decision from the system operator was satisfactory, it can be reused. On the other hand, if the decision he made had negative impact on the system performance, it can be shown as a warning to avoid similar negative decisions.

The entire KB therefore, is composed from:

- Raw knowledge – it is a result of all the digitally recorded data from all the shop floor elements. Each of them, during their operation, generates data over a long period of time. As a result, there are large amounts of data which can be viewed as knowledge in its raw form. However, not all this data is useful, accurate, complete or can be a basis for applicable extracted knowledge. In fact, only a small part of this data can be used to generate new useful knowledge.
- External knowledge – this type of knowledge contains facts which can be specified even before the assembly system starts operating. It is derived from external data which includes hardware specifications, operational facts from manuals, troubleshooting guides and other documentation. It is predefined and easily transferable to IAM Knowledge Base.

- Extracted knowledge – as stated, only a small part of raw data can be used to extract new knowledge. However, once defined, it can be very useful because it reflects the system specific knowledge which can be very difficult or impossible to define before a system even exists. It can contain unique, variable occurrences which happen during the execution of working scenarios (faulty mobile robot's wheels, power fluctuations, varying component quality from different suppliers, etc.).
- Expert knowledge – It is a result of years of experience of human experts. General experiences from domain experts as well as accumulated "situation-decision-results" cases from the past.

However, the goal for the IAM is to have an ability to chain all these knowledge types. This ability is called metaknowledge or knowledge about knowledge as shown in Fig. 5.5. With it, the IAM "knows" what it "knows. It represents learning on a higher level where the IAM should be able to correct itself. Even in cases where a wrong proposal was given it should use those facts to improve itself for future use.

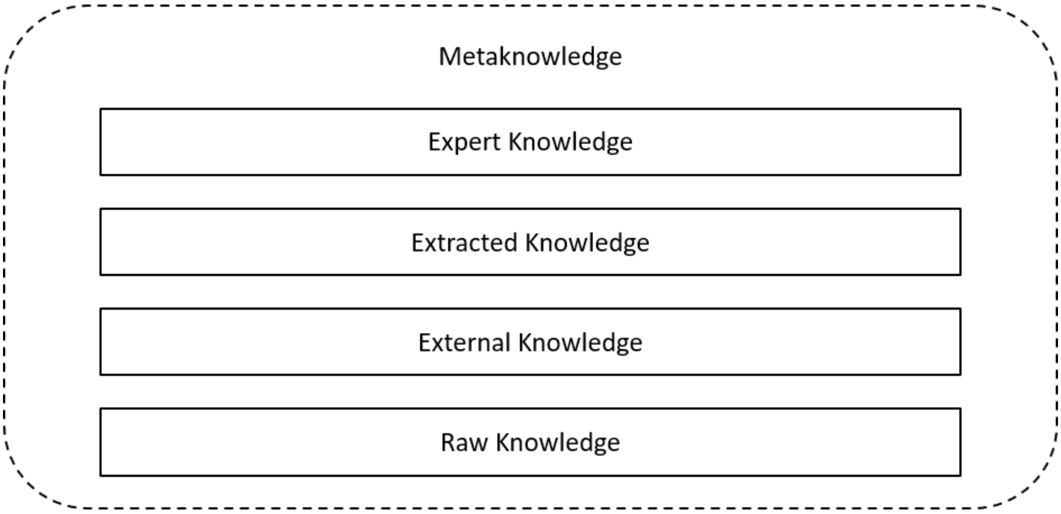


Fig. 5.5. Metaknowledge

5.2.3 Decision support

Decision support in the IAM is based on:

- Decision engine
- User interface

5.2.3.1 Decision engine

Intelligence of a system is realised through its ability to reason about knowledge. Once the KB has been set, it is connected to the IAM reasoning core also known as the Decision Engine or the Rule Interpreter in case of a rule-based knowledge representation. It is an automated computer software which uses, acquires and manipulates knowledge in order to define a solution to a specific problem as well as asserting new knowledge back to the KB.

The assembly process within BAS is very complex and it is hard to cover all scenarios and disturbances. One of IAM advantages is that it becomes more accurate as more data is accumulated. However, as with any other system, mistakes are possible. This can happen because its KB is incomplete or contains incorrect facts or rules. As a result, conclusions can be inaccurate.

Therefore, it is very important that the IAM can demonstrate the logic behind its reasoning. This is completed using the explanation capabilities of the decision engine. During his decision-making process, the system operator has several options when presented with a proposal:

- Acceptance – In this case the system operator accepts the IAM proposal without considering the reasoning logic. It can happen because of multiple reasons. This includes high stress situations, when he is not concentrated, when a decision is needed immediately, when his knowledge is not sufficient or simply when he agrees with the proposal.
- Inspection – If the system operator is not satisfied or suspects the validity of the proposal, he will inspect it using the explanation mechanism. It should show every step of the reasoning process through the graphical user interface. The explanation should always be presented to the system operator in a natural and understandable way.
- Rejection – The system operator can reject proposals if he has enough knowledge about the specific problem or if he decides that the reasoning is flawed after performing an inspection. In either case, his decision is recorded for future reference. Based on his past decisions, new rules can be extracted from the recorded data and the decision engine reasoning should be able to improve.

Additionally, explanation capabilities can be utilized for:

- Verification, maintenance and improvement – During the development of IAM it is important to verify the system accuracy. This is completed by controlled reasoning tests in order to see if a specific rule is fired when expected.
- Trust – In the early stages of IAM utilization, the system operator does not completely trust the system. He can use the explanation mechanism to check the reasoning and slowly acquire confidence in the reliability of the IAM.
- Inspiration – Even if the system operator decides that the proposal is invalid, he can inspect the reasoning logic and identify possible wrong steps. Doing this, it is possible for him to reach his own conclusions.

5.2.3.2 User interface

There are specific requirements during the realisation of a graphical user interface (GUI). These include user friendliness, simplicity, consistency, hierarchy, logical structure, etc. This is especially important during the interaction between the system operator and the IAM.

In a complex system such as BAS, GUI needs to assist the system operator to reach quality decisions in good time with minimal effort. The system operator does not have to be an expert in AI or to be able to resolve technical errors of the IAM itself.

The interface needs to be flexible and adaptable according to the user's requirements, habits and work practices. Input methods can include keyboard, mouse and fingers via the touch sensitive screen. Each method offers advantages in specific use case scenarios. Keyboard is used to quickly and precisely describe problems in the dialog manager as well as for custom user shortcuts. Mouse inputs are precise and offer contextual menus via right / left clicks. On the other hand, using fingers is intuitive and offers effortless sliding through menus as well as allowing for gestures through multi touch screen technology. The interface needs to be developed to take advantages of each of those methods. The on-screen elements need to be large and clear to allow easy selection using fingers or a mouse.

Menus need to be organized and a help function always available. The information needs to be presented using clear and understandable methods. These can include, reports, tables, graphics, schemes, graphs, simple text-based instructions etc. All user interface elements have a common purpose – to present most useful and relevant information upon system operator request.

5.3 Functions

The IAM is a support tool. It is used when the need for the system operator to make decisions is the highest. This is during planning and when the assembly is not running according to plan, which includes minor fluctuations and major disturbances. Based on these situations, the system operator selects a corresponding function which could help him.

The IAM has the following main functions:

- Planning and simulation
- Diagnostic
- Adviser

5.3.1 Planning and simulation

Main goal of planning in BAS is to achieve the highest productivity by organizing assembly as a continuous stream of one or more parallel assembly orders. One assembly order means to assemble one run of product. Highest productivity means that a maximum number of assembled products are finished within a set timeframe with regards to all system states, external priorities of BAS orders, system bottlenecks, limited hardware and human resources, limited throughput capacity, disturbances and availability of all supplementary resources. To accomplish this, all necessary assembly activities need to be completed in the shortest time possible. All system resources, which include hardware, software and human resources need to be synchronised and complete their tasks simultaneously.

Planning has to ensure that the customer orders are finished in acceptable time. When a customer orders different quantities of various product types, they need to be finished and sent at a specific deadline (Year-Month-Day-Hour-Minute).

Planning can be a basis for a simulation where any possible scenarios could be created and analysed for a specific time period in advance. When planning results (queues and reserved resources) are used for simulation purposes the result is a virtual scenario. However, such scenarios only show a possibility how the assembly process could take place within the entire assembly system. The precision of simulations decreases as the number of orders is completed, because the probability that disturbances will occur increases with time.

Planning has to always take place within a defined future time period. The boundary is represented with a planning horizon. This horizon can be set as desired so that the simulation can cover hourly, daily, weekly or monthly time periods in advance. The IAM work should be based on the forecast of execution of working scenarios for a short time horizon. Using simulation methods, it is possible to make assumptions about the progress of assembly and system behaviour.

This can include when the orders will be finished, what the status of resources will be, what the utilization of hardware and shop floor operators will be, etc. Such scenarios are assessed from different aspects by the system operator. If he decides that a particular virtual scenario is suitable, it will be enabled as a working scenario and used for actual scheduling control.

Simulation makes it possible to define specific or all resources as unlimited. This makes it possible to check if the resources will cause a bottleneck during the eventual execution of working scenarios. If yes, it is possible to identify which type, when and what quantity of resources need to be increased so that the scenarios are achievable.

This gives a new meaning to planning as it can lead to development and increase of system possibilities. Simulations can also be useful when major disturbances occur where affected assembly orders need to be cancelled. Unaffected orders and system status are basis for the simulation which is used to build alternative virtual scenarios. If the system operator decides he can activate such rescue scenarios for actual scheduling control.

5.3.2 Diagnostic

Diagnostic serves to identify problems with the following properties (Puppe, 1991):

- A problem is defined with a number of attributes also known as symptoms.
- There is a connection between symptoms and diagnoses which needs to be determined.
- A problem is defined through a selection of one diagnosis from a number of existing alternatives.
- It is not always possible to identify all symptoms. In that case it is necessary to define alternative, plausible hypotheses which can be modified or rejected if new symptoms are identified.

The diagnostic process components are shown in Table 5.1. Diagnostic starts when unusual symptoms are detected. A set of diagnostic tests is performed which search through gathered system data in order to detect any symptoms, signs or evidence. The main goal is to identify what is causing the problem based on a number of matched symptoms.

Table 5.1. Diagnostic components, (Rychener, 2012)

Start	Something is wrong (unusual symptoms)
	Gather data about symptoms and system status
	Standard set of diagnostic tests
Tasks	Match a set of symptoms with a known problem definition
	Find probable causes of symptoms
	Propose the most appropriate hypotheses / Recommend solutions
Constraints	There are many test types – selectivity needs to be applied
	Tests may be expensive (Time / Money)
	Tests may not be reliable or precise

Diagnostic will be simple if a symptom is unique to a specific problem. In that case, the precision of the diagnostic increases with the number of identified unique symptoms. The problem can be identified and a definite solution can be recommended if it was already solved or described in the Knowledge Base. For example, if a mobile robot stops moving and the data shows that its battery is empty.

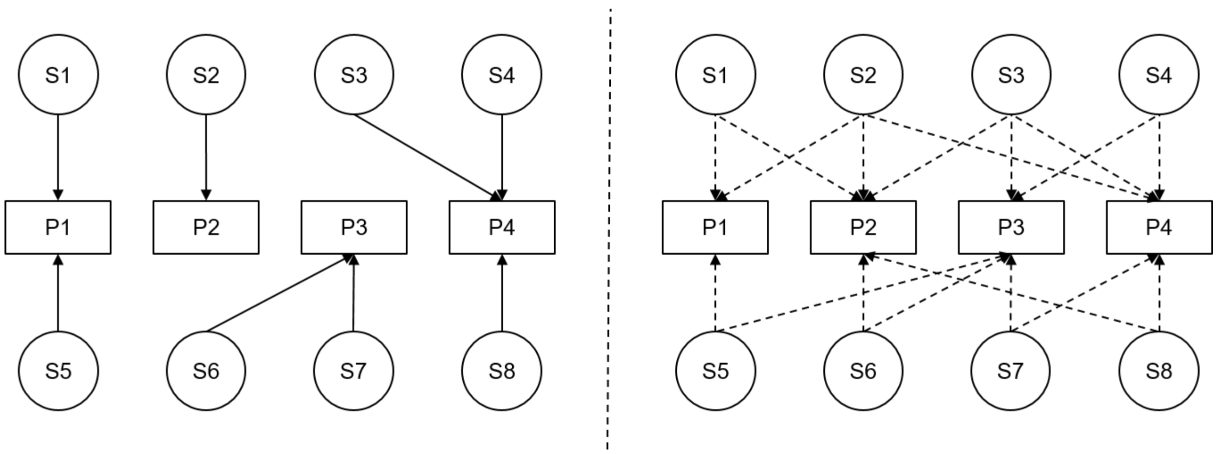


Fig. 5.6. Unique Symptoms vs. Common Symptoms (S – Symptom, P – Problem)

However, it is much more likely that a single symptom or a group of symptoms can be a cause for many different problems. These are called common symptoms. Fig. 5.6. shows the comparison between unique and common symptoms.

For this reason, the diagnostic has to group all symptoms into applicable hypotheses (explanations) which could potentially describe possible causes. Between them, the diagnostic has to evaluate their probabilities.

The IAM diagnostic function helps the system operator to identify problems before or during the realization of BAS working scenarios. During the execution of working scenarios there can be any number of problems with assembly stations, mobile robots, shop floor operators, resources, software etc. The diagnostic needs to show the problem and its possible causes. It finds symptom patterns in order to detect irregularities.

Before the beginning of actual assembly, it is necessary to verify that it is possible to complete the order with the available system resources. The diagnostic applies a set of tests which verify if the system is ready. This includes operational status of hardware, setup status of assembly stations, assembly order completion status, pool of orders status, status of resources etc.

If all tests are positive, it is possible to complete an order. If not, the diagnostic gives an overview of all test results and shows detailed information about what is necessary to complete the orders as well as what is currently available within the system. For example, it has to be verified if the problem can be solved by partitioning the batch size, by replacing a mobile robot or an assembly station etc.

5.3.3 Adviser

The system operator needs to ensure an uninterrupted execution of working scenarios by making final decisions. However, he has limited ability to make such decisions in good time. Good time means that the shop floor elements do not wait for his decision during the executions of BAS working scenarios.

Generally, if the system operator considers a problem to be simple or that he has enough time to solve it, he would not use the adviser function. Instead, he would directly apply the solution he sees acceptable. On the other hand, if he considers a problem to be complex, or that he does not have enough time or knowledge, he has the adviser function at his disposal as shown in Fig. 5.7.

The adviser function presents a dialog manager which serves to exchange information with the system operator and to define a problem. This is realised as a series of questions on which the system operator answers. Such an information exchange is performed like a conversation as shown in Fig. 5.8. This makes it more natural and intuitive for the system operator. Many different data collection methods can be utilized. This includes Boolean (Yes / No) questions, input/output values (time, speed, percentage…), multiple choice, etc. An experimental IAM interface was developed for demonstration purposes in Clips 6.3. Complete code is available in appendix A.

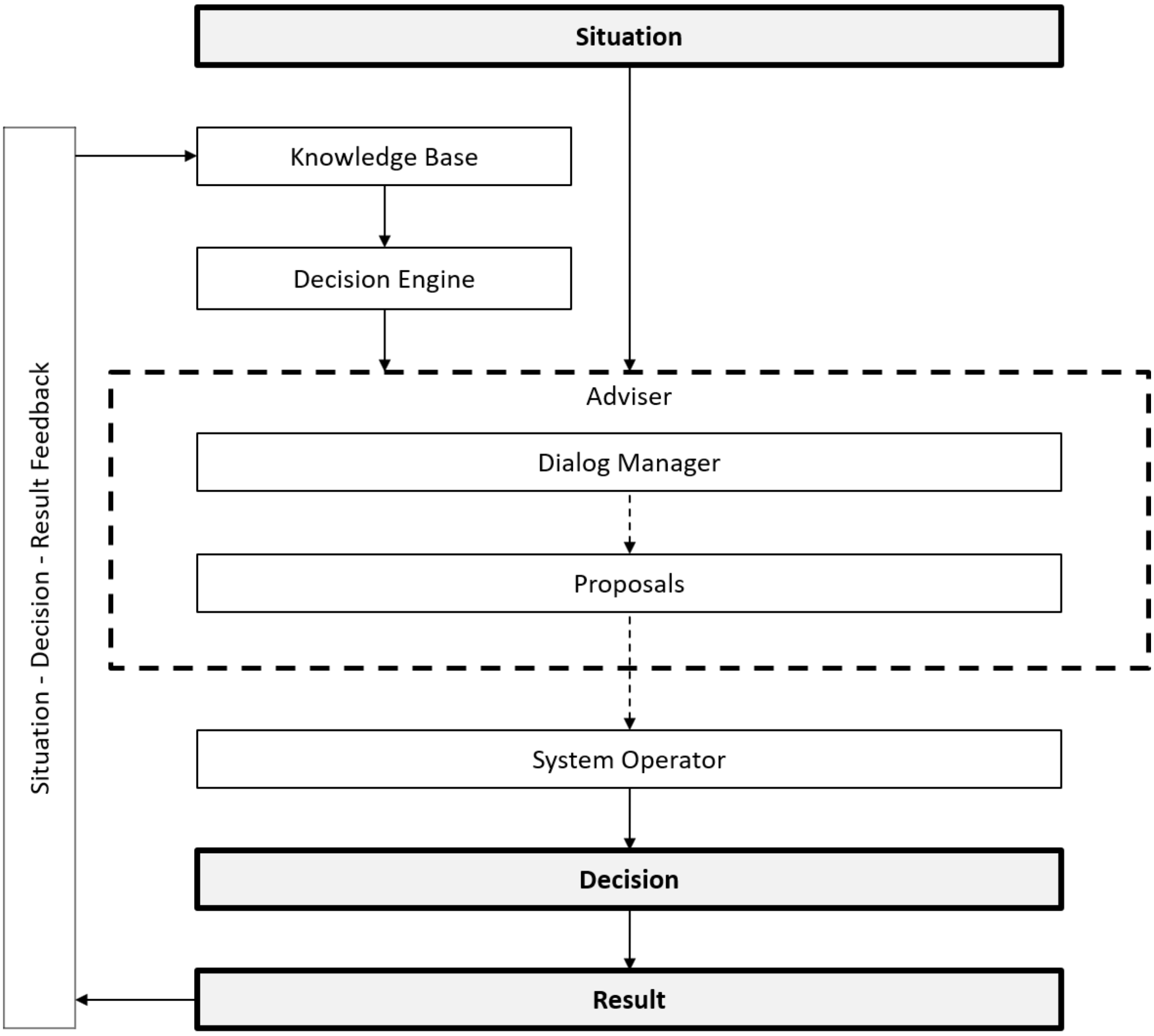


Fig. 5.7. Adviser and Situation – Decision – Result Feedback

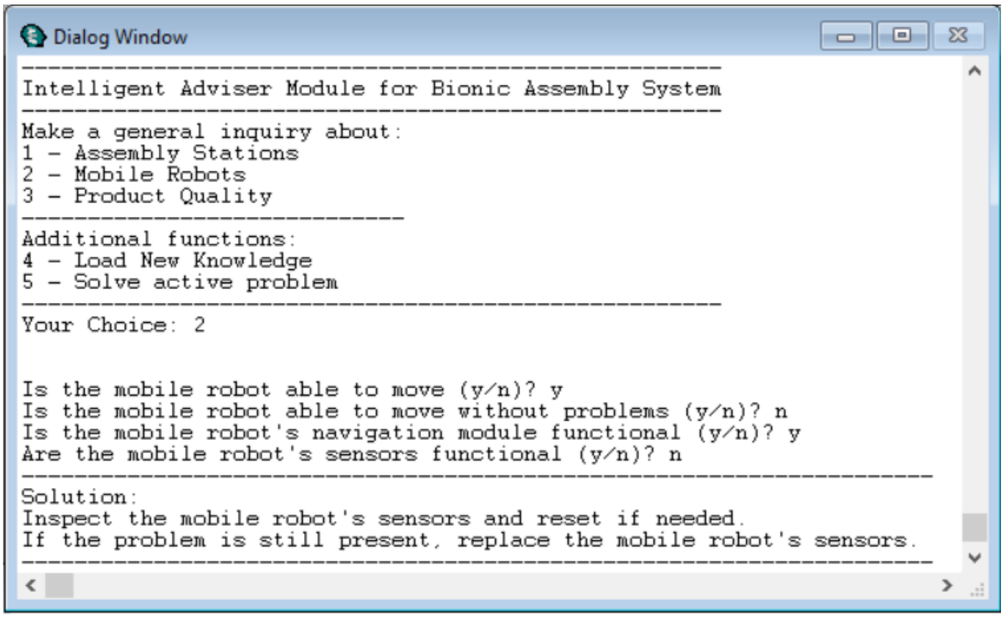


Fig. 5.8. Adviser Dialog Manager (Appendix A)

Such inputs from the dialog manager are called problem descriptors. The adviser completes its function based on its knowledge base, decision engine and these newly acquired descriptors. The results of this procedure are proposals. The system operator decides if he will accept, inspect or reject them. If possible, it is recommended that the system operator gives feedback about his choice right afterwards. This data is recorded for future reference. This way the IAM learns and expands its KB. All this information can be useful if a similar problem repeats.

After each decision, the results are recorded back to the KB as a situation – decision – result (SDR) feedback:

- A situation represents the circumstances in which the adviser is used. It includes the problem type and description, where the problem originates, when it occurred, list of all the shop floor elements involved.
- Decision represents the final solution which was applied by the system operator and why. After the exchange of information using the dialog manager, he alone decides what to do.
- Result represents the applied solution, when it was applied and if the system operator's decision was positive, neutral or negative.

5.4 Characteristics

Quick, quality and effective system operator decisions have a high impact on the overall system performance and efficiency. The main workflow within the control system is based on the interaction between the IAM and the system operator. This cooperation can be viewed as a human operator – AI system symbiosis, where each side has its own advantages and limitations.

In critical decision moments the system operator is in a high stress environment. This can lead to rushed, bad quality or less optimal decisions. IAM proposes what to do but as with any other software, errors are possible. This can include false readouts, incomplete database, programming bugs, etc. Overall, the IAM as a system has characteristics which define it.

IAM has the following advantages:

- Flexible – such a system can be implemented in other production systems with high technical similarity. Here, large amounts of data are produced and where fast and quality decisions need to be made. Additionally, it can contain valuable knowledge from multiple experts, which would otherwise take a long time to acquire.
- Dynamic – the IAM capabilities to learn means that the system adapts to the working environment. The knowledge base constantly increases and as a result the adviser function becomes more accurate and helpful.

- Error detection – descriptive and predictive data analysis could enable detection of irregularities in the early stages or even before the start of assembly.
- Promotes human centric systems – the skill, experience and intuition of the system operator can be combined with the analytical and interconnected functions of the IAM in order to achieve quick and effective solutions.
- Reduces stressful decision-making environment for the system operator, improves the quality of decisions and reduces time needed to make them. Additionally, it offers consistent expert advice available all the time, and does not get tired, overworked or forget facts.

IAM can have the following disadvantages:

- False warnings are possible – this can happen due to faulty sensors, invalid prediction models, software bugs, etc. Additionally, mistakes in the knowledge base can lead to incorrect decisions – (error in / error out).
- Expensive implementation – setup time, infrastructure and maintenance issues can demand high initial costs.
- Knowledge extraction problem – it can be difficult to define and to maintain all KB. Additionally, it can be difficult to extract or represent all types of knowledge as well as to transfer the context and the content of the human expert knowledge to the IAM.
- Accuracy – the effectiveness of the system improves with higher amounts of stored data. This means that younger systems are less accurate. Additionally, due to technical limitations, the questions can be misunderstood and there is lack of human capabilities (common sense, emotions).
- Human rejection – human operators reject the suggestions of the IAM. It is not uncommon that humans do not completely trust or want to rely on computer assistance (“I know better than a machine” effect).

The characteristics comparison between the system operator and the IAM are shown in Table 5.2. Human characteristics of the system operator are defined by his high intuition, natural ability to learn and context awareness. His high flexibility is realized through the ability to adapt to each situation and to find a solution even if faced for the first time. However, he has limited mental and physical capacity to perform the same over longer period of time (Machizawa et al., 2012).

His concentration declines even faster if the decision-making environment is stressful (Flin, 1997). There is always a possibility of a human error because he becomes easily tired. On the other hand, the IAM is a robust system whose performance is improving the longer it is operational. Although it has a limited learning capacity, it has high analytical and (theoretically infinite) memory capabilities. In such a human – machine interaction, the strengths of one side should reduce the weaknesses of the other side.

Table 5.2. Characteristics Comparison

Characteristic	System Operator	Intelligent Adviser Module
Intuition	High	Low
Flexibility	High	Low
Context awareness	High	Low
Learning ability	High	Limited
Short term memory	Medium	High
Long term memory	Low	High
Information capacity	Low	High
Analysis	Medium	High
Robustness	Easily tired	Does not need rest

5.5 Realization

The operation of the IAM is set within the digital domain. This means that computer hardware and software need to be used to facilitate its realization. The system operator is not expected to fix hardware or to debug software. These tasks are to be completed by IT experts and knowledge engineers respectively. The main system operator task is to ensure that the execution of BAS working scenarios is realized as close as possible to what is planned.

The discussion about hardware (CPU, memory, storage etc.) as well as software specifics (code type, OS platform, etc.) is beyond the scope of this dissertation. This is because the state of the art in this domain changes rapidly as demonstrated by Moore's Law (Schaller, 1997) and by the observed increase of performance to price ratio in the last few decades of computer utilization (Gray & Shenoy, 2000).

Instead, it makes much more sense to analyse the current IT demands and infrastructure. Based on the observations (Iyengar et al., 1997) and (Nakajima et al., 2002), it can be concluded that server grade hardware and software are the most optimal platform for the realization of the IAM. This is because they need to respond to the following challenges:

- Durability – it needs to have the ability to endure prolonged and high demands over a period of its operating lifetime.
- Stability – during its operating lifetime it needs to be able to perform with minimal setbacks or down times.
- Reliability – it needs to be able to maintain a constant level of performance during its operating lifetime.
- Scalability – it needs to be able to accommodate any future updates, expansions or increase of resources according to demand or availability.
- Speed – any bottlenecks need to be avoided and its operations should be as fast and optimized as possible.
- Redundancy – if and when a failure occurs, it has to be easily replaceable so that any downtime can be avoided or at very least minimized.

- Safety – the system needs to be protected from any external (or internal) unauthorised access as well as all data needs to be encrypted.
- Backup –a backup has to be always made so that in case of serious malfunctions data can be quickly restored.

5.5.1 Types of proposals and their use by the system operator

For the system operator, the most useful outputs from the IAM are proposals. He can use them in various scenarios. Based on problem complexity, information accuracy and knowledge amount, there are two principle types of proposals which can be defined :

- Implicit proposals – If a problem is very complex, or there is insufficient data or the IAM Knowledge Base does not contain any applicable knowledge, the IAM outputs implicit proposals. These can be in form of an overview, statistics, comparisons, etc. The main goal of such proposals is to give an inspiration or an idea to the system operator in which case they can be more generalized and open to interpretation (For example: the battery health of MR1 is at 93% which is higher when compared to the battery health of MR2 which is 15%). This information does not specify what the problem might be exactly, but it might be useful to the system operator to come closer to a solution.
- Explicit proposals – these proposals are a result of the IAM's applicable Knowledge Base and sufficient information. In this case, the IAM is able to specify what the problem is and how to resolve it. As a result, the system operator is given explicit proposals. These can be in form of necessary steps, precise descriptions or problem definitions. The main goal of such proposals is to free the system operator from active contemplating in cases where he does not have the time, knowledge or will.

Fig. 5.9. shows the position of the system operator in contrast to proposals. In principle, he can accept, reject or inspect the validity of proposals. Which action he will choose, depends on the situation context and other certain factors.

When the system operator is faced with a new problem, the first factor which comes in play is does he have sufficient level of knowledge to recognize and solve it. If no, he does not have any choice but to accept the proposal.

If yes, then the context of the situation plays a vital role. In other words, is he exposed to low stress levels or does he have enough time to contemplate about a decision. If no, he does not have any choice but to accept the proposal.

If yes, does he suspect in the validity of the proposal. If no, does he agree with the proposal. If he has the knowledge, time and agrees with the proposal, then naturally he accept it. On the other hand, if he does not agree, he will reject it.

If the system operator suspects the validity of the proposal, that means that he has sufficient knowledge and time to inspect the proposal. According to the inspection, if the proposal is valid it is accepted and if it is not valid, it is rejected.

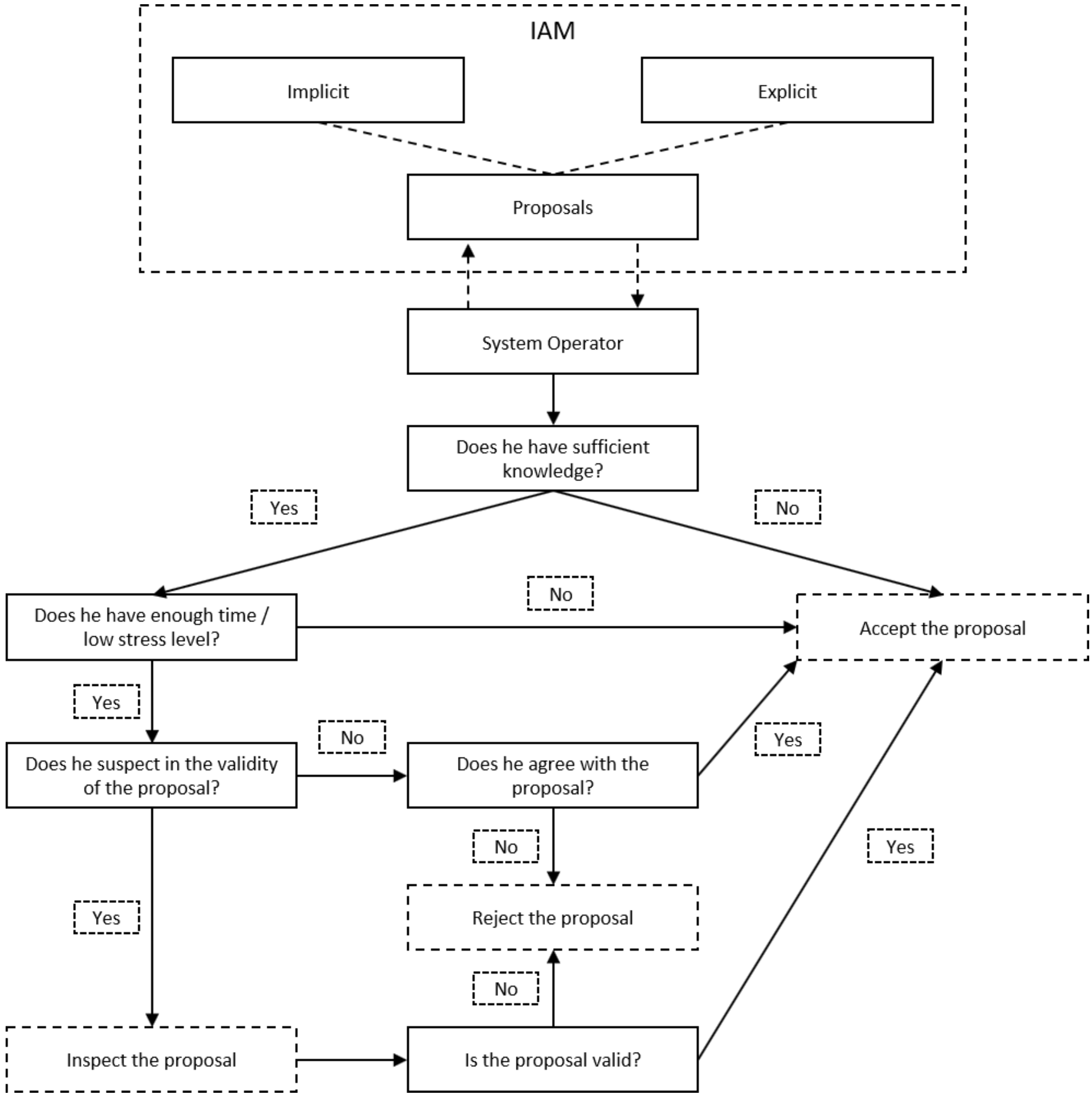


Fig. 5.9. Types of proposals and their use by the system operator

5.5.2 Adaptive learning

BAS shop floor elements represent a set of comparable hardware and personnel (Haskovic et al., 2017). This includes same or at the very least, similar mobile robots and assembly stations as well as similarly trained shop floor operators. During the execution of BAS working scenarios, these elements can be performing similar tasks respective to their function. Additionally, during their simultaneous operations they can have similar performance and statuses.

During their operational lifetime in BAS, these elements produce digital data which is recorded. This data is used to produce and accumulate new knowledge. However, hardware can be replaced with new type of machines, each shop floor operator can have individual work habits or methods etc.

The issue here is – from where does the new knowledge accumulation begin? When a new shop floor element is introduced how does the IAM compensate for the potential new element behaviour? In other words, how to be sure that the used, already accumulated knowledge is up to date and useful? This is very important because if the IAM is using inadequate knowledge the proposals could be wrong and cause even more serious problems.

In order to avoid this, all new sources of knowledge have to be documented, indexed and verified through adaptive learning as shown in Fig. 5.10. IAM's adaptive learning algorithm automatically forms groupings from digitally recorded data. These groupings are based on similar operating parameters from individual shop floor elements and their working conditions. All groupings are contained and organized within the IAM Knowledge Base.

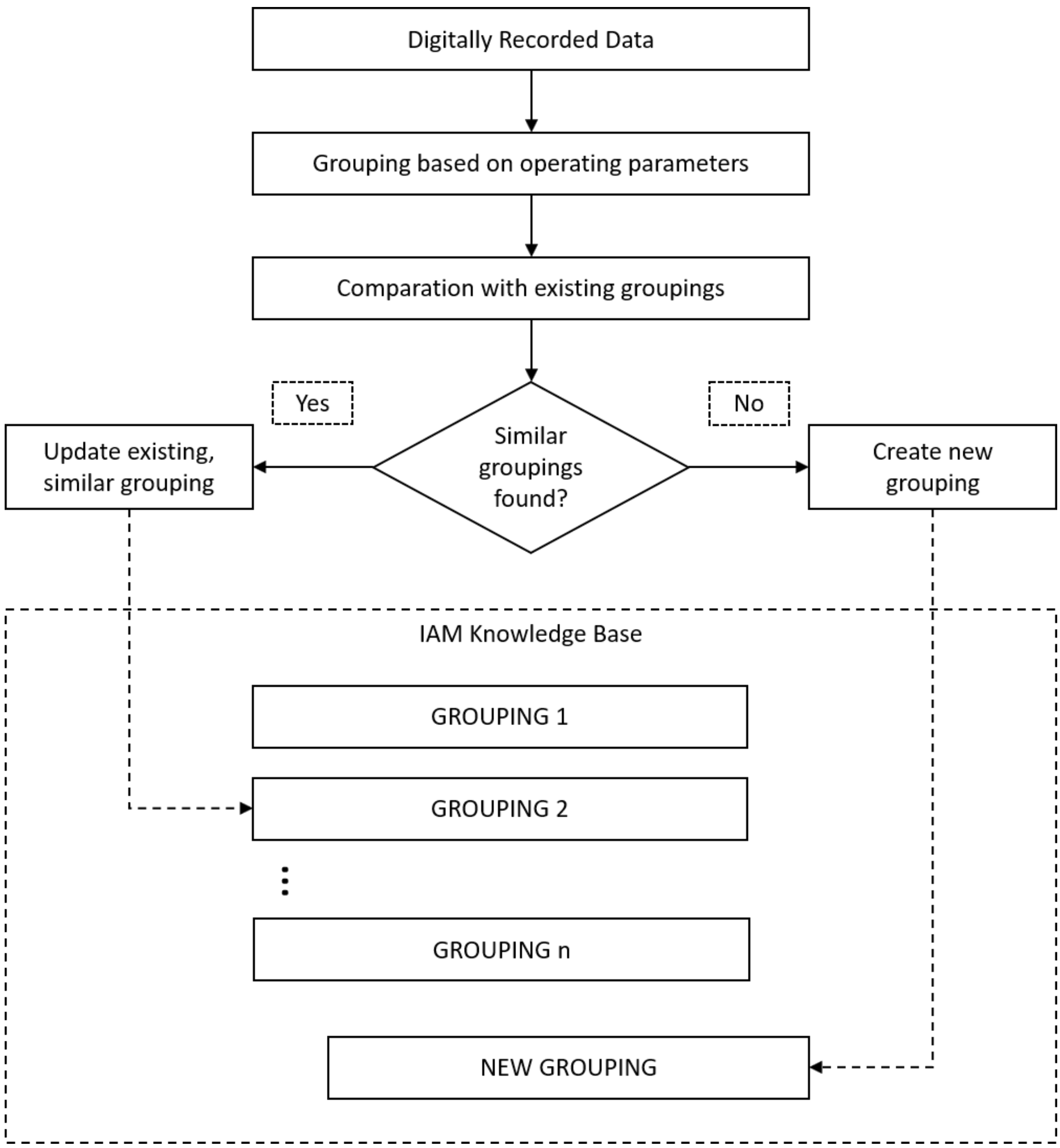


Fig. 5.10. New source of knowledge verification through adaptive learning

This algorithm continuously compares the newly created input data with existing groupings and tries to find the most similar grouping to that input. This search can have the following two results:

- Similar grouping found. The element which produced this input data will be associated with the performance of the identified grouping. That means its behaviour is within specified tolerances and previous knowledge can be applied. Any possible, minor differences between this input data and the grouping are used to update the grouping.
- Similar grouping not found. It means that the element which produced this data is operating in a new manner and previous knowledge should not be applied. When this happens, this input data becomes a base for a new grouping. When there are enough of such out-of-bound inputs it means a new behaviour is emerging from that specific shop floor element. This defines the adaptive nature of the IAM where its accumulated knowledge base can be used even when elements are changing. A similar monitoring principle was successfully implemented by the Komatsu company for the assessment and prediction of their heavy equipment engine health status (Lee et al., 2014).

5.6 Working modes

During his shift in BAS, the system operator needs to constantly reach decisions. These decisions are based on fragmented and incomplete information about the actual system states and its components. On one side, the main source of information is based on his communication with other humans and on the other side, from the audio / visual feedback as shown in Fig. 5.11. The main digitally recorded data stream in BAS represents a vast collection of data. It is formed between the control and the controlled systems. This includes all the commands, responses and status messages.

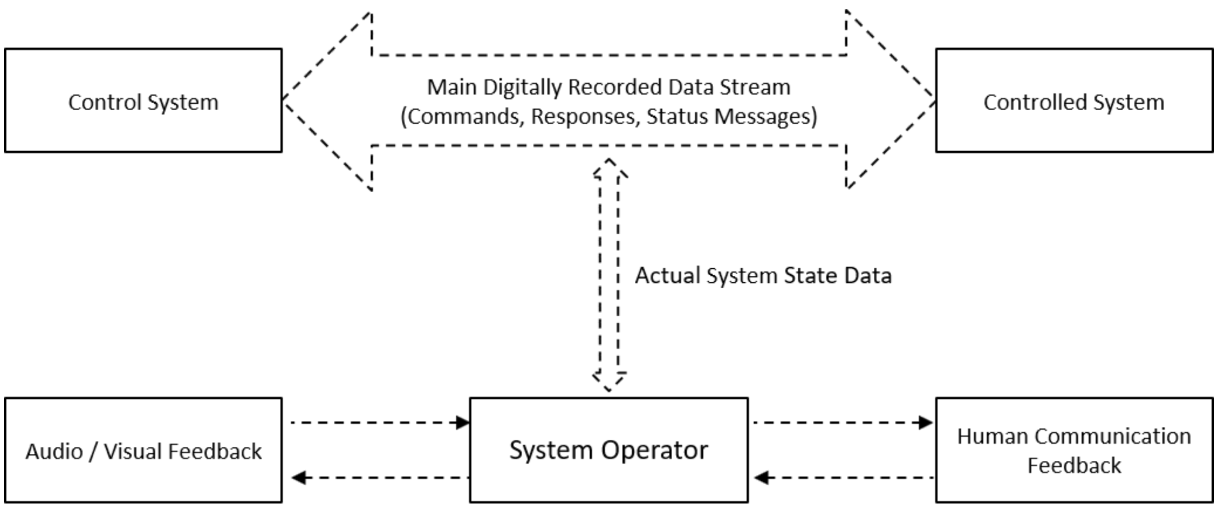


Fig. 5.11. Main sources of information for the system operator

It is impractical and moreover, impossible for the system operator to completely utilize all this data. For this reason, only a small part of this data stream is actively presented to the system operator in real time. The main purpose of this data selection is to give data about actual system states at the system operator's disposal.

In order for the IAM to give relevant and useful information to the system operator, the target and the actual state of the system are constantly monitored.

5.6.1 Target and the actual state of the system

Functionality and efficiency are the two most important requirements of BAS. The main goal of the entire system is to achieve efficient processing of continuous stream of one or more parallel assembly orders. One assembly order means to assemble one run of product. Each product is assembled according to a defined sequence of operations. For every operation, there is a group of assembly stations which can complete them.

To ensure that BAS is working with the highest possible efficiency, it has to be organised so that the minimal sum of lost assembly station time is achieved. This can be accomplished through the organisation of the system according to the following secondary goals:

- Assembly station standstills are avoided by ensuring their uninterrupted work. To achieve this, every station has to have the necessary NC programs, tools, workpieces and various other resources at their disposal at the right time.
- Assembly station setups should be as short as possible. Setup procedures should start without delay and all necessary resources have to be available at the right time.
- Removal of assembly station standstills. In case that a station is not working, it has to be brought back to a working state as soon as possible.

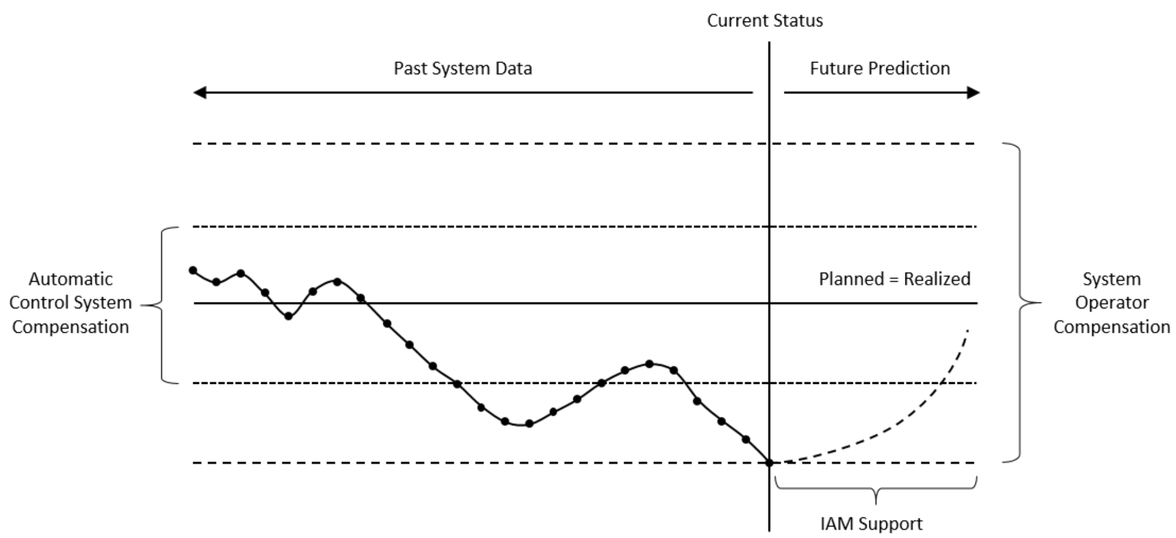


Fig. 5.12. Intelligent Adviser Module working modes

For this reason, target and the actual state of the system are monitored. There is always a difference between planned and realized working scenarios as shown in Fig. 5.12. System efficiency directly depends on this difference. Smaller difference means higher system efficiency and smaller number of minor disturbances. Depending on the size of the difference, there are three main working modes during the execution of BAS working scenarios: normal, transition and disturbance modes. In any case, the IAM is always available. The system operator chooses if he will accept, partially accept or ignore the IAM proposals.

5.6.2 Normal working mode

Small differences between planned and realized working scenarios are compensated by the automatic control system and the self-organizing nature of the shop floor. In this mode, the entire assembly process is regulated without the intervention from the system operator. These small differences are caused by chance causes. They are negligible and inevitable variations which occur in random manner. Such variations cannot be anticipated, detected, identified or eliminated from practical or economic reasons (****, 2016).

Causes of small differences can include (Deming & Edwards, 1982):

- minor variations in operation start / end times
- mobile robot traveling times
- occasional failed quality control checks
- slight variations in raw material
- vibrations caused by operating hardware
- normal wear and tear
- computer latency
- minor imprecise shop floor operator actions
- working conditions (light, noise, temperature, humidity, dust, ventilation...)

For example, if a product during its assembly did not pass a quality control check, the system reroutes it to a repair station. However, a certain percentage of errors are non-reparable. In that case the automatic compensation system needs to ensure that the exact number of ordered products is completed. That means that the defective product is discarded, and an automatic replacement order takes place. If there are multiple failed products it implies that there are more serious problems in the system. The automatic control system cannot compensate for it.

5.6.3 Transition mode

The automatic control system is limited. If the difference is increasing beyond its ability to compensate it, the system operator is notified to actively bring the system back to a normal working mode.

The cause of these more serious differences are called assignable causes. They result in a larger amount of variations.

They can be detected, identified and eliminated and as such do not present a significant problem for the system operator. If he chooses, he can solve them himself or he can use the help from the IAM.

Transition mode can be caused by:

- Defective raw material
- Shop floor operator is absent or asleep
- Wrong equipment adjustment or calibration
- Defective mobile robots
- Assembly station malfunction
- Power fluctuation

5.6.4 Disturbance mode

Extreme differences between planned and realized working scenarios mean that there is a problem caused by multiple errors. In such a disturbance mode, the system operator does not have enough time or ability to make a decision. He needs to be presented with the most important information which includes problem description, possible explanations and a solution.

For this reason, he uses the IAM as a support tool which helps him to reach higher quality decisions in good time. Disturbance mode can be caused by previously described assignable causes when they occur at the same, and can result with:

- Bad consecutive quality of multiple assembled products
- Inability to keep up with deadlines
- Multiple assembly station failures

In order for the IAM to successfully operate for what it was designed, its performance needs to be repeatable and predictable. Additionally, it is very important that there is an efficient interaction with the system operator.

However, it can be assumed that there will be multiple different workers performing the duty of a system operator during the lifetime of BAS and the IAM. All system operators have varied levels of experience. As a result, their experience has a direct influence on the quality of decisions and time needed to make them.

BAS and IAM are in their concept stage. For this reason, real-world performance data is unavailable. An experiment needs to be set-up to investigate the contribution of IAM proposals on the quality and time of decisions.

5.7 Summary

This chapter described the IAM as an integral part of the BAS control system. As a result of this integration, IAM is able to learn and to improve the accuracy of its proposals over time. This makes it possible for the IAM to operate during the execution of various working scenarios which can occur during BAS lifetime.

The IAM structure is composed from system monitoring, knowledge management and decision support submodules. The main IAM functions include planning and simulation, diagnostic and advising. The IAM needs to utilize a familiar, simple, consistent, and logical graphical user interface which enables an efficient interaction with the system operator.

The work of the IAM is based on the actual system state data derived from the digitally recorded data stream between the control and the controlled systems. According to the size of the difference between the planned and the realized execution of the working scenarios, there are three main working modes: normal, transition and disturbance.

Next chapter will describe the interface experiment set-up. The purpose of this experiment is to investigate the contribution of IAM proposals on the quality and time of decision in conflict situations and with multiple human operators.

Chapter 6

IAM Experiment Set-up

The main purpose of the IAM is to support the system operator during his decision-making process. To achieve this, the IAM is conceptualized as an integral part of the BAS control system. The following IAM key components have been defined: Structure, functions, characteristics and working modes.

The initial results show that the IAM could be successfully integrated in the control system of BAS. The main advantage is the possibility to include actual system states, predefined expert knowledge, external facts and definitions with new system specific knowledge. This makes it possible for the quality of IAM proposals to constantly improve.

All BAS production activities which should be completed during a set timeframe (Year, month, week…) and intensity (shifts) are defined at the factory level. Regardless of the set plan, it can be assumed that there will be multiple, different workers performing the duty of a system operator (as a single, final decision maker) during the operational lifetime of BAS and IAM.

Workers do not have same performances. Each system operator is defined by its own capabilities, age, gender, concentration, knowledge, personality, habits and most important, level of experience. The latter has a direct influence on the quality of decisions and time needed to make them.

BAS and IAM are in their concept stage. For this reason, real-world performance data is unavailable. An experiment needs to be set-up. Therefore, this chapter will describe the experiment set-up, in which the interaction between multiple system operators and a simulated IAM will be examined in various case studies. The purpose of such an experiment is to investigate the contribution of IAM proposals on the quality and time of decisions according to specific situations as shown in Fig. 6.1.

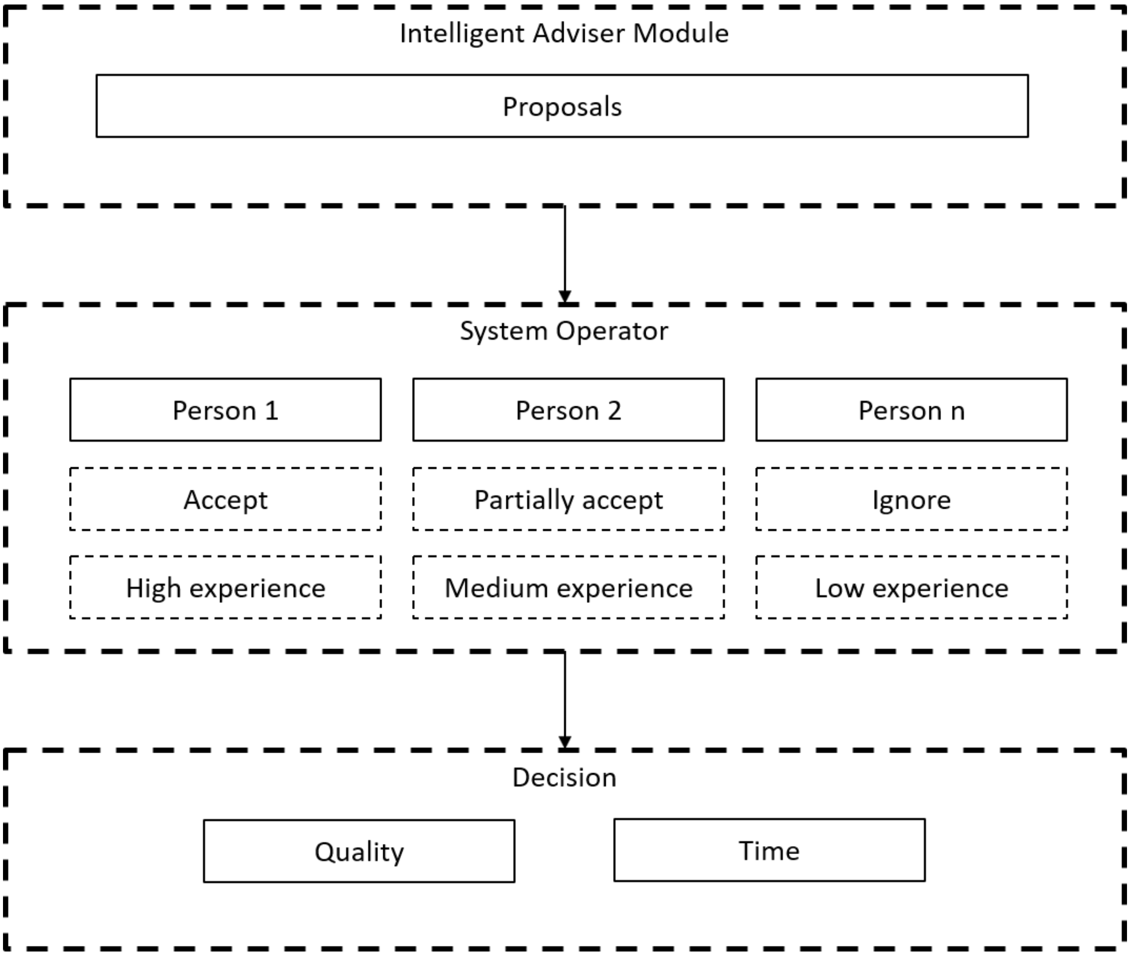


Fig. 6.1. Investigation of contribution of IAM proposals

6.1 Method

A laboratory "*2D problem compensation*" experiment has been devised. Its main purpose is to investigate the following:

1. Do IAM proposals help the system operator to reach higher quality decisions in shorter time with regards to situation complexity?
2. How to achieve a good balance between higher quality of decisions and shorter time with regards to multiple system operators with varying levels of experience?

Before the experiment is set up, there are a number of assumptions which will be incorporated into the experiment. These include:

- IAM is fully functional and implemented in the BAS control structure
- IAM is operating according to expected real life performance
- There are multiple workers performing the duty of the system operator
- System operators do not have the same level of experience

The main experiment concept is shown in Fig. 6.2. The human subject needs to move the red square which represents an error to the zero position which represents the solution. The movement is completed using the 4 arrow keys (left, right, up, down) on the keyboard. Each keyboard press is registered as one user movement. The movement is not autonomous. That means that the red square will not move on its own in the selected direction. The movement is not continuous. That means even if a key is constantly being pressed, that results only with a single movement.

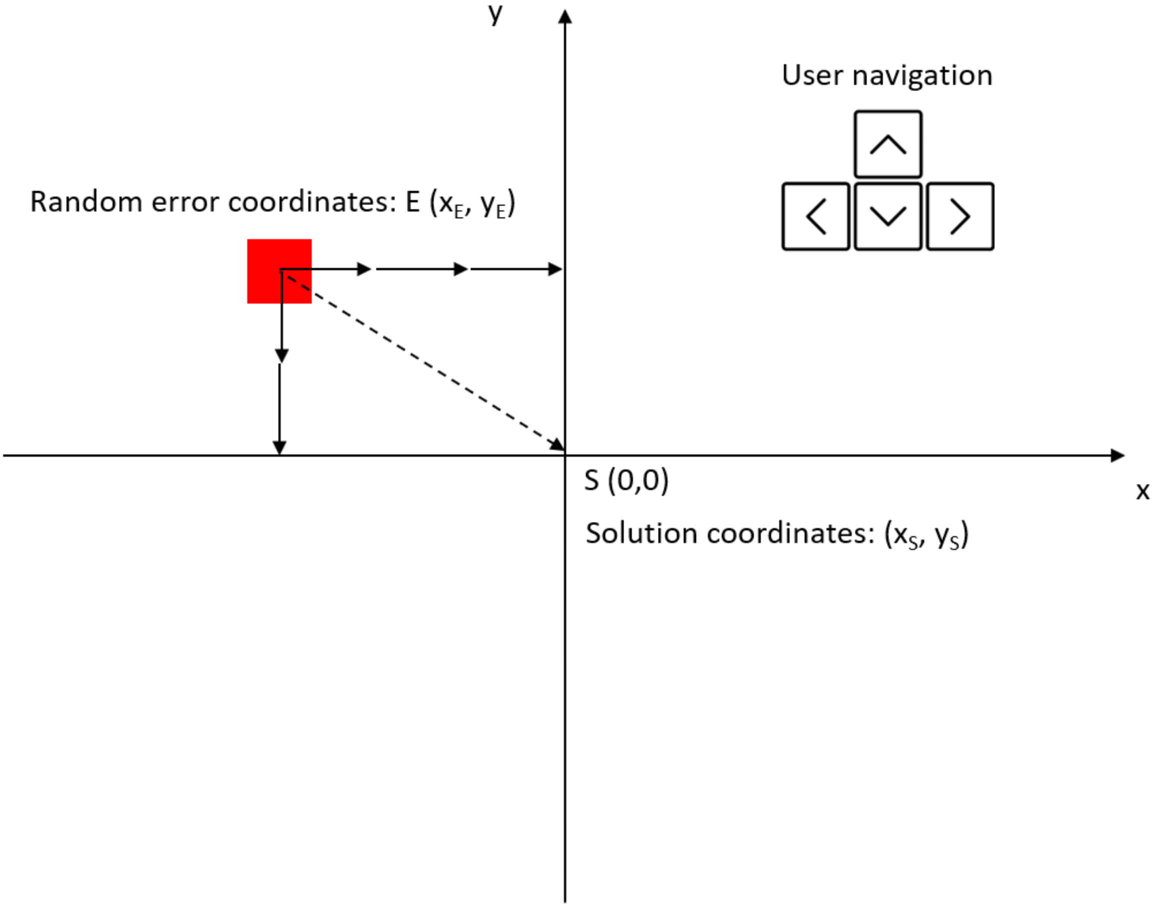


Fig. 6.2. Experiment concept: "*2D error compensation*" task

The entire area is divided into an invisible grid area. Counting from the solution coordinates (0,0), there are 10 moves possible in each direction before reaching the edge of the solving field. Diagonal movement is achieved when pressing a combination between the appropriate up/down and right/left arrow keys. In any case, one diagonal movement is counted as 2 steps.

When the edge of the field is reached, each further movement in the direction of the edge is not counted as a step – the red square needs to change its position in order to count the steps. Each time the square is brought to the zero position (0,0), its colour turns into green, and after a half of second pause, it resets itself. The reset means that it changes its colour back to red and its new coordinates (X_E, Y_E) are randomly generated.

6.2 Tools

The previously described experiment test platform was developed in a source code editor called Notepad ++ (*****, 2018) and tested in a standard web browser. The entire code is available in Appendix B. To achieve the set look and behaviour of the experiment, a number of development tools was used as shown in Fig. 6.3.

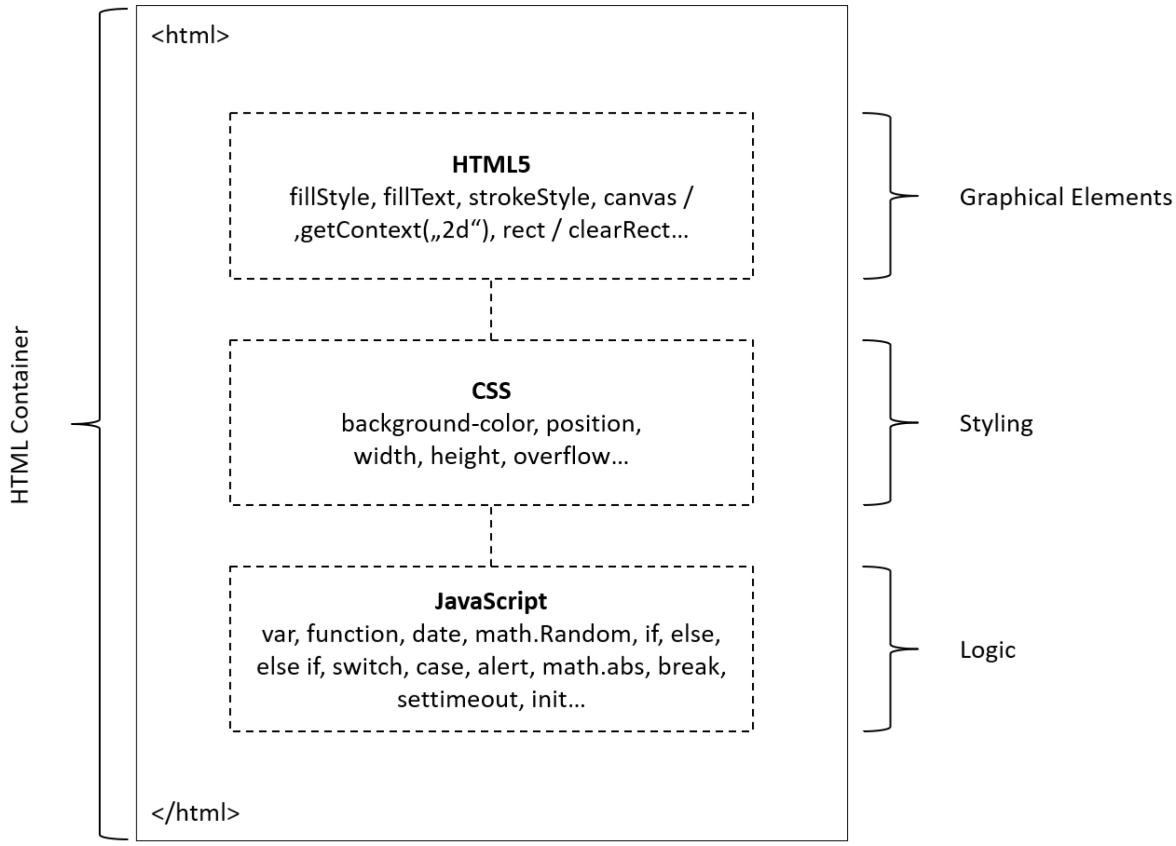


Fig. 6.3. Development tool for the experiment platform

These include:

- HTML format was used for the web site container. This allows the experiment to be easily sent and opened by any computer equipped with a modern web browser.
- HTML5 was used to represent the graphic elements. This includes the X and Y axis movable square, refresh rate, counter elements etc.
- Internal CSS was used to define the style of all the presented elements. This includes positions, colours, fonts etc.
- JavaScript was used to program the logic. This includes keypress actions, counting, moving, data generation, IAM simulation etc.

6.3 Description

Fig. 6.4. shows how the final, developed test field looks like when rendered in a web browser. There are 3 windows. First window is the already described problem solving area, where the subjects perform the compensation task. In it, a run counter is present. It has a simple function to display how many runs in a particular case study have been solved. Once the counter has reached the predetermined value, it presents a visual warning to the subject that the test is over. The more runs in a specific case there are, the more accurate the collected data will be.

The second window, the bottom left, light blue area is used for performance data collection. Each time the subject successfully brings the red square to the zero position (0,0) a set of performance data for that run is recorded for later analysis.

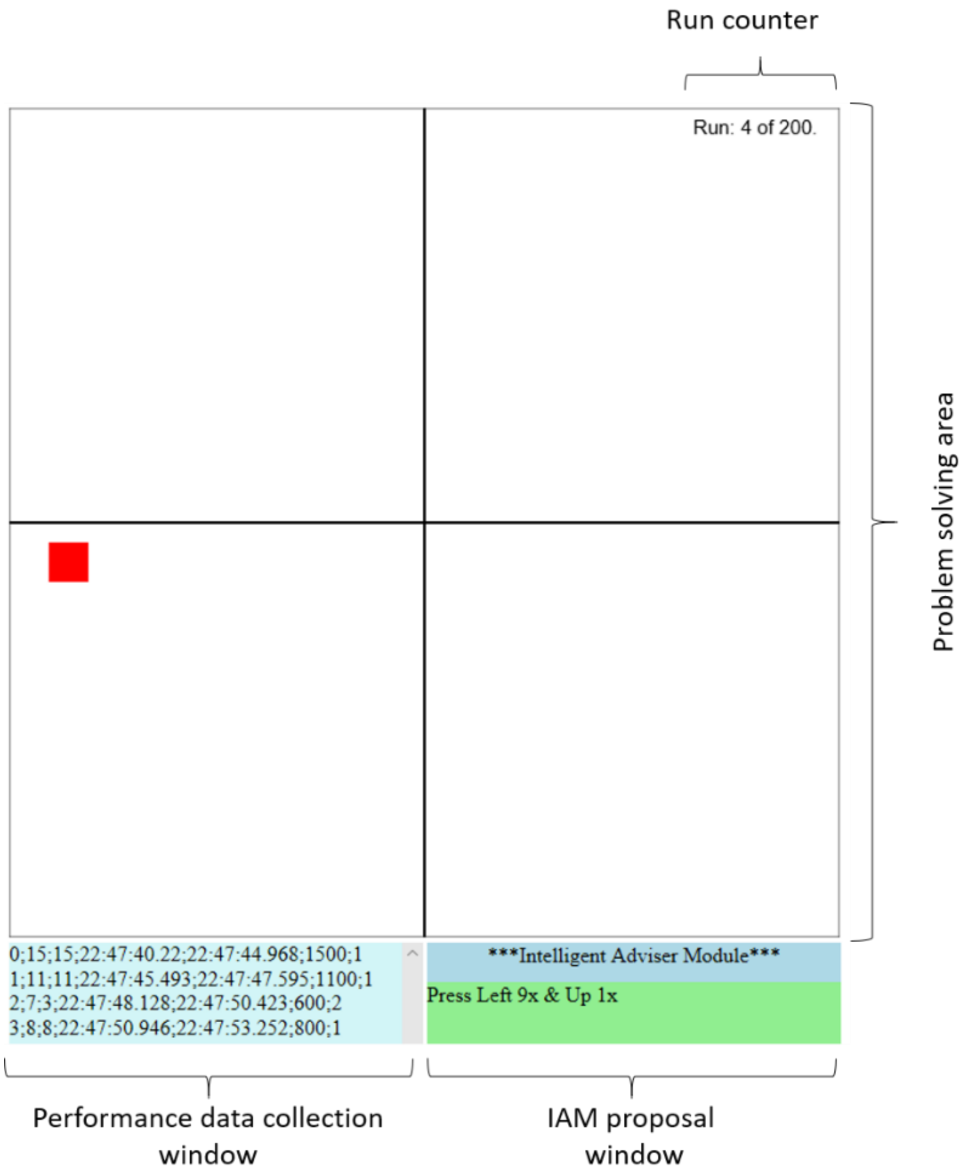


Fig. 6.4. Experiment in browser overview

The third window, the bottom right, light green area is used to simulate the IAM proposals. According to the needed case study, it can be turned on or off. These proposals are always displayed after a specific time delay.

This simulates the real-life performance of the IAM. It is to be expected that there will be always a specific delay according to the complexity of the situation. This delay depends on the size of the error and the level of disturbance.

With increase of the size of the error or the level of disturbance, the delay will be greater. The size of the error represents how far the red square position (X_E, Y_E) is from the zero position (0,0).

Level of disturbance is an additional variable which can be adjusted according to the needed case study. There are 3 levels of disturbance:

- Level D1 represents normal working scenarios. Here, all arrow keys have the same movement as displayed on the keyboard. For example, if the subject wishes to move the square to diagonally lower-right, he/she would press the down and right keys.
- Level D2 represents transition mode. Here, only two arrow keys remain the same. The other two have their function randomly reversed. For example, if the subject wishes to move diagonally upper-left, he/she would press the up and right keys.
- Level D3 represents disturbance mode. Here all arrow keys have a different function randomly assigned to them. For example, if the subject wishes to move diagonally lower/left, he/she would press the up and right keys.

The higher the level of disturbance, the more the subject will have to use his experience and "relearn" the arrow key functions. This will have an affect on time and quality to solve the problem. To help the subjects to minimize such affect, the proposals in the IAM window are shown in the form of "*Press Key 1 x times & Key 2 x times*". That means that the subjects will need to press exactly that combination of keys that many times even if it is not intuitive according to what they see.

However, the IAM is assumed to operate according to expected real life performance. For this reason, an internal logic has been implemented into the experiment. That means that sometimes the IAM can give a wrong proposal or that sometimes the proposal may never come. It is up to the subject to realize when this happens.

If the subject performs movement based on the proposal and this is not decreasing the error, or if it takes unusually long time for the proposal to appear, the subject should react accordingly.

6.4 Case studies

The experiment is organised through following case studies as shown in Fig 6.5.:

- Case A: Simple problems without IAM (200 data entries) – Subjects perform error compensation tasks with constant disturbance level D1 and no assistance from the IAM.
- Case B: Simple problems with IAM (200 data entries) – Subjects perform error compensation tasks with constant disturbance level D1 and with assistance from the IAM.
- Case C: Complex problems without IAM (200 data entries) – Subjects perform error compensation tasks with random disturbance levels D1, D2 and D3 and no assistance from the IAM.
- Case D: Complex problems with IAM – Individual approach (200 data entries) – Subjects perform error compensation tasks with random disturbance levels D1, D2 and D3 with assistance from the IAM. Subjects are instructed to try and reach a balance between quality decisions and good time during their problem solving and interaction with the IAM.
- Case E: Complex problems with IAM – Wait for advice (200 data entries) – Subjects perform error compensation tasks with random disturbance levels D1, D2 and D3 with assistance from the IAM. Subjects are instructed to exclusively wait for an advice from the IAM during their problem solving.
- Case F: Complex problems with IAM – Adjusted for balance (200 data entries) – Subjects perform error compensation tasks with random disturbance levels D1, D2 and D3 with assistance from the IAM. Additional case study adjusted for balance between higher quality and good time.

The experiment is held in the following two runs:

- Run 1 - 10 human subjects marked as (S1.1 – S1.10) perform the following case studies A, B, C, D, E, F.
- Run 2 – additional 15 human subjects marked as (S2.1 – S2.15) perform the following case studies A, B, C, D, E, F.

Experiment expectations:

- IAM proposals should not have a high contribution on the system operator and / or significantly his performance during simple problems.
- IAM proposals should have a higher contribution on the system operator and / or improve his performance during transition and disturbance modes.
- Subject interactions with the IAM will be different.
- Different human subject performances will vary in quality and time – this should simulate varying levels of system operator experience.

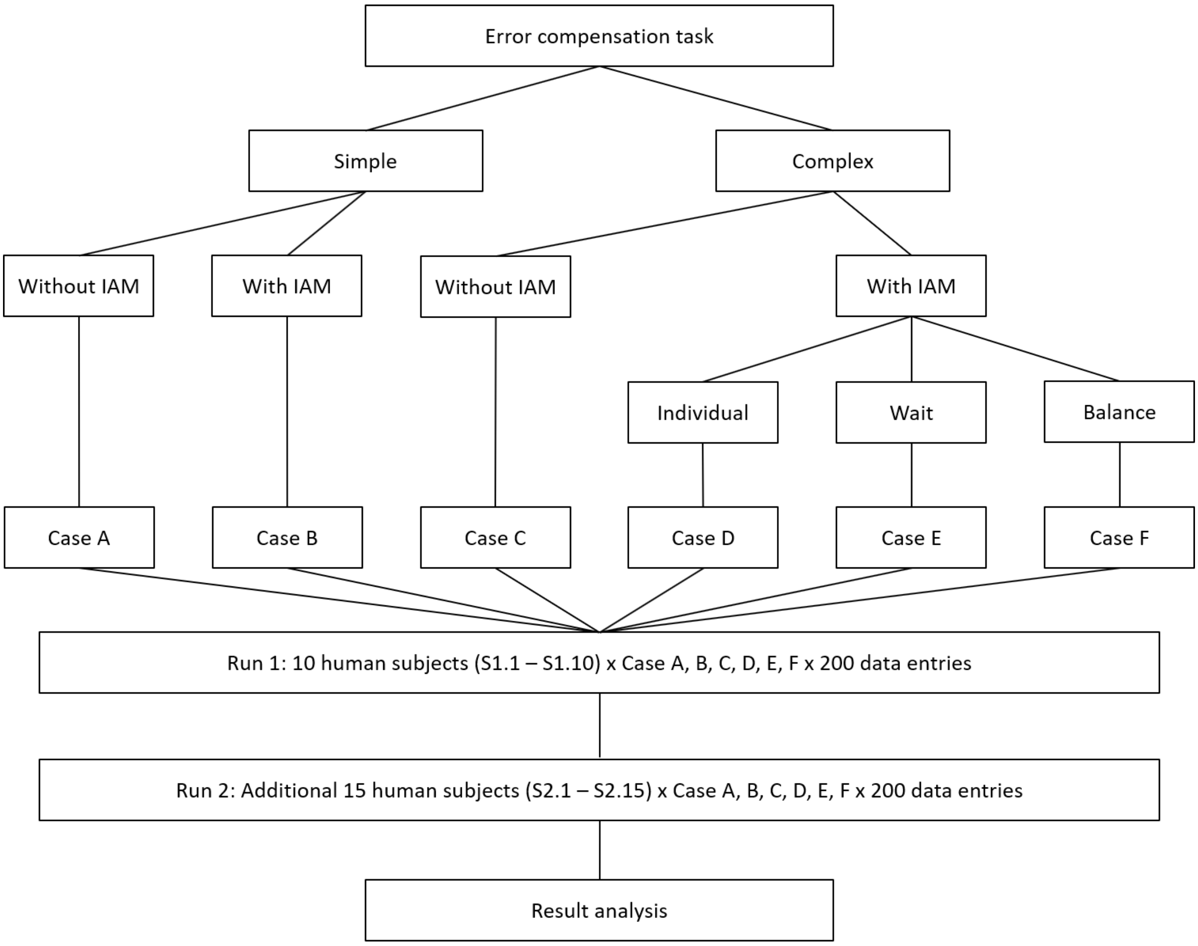


Fig. 6.5. Case studies

6.5 Data

For each case study, during each run, the following data is recorded:

- Run number
- Number of user inputs
- Size of error
- Time when the error occurred
- Time when the error was solved
- Delay
- Disturbance level

All additional data is calculated as:

$$T = T_S - T_E \tag{6.1}$$

where

T – Total time (s)

T_E – Time when random error occurred (hh:mm:ss.ms)

T_S – Time when random error was solved (hh:mm:ss.ms)

$$E = |X_E| + |Y_E| \tag{6.2}$$

where

E – Size of error (minimum number of user inputs)

X_E – Absolute distance between error and solution at the X-axis

Y_E – Absolute distance between error and solution at the Y-axis

$$Q = \frac{E}{n} \tag{6.3}$$

where

Q – Quality of decision (%)

n – number of user inputs (one keypress = one movement in either direction)

$$v = \frac{n}{T} \tag{6.4}$$

where

v – speed of user movement (inputs/s)

$$d = E * D \tag{6.5}$$

where

d – IAM proposal delay (s)

D – Disturbance level (D1, D2, D3)

$$R = \frac{Q * T_{Ideal}}{T} \tag{6.6}$$

where

R – Case ranking (%)

T_{Ideal} – Theoretical ideal time to solve a problem (s) (90% of best case scenario)

Generated data for analysis in table form as shown in Table 6.1:

- 10 subjects x 6 runs x 200 data entries x 11 categories = 132,000 analysis values
- 15 subjects x 6 runs x 200 data entries x 11 categories = 198,000 analysis values

Table 6.1. Example of collected data for analysis

Run	n	E	T_E	T_S	d	D	T	Q	T	*v*
0	17	11	19:23:40.566	19:23:47.523	2200	2	00:00:06.957	64.71%	6.96	2.44
1	16	14	19:23:48.220	19:23:52.379	2800	2	00:00:04.159	87.50%	4.16	3.85
2	5	5	19:23:52.892	19:23:55.253	500	1	00:00:02.361	100.00%	2.36	2.12
3	17	13	19:23:55.760	19:24:01.808	2600	2	00:00:06.048	76.47%	6.05	2.81
4	15	13	19:24:02.314	19:24:07.170	2600	2	00:00:04.856	86.67%	4.86	3.09
5	14	10	19:24:07.517	19:24:12.963	4500	3	00:00:05.446	71.43%	5.45	2.57
6	13	13	19:24:13.470	19:24:18.842	2600	2	00:00:05.372	100.00%	5.37	2.42
7	13	11	19:24:19.356	19:24:23.283	2200	2	00:00:03.927	84.62%	3.93	3.31
8	5	1	19:24:23.793	19:24:28.558	450	3	00:00:04.765	20.00%	4.76	1.05
9	17	15	19:24:29.620	19:24:33.729	3000	2	00:00:04.109	88.24%	4.11	4.14
10	11	11	19:24:34.232	19:24:37.703	1100	1	00:00:03.471	100.00%	3.47	3.17
11	14	12	19:24:38.200	19:24:42.610	5400	3	00:00:04.410	85.71%	4.41	3.17
12	15	11	19:24:42.570	19:24:49.306	4950	3	00:00:06.736	73.33%	6.74	2.23
13	10	10	19:24:49.807	19:24:53.458	2000	2	00:00:03.651	100.00%	3.65	2.74
14	6	4	19:24:53.960	19:24:56.744	800	2	00:00:02.784	66.67%	2.78	2.16
15	16	14	19:24:57.245	19:25:01.263	2800	2	00:00:04.018	87.50%	4.02	3.98
16	7	7	19:25:01.765	19:25:04.400	700	1	00:00:02.635	100.00%	2.64	2.66
17	18	14	19:25:04.516	19:25:09.907	6300	3	00:00:05.391	77.78%	5.39	3.34
18	3	3	19:25:10.419	19:25:12.246	300	1	00:00:01.827	100.00%	1.83	1.64
19	14	12	19:25:12.754	19:25:16.893	2400	2	00:00:04.139	85.71%	4.14	3.38
20	15	13	19:25:17.407	19:25:22.810	5850	3	00:00:05.403	86.67%	5.40	2.78
21	13	13	19:25:22.593	19:25:26.535	1300	1	00:00:03.942	100.00%	3.94	3.30
22	5	5	19:25:27.460	19:25:28.784	500	1	00:00:01.324	100.00%	1.32	3.78
23	6	4	19:25:29.296	19:25:31.817	800	2	00:00:02.521	66.67%	2.52	2.38
24	5	5	19:25:32.315	19:25:34.933	1000	2	00:00:02.618	100.00%	2.62	1.91
25	13	13	19:25:35.434	19:25:39.402	5850	3	00:00:03.968	100.00%	3.97	3.28
26	12	8	19:25:39.904	19:25:43.913	1600	2	00:00:04.009	66.67%	4.01	2.99
27	7	5	19:25:44.422	19:25:47.693	2250	3	00:00:03.271	71.43%	3.27	2.14
…	…	…	…	…	…	…	…	…	…	…

6.6 Summary

This chapter presented the contribution of IAM proposals on the quality and time of decisions as the focus of investigation. For this reason, a laboratory "*2D problem compensation*" experiment has been devised. Additionally, the approach method, used development tools as well as experiment workflow and organisation were presented. As a result, a large dataset was generated and acquired. This data will be analysed, and the results will be presented in the next chapter.

Chapter 7

Experiment Analysis

As previously established, the main focus of investigation is the contribution of IAM proposals on the quality and time of decisions. This chapter presents the "*2D problem compensation*" experiment analysis. This experiment is based on the custom tool developed in the previous chapter. Such a tool allows to perform multiple different case studies. This makes it possible to test how the presence or absence of IAM proposals has an influence on the system operator in a variety of situations, where the intensity of the problem is variable.

After the experiment was performed, a large dataset was generated. The data analysis was performed in Microsoft Excel. As stated, the experiment was performed in two runs. The 10 human subjects from the first run consist from 50% males and 50% females with the overall average age of 37.9 years, and the 15 human subjects from the second run consist from 53% males and 47% females with the overall average age of 39.2 years.

The analysis of such acquired data should allow to reach conclusions and to possibly improve the operation of the IAM. The experiment deals with the following main points:

1. **The use of IAM during simple problems**: Comparison between achieved quality of decisions and time needed to reach them, with and without the presence of the IAM. Analysis performed in order to verify if there is a substantial performance difference.
2. **The use of IAM during complex problems**: Comparison between achieved quality of decisions and time needed to reach them, with and without the presence of the IAM. Analysis performed in order to identify the level of performance differences in various situations.
3. **Wait for IAM advice**: Control comparison between achieved quality of decisions and time needed to reach them, with the presence of IAM where the subjects are told to exclusively wait for an advice from the IAM – Note: this is not how the IAM operates in real application – the system operator, as the final decision maker chooses if he will accept, partially accept or ignore the IAM proposals.
4. **Adjustment for improved balance**: Additional experiment performed with an adjustment in the function of the IAM. This adjustment serves to try and achieve an improved balance between the quality of the decision and time needed to reach them. Results are compared and analysed.

7.1 The use of IAM during simple problems

7.1.1 Quality Comparison – Cases A & B

First run comparison of the quality of decision between cases A and B where the higher values on the graph are better is shown in Fig. 7.1. For each of the 10 subjects (designated from 1.1 to 1.10) there is a blue bar representing the problem solving for simple cases without the help of the IAM and the orange bar representing the problem solving for simple cases with the help of the IAM. Average values from each of the subject's 200 runs in a specific case were taken and compared.

With 50% of the subjects, the quality of the decisions decreased in the range from min. 0.3% till max. 1.7%. With the other 50% of the subjects, the quality of the decisions increased in the range from min. 0.02% till max. 5%.

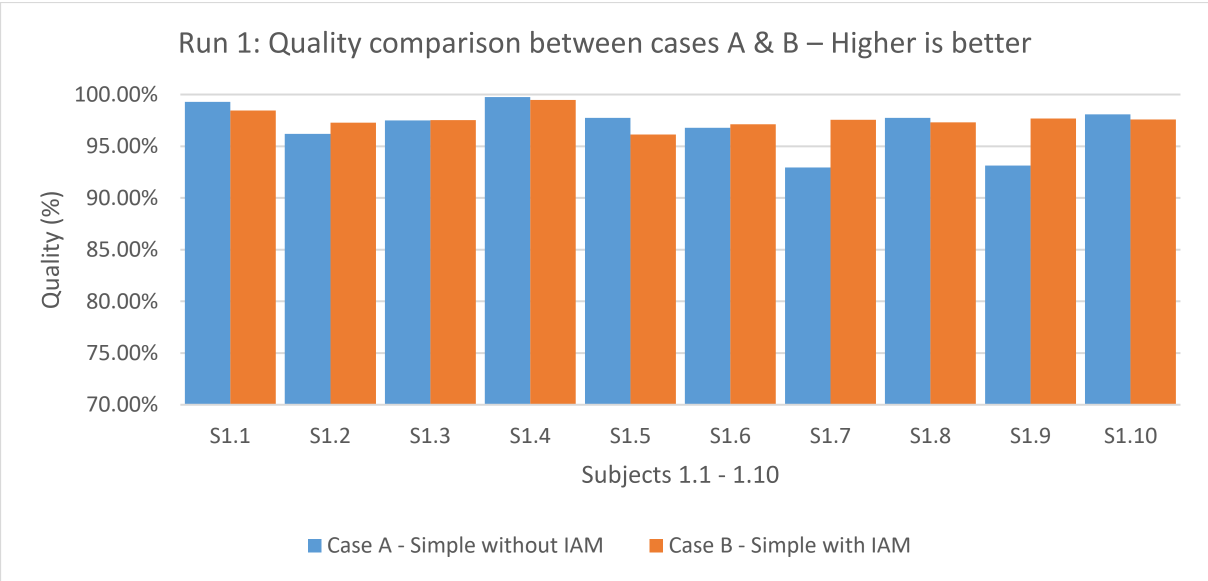


Fig. 7.1. Run 1: Quality comparison – Cases A & B

As previously stated, a second identical experiment run is always performed in order to verify the results. Second run comparison of the quality of decision between cases A and B is shown in Fig. 7.2. For each of the 15 subjects (designated from 2.1 to 2.15) there is a blue bar representing the problem solving for simple cases without the help of the IAM and the orange bar representing the problem solving for simple cases with the help of the IAM. Average values from each of the subject's 200 runs in a specific case were taken and compared.

With 35% of the subjects, the quality of the decisions decreased in the range from min. 0.06% till max. 1.8%. With the other 65% of the subjects, the quality of the decisions increased in the range from min. 0.02% till max. 2.8%.

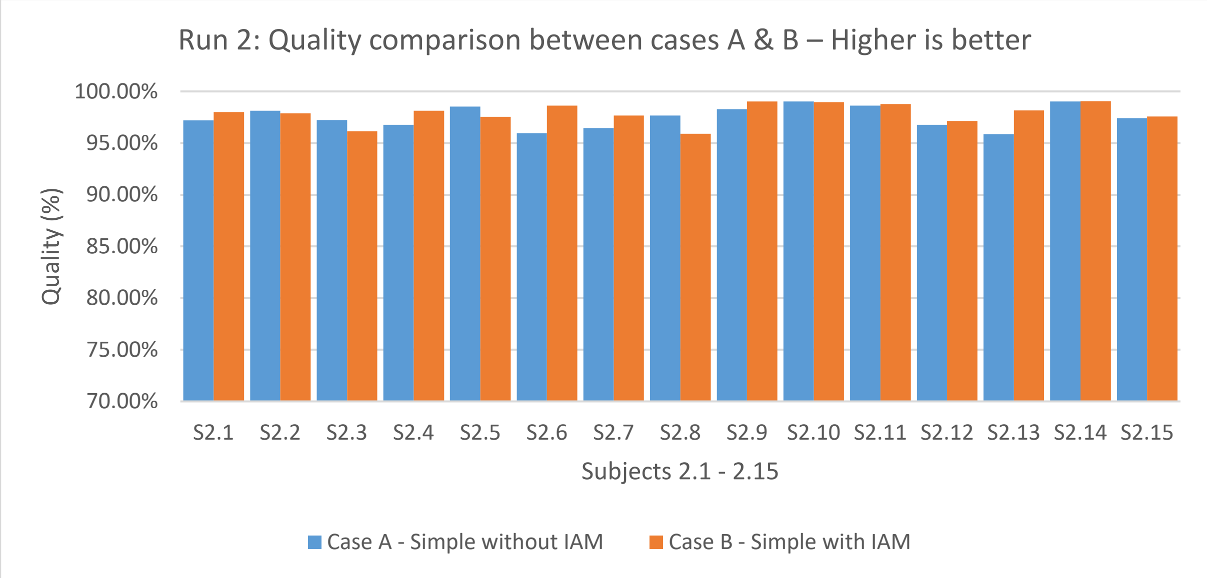


Fig. 7.2. Run 2: Quality comparison – Cases A & B

Finally, an average of all the subjects from the first run and an average of all the values from the second run were taken and compared as shown in Fig. 7.3. Overall, there was an increase of the quality of decisions with the IAM in both runs. However, this increase was not statistically noticeable. In the first run there was an increase of 0.71% and in the second run of 0.56%. According to these values, the IAM does not introduce a noticeable difference when solving simple problems.

This is because, in an ideal world, the system operator would know the type of disturbance in advance. This allows him to recognize the solution and apply it quickly and effectively without having a need for any assistance.

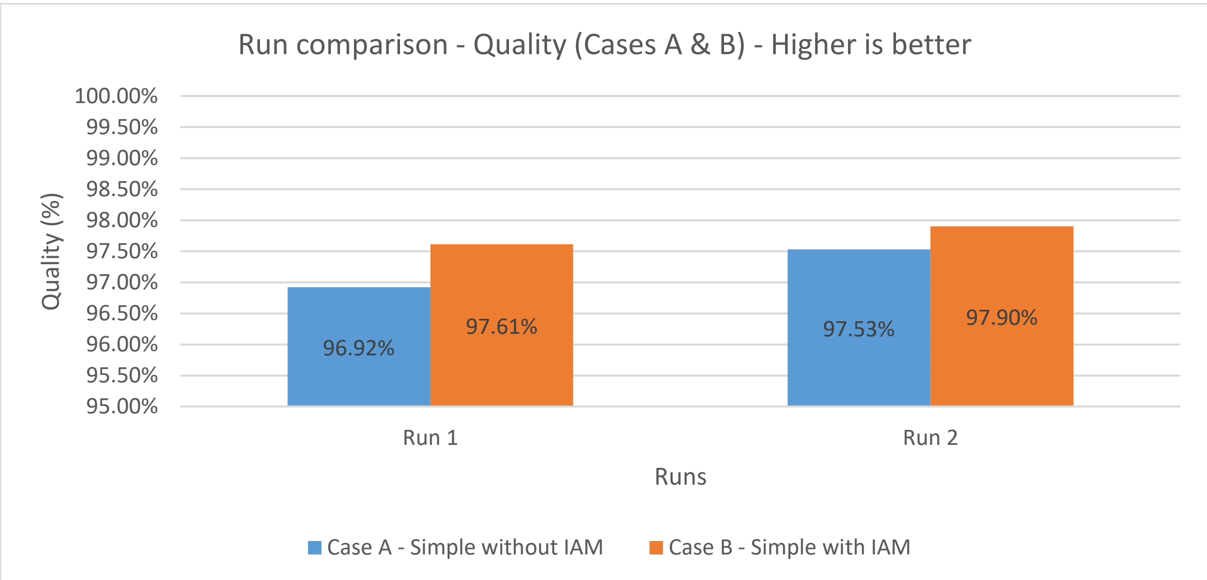


Fig. 7.3. Overall run comparison -Quality (Cases A & B)

7.1.2 Time Comparison – Cases A & B

First run comparison of the time needed to reach a decision between cases A and B where the lower values on the graph are better is shown in Fig. 7.4. For each of the 10 subjects (designated from 1.1 to 1.10) there is a blue graph line representing the problem solving for simple cases without the help of the IAM and the orange graph line representing the problem solving for simple cases with the help of the IAM. Average values from each of the subject's 200 runs in a specific case were taken and compared.

Only 10% of the subjects (S1.7) had a slightly longer time of 0.16s needed to reach a decision with the IAM. This represents an increase of 6.9%. With the 80% rest of the subjects, there was an improvement of the time needed to reach a decision with the IAM. They achieved a lower time ranging from 0.06s till 0.7s or improvement from 2.3% till 23%. One subject (S1.8) achieved a lower time of 2.3s or 56% improvement with the IAM. This is an outlier in the data and does not reflect the overall performance. One possible explanation is that his performance from the first run, without the IAM was lower due to even larger lack of experience with the task than the rest of the subjects.

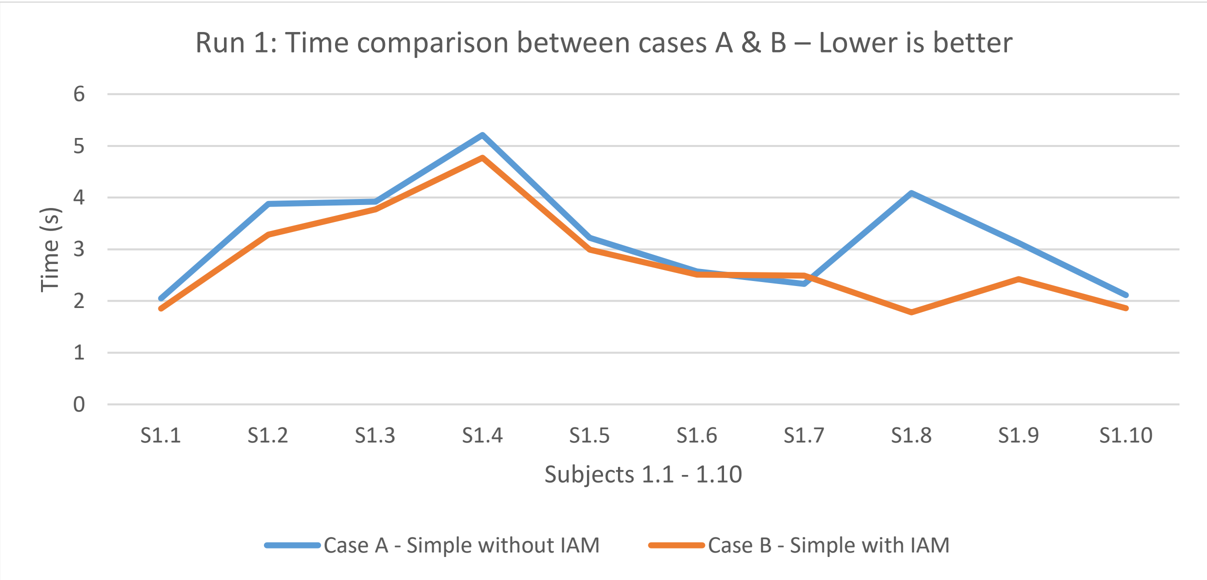


Fig. 7.4. Run 1: Time comparison – Cases A & B

Second run comparison of the time needed to reach a decision between cases A and B where the lower values on the graph are better is shown in Fig. 7.5. For each of the 15 subjects (designated from 2.1 to 2.15) their colour representation is the same as in the first run. Again, average values from each of the subject's 200 runs in a specific case were taken and compared.

35% of the subjects had a slightly longer time ranging from 0.07s till 0.47s needed to reach a decision with the IAM. This represents an increase of 1.73% till 15%. With the 65% rest of the subjects, there was an improvement of the time needed to reach a decision with the IAM. They achieved a lower time ranging from 0.06s till 0.5s or improvement from 1.9% till 20%.

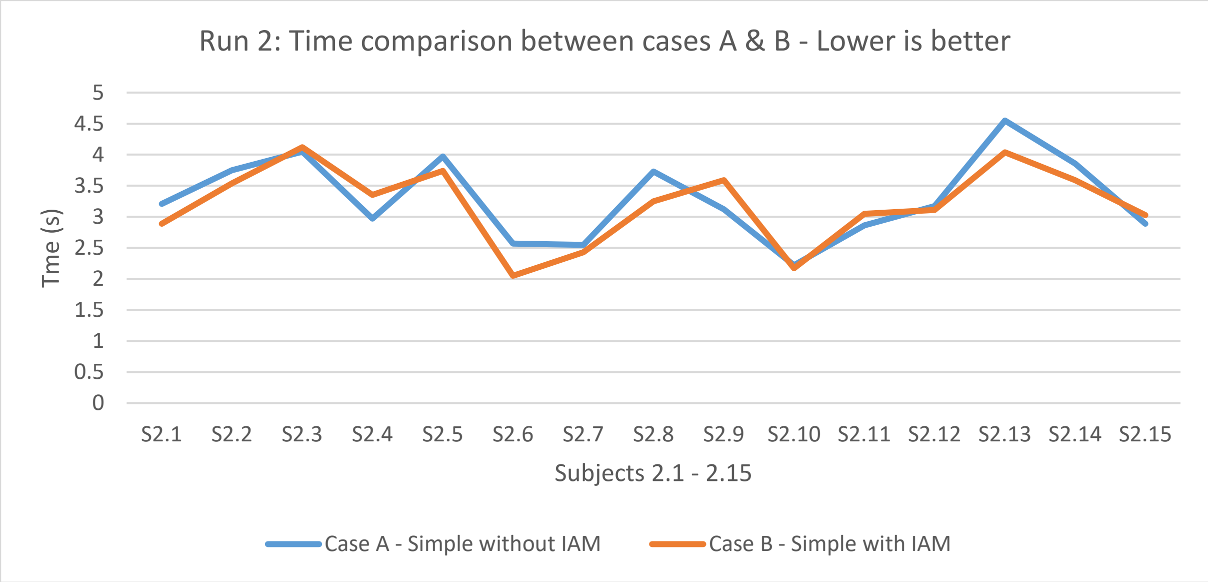


Fig. 7.5. Run 2: Time comparison – Cases A & B

Finally, an average of all the subjects from the first run and an average of all the values from the second run were taken and compared as shown in Fig. 7.6. Overall, there was an improvement if time needed to reach a decision with the IAM in both runs.

The first run saw an average increase of 0.5s or 15%. However, in the second run there was only a slight increase of 0.1s or 3%. These results clearly show that the subjects ignored the IAM proposals. Any potential time improvement was because the subjects have improved their experience with the mechanics of the test.

Fig. 7.6. Overall run comparison -Time (Cases A & B)

7.2 The use of IAM during complex problems

First run combined comparison of the quality of decision and the time needed to reach it, between cases C and D is shown in Fig. 7.7. For each of the 10 subjects (designated from 1.1 to 1.10) there is a blue bar representing the problem solving for complex cases without the help of the IAM and the orange bar representing the problem solving for complex cases with the help of the IAM. Higher values represent better quality.

Additionally, there is a gray graph line representing the problem solving for complex cases without the help of the IAM and the yellow graph line representing the problem solving for complex cases with the help of the IAM. All average values from each of the subject's 200 runs in a specific case were taken and compared. Lower values represent better time.

Only 10% of the subjects (S1.5) had a decreased quality of the decisions by 1.84% with the IAM. Further 20% of the subjects saw an insignificant increase from 0.12% till 0.73%. The following 40% of the subjects saw a slight increase from min. 3.40% till 10%. The final 30% of the subjects saw a significant increase from 15% till 23%.

On the other side, even with the IAM, 50% of the subjects achieved a longer time to reach a decision, from 0.11s till 1.5s or an increase of 2.4% till 22%. The other half achieved a better time from 0.01s till 0.64s or an improvement from 0.2% till 15%. In general, it can be seen that the results are not uniform. In half the cases, there is a higher quality but at the expense of higher time (S1.8). After the analysis of the second run it will be possible to better understand what is happening.

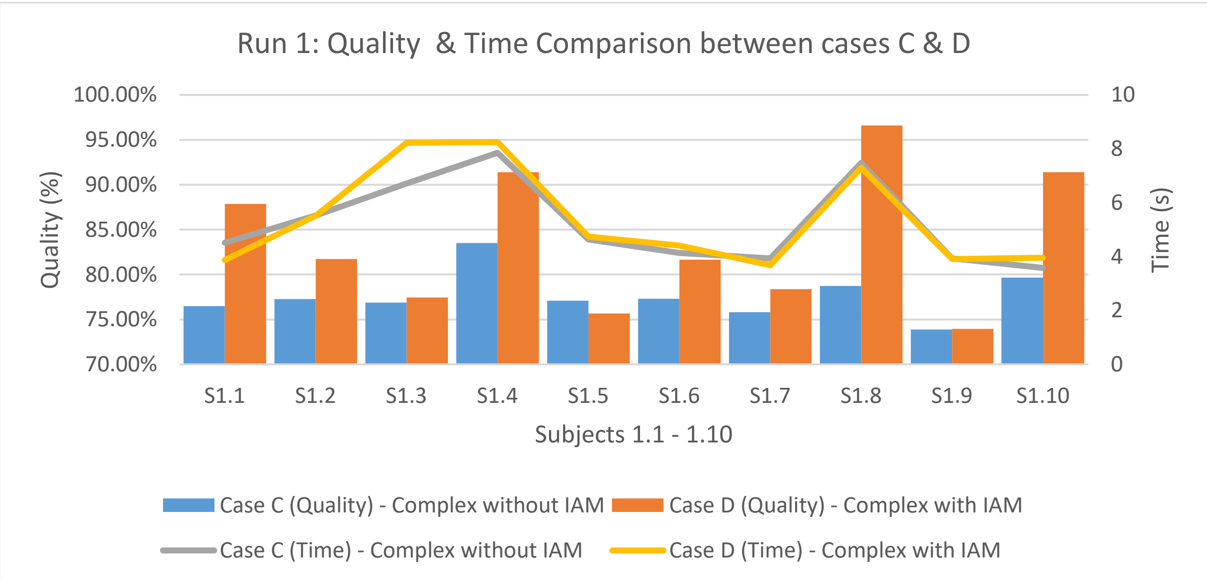


Fig. 7.7. Run 1: Quality & Time comparison – Cases C & D

Second run combined comparison of the quality of decision and the time needed to reach it, between cases C and D is shown in Fig. 7.8. For each of the 15 subjects (designated from 2.1 to 2.15) their colour representation is the same as in the first run. Average values from each of the subject's 200 runs in a specific case were taken and compared. Higher quality values represent better quality. Lower time values represent better time.

35% of the subjects had a decreased quality of the decisions in the range from 3% till 5% even with the IAM. Further 50% of the subjects saw a slight increase from min. 1.5% till 10%. The final 15% of the subjects saw a significant increase of 17%.

On the other side, even with the IAM, 25% of the subjects achieved a longer time to reach a decision, from 0.15s till 1.5s or an increase of 3.2% till 37%. The other 75% of subjects achieved a better time from 0.14s till 0.9s or an improvement from 2.6% till 18%. These results are similar as in the previous run. In majority of the cases, the subjects are achieving a higher quality, but their time varies. Therefore, an additional control case needs to be performed, where the subjects are instructed to exclusively wait for an advice. The main purpose is to find out if it will be possible to achieve a better balance between the quality and time.

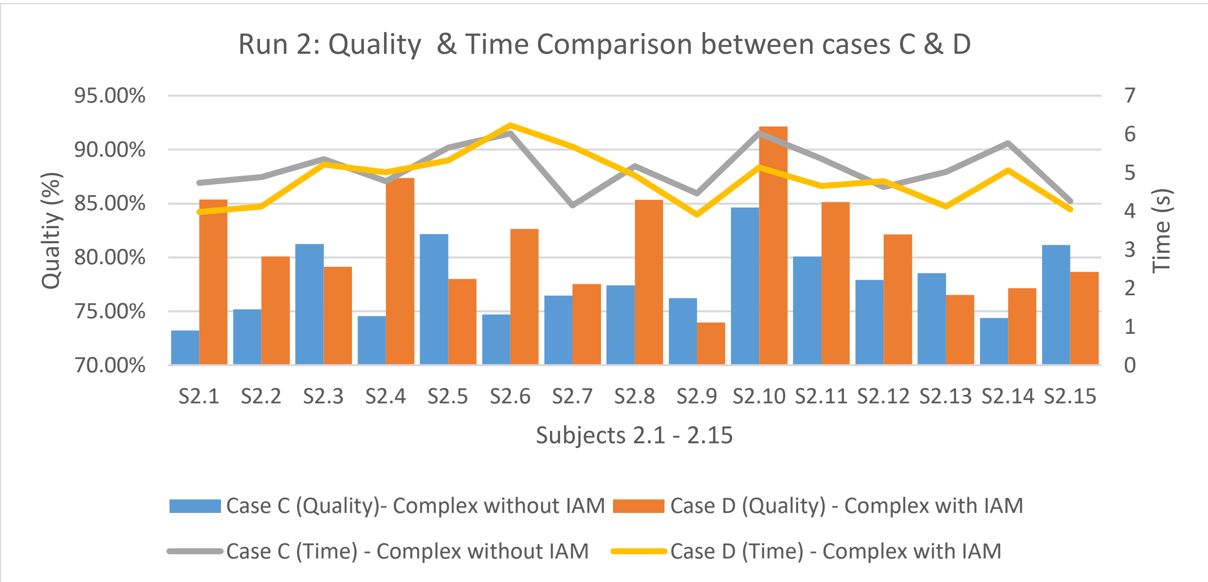


Fig. 7.8. Run 2: Quality & Time comparison – Cases C & D

7.3 Wait for IAM advice

First run combined comparison of the quality of decision and the time needed to reach it, between cases C, D and E is shown in Fig. 7.9. For each of the 10 subjects (designated from 1.1 to 1.10) their colour representation is the same as in the previous runs. Newly added grey bar represents the quality and the green graph line represents time for complex cases with the help of the IAM where the subjects wait for the IAM advise.

When compared with case D, only 10% of the subjects (S1.8) had a decreased quality of decisions by 0.60% with the IAM. That means that the 90% of the subjects saw an increase from 5% till 28% when waiting for an advice from the IAM.

This was expected. The question is at what time price?

Interestingly, 30% of the subjects achieved a better time, from 0.24s till 2.9s or an improvement of 5% till 25%. Again, subject S1.8 falls outside of the norm with an improvement of 40% when waiting for an advice. The rest of the 60% of subjects have, as expected achieved a longer time from 0.34s till 1.69s or an increase of 0.35% till 45%.

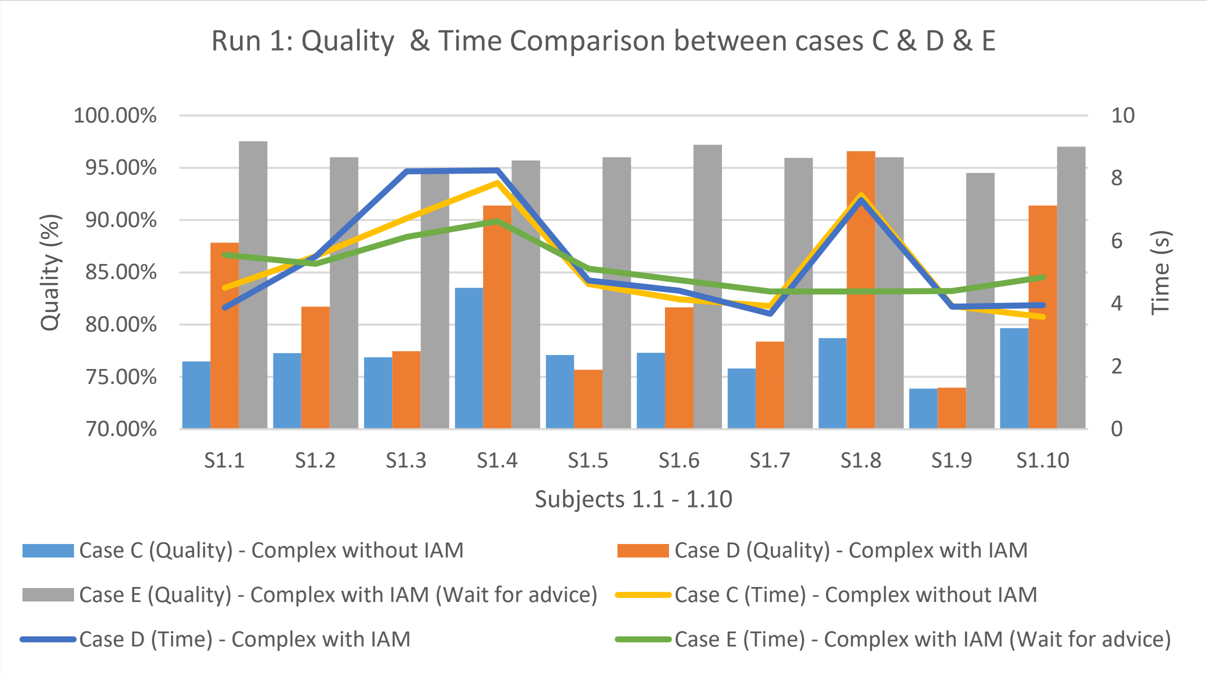


Fig. 7.9. Run 1: Quality & Time comparison – Cases C & D & E

Second run combined comparison of the quality of decision and the time needed to reach it, between cases C, D and E is shown in Fig. 7.10. For each of the 15 subjects (designated from 2.1 to 2.15) their colour representation is the same as in the previous runs.

As expected, all of the subjects achieved a better quality, from 5% till 30% when waiting for an advice from the IAM (compared to case D).

15% of the subjects achieved a slightly better time of 5%. The rest of the 85% achieved a longer time from 0.7s till 2.23s or an increase of 11% till 55%.

In order to reach a conclusion, all three cases will be compared according to their respective runs.

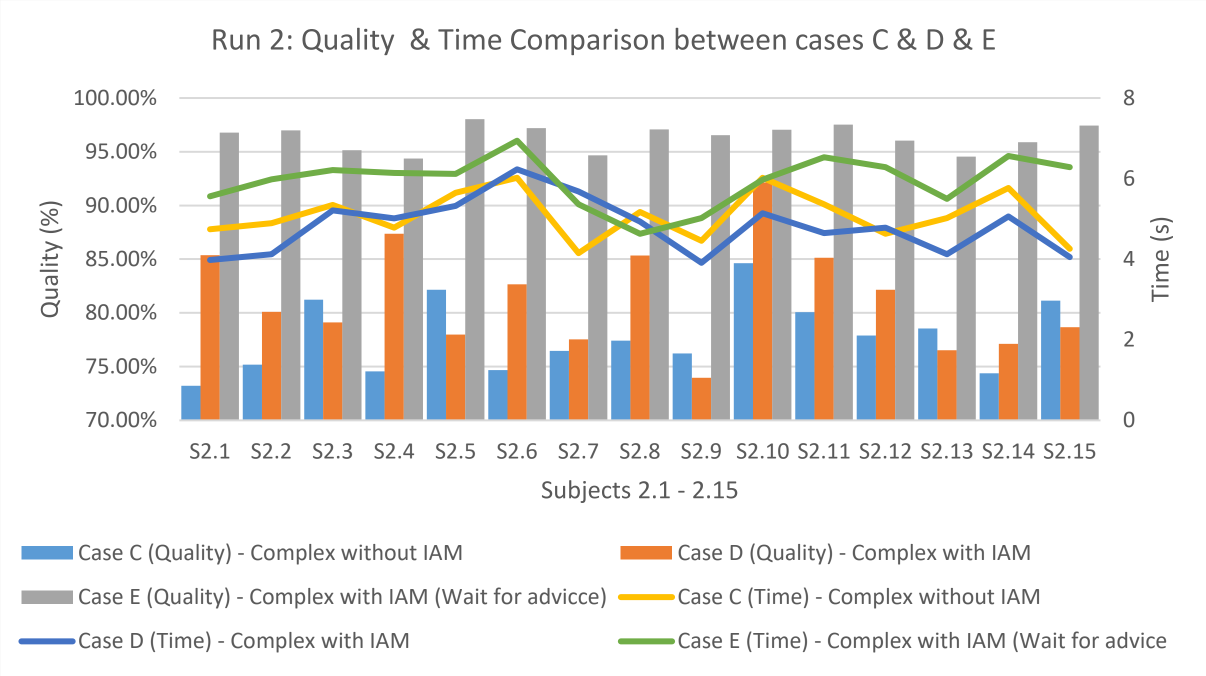


Fig. 7.10. Run 2: Quality & Time comparison – Cases C & D & E

An average of all the subjects from the first run and an average of all the values from the second run were taken and compared as shown in Fig. 7.11.

When compared with case C (without the help from the IAM), there was an increase of the quality of decisions with the IAM in both runs for cases D and E. When the performance depends on the ability of the system operator to wait and accept an advice from the IAM, there is an increase of quality of decisions of 4% till 8%. When the system operator has to strictly wait for the advice from the IAM, there is an increase of quality of decision of 24%.

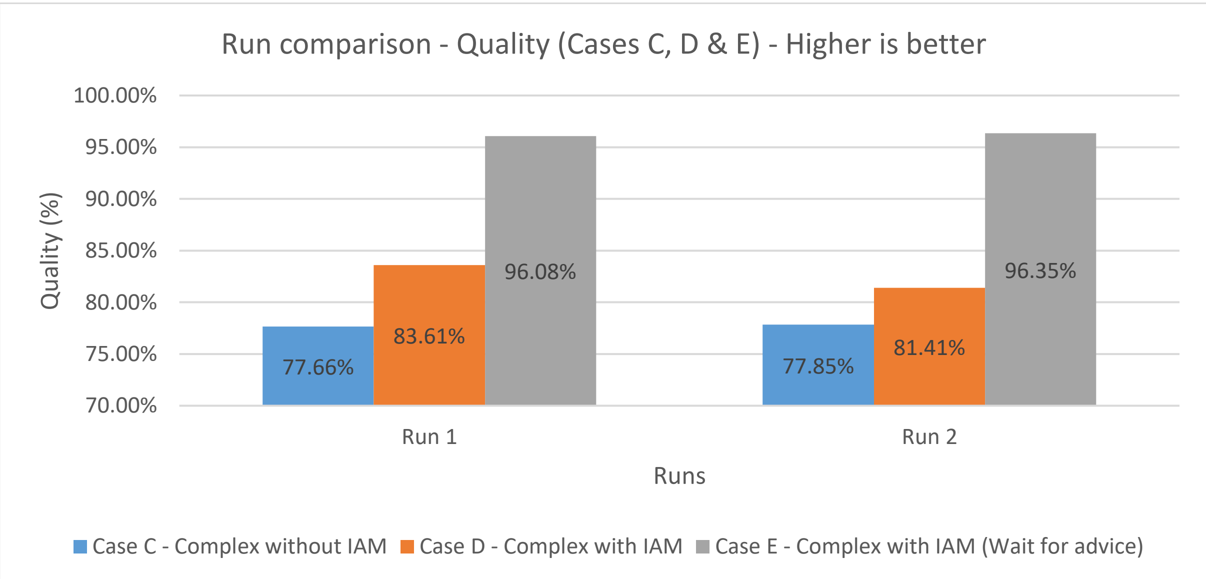


Fig. 7.11. Overall run comparison -Quality (Cases C, D & E)

An average of all the subjects from the first run and an average of all the values from the second run were taken and compared as shown in Fig. 7.12. When compared with case C (without the help from the IAM), there were slight inconsistencies between the 2 runs.

Generally, when the performance depends on the ability of the system operator to wait and accept an advice from the IAM, there is a slight improvement of needed time of 5%. When the system operator has to strictly wait for the advice from the IAM, there is an increase of time of 17%.

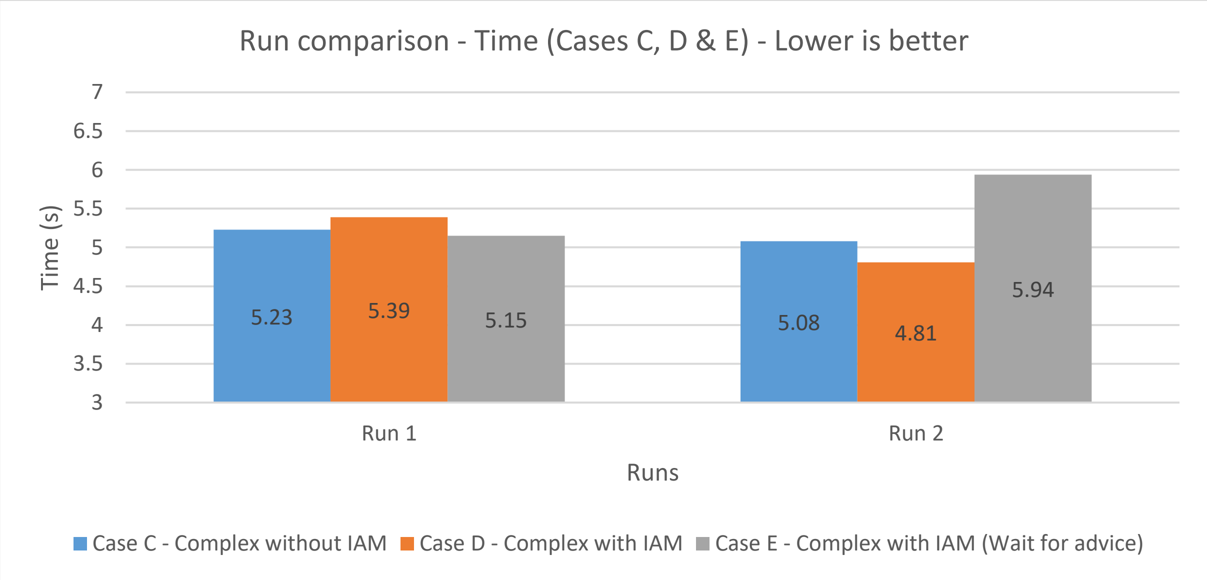


Fig. 7.12. Overall run comparison -Time (Cases C, D & E)

7.4 Adjustments

Based on all the previous results it can be concluded that the subjects in the role of a system operator achieve variable performances using IAM, according to their experience. In all experiment cases with complex problems (C, D, E, F), disturbances D1, D2 and D3 are generated randomly, where each has a one third chance to occur. As shown in simple cases A and B (with a constant level of disturbance - D1), the subjects achieve a high quality with good time and by ignoring IAM proposal.

When instructed to strictly wait for an advice, the majority will achieve higher quality of decisions but in longer time. However, this is not a solution because this is not how IAM works. System operator decides if he will accept, partially accept or ignore the IAM proposal.

The interaction between IAM and the system operator needs to be adjusted. The IAM operation needs to be comparable to a navigation system: it should offer proposals of ways based on input data: actual position, present time and the target. Proposal are continuously generated with any change of input data.

IAM continuously generates proposals which answer on the question: what is right to do in this situation here and now? Here, the key adjustment is that the IAM detects if the actions of the system operator are not decreasing the error. Direct consequence of this is a lower quality decision and a longer time needed. Therefore, when this happens, the IAM should audio/visually notify the system operator that he is moving away from the target state and to define the "solution way" more clearly.

Fig. 7.13. shows the adjusted IAM interaction. This adjustment is a basis for an additional case study F. The experiment is performed as described in a previous chapter, with an addition: as soon as the subject stops decreasing the distance of the red square (error) from the origin position (0,0), the entire problem-solving area turns its colour to yellow, and the IAM advice is duplicated next to the red square – the "*escape route*" is more clearly defined. The main aim is to try and achieve a better balance between higher quality of decisions and time needed for them, especially during: solving of conflict situations, solving of complex situations, solving of non-standard situations, support of less experienced system operators.

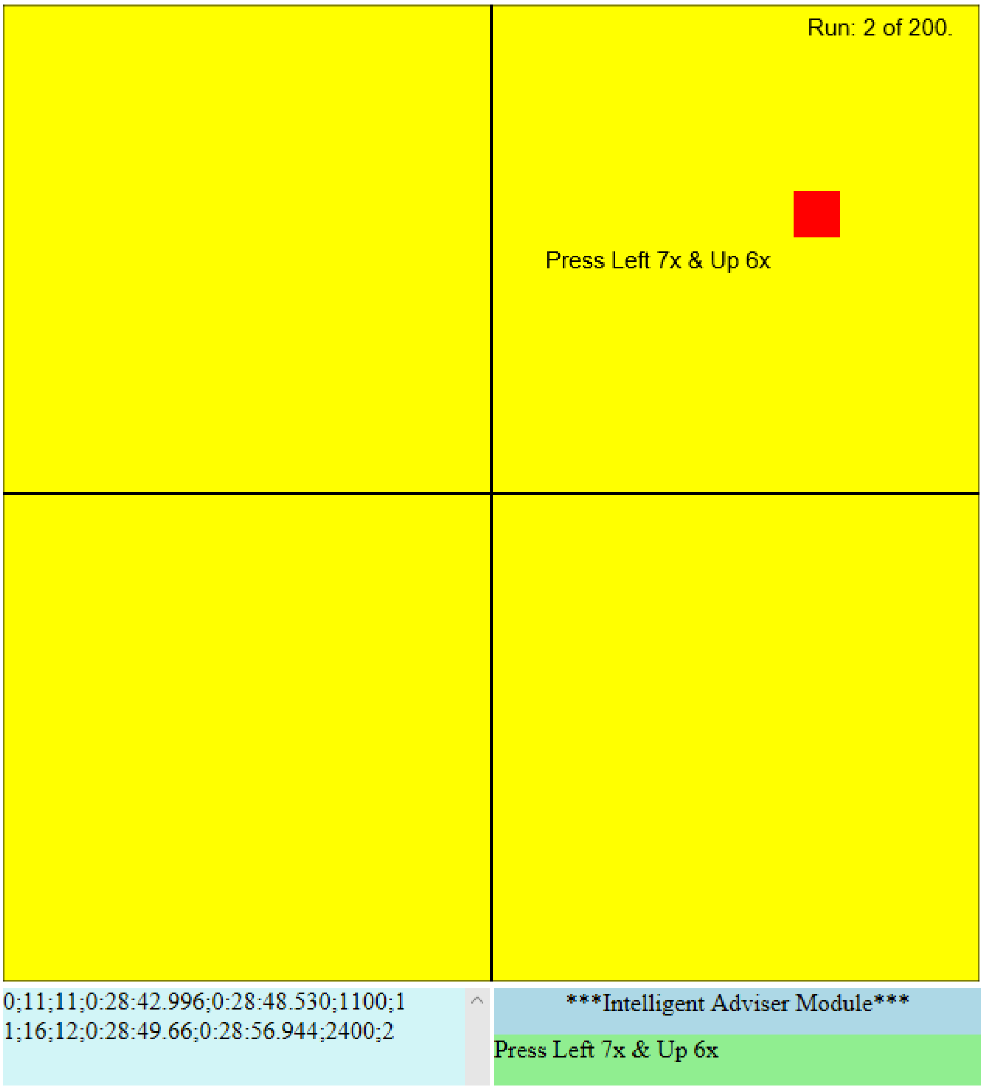


Fig. 7.13. Adjusted IAM interaction with the system operator

In the first run, when compared with case D, 30% of the subjects had an insignificant decrease of quality of decisions by around 3% when using the IAM with the adjusted interaction. Further 30% did not have any change of quality. The rest of the 40% of the subjects had an improvement of 8% till 15%.

30% of the subjects achieved a longer time, from 4% till 8%. 10% did not have any change if time. 60% of the subjects saw an improvement of their time from 3% till 60%.

As shown in Fig. 7.14. average results analysed from case F produced a good balance between quality and time when compared to all other cases with a complex problem (C, D, E).

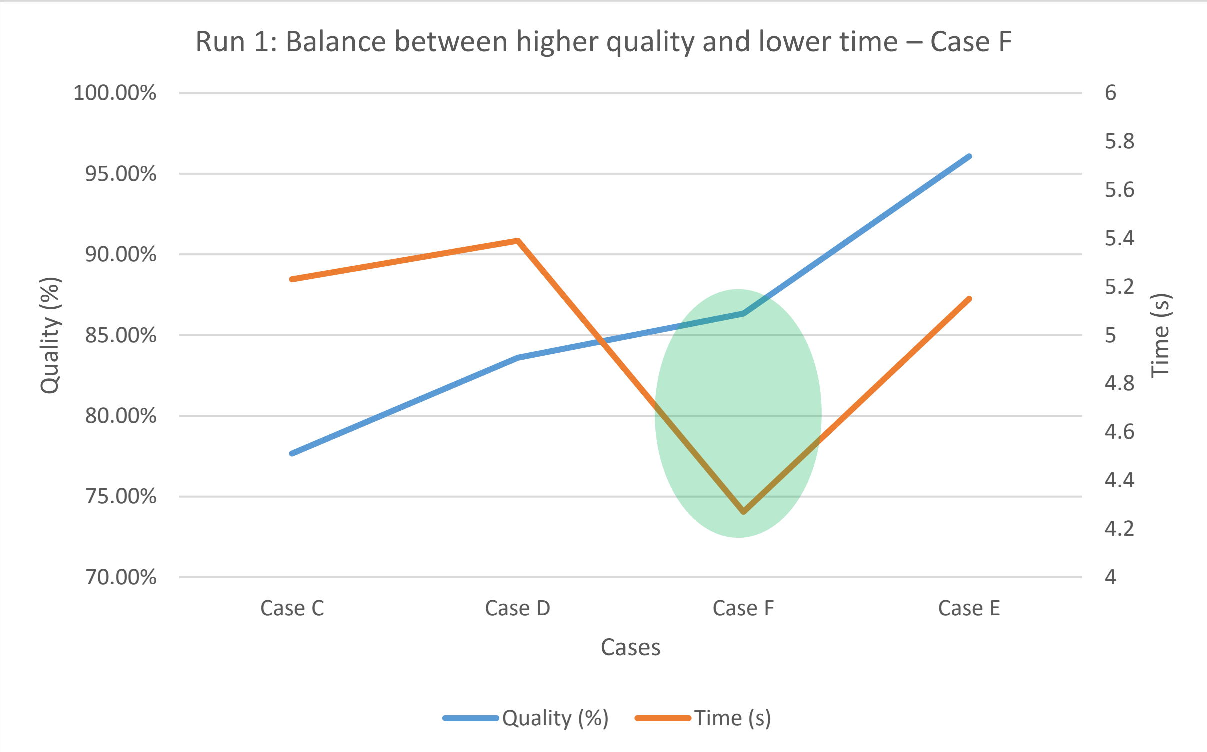


Fig. 7.14. Run 1: Case F –balance between higher quality and good time

In the second run, when compared with case D, 10% of the subjects had an slight decrease of quality of decisions by around 5% when using the IAM with the adjusted interaction. Further 75% had a slight increase from 1% till 10%. The rest of the 15% of the subjects had an improvement of 15% till 17%.

40% of the subjects achieved a longer time, from 3% till 25%. 60% of the subjects saw an improvement of their time from 5% till 35%.

As shown in Fig. 7.15. average results analysed from case F also produced a good balance between quality and time when compared to all other cases with a complex problem (C, D, E).

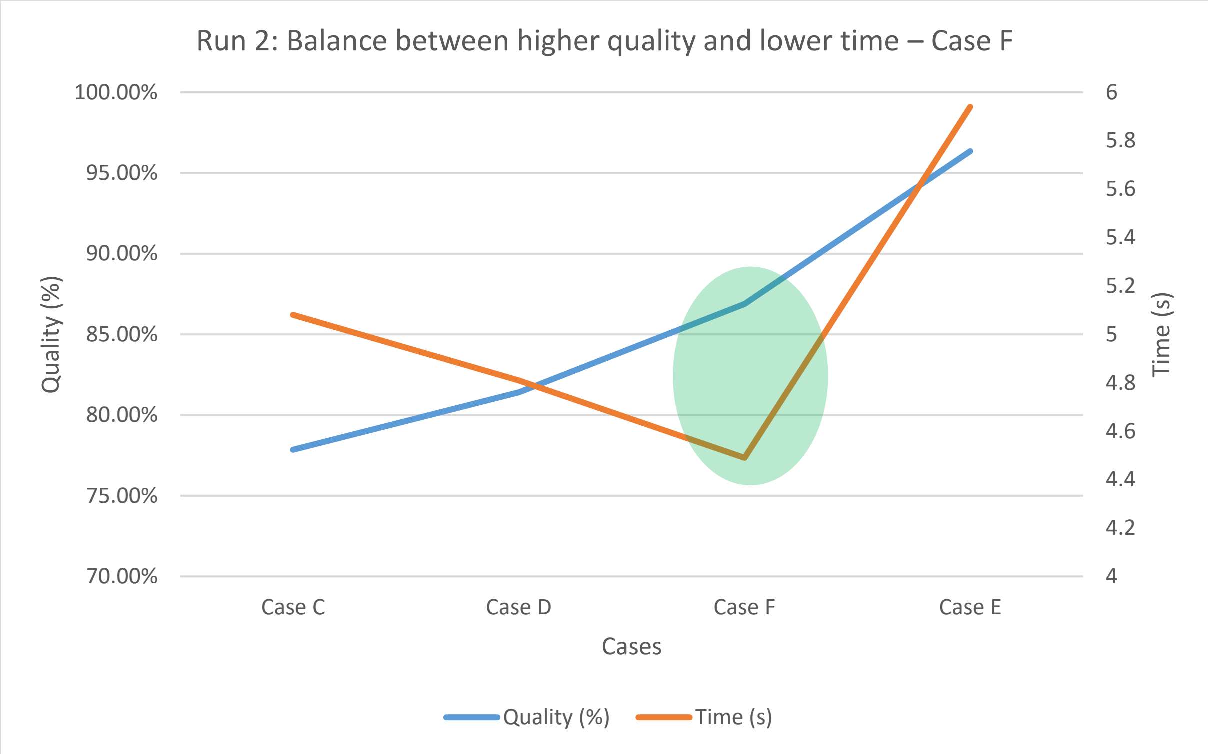


Fig. 7.15. Run 2: Case F –balance between higher quality and good time

7.5 Ranking

Ranking of all the cases is calculated as quality over time as presented in the previous chapter. It represents a metric which shows how much time was needed to achieve a specific level of quality. A perfect ranking of 100% would be achieved when a perfect quality of 100% is achieved in the ideal time. Ideal time is represented as the 90% of the best achieved time from all the case studies. In this experiment, ideal time is 2 seconds.

Fig. 7.16. and Fig. 7.17, show case rankings for run 1 and run 2. The results show that:

- For simple problems, case B (with IAM) resulted in 15% better results than case A (without IAM). However, in the second run, although case B, again has a better ranking, it is only about 3% better than case A.
- For complex problems, case F (with the adjusted IAM interaction) resulted in 20% to 26% (both runs) better results than case C (without IAM).
- For complex problems, case F (with the adjusted IAM interaction) resulted in 8% to 13% (both runs) better results than case D (with IAM, before IAM interaction adjustment).

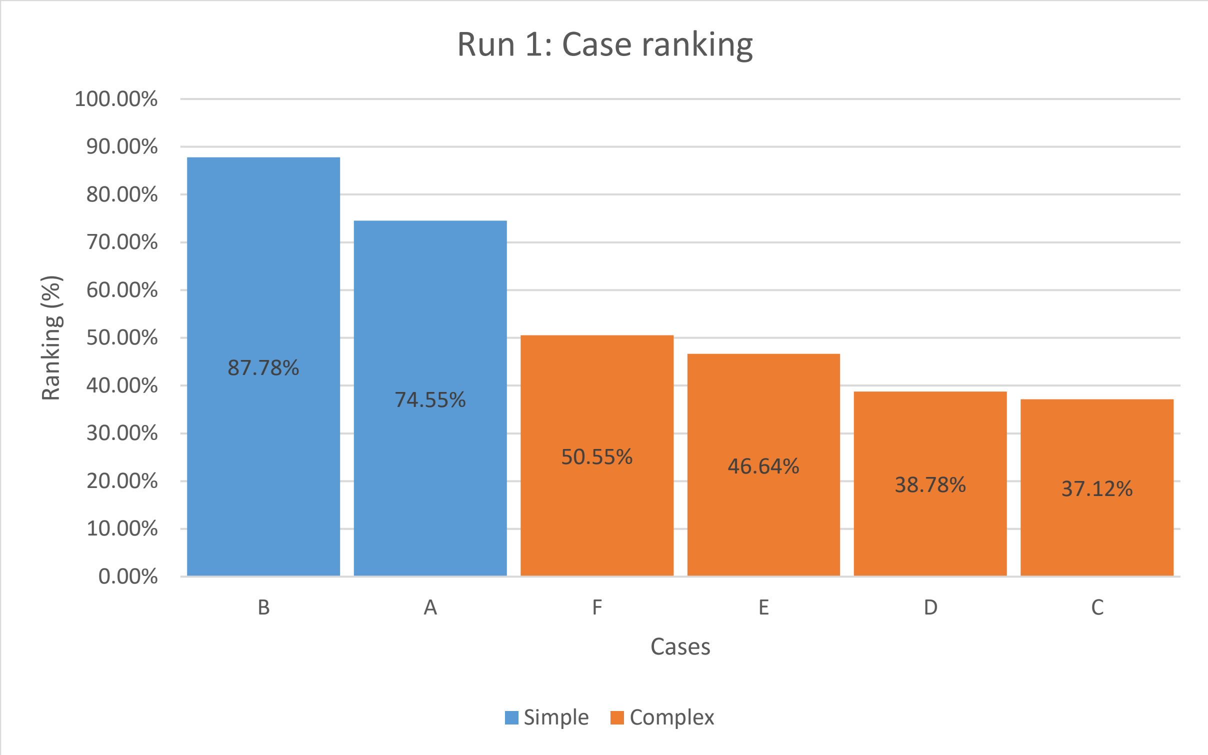


Fig. 7.16. Run 1: Case ranking as quality over time

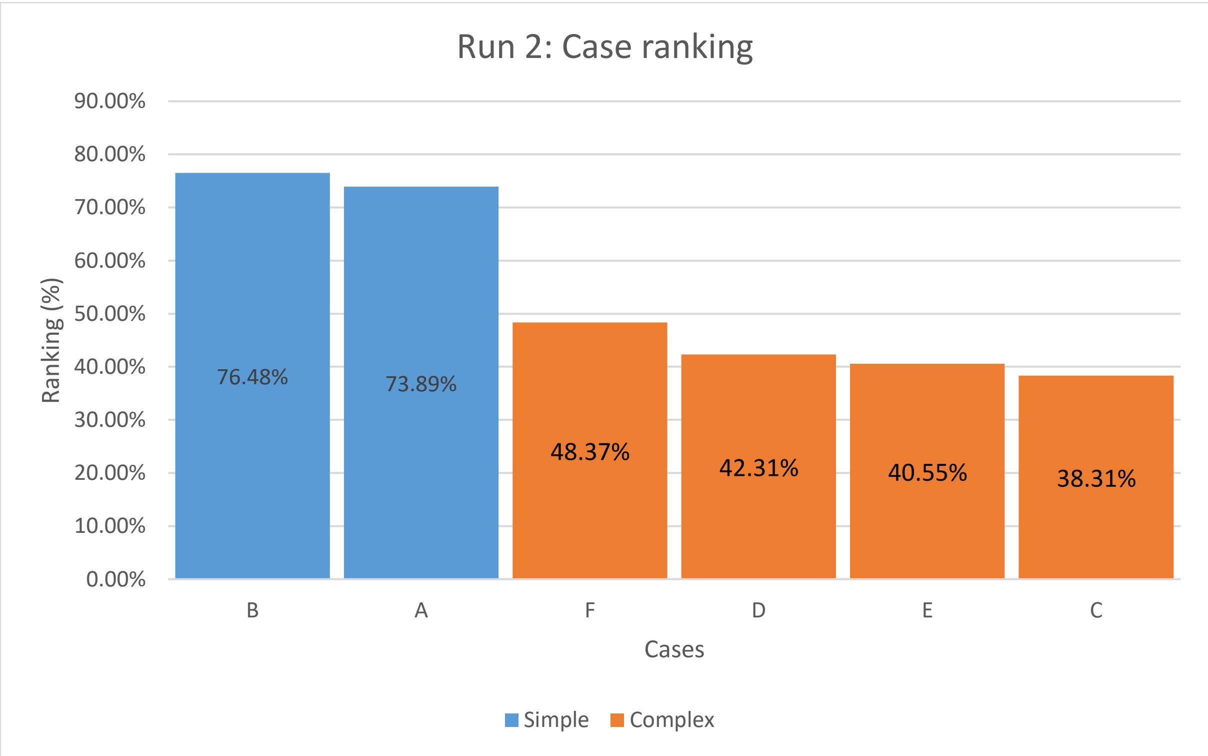


Fig. 7.17. Run 2: Case ranking as quality over time

7.6 Summary

This chapter presented the analysis of the data acquired from the “*2D problem compensation*” experiment. Data was generated after two runs of experiments for 6 different case studies. The analysis confirms that the adjusted approach shows promising results in the continued development of the IAM.

The main results of the experiment are:

1. The use of the IAM does not significantly have an influence on the system operator’s performance during simple problems.
2. The use of the IAM helps the system operator to achieve higher quality decision in shorter time.
3. System operator experience has a direct influence on the quality of decision and time needed to reach them
4. A good balance between higher quality of decisions and a lower time to reach them can be achieved by adjusting the IAM interaction with the system operator.
5. The IAM has to interact with the system operator as a navigation system. It offers proposals of “*escape routes*” based on input data: actual position, present time and the target. Proposal are continuously generated with any change of input data.

Next chapter will present the IAM concept realized in the frame of “Smart production” project. This is a result of cooperation between the company Festo and the International laboratory "Sensorika", together with specialists and graduate students from MSTU “Stankin”, INET RSUH, KIAM Russian Academy of sciences, JSC “TechInvest”. Main goal is to verify if the IAM concept can be realized in an industrial application.

Chapter 8

IAM in “Smart production”

Each of the 4 global industrial revolutions can be characterized through mechanization, electrification, automation and intellectualization which represents the current Industry 4.0. It is based on “Internet of Things”, “Big Data” and "Cyber-physical systems". As a result of all these technological advancements, the productivity is increasing stronger and faster.

To achieve this, it was necessary to develop new methods for hardware fault prediction, integration of self-repairing functionality, search of alternative solutions for current production problems and efficient maintenance to name a few.

By using built-in sensors and actuators which are connected to the network there is a need for new optimizing methods. These methods need to ensure better resource utilization in a production environment. The creation of a new multi-agent control technology for industrial automation presents several challenges. These include the creation of the automated system and a realisation of a highly variable production which includes logical processing of contradictions.

It was necessary to find adequate software-based decision methods with remote access capabilities, as well as reprogramming and monitoring capabilities for a live production line (Panfilov et al., 2016). As an answer to these requirements, a logical processing system was created. It is based on branching time logic, parallel computing and production control simulation which represent a few new methods from the intelligent industry 4.0.

One of the leading companies in the implementation of Industry 4.0 concepts is the concern FESTO AG, which operates in more than 100 countries with 300,000 users by realizing and providing just-in-time production.

In cooperation with FESTO, the concepts of Industry 4.0, Bionic Assembly System (BAS) and its integrated Intelligent Adviser Module (IAM) are represented and realized in the frame of “Smart production” project. International laboratory "Sensorika", together with specialists and graduate students from MSTU “Stankin”, INET RSUH, KIAM Russian Academy of sciences, JSC “TechInvest” have developed a technology for implementation and deployment of "Smart production" by building a new automatic line for foam-glass product manufacturing in the city of Kimry.

8.1 “Smart production” concept overview

The proposed automation system is developed according to the patented methods for automated resolution of logical contradictions (Pryanichnikov et al., 2017). By using these methods, the system is able to adapt based on the particular tasks and the environment. This allows it to predict possible failures, errors or shutdowns during operations and is the basis of reliability improvement in a project.

Such automation represents an integration of a digital network within a decentralised system. This makes it possible to switch between an automated or manual control, according to the need.

Such an intelligent control concept makes it possible to execute complex tasks during a production process. This includes valve switching, organization of production lines according to flexible manufacturing principles, motor control, damping etc. The overview of the entire “Smart production” is shown in Fig. 8.1.

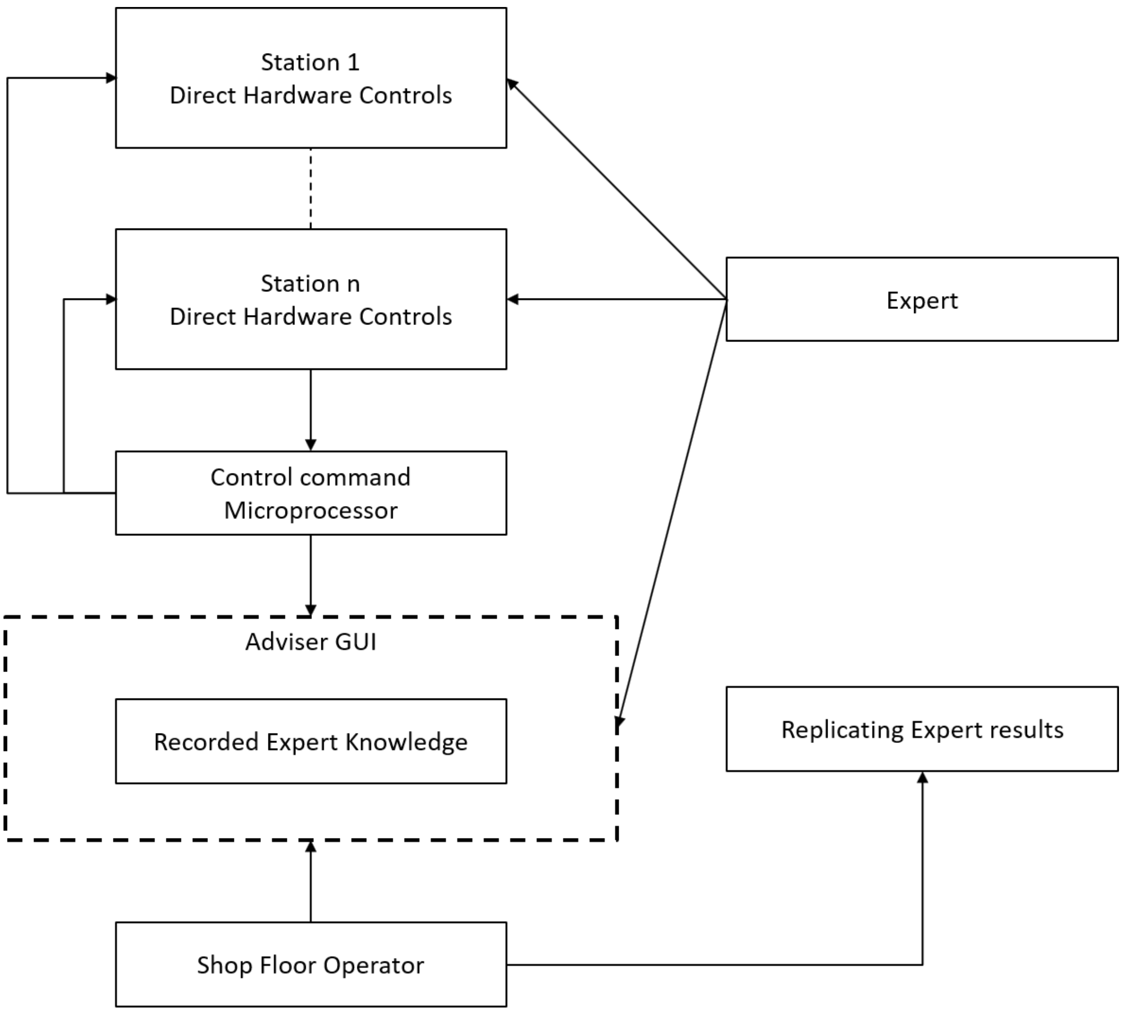


Fig. 8.1. “Smart production” overview

8.1.1 Technology, implementation and practical application

During the development, the technology of Industry 4.0 concept was realized. The implementation of sensory and peripheral devices necessary for providing "intelligent data input" and for constructing feedbacks was investigated. The general overview of the mounted cabinets with industrial automation control and programmable logic controllers is shown in Fig. 8.2. Top-level computers provide logical analysis of contradictions, identification of possible violations of the technological cycle, analysis of the actions of expert technologists and correction of prepared recipes or the generation of new ones.

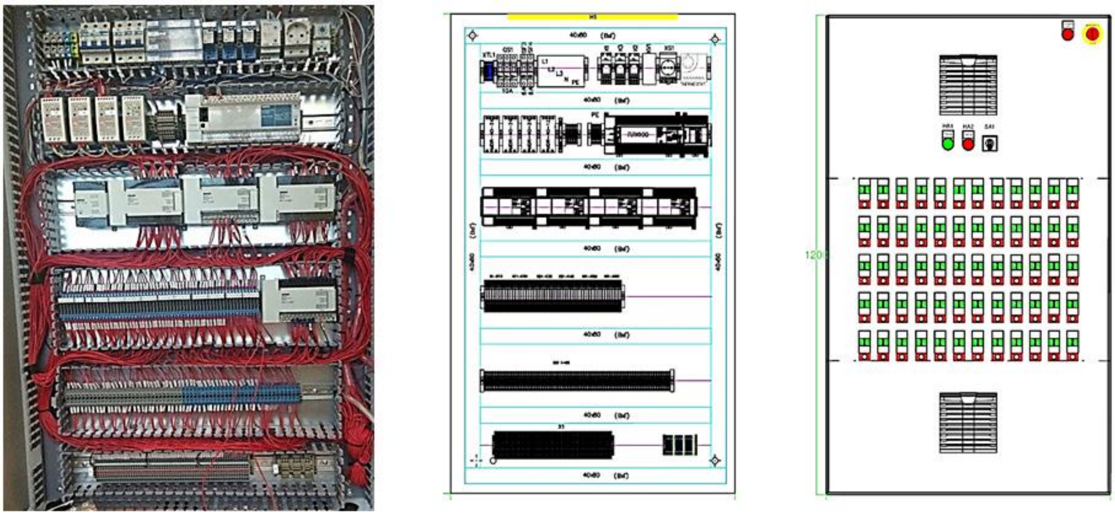


Fig. 8.2. Hardware implementation of shop management - design of control panels and their installation

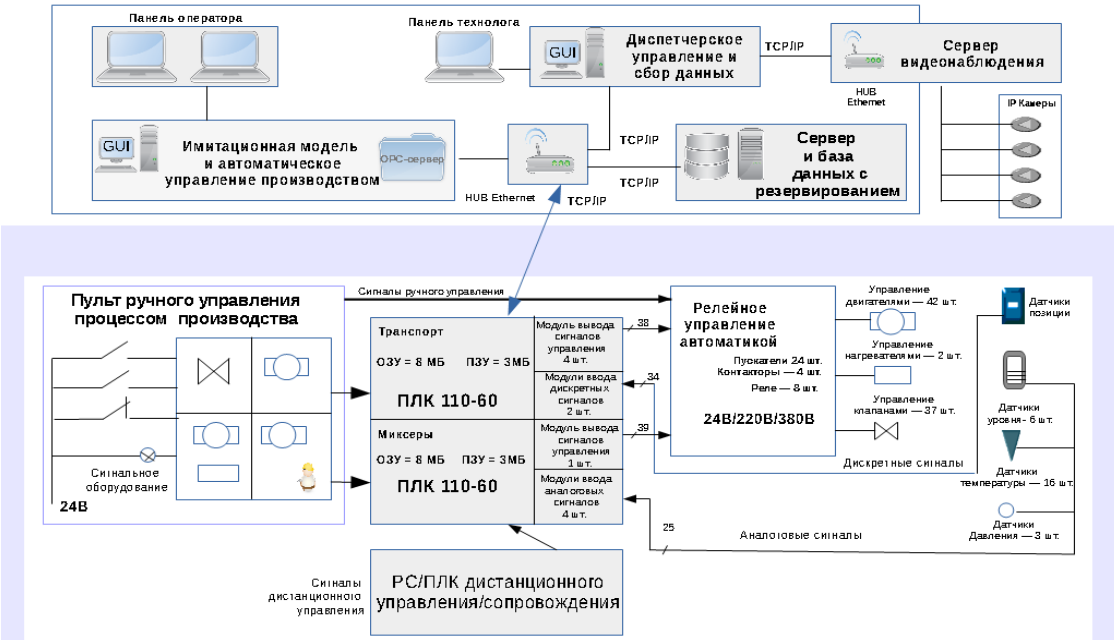


Fig. 8.3. Adviser interface - technological scheme with logical control of failures

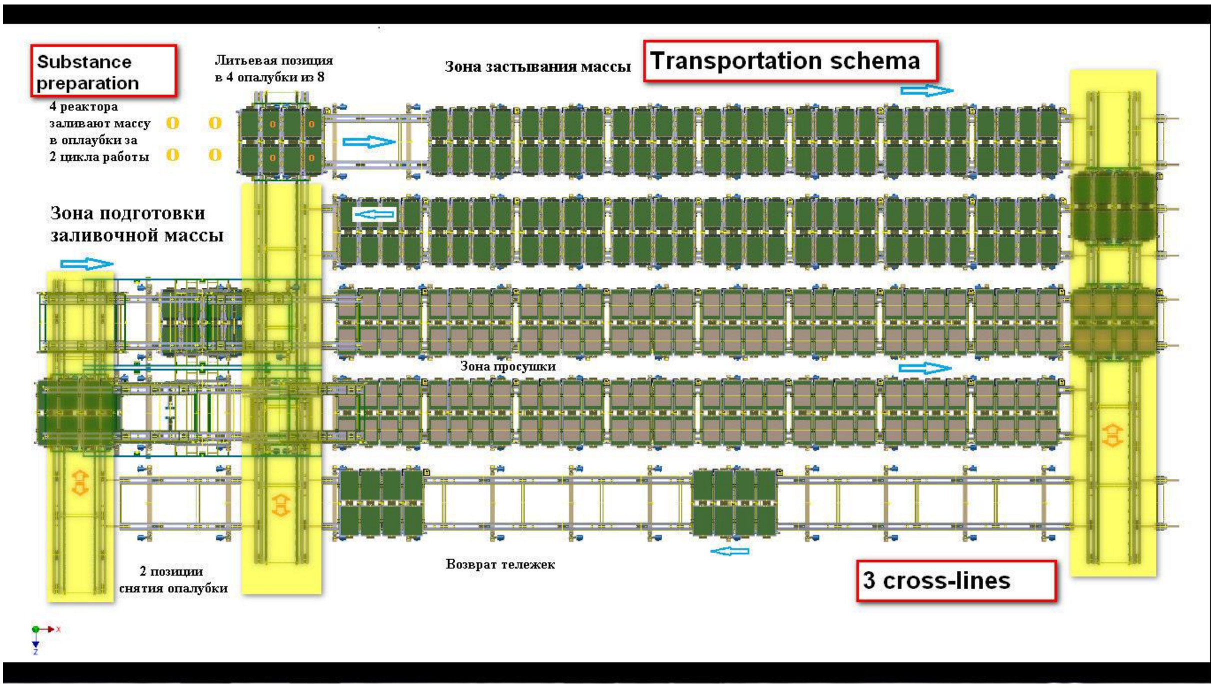
As shown in Fig. 8.3. the adviser interface is used to overview and manage the entire production cycle. It enables remote monitoring and control, logical testing and update of technological processes by the expert technologist (recording and embedding new recipes through the keypad and through the central PC)

Additionally, it performs “intelligent data recording” which is based on feedback from the sensor system. This makes it possible to control the operation of all actuators according to logical analysis and through the detection of any irregularities in the manufacturing processes. This significantly reduces the chance of errors and disturbances during the technological cycle.

The control system ensures complete automatic production, realised in two separate parts of the workshop. First part represents the preparation of a sodium glass material with the addition of required additives. Second part represents moulding, drying, final processing, packaging and shipment.

A specialised transport system co-developed with the International laboratory "Sensorika" is used for drying and to move the materials and finished products in the production space with the size of about 1400 m^2.

The entire transport system is based on individual robotic trucks, which are driven by electric motors and are equipped with a set of sensor devices. These transport units are able to move along a complicated path, weigh up to 2-3 tons and are designed to operate in hard conditions with high humidity as shown in Fig. 8.4.



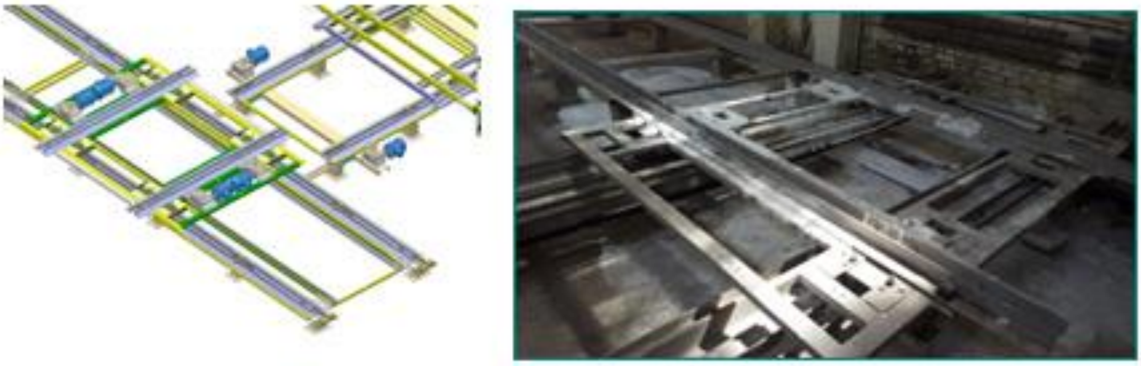


Fig. 8.4. Diagram of the reconfigurable transport shop system for precise positioning

The technological scheme for the main material preparation is formalised and described with around 200 logical variables. There are 3 large capacity reactors. The main material components are mixed in the required proportions at a specified temperature. First stage mixture heating requires a temperature of above 100 C. This heat is provided from a special steam generator which produces steam at 160 C with a capacity of 250 kg/h. The prepared mixture enters the storage tanks. As a result of mixing, an exothermic chemical reaction takes place, which needs to be cooled through a heat exchanger which uses tanks with a reserve of cold and hot water. Once cooled mixture is mixed with a fiberglass suspension which gives the required final product characteristics.

All valves are remotely controlled. The control system is equipped with microprocessors which execute control commands. The temperature regime, the filling levels of tanks and the pressure in them are controlled by required monitoring sensors. These sensors have a backup during the most critical phases of the production cycle. They ensure the execution of all the steps at the required time intervals.

The plant technologist is a human expert with the required knowledge about the entire process. Initially he prepares the recipe for the substance preparation. A recipe represents all the necessary steps, times and components. After the initial, test product batch has been prepared and the quality control has been passed, the recipe is recorded within the database. Afterwards, this recipe can be called up by any shop floor operator on duty and can be executed without the help of the expert technologist.

Carriages are moved along the specified trajectories. These trajectories are executed by the control system based on the feedback information from optical sensors and from the video monitoring system. Additionally, all carriages are equipped with a collision avoidance system.

The development of the control system is based on the expert-technologist knowledge. Such knowledge is represented using the mathematical approaches conventionally called Ancient Greek, Ancient Egyptian, Ancient Indian and Ancient Chinese or IGEC. These methods allow to construct evidence, as well as to make informal knowledge computational and to eliminate contradictions arising from different approaches.

8.2 Logical analysis of the feasibility of operations

One of the main problems of modern swarm robotics and self-organized production systems is the need for more intelligent management of system resources. This is especially true in the implementation of adaptive behaviour of mobile robots. This includes avoiding conflicts, contradictions, emergency situations and blocking during the joint work of a large number of assembly stations, mechatronic devices, mobile robots and shop floor operators.

To describe this behaviour of mobile robots, an essential tool is the construction of expert schemes embedded in the control system. Additionally, an algorithm superstructure is needed as well. It connects conflicting expert schemes into a single algorithm that implements ambiguous decisions and expert knowledge. These schemes ensure the maximization of quality according to set criteria as well as to replace missed measurements through logical filters. This allows the mobile robots to remain functional even with incomplete data and to provide, either exact trajectory measurements, or approximations of the areas in which they are located, generating sufficient data to solve locomotive problems.

8.2.1 Methods of logical analysis

The described logical analysis techniques include a special logical device and IGEC technology tools as presented in (Pryanichnikov, 2017) (Helemendik, 2006) and (Kirilchenko, 2013). In a logical approach of mobile robot analysis, the entire set of parameters is considered as a task. Depending on the nature, type and complexity, the problem is described with formula Θ using a special logical language, after which the formula is investigated for feasibility as shown in Fig. 8.5.

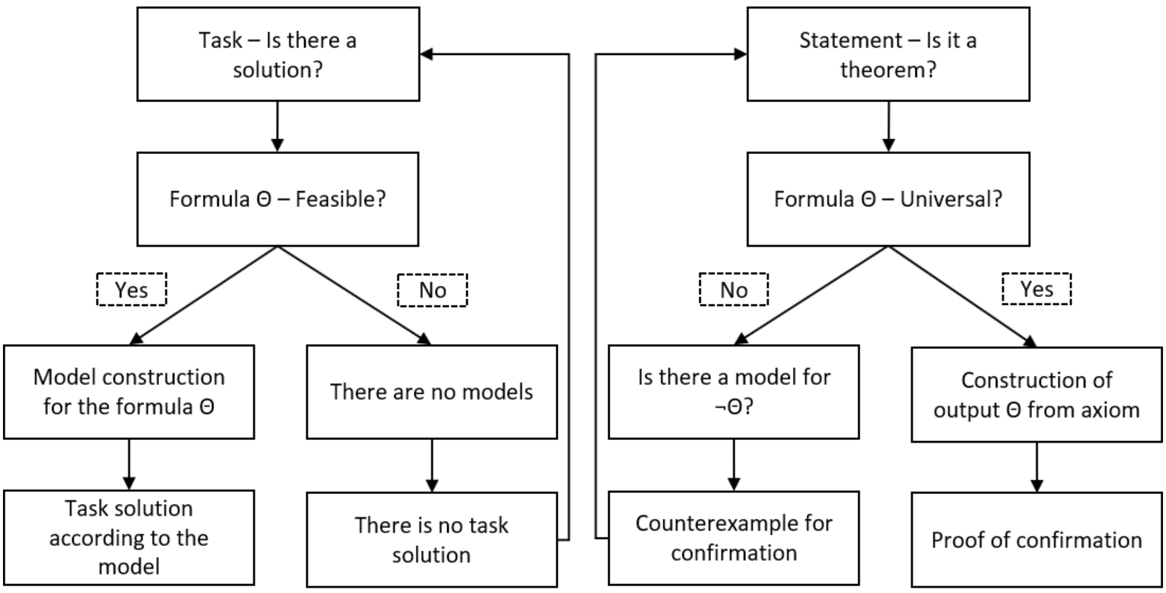


Fig. 8.5. Schemes for constructing reasoning within the framework of mathematical logic

In this case, the formula that describes the problem is feasible if and only if the problem has a solution, and in case of feasibility, the solution is obtained from the model for the formula. Depending on the type of task, this solution can be a suitable set of values of Boolean variables (switches), a production system, a program of optimal (according to the goal) autonomous robot functions, specific interactions with the external environment, etc.

In the case that the formula Θ is not feasible, the correction of the problem can be updated through refinement or weakening of its conditions as represented with an arrow (up) in the form of feedback. If the answer remains negative, then the entire performed analysis (formula Θ feasibility test) still gives a useful result from two points of view:

1. From a practical point of view, it helps to understand that, under the given conditions, the problem does not have a solution, which often makes it possible to find the causes of error, as well as not to invest further resources in a deadlock implementation, but rather to concentrate on a more promising direction.
2. From the theoretical point of view, the impossibility of the formula ¬Θ, means the general validity of the formula Θ, which allows us to obtain the most reliable proof of the true assertion (in particular, by means of the derivation of the theorem from axioms).

Such technology was successfully utilized in the "Intellectual Robotronics" project, in which the development of software and hardware solutions served as a basis for network of associative laboratories and operational centres, as proposed by IAE – CEB (International Academy of Engineering – Central European Branch) in cooperation with different universities from Austria, Russia and Croatia.

Therefore, IGEC method gives promising results when implemented in a management system within the context of Industry 4.0 (Pryanichnikov, 2016). The main problems include testing and control program development as well as formalisation and construction of suitable imitation models.

Further research is directed towards analysis of data derived from sensors and hardware feedbacks. The main purpose is to identify any irregularities during the production process. This can make it possible to predict and prevent disruption in the technological cycle.

The use of the adviser interface provides a continuous remote access to it through cloud services and the internet. This makes it possible to remotely monitor, test or update any technological process. Additionally, the interface allows to create and merge several production processes into a single system. The described production includes a set of common technological operations, which make it possible to clearly illustrate the application of the principles of the concept Industry-4.0.

Conclusion

The research presented in this thesis focused on the investigation of working scenarios and efficiency of next generation of modern assembly systems. These systems are known as Bionic Assembly System (BAS). The main focus was the development of the Intelligent Adviser Module (IAM) concept as a decision-making support tool for the system operator in BAS.

The main contribution consists from the following parts:

- The theoretical part analysed modern production challenges, development directions of new modern assembly systems, BAS concept, IAM concept
- The practical part included two fields of investigation:
    1. **Experiment**: BAS and IAM are in their concept stage of development. For this reason, real-world performance data was unavailable. An experiment has been set-up for the investigation of contribution of IAM proposals on the quality and time of decisions.
    2. **Practical implementation**: Main goal was to verify if the IAM concept could be successfully realized in an industrial application with a high technical similarity to BAS. Another goal was to determine if the Industry 4.0 technology could be used to facilitate main IAM functions: system monitoring, data collection, knowledge discovery and decisions support

The main results of the investigation are:

1. The quality of decisions is higher, and / or time is shorter, especially during: solving of conflict situations, solving of complex situations, solving of non-standard situations, support of less experienced system operators.
2. System operator is the final decision maker. He alone decides if he will accept, partially accept or ignore the IAM proposals.
3. System operator experience has a direct influence on the quality of decision and time needed to reach them.
4. IAM continuously generates proposals which answer on the question: what is right to do in this situation here and now?

5. The IAM operation is comparable to a navigation system: it offers proposals of ways based on input data: actual position, present time and the target. Proposal are continuously generated with any change of input data.

The main results of the IAM implementation in the frame of “Smart production” project are:

1. Successful realisation of IAM in an industrial application with a high technical similarity to BAS - automated production of a glass material. Instead of mobile robots, the transport units are trucks with electric motors. Instead of assembly stations, there are stations for preparation, moulding, drying, processing, drying, packaging and shipping. Instead of assembly, the final product is prepared according to a specified operation order.
2. Production systems with high technical similarity to BAS can increase their efficiency using the IAM concept.
3. Modern technology and hardware are ready to facilitate logical analysis of contradictions, identification of conflict situations, extraction and representation of expert knowledge.

Based on the results presented in this dissertation, future research can focus on the further investigation of the IAM performance in real industry working scenarios. Main topics of research can include IAM stability, maintenance, optimization and general usability. Additionally, different methods of information exchange between the IAM and the system operator can be examined. One possible direction of further development is the implementation of machine learning and new man-machine interfaces which can include speech recognition and voice assistant capabilities.

References

* (2016) “Knowledge.” Oxforddictionaries.com, Oxford Dictionary, n.d. Web. 25 May. 2016. http://www.oxforddictionaries.com/us/definition/american_english/knowledge

** (2016) “Learning.” Oxforddictionaries.com, Oxford Dictionary, n.d. Web. 30 May. 2016. <http://www.oxforddictionaries.com/definition/english/learning

*** (2016). Orange data mining software project. Available online at http://orange.biolab.si/

**** (2016). Weka data mining software project. Available online at http://www.cs.waikato.ac.nz/ml/weka/

**** Random variation. BusinessDictionary.com. WebFinance, Inc. July 26, 2016 <http://www.businessdictionary.com/definition/random-variation.html>

***** (2018). Notepad++ source code editor. Available online at < https://notepad-plus-plus.org/ >

Abraham, A., Corchado, E. & Corchado, J. M. (2009). Hybrid learning machines. Neurocomputing, 72(13), 2729-2730

Agre, G. & Peev, S. (2002). On supervised and unsupervised discretization. Cybernetics and information technologies, 2(2), 43-57

Amen, M. (2000). Heuristic methods for cost-oriented assembly line balancing: A survey. International Journal of Production Economics, 68(1), 1-14

Bentley, P. (1999). Evolutionary Design by Computers, Morgan Kaufmann Publishers, Inc. San Francisco, California ISBN 1-55860-605-X

Bonabeau, E., Dorigo, M. & Theraulaz, G. (1999). Swarm Intelligence: From Natural to Artificial Systems”, New York, NY: Oxford University Press

Boothroyd, G. (1988). Fundamentals of Metal Machining and Machine tools (Vol. 28). Crc Press

Botti, V. & Boggino, A. G. (2008). ANEMONA: A multi-agent methodology for Holonic Manufacturing Systems. Springer Science & Business Media

Brachman, R. J., Levesque, H. J. & Reiter, R. (1992). Knowledge representation. MIT press

Browne, J., Dubois, D., Rathmill, K., Sethi, S. P. & Stecke, K. E. (1984). Classification of flexible manufacturing systems. The FMS magazine, 2(2), 114-117

Camazine, S. (2003). Self-organization in biological systems. Princeton University Press

Caruana, R. & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning (pp. 161-168). ACM

Colombo, A. W., Schoop, R., Leitao, P. & Restivo, F. (2004). A collaborative automation approach to distributed production systems," Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on, Berlin, 2004, pp. 27-32. doi: 10.1109/INDIN.2004.1417297

Davis, R., Shrobe, H. & Szolovits, P. (1993). What is a knowledge representation? AI magazine, 14(1), 17

De Jong, T. & Ferguson-Hessler, M. G. (1996). Types and qualities of knowledge. Educational psychologist, 31(2), 105-113

Deming, W. E. & Edwards, D. W. (1982). Quality, productivity, and competitive position (Vol. 183). Cambridge, MA: Massachusetts Institute of Technology, Center for advanced engineering study

Dhatt, G., Touzot, G. & Lefrancois, E. (2012). Finite Element Method, ISTE Ltd and John Wiley & Sohns Inc., ISBN 978-1-84821-368-5

Dicken, P. (1998). Global Shift: Transforming the World Economy. Third Edition, Guilford Publications, Incorporated, ISBN 1572-303034

Dougherty, J., Kohavi, R. & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In Machine learning: proceedings of the twelfth international conference (Vol. 12, pp. 194-202)

Dunn, M. (2013). What other types of reasoning are there? theoryofknowledge.net. http://www.theoryofknowledge.net/ways-of-knowing/reason/what-other-types-of-reasoning-are-there/ Last accessed: 26th May 2016

Entezari-Maleki, R., Rezaei, A. & Minaei-Bidgoli, B. (2009). Comparison of classification methods based on the type of attributes and sample size. Journal of Convergence Information Technology, 4(3), 94-102

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases, AI Magazine, Volume 17, Number 3, © AAAI

Feigenbaum, E. A. & McCorduck, P. (1983). The fifth generation. Addison-Wesley Pub.

Feng, S. & Zhang, Y. (1999). Conceptual Process Planning-A definition and functional decomposition. Manufacturing Science and Engineering, 10, 97-106

Fikes, R. & Kehler, T. (1985). The role of frame-based representation in reasoning. Communications of the ACM, 28(9), 904-920

Fiorentino, A. (2014). Cost drivers-based method for machining and assembly cost estimations in mould manufacturing. The International Journal of Advanced Manufacturing Technology, 70(5-8), 1437-1444

Flin, R. H. (Ed.). (1997). Decision making under stress: Emerging themes and applications. Gower Technical

Fogel, D. B. (2000). What is evolutionary computation? Spectrum, IEEE, 37(2), 26-28

Ford, H. (1926). Mass production. Encyclopaedia Britannica, 13, 821-823

Getty, David J., et al. (1995). System operator response to warnings of danger: A laboratory investigation of the effects of the predictive value of a warning on human response time, Journal of Experimental Psychology: Applied 1.1 (1995): 19

Giarratano, J. C. (1993). CLIPS User's guide. NASA Technical Report, Lyndon B Johnson Center

Giarratano, J. C., & Riley, G. (1998). Expert systems. PWS Publishing Co.

Giarratano, J. (2002). CLIPS: User Guide version 6.20

Glendening, J. (2013). The Evolutionary Imagination in Late-Victorian Novels: An Entangled Bank. Ashgate Publishing, Ltd.

Goldratt, Eliyahu M. (1988). Computerized shop floor scheduling, The International Journal of Production Research 26.3, Pages: 443-455

Gray, J., & Shenoy, P. (2000). Rules of thumb in data engineering. In Data Engineering, 2000. Proceedings. 16th International Conference on (pp. 3-10). IEEE.

Groover, M. P. (2007). Fundamentals of modern manufacturing: materials processes, and systems. John Wiley & Sons

Haken, H. & Jumarie, G. (2006). A macroscopic approach to complex system. Springer-Verlag. Berlin, Heidelberg, New York, 3th

Hartmann, D. & Lehner, K. (1990). Technical Expert systems, Springer-Verlag

Haskovic D., Katalinic B., & Kukushkin I. (2015). Role of the Adviser Module in the Hybrid Assembly Subordinating Control Structure, Procedia Engineering, Volume 100, Pages 1706-1713, ISSN 1877-7058, http://dx.doi.org/10.1016/j.proeng.2015.01.546

Haskovic, D., Katalinic, B., Kildibekov, A. & Kukushkin, I. (2016). Intelligent Adviser Module for Bionic Assembly Control System: Functions and Structure Concept, Proceedings of the 26th DAAAM International Symposium, pp.1158-1165, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-07-5, ISSN 1726-9679, Vienna, Austria

Haskovic, D., Katalinic, B., Zec, I., Kukushkin, I. & Zavrazhina, A. (2016). Structure and Working Modes of the Intelligent Adviser Module, Proceedings of the 27th DAAAM International Symposium, pp.0866-0875, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-08-2, ISSN 1726-9679, Vienna, Austria

Haskovic, D., Katalinic, B-, Zec, I., Kukushkin, I. & Zavrazhina, A. (2017). Intelligent Adviser Module: Proposals and Adaptive Learning Capabilities, Proceedings of the 28th DAAAM International Symposium, pp.1191-1196, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-11-2, ISSN 1726-9679, Vienna, Austria, DOI: 10.2507/28th.daaam.proceedings.165

Hays, R. & Wheelwright (1984). S. Restoring Our Competitive Edge: Competing Through Manufacturing, Wiley, New York

Heilala, J. & Voho, P. (2001). Modular reconfigurable flexible final assembly systems. Assembly Automation, 21(1), 20-30

Helemendik R.V. (2006) Algorithm for recognizing the feasibility of formulas for the logic of branching time and an effective algorithm for constructing the conclusions of universally valid formulas from axioms // Mathematical problems in cybernetics. Issue. 15: Collection of articles / Ed. ABOUT. Lupanov. - Moscow: Fizmatlit,. - P. 217-266.

Hobday, M. (1998). Product complexity, innovation and industrial organisation, Research Policy, Volume 26, Issue 6, February, Pages 689-710, ISSN 0048-7333, http://dx.doi.org/10.1016/S0048-7333(97)00044-9

Hollnagel, E. (1989). The reliability of expert systems. Prentice-Hall, Inc..

Holsapple, C.W. & A. B. Whinston. (1996). Decision Support Systems: A Knowledge-Based Approach. St. Paul: West Publishing. ISBN 0-324-03578-0

Holsapple, C. & Whinston, A. B. (Eds.). (2013). Decision support systems: theory and application (Vol. 31). Springer Science & Business Media

Hsueh, Y. W. & Yang, C. Y. (2008). Prediction of tool breakage in face milling using support vector machine. The International Journal of Advanced Manufacturing Technology, 37(9-10), 872-880

Hu, S. J., Zhu, X., Wang, H. & Koren, Y. (2008). Product variety and manufacturing complexity in assembly systems and supply chains. CIRP Annals-Manufacturing Technology, 57(1), 45-48

Iyengar, A., MacNair, E., & Nguyen, T. (1997, November). An analysis of web server performance. In Global Telecommunications Conference, 1997. GLOBECOM'97., IEEE (Vol. 3, pp. 1943-1947). IEEE.

Jackson, P. (1998). Introduction To Expert Systems (3 ed.), Addison Wesley, p. 2, ISBN 978-0-201-87686-4

Janis, I. L. & Mann, L. (1977). Decision making: A psychological analysis of conflict, choice, and commitment. Free Press

Kamrani, A. K., Sferro, P. & Handelman, J. (1995). Critical Issues in Design and Evaluation of Computer Aided Process Planing Systems. Computers & Industrial Engineering, Volume 29, Issues 1–4, September 1995, Pages 619-623, ISSN 0360-8352, http://dx.doi.org/10.1016/0360-8352(95)00144-P

Katalinic, B. (1990). Industrieroboter und flexible Fertigungsysteme für Drehteile, VDI, Verlag, ISBN 3-18-40-1027-9, Duesseldorf 2

Katalinic, B. (1997). Design Methodology of Scheduling Strategies and Scenarios of Complex Flexible Systems, Proc. 29th CIRP International Seminar on Manufacturing Systems (Ed. K. Iwata & K. Ueda), pp 179-185, Osaka University, May 11-13, 1997, Osaka, Japan

Katalinic, B., Visekruna, V. & Kordic, V. (2002). Bionic Assembly Systems: Design and Scheduling of Next Generation of Self-organising Complex Flexible Assembly System in CIM environment, Proceedings of The 35th CIRP-International Seminar on Manufacturing Systems, 12-15 May 2002, Seoul, Korea

Katalinic, B., Pryanichnikov, V., Ueda, K., Torims, T., Kukushkin, I., Cesarec, P. & Kettler, R. (2013). Control Structure and Scheduling of a Hybrid Assembly System. Estonian Journal of Engineering, 2013, Vol.19, Iss.1, pp.18-29. ISSN 1736-7522. doi:10.3176/eng.2013.1.03

Katalinic, B., Kukushkin, I., Pryanichnikov, V. & Haskovic, D. (2014). Cloud Communication Concept for Bionic Assembly System, Procedia Engineering, Volume 69, 2014, Pages 1562-1568, ISSN 1877-7058, doi:10.1016/j.proeng.2014.03.156

Katalinic, B., Kukushkin, I. & Haskovic, D. (2014). Bionic Assembly System Cloud: Functions, Information Flow and Behavior. Proceedings of the 9th International Conference of DAAAM Baltic, Industrial Engineering, 24-26 April 2014, Tallinn, Estonia , pp 103 – 108, ed. T. Otto, Tallinn, Estonia

Kirilchenko A., Pryanichnikov V., Rogozin K.V. (2013) Limits of reliability and reliability of evidence. Skepticism in mathematics, functions, traditions // Information-measuring and control systems. - Vol. 11, No. 4. - P. 57-65.

Klir, G. & Yuan, B. (1995). Fuzzy sets and fuzzy logic (Vol. 4). New Jersey: Prentice hall

Koestler, A. (1967). The Ghost in the machine (Arcana books, London)

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. & Van Brussel, H.(1999). Reconfigurable Manufacturing Systems, CIRP Annals - Manufacturing Technology, Volume 48, Issue 2, 1999, Pages 527-540, ISSN 0007-8506, http://dx.doi.org/10.1016/S0007-8506(07)63232-6

Kukushkin, I. (2014). Development of working scenarios and strategies for self-organizing assembly systems. Wien, Techn. Univ., Dissertation

Kumar, A. (2013). Types of Production Processes, Published on Jan 8, 2013, Class Notes for PGDM Second Semester students

Kusiak, A. (1987). Artificial intelligence and operations research in flexible manufacturing systems. INFOR: Information Systems and Operational Research, 25(1), 2

Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. META Group Research Note, 6, 70

Langton, C. G. (1997). Artificial life: An overview. MIT Press

Lee, J., Kao, H. A., & Yang, S. (2014). Service innovation and smart analytics for industry 4.0 and big data environment. Procedia Cirp, 16, 3-8.

Lee, K. (1999). Principles of CAD/CAM/CAE Systems. Addison-Wesley Longman Publishing Co., Inc.

Leitao, P. & Restivo, F. (2003). Towards Autonomy, Self-Organisation and Learning in Holonic Manufacturing, Multi-Agent Systems and Applications III, Volume 2691 of the series Lecture Notes in Computer Science, pp 544-553, doi: 10.1007/3-540-45023-8_52

Leitao, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey, Engineering Applications of Artificial Intelligence, Volume 22, Issue 7, October 2009, Pages 979-991, ISSN 0952-1976, http://dx.doi.org/10.1016/j.engappai.2008.09.005

Liang, S. Y., Hecker, R. L. & Landers, R. G. (2002). Machining Process Monitoring and Control: The State–of–the–Art. In ASME 2002 International Mechanical Engineering Congress and Exposition (pp. 599-610). American Society of Mechanical Engineers

Lipson, H. & Kurman, M. (2013). Fabricated: The New World of 3D Printing, John Wiley & Sohns Inc., ISBN 978-1-118-35063-8

Lotter, B. & Wiendahl, H. P. (2006). Montage in der industriellen Produktion. Springer, Verlag Berlin Heidelberg

Machizawa, M. G., Goh, C. C. & Driver, J. (2012). Human visual short-term memory precision can be varied at will when the number of retained items is low. Psychological science,23(6), 554-559

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. & Hung Byers, A. (2011). Big data: The next frontier for innovation, competition, and productivity, McKinsey Global Institute

McCormick, R. (1997). Conceptual and procedural knowledge. International journal of technology and design education, 7(1-2), 141-159

Menzies, T., Greenwald, J. & Frank, A. (2007). Data mining static code attributes to learn defect predictors. IEEE transactions on software engineering, 33(1), 2-13

Michalski, R. S., Carbonell, J. G. & Mitchell, T. M. (Eds.). (2013). Machine learning: An artificial intelligence approach. Springer Science & Business Media

Minsky M, (1975). A Framework for Representing Knowledge, in P. WINSTON, Ed., The Psychology of Computer Vision, pp. 211-277., McGrawHill, New York

Minsky, M. (1985). The society of Mind, Simon and Schuster, New York

Mitchell, M. (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press. ISBN 9780585030944

Monge, G. (1811). Géométrie descriptive. J. Klostermann fils

Müller, B., Reinhardt, J. & Strickland, M. T. (2012). Neural networks: an introduction. Springer Science & Business Media

Nakajima, T., Ishikawa, H., Tokunaga, E., & Stajano, F. (2002). Technology challenges for building internet-scale ubiquitous computing. In Object-Oriented Real-Time Dependable Systems, 2002.(WORDS 2002). Proceedings of the Seventh International Workshop on (pp. 171-179). IEEE.

National Research Council (US). Committee on Innovations in Computing, Communications, & Lessons from History. (1999). Funding a revolution: government support for computing research. National Academies Press

Newell, A. & Simon, H. A. (1972). Human problem solving (Vol. 104, No. 9). Englewood Cliffs, NJ: Prentice-Hall

Owen, A.E. (1984). Flexible Assembly Systems Assembly by Robots and Computerized Integrated Systems, Plenum Press, New York, NY

Panfilov, P., Salibekyan, S. & Leokhin, Y. (2016). Simulation Modeling Environment for Complex Data Driven Distributed Systems Design, Proceedings of the 27th DAAAM International Symposium, pp.0391-0399, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-08-2, ISSN 1726-9679, Vienna, Austria, DOI: 10.2507/27th.daaam.proceedings.058

Pine, B. J. (1991). Paradigm shift--from mass production to mass customization (Doctoral dissertation, Massachusetts Institute of Technology)

Polckofl, J. C. (1990). Computer Integrated Manufacturing. Journal of Accountancy, 25

Poole, D. L. & Mackworth, A. K. (2010). Artificial Intelligence: foundations of computational agents. Cambridge University Press

Post, P. (2014). "Smart Systems for Intelligent Manufacturing - Industry 4.0" Plenary Lecture, 25th DAAAM International Symposium, Vienna, Austria

Pryanichnikov, V., Aryskin, A., Eprikov, S., Kirsanov K., Khelemendik, R., Ksenzenko, A., Prysev, E. & Travushkin, A. (2017). Technology of Multi-Agent Control for Industrial Automation with Logical Processing of Contradictions, Proceedings of the 28th DAAAM International Symposium, pp.1202-1207, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-11-2, ISSN 1726-9679, Vienna, Austria, DOI: 10.2507/28th.daaam.proceedings.167

Pryanichnikov V., Helemendik R.V. (2016) Information technology IGEC and pentalogy / / Computer Science and Information Technology: Materials of International. sci. Conf. - Saratov: The publication. center "Science" - P. 331-333.

Puppe, F. (1991). Einführung in Expertensysteme. Springer-Verlag

Rodriguez-Toro, C., Jared, G. & Swift, K. (2004). Product-Development Complexity Metrics: A Framework for Proactive-DFA Implementation, Pages 483-490, International Design Conference - Design 2004, Dubrovnik, May 18 - 21, 2004

Russell, S.J. & P. Norvig. (2002). Artificial Intelligence: A Modern Approach, 2nd ed. Prentice Hall

Rychener, M. (Ed.). (2012). Expert systems for engineering design. Elsevier

Saaksvuori, A. & Immonen, A. (2008). Product Lifecycle Management, Springer, ISBN 978-3-540-78173-8

Saed Sayad, S. (2011). Real Time Data Mining, Paperback, ISBN-13: 978-0986606045 Self-Help Publishers

Schaller, R. R. (1997). Moore's law: past, present and future. IEEE spectrum, 34(6), 52-59.

Schank, R. (1991). Where is the AI, AI Magazine, 12(4):38-49

Smith, A. (2007). An Inquiry into the Nature and Causes of the Wealth of Nations. Edited by Sálvio M. Soares. MetaLibri, v.1.0p.

Spath, D., Scholz, O. (2007). Mutability for a Profitable Assembly in Germany (in German). Industrie Management, Vol. 23, No. 2, pp. 61-64

Stecke, K. E. (1983). Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. Management Science, 29(3), 273-288.

Stevens, S. S. (1946). On the theory of scales of measurement

Tharumarajah, A., Wells, A. J. & Nemes, L., (1998). Comparison of emerging manufacturing concepts. In Systems, Man, and Cybernetics, 1998 IEEE International Conference on (Vol. 1, pp. 325-331). IEEE

Troy, T. F. (1991). The “correct” definition of intelligence. International Journal of Intelligence and Counter Intelligence, 5(4), 433-454

Tseng, M., Law, P. & Cerva, T. (1992). Knowledge Based Systems. Industrial Engineering Handbook, Chapter 7

Waterman, D. (1986). A guide to expert systems

Whitney, D. E. (2004). Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development. New York, NY: Oxford University Press, 2004. ISBN:0195157826

Woodbury, R. S. (1960). The legend of Eli Whitney and interchangeable parts. Technology and Culture, 1(3), 235-253

Yates, F. E. (Ed.). (2012). Self-organizing systems: The emergence of order. Springer Science & Business Media

Zharova, A., Elin, V. & Panfilov, P. (2017). Personal Data in Cloud. Russia Experience, Proceedings of the 28th DAAAM International Symposium, pp.1136 1142, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-11-2, ISSN 1726-9679, Vienna, Austria, DOI: 10.2507/28th.daaam.proceedings.158

List of Figures

List of Tables

Appendix A

IAM Interface Source Code

Appendix A contains the full source code for the Intelligent Adviser Module. Clips 6.3 platform was used to facilitate the development of the IAM rule-based decision support functions. Main purpose is to demonstrate the IAM interface. The code contains rules, fact assertions, salience declarations, menu definitions, load new knowledge via Knowledge Record File loading functions, solve active problems via Problem Record File loading functions and printout functions.

```
;======================================================
; Intelligent Adviser Module for Bionic Assembly System
; Author: Damir Haskovic, 2016/2017 - TU WIEN / IFT - IMS
; Written for CLIPS Version 6.3
;======================================================
;1) Assembly Station*

(defrule ass_station_question_turn_on
(main_choice 1)
(not (solution ?))
=>
(printout t "Is the assembly station able to turn on (y/n)? ")
(assert (ass_station_turn_on (read))))

(defrule ass_station_question_operate_normally
(main_choice 1)
(ass_station_turn_on y)
(not (solution ?))
=>
(printout t "Is the assembly station operating normally (y/n)? ")
(assert (ass_station_operating_normally (read))))

(defrule ass_station_no_repair_needed
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally y)
(not (solution ?))
=>
(printout t "---------------------------------------------------"crlf)
(printout t "Congratulations - the assembly station is in order!"crlf)
(printout t "No further action is required!" crlf)
(printout t "---------------------------------------------------"crlf))

(defrule ass_station_question_servo_drive
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally n)
(not (solution ?))
=>
(printout t "Is the assembly station's servo drive operating irregularly (y/n)? ")
(assert (ass_station_servo_drive (read))))

(defrule ass_station_question_start_operating
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally n)
(ass_station_servo_drive n)
(not (solution ?))
=>
(printout t "Can the assembly station start operating (y/n)? ")
(assert (ass_station_start_operating (read))))

(defrule ass_station_choice_servo_drive_yes
```

```
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally n)
(ass_station_servo_drive y)
(not (solution ?))
=>
(printout t "---------------------------------------------------------------------------------"crlf)
(printout t "Perform a visual inspection of the servo drive and select one of the following:"crlf)
(printout t "1 - Shorted power transistor"crlf)
(printout t "2 - Improper grounding or shielding"crlf)
(printout t "3 - The amplifier current is incorrect"crlf)
(printout t "4 - None"crlf)
(printout t "---------------------------------------------------------------------------------"crlf)
(printout t "Your Choice: " )
(assert (servo_drive_choice (read))))

(defrule ass_station_solution_servo_drive_choice_1
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally n)
(ass_station_servo_drive y)
(servo_drive_choice 1)
(not (solution ?))
=>
(printout t "--------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Replace the shorted power transistor module."crlf)
(printout t "--------------------------------------------"crlf)
(assert (solution ok)))

(defrule ass_station_solution_servo_drive_choice_2
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally n)
(ass_station_servo_drive y)
(servo_drive_choice 2)
(not (solution ?))
=>
(printout t "--------------------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Improper grounding or shielding is causing noise."crlf)
(printout t "Perform a proper grounding and install sufficient shielding."crlf)
(printout t "--------------------------------------------------------------"crlf)
(assert (solution ok)))

(defrule ass_station_solution_servo_drive_choice_3
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally n)
(ass_station_servo_drive y)
(servo_drive_choice 3)
(not (solution ?))
=>
(printout t "--------------------------------------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Servo amplifier current rating is insufficient to supply the sufficient torque."crlf)
(printout t "Ensure that the amplifier is supplying sufficent torque."crlf)
(printout t "--------------------------------------------------------------------------------"crlf)
(assert (solution ok)))

(defrule ass_station_solution_servo_drive_choice_4
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally n)
(ass_station_servo_drive y)
(servo_drive_choice 4)
(not (solution ?))
=>
=>
(printout t "-----------------------------------------"crlf)
(printout t "Unable to determine the cause of problem."crlf)
(printout t "The Knowledge Base needs to be updated." crlf)
(printout t "-----------------------------------------"crlf)
(assert (solution ok)))

(defrule ass_station_solution_start_operating_yes
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally n)
(ass_station_start_operating y)
(not (solution ?))
=>
(printout t "-----------------------------------------"crlf)
(printout t "Unable to determine the cause of problem."crlf)
(printout t "The Knowledge Base needs to be updated." crlf)
(printout t "-----------------------------------------"crlf)
(assert (solution ok)))
```

```
(defrule ass_station_solution__start_operating_no
(main_choice 1)
(ass_station_turn_on y)
(ass_station_operating_normally n)
(ass_station_start_operating n)
(not (solution ?))
=>
(printout t "---------------------------------------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Safety door interlock circuit can disable the operation of the assembly station."crlf crlf)
(printout t "Complete the following:"crlf)
(printout t "- check if all the interlocking keys are fully engaged"crlf)
(printout t "- try reseting the interlocking keys (vibration can activate the keys prematurily)"crlf)
(printout t "---------------------------------------------------------------------------------"crlf)
(assert (solution ok)))

(defrule ass_station_question_connected_to_power
(main_choice 1)
(ass_station_turn_on n)
(not (solution ?))
=>
(printout t "Is the assembly station connected to power (y/n)? ")
(assert (ass_station_connected_to_power (read))))

(defrule ass_station_solution_connected_to_power_no
(main_choice 1)
(ass_station_turn_on n)
(ass_station_connected_to_power n)
(not (solution ?))
=>
(printout t "--------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Re-connect the assembly station to a power source."crlf)
(printout t "--------------------------------------------------"crlf)
(assert (solution ok)))

(defrule ass_station_question_fuse_holder
(main_choice 1)
(ass_station_turn_on n)
(ass_station_connected_to_power y)
(not (solution ?))
=>
(printout t "Is the assembly station's fuse holder broken (y/n)? ")
(assert (ass_station_fuse_holder (read))))

(defrule ass_station_solution_fuse_holder_yes
(main_choice 1)
(ass_station_turn_on n)
(ass_station_connected_to_power y)
(ass_station_fuse_holder y)
(not (solution ?))
=>
(printout t "-----------------------------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Fuse holders are prone to breaking. If broken, replace the fuse holder."crlf)
(printout t "-----------------------------------------------------------------------"crlf)
(assert (solution ok)))

(defrule ass_station_question_wire_terminals
(main_choice 1)
(ass_station_turn_on n)
(ass_station_connected_to_power y)
(ass_station_fuse_holder n)
(not (solution ?))
=>
(printout t "Are the wire terminals inside the electrical box connected? (y/n)? ")
(assert (ass_station_wire_terminals (read))))

(defrule ass_station_solution_wire_terminals_no
(main_choice 1)
(ass_station_turn_on n)
(ass_station_connected_to_power y)
(ass_station_fuse_holder n)
(ass_station_wire_terminals n)
(not (solution ?))
=>
(printout t "-------------------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Reconnect the wire terminals and secure that they are tight."crlf)
(printout t "-------------------------------------------------------------"crlf)
(assert (solution ok)))

(defrule ass_station_solution_wire_terminals_yes
(main_choice 1)
(ass_station_turn_on n)
(ass_station_connected_to_power y)
(ass_station_fuse_holder n)
(ass_station_wire_terminals y)
```

```
(not (solution ?))
=>
(printout t "-----------------------------------------"crlf)
(printout t "Unable to determine the cause of problem."crlf)
(printout t "The Knowledge Base needs to be updated." crlf)
(printout t "-----------------------------------------"crlf)
(assert (solution ok)))

;2) Mobile Robot*

(defrule mob_robot_question_able_to_move
(main_choice 2)
(not (solution ?))
=>
(printout t "Is the mobile robot able to move (y/n)? ")
(assert (mobile_robot_move (read))))

(defrule mob_robot_question_battery_status
(main_choice 2)
(mobile_robot_move n)
(not (solution ?))
=>
(printout t "Is the mobile robot's battery discharged (y/n)? ")
(assert (mob_robot_battery (read))))

(defrule mob_robot_solution_battery_status_yes
(main_choice 2)
(mob_robot_battery y)
(not (solution ?))
=>
(printout t "------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Replace or recharge the mobile robot's battery." crlf)
(printout t "------------------------------------------------"crlf)
(assert (solution ok)))

(defrule mob_robot_solution_battery_status_no
(main_choice 2)
(mob_robot_battery n)
(not (solution ?))
=>
(printout t "------------------------------------------"crlf)
(printout t "Unable to determine the cause of problem."crlf)
(printout t "Replace the mobile robot unit." crlf)
(printout t "------------------------------------------"crlf)
(assert (solution ok)))

(defrule mob_robot_question_move_no
(main_choice 2)
(mobile_robot_move n)
(not (solution ?))
=>
(printout t "Is the mobile robot's battery discharged (y/n)? ")
(assert (mob_robot_battery (read))))

(defrule mob_robot_question_move_normally
(main_choice 2)
(mobile_robot_move y)
(not (solution ?))
=>
(printout t "Is the mobile robot able to move without problems (y/n)? ")
(assert (mob_robot_move_normally (read))))

(defrule mob_robot_no_repair_needed
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally y)
(not (solution ?))
=>
(printout t "-----------------------------------------------"crlf)
(printout t "Congratulations - the mobile robot is in order!"crlf)
(printout t "No further action is required!" crlf)
(printout t "-----------------------------------------------"crlf))

(defrule mob_robot_question_navigation
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
(not (solution ?))
=>
(printout t "Is the mobile robot's navigation module functional (y/n)? ")
(assert (mob_robot_navigation (read))))

(defrule mob_robot_solution_navigation
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
(mob_robot_navigation n)
```

```
(not (solution ?))
=>
(printout t "--------------------------------------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Reset the mobile robot's navigation module." crlf)
(printout t "If the problem is still present, replace the mobile robot's navigation module." crlf)
(printout t "--------------------------------------------------------------------------------"crlf)
(assert (solution ok)))

(defrule mob_robot_question_sensors
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
(not (solution ?))
=>
(printout t "Are the mobile robot's sensors functional (y/n)? ")
(assert (mob_robot_sensors (read))))

(defrule mob_robot_solution_sensors
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
(mob_robot_sensors n)
(not (solution ?))
=>
(printout t "----------------------------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Inspect the mobile robot's sensors and reset if needed." crlf)
(printout t "If the problem is still present, replace the mobile robot's sensors." crlf)
(printout t "----------------------------------------------------------------------"crlf)
(assert (solution ok)))

(defrule mob_robot_question_speed
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
(not (solution ?))
=>
(printout t "Is the mobile robot's speed low (y/n)? ")
(assert (mob_robot_speed (read))))

(defrule mob_robot_solution_speed_no
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
(mob_robot_speed n)
(not (solution ?))
=>
(printout t "------------------------------------------"crlf)
(printout t "Unable to determine the cause of problem."crlf)
(printout t "Replace the mobile robot unit." crlf)
(printout t "------------------------------------------"crlf)
(assert (solution ok)))

(defrule mob_robot_solution_speed_yes
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
(mob_robot_speed y)
(not (solution ?))
=>
(printout t "---------------------------------------------------------------------------------"crlf)
(printout t "Perform a visual inspection of the mobile robot and select one of the following:"crlf)
(printout t "1 - Faulty mobile robot's wheel"crlf)
(printout t "2 - Malfunctioning mobile robot's motor"crlf)
(printout t "3 - None"crlf)
(printout t "---------------------------------------------------------------------------------"crlf)
(printout t "Your Choice: " )
(assert (speed_choice (read))))

(defrule mob_robot_solution_speed_yes_1
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
(mob_robot_speed y)
(speed_choice 1)
(not (solution ?))
=>
(printout t "---------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Replace the faulty mobile robot's wheel." crlf)
(printout t "---------------------------------------"crlf)
(assert (solution ok)))

(defrule mob_robot_solution_speed_yes_2
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
```

```
(mob_robot_speed y)
(speed_choice 2)
(not (solution ?))
=>
(printout t "--------------------------------------------------"crlf)
(printout t "Solution:"crlf)
(printout t "Replace the malfunctioning mobile robot's motor." crlf)
(printout t "--------------------------------------------------"crlf)
(assert (solution ok)))

(defrule mob_robot_solution_speed_yes_3
(main_choice 2)
(mobile_robot_move y)
(mob_robot_move_normally n)
(mob_robot_speed y)
(speed_choice 3)
(not (solution ?))
=>
(printout t "-----------------------------------------"crlf)
(printout t "Unable to determine the cause of problem."crlf)
(printout t "Replace the mobile robot unit." crlf)
(printout t "-----------------------------------------"crlf)
(assert (solution ok)))

;3) Product Quality!*

(defrule product_quality_load
(main_choice 3)
(not (solution ?))
=>
(load-facts "D:/Google Drive/2. Doktorat/1. Dissertation/1.Haskovic_Damir/Clips/Code/PRF_2016-10-22-10-
04-17_0036_product_quality.clp"))


(defrule product_quality_question_no
(main_choice 3)
(not (solution ?))
=>
(printout t "-----------------------------------------"crlf)
(printout t "Unable to determine the cause of problem."crlf)
(printout t "The Knowledge Base needs to be updated." crlf)
(printout t "-----------------------------------------"crlf)
(assert (solution ok)))

;4) Load New Knowledge*

(defrule load_new_knowledge
(declare (salience 20))
(main_choice 4)
=>
(load "D:/Google Drive/2. Doktorat/1. Dissertation/1.Haskovic_Damir/Clips/Code/KRF_2016-11-19-16-01-
29_340.clp"))

;5) Solve active problem!*

(defrule solve_active_problem
(main_choice 5)
=>
(load-facts "D:/Google Drive/2. Doktorat/1. Dissertation/1.Haskovic_Damir/Clips/Code/PRF_2016-09-10-12-
55-33_0022.clp"))

;*Main Menu*
(defrule main-menu "Lists all the current IAM functions"
 (declare (salience 10))
(or (main_choice 4) (initial-fact))
 (not (solution ?))
  =>
    (printout t crlf)
    (printout t "------------------------------------------------------"crlf)
    (printout t "Intelligent Adviser Module for Bionic Assembly System"crlf)
    (printout t "------------------------------------------------------"crlf)
    (printout t "Make a general inquiry about:"crlf)
    (printout t "1 - Assembly Stations"crlf)
    (printout t "2 - Mobile Robots"crlf)
    (printout t "3 - Product Quality"crlf)
    (printout t "----------------------------"crlf)
    (printout t "Additional functions:"crlf)
    (printout t "4 - Load New Knowledge"crlf)
    (printout t "5 - Solve active problem"crlf)
    (printout t "------------------------------------------------------"crlf)
    (printout t "Your Choice: " )
    (assert (main_choice (read)))
    (printout t crlf crlf))
```

Appendix B

Experiment Source Code

This listing presents the code for the "*2D problem compensation*" experiment. A source code editor called Notepad ++ (*****, 2018) was used to facilitate the development. Testing was performed in a standard web browser.

```
<script>
/*
;===========================================================
; 2D Error Compensation Experiment:
; Intelligent Adviser Module for Bionic Assembly System
; Author: Damir Haskovic, 2017/2018 - TU WIEN / IFT - IMS
;===========================================================
*/
</script>

<!doctype html>
<html>
<head>
<meta charset="UTF-8" />
<title>IAM - Case Study</title>
</head>
<body>
<section>
<div><canvas id="canvas" width="630" height="630"></canvas></div>

<style>
    div.output {
    background-color: #d2f5f7;
    width: 314px;
    height: 200px;
    overflow: scroll; }
    position: absolute;
</style>

<div class="output" id="log" shape="rect"></div>
<div  id="IAM" style="position: absolute; top: 672px; left: 325px; width: 314px; height: 170px;
background-color: lightgreen;"></div>
<div  id="Title" style="position: absolute; top: 642px; left: 325px; width: 314px; height: 30px;
background-color: lightblue;" align="center">
***Intelligent Adviser Module***
</div>

<script>

var counter=0;
var disturbance, canvas, ctx, compare;
var text = "";
var dx, dy = 30;
var WIDTH, HEIGHT = 630;
var n = 0;
var color = "red";
var background = "white";
var solved = 0;
var directions ="";
var attention_color = "transparent";

var x = (Math.floor((Math.random() * 20) + 0))*30;
var y = (Math.floor((Math.random() * 20) + 0))*30;

var error = (Math.abs(x-300)+Math.abs(y-300))/30;
var error2 = (Math.abs(x-300)+Math.abs(y-300))/30;
var d,h,m,s,ms,occured; //timeGenerate function variables
```

```
var keyArrow1, keyArrow2,keyArrow3,keyArrow4, storage, randomnumber,randomDisturbance; //randomKey
function variables
var direction_a, direction_b, direction_c, direction_d;
var whereX, whereY,delay;

function timeGenerate () {
d = new Date();
h = d.getHours();
m = d.getMinutes();
s = d.getSeconds();
ms = d.getMilliseconds();
occured = h + ":" + m + ":" + s + "." + ms;
time1=m*60000+s*1000+ms;
}

function randomKey () {
document.getElementById("IAM").innerHTML = "";
randomDisturbance = (Math.floor(Math.random()*3) + 1); //*1 for simple, *3 for complex

if (randomDisturbance == 1) {
    keyArrow1 = 38; //otherwise up 38
    keyArrow2 = 40; //otherwise down 40
    keyArrow3 = 37; //otherwise left 37
    keyArrow4 = 39; //otherwise right 39
    }

if (randomDisturbance == 2) {
    storage = [];

    randomnumber = (Math.floor(Math.random()*2) + 1);

        if (randomnumber == 1)  {
        keyArrow1 = 40; //otherwise up 38
        keyArrow2 = 38; //otherwise down 40
        keyArrow3 = 37; //otherwise left 37
        keyArrow4 = 39; //otherwise right 39
        }

        if (randomnumber == 2)  {
        keyArrow1 = 38; //otherwise up 38
        keyArrow2 = 40; //otherwise down 40
        keyArrow3 = 39; //otherwise left 37
        keyArrow4 = 37; //otherwise right 39
        }
    }

if (randomDisturbance == 3) {
    storage = [];

    while(storage.length < 4){
    randomnumber = (Math.floor(Math.random()*4) + 0)+37;
        if(storage.indexOf(randomnumber) > -1) continue;
        storage[storage.length] = randomnumber;
    }

    keyArrow1 = storage[0]; //otherwise up 38
    keyArrow2 = storage[1]; //otherwise down 40
    keyArrow3 = storage[2]; //otherwise left 37
    keyArrow4 = storage[3]; //otherwise right 39
    }
}

//turn on IAM
function IAM_Help () {
directions ="";
switch (keyArrow1) {
        case 38:
            direction_a="Up";
            break;
        case 40:
            direction_a="Down";
            break;
        case 37:
            direction_a="Left";
            break;
        case 39:
            direction_a="Right";
            break;
        }

 switch (keyArrow2) {
        case 38:
            direction_b="Up";
            break;
        case 40:
            direction_b="Down";
            break;
        case 37:
```

```
            direction_b="Left";
            break;
        case 39:
            direction_b="Right";
            break;
        }

    switch (keyArrow3) {
        case 38:
            direction_c="Up";
            break;
        case 40:
            direction_c="Down";
            break;
        case 37:
            direction_c="Left";
            break;
        case 39:
            direction_c="Right";
            break;
        }

  switch (keyArrow4) {
        case 38:
            direction_d="Up";
            break;
        case 40:
            direction_d="Down";
            break;
        case 37:
            direction_d="Left";
            break;
        case 39:
            direction_d="Right";
            break;
        }

whereX = (x-300)/30;
whereY = (y-300)/30;

    if (whereX > 0 && whereY < 0) {
    document.getElementById("IAM").innerHTML = "Press " + direction_c +" " + Math.abs(whereX) + "x & "
+direction_b +" "+ Math.abs(whereY)+"x"; //left down
    directions = "Press " + direction_c +" " + Math.abs(whereX) + "x & " +direction_b +" "+
Math.abs(whereY)+"x";
    }
    else if (whereX > 0 && whereY > 0) {
    document.getElementById("IAM").innerHTML = "Press " + direction_c +" " + Math.abs(whereX) + "x & "
+direction_a +" "+ Math.abs(whereY)+"x";  // left up
    directions = "Press " + direction_c +" " + Math.abs(whereX) + "x & " +direction_a +" "+
Math.abs(whereY)+"x";
    }
    else if (whereX < 0 && whereY < 0)  {
    document.getElementById("IAM").innerHTML = "Press " + direction_d +" " + Math.abs(whereX) + "x & "
+direction_b +" "+ Math.abs(whereY)+"x"; // right down
    directions = "Press " + direction_d +" " + Math.abs(whereX) + "x & " +direction_b +" "+
Math.abs(whereY)+"x";
    }
    else if (whereX < 0 && whereY > 0)  {
    document.getElementById("IAM").innerHTML = "Press " + direction_d +" " + Math.abs(whereX) + "x & "
+direction_a +" "+ Math.abs(whereY)+"x"; // right up
    directions ="Press " + direction_d +" " + Math.abs(whereX) + "x & " +direction_a +" "+
Math.abs(whereY)+"x";
    }
    else if (whereX == 0 && whereY < 0)  {
    document.getElementById("IAM").innerHTML = "Press " + direction_b +" " + Math.abs(whereY)+"x"; //
down
    directions = "Press " + direction_b +" " + Math.abs(whereY)+"x";
    }
    else if (whereX == 0 && whereY > 0)  {
    document.getElementById("IAM").innerHTML = "Press " + direction_a +" " + Math.abs(whereY)+"x";  //up
    directions = document.getElementById("IAM").innerHTML = "Press " + direction_a +" " +
Math.abs(whereY)+"x";
    }
    else if (whereX > 0 && whereY == 0)  {
    document.getElementById("IAM").innerHTML = "Press " + direction_c +" " + Math.abs(whereX)+"x"; //left
    directions = "Press " + direction_c +" " + Math.abs(whereX)+"x";
    }
    else if (whereX < 0 && whereY == 0)  {
    document.getElementById("IAM").innerHTML = "Press " + direction_d +" " + Math.abs(whereX)+"x";  //
right
    directions = "Press " + direction_d +" " + Math.abs(whereX)+"x";
    }

    else {
    //alert("ok");
    }
```

```
 }

function again () {
    x = (Math.floor((Math.random() * 20) + 0))*30;
    y = (Math.floor((Math.random() * 20) + 0))*30;

    randomKey();
    error = (Math.abs(x-300)+Math.abs(y-300))/30;
    error2 = (Math.abs(x-300)+Math.abs(y-300))/30;
    solved = 0;
    timeGenerate();
    color = "red";
    background ="white";
    attention_color = "transparent";
    n = 0;
    compare = error;
}

function check () {

    if (keyArrow1 ==38 && keyArrow2 == 40 && keyArrow3 == 37 && keyArrow4 == 39) {
    disturbance = 1;
    delay = (error*100)*disturbance; //maximum delay is 2 seconds for no disturbance
    }

    else if (keyArrow1 ==40 && keyArrow2 ==38 && keyArrow3 ==37 && keyArrow4 ==39) { //maximum delay is 4
seconds for medium disturbance
    disturbance = 2;
    delay = (error*100)*disturbance;
    }

    else if (keyArrow1 ==38 && keyArrow2 ==40 && keyArrow3 ==39 && keyArrow4 ==37) { //maximum delay is 4
seconds for medium disturbance
    disturbance = 2;
    delay = (error*100)*disturbance;
    }

    else {
    disturbance = 3;   //maximum delay is 9 seconds for highest disturbance
    delay = (error*150)*disturbance;
    }
}

function doKeyDown(evt){

    switch (evt.keyCode) {
        case keyArrow1: //up 38
            if (y - dy >= 0){
            y -= dy;
            n+=1;
        }
        break;

        case keyArrow2: //down 40
        if (y + dy < HEIGHT){
        y += dy;
        n+=1;
        }
        break;

        case keyArrow3: //left 37
        if (x - dx >= 0){
        x -= dx;
        n+=1;
        }
        break;

        case keyArrow4: //right 39
        if (x + dx < WIDTH){
        x += dx;
        n+=1;
        }
        break;
    }

if (x==300 && y==300) {

                var d = new Date();
                var h = d.getHours();
                var m = d.getMinutes();
                var s = d.getSeconds();
                var ms = d.getMilliseconds();

                solved = h + ":" + m + ":" + s + "." + ms;
                document.getElementById("log").innerHTML += counter + ";" + n + ";" + error + ";" +
occured + ";"+ solved + ";" + delay +";"+disturbance+"<br>";
                color = "green";
```

```
                setTimeout(again, 500);
                setTimeout(IAM_Help, delay); //turn on IAM
                counter ++;

                    if (counter ==200) { //set number
                        alert("You have completed your session - Thank you!");
                    }
                }

                    else {
                    color = "red";
                    }

error2 = (Math.abs(x-300)+Math.abs(y-300))/30;

if (compare<error2) {  // error increase tolerance
    attention_color = "black";
    background ="yellow";
    }

    else {
        compare = error2;
        }

check ()

var d = new Date();
var m = d.getMinutes();
var s = d.getSeconds();
var ms = d.getMilliseconds();
var miliseconds = m*60000+s*1000+ms
var result = miliseconds-time1;

    if (result > delay)  {
    IAM_Help();  //turn on IAM
    }
}

function target(x,y,w,h) {
ctx.beginPath();
ctx.rect(x,y,w,h);
ctx.fill();
//ctx.font = "15px Arial";
//ctx.fillText("Solved");
}

function rect(x,y,w,h) {
ctx.beginPath();
ctx.rect(x,y,w,h);
ctx.closePath();
ctx.fill();
ctx.stroke();
}

function axis(x,y,w,h) {
ctx.beginPath();
ctx.rect(x,y,w,h);
ctx.fill();
}

function clear() {
ctx.clearRect(0, 0, WIDTH, HEIGHT);
}

function init() {
canvas = document.getElementById("canvas");
ctx = canvas.getContext("2d");
return setInterval(draw, 20);
}

function draw() {
clear();
ctx.fillStyle = background;
ctx.strokeStyle = "black";
rect(0,0,WIDTH,HEIGHT);

ctx.fillStyle = color;
target(x, y, 30,30); //Error coordinates

ctx.fillStyle = "black";
axis(0, 314, 630,2);
axis(314, 0, 2,630);

ctx.font = "15px Arial";
ctx.fillText("Run: " + counter +" of 200.",519,20);

if (x<315 && y<315) {
ctx.fillStyle = attention_color;
```

```
//ctx.font = "15px Arial bold";
ctx.fillText(directions,x,y+50);
}

if (x<315 && y>315) {
ctx.fillStyle = attention_color;
//ctx.font = "15px Arial bold";
ctx.fillText(directions,x,y-20);
}

if (x>315 && y<315) {
ctx.fillStyle = attention_color;
//ctx.font = "15px Arial bold";
ctx.fillText(directions,x-160,y+50);
}

if (x>315 && y>315) {
ctx.fillStyle = attention_color;
//ctx.font = "15px Arial bold";
ctx.fillText(directions,x-160,y-20);
}
/* variable testing*/
ctx.fillStyle = "black";

}

alert("If the area turns yellow - Wait for advice!");
init();
randomKey();
check ();
setTimeout(IAM_Help, delay);   //turn on IAM
timeGenerate();
compare = error;

window.addEventListener('keyup',doKeyDown,true);
window.removeEventListener('keydown',doKeyDown,true);

</script>
</section>
</body>
</html>
```

Angaben zur Person

Vorname, Nachname	Damir Hasković, mag. ing. mech.
Adresse	Arnethgasse 31/28, 1160 Vienna, Austria
Kontakt (Österreich)	+43 680 2248322
Kontakt (Kroatien)	+385 98 9003114
E-Mail	haskovic.damir@gmail.com
Staatsbürgerschaft	Kroatien, EU
Geburtsdatum, -ort	12. 05. 1987, Rijeka (Kroatien)
Interessen	Autos, Technologie, E-Gitarre, Musik- und Videobearbeitung, Schwimmen, Laufen

Ausbildung

01/2013 – 09/2018 (voraussichtlich)

Doktor der Technischen Wissenschaften (Dr. techn.)

- Technische Universität Wien, Wien (Österreich)
- Maschinenbau: Institut für Fertigungstechnik und Hochleistungslasertechnik
- Forschung: Arbeitsszenarien des hybriden selbstorganisierenden Montagesystems

09/2009 - 09/2011

Master of Mechanical Engineering (mag. ing. mech)

- Technische Universität Rijeka (Kroatien)
- Maschinenbau: Technologisch - informatisches Ingenieurwesen; magna cum laude und in Mindeststudienzeit abgeschlossen; Leistungsstipendium erhalten
- Forschung: Entwicklung eines virtuellen Modells und Simulation des Produktionsprozesses für die Firma Alpron

09/2006 - 09/2009

Bachelor of Mechanical Engineering (bacc. ing. mech)

- Technische Universität Rijeka (Kroatien)
- Maschinenbau; magna cum laude und in Mindeststudienzeit abgeschlossen
- Forschung: Entwicklung eines Konzeptes für die Online-Maschine-Steuerung auf Basis von Open-Source-Technologie

Arbeitserfahrung

01/2013 – 07/2018 (voraussichtlich)

Assistent des Professors und Wissenschaftler

Technische Universität Wien, Wien (Österreich)

- Vorlesungen, Prüfungen, Bewertungen, Konsultationen
- Mitglied des Intelligent Manufacturing Systems Group von Prof. Dr.sc. Dr.mult. h.c. Prof.h.c. Branko Katalinic
- Optimierung von Produktionssystemen der nächsten Generation

03/2015 – 08/2015

Gastwissenschaftler

- Boston University, Boston (Vereinigte Staaten)
- Gastwissenschaftler durch Marshallplan Stipendium / Boston University, USA
- Gastinstitution: Hybrid and Networked Systems (HyNeSs) Lab, Prof. Calin Belta

01/2013 – 09/2018 (voraussichtlich)

Generalsekretär

- DAAAM International Vienna, Wien (Österreich)
- - Organisation der Konferenzen, Programmierung, Design, International Team – Leader, web Administrator

06/2010

Maschinenbau - Techniker (Praktikum)

- Elcon Gerätebau, Rijeka (Kroatien)
- Arbeit mit CNC-Maschinen, Erstellung von technischen Dokumentationen in Fimensoftware, Prüfung der Umsetzung von technischen Entwürfen in AutoCAD, Erstellung von 3D-Modellen in MasterCAM.

06/2008	**Assistent des technischen Inspektors (Praktikum)** - American Bureau of Shipping, Rijeka (Kroatien) - Aufsicht der Produktion von Schiffsteilen, Kontrolle der Einhaltung von Sicherheits-, Konstruktions- und Instandhaltungsstandards.

Persönliche Erfolge

2013 - heute	Technischer Redakteur und Rezensent für „*DAAAM Proceedings*“ and „*DAAAM Scientific Books*“
2013 - heute	Hauptdesigner für die visuelle Identität der 24 - 29. DAAAM-Konferenzen (Bucheinbände, Poster, Zertifikaten, Flaggen, Dokumenten etc.)
2011	Präsentation der Bachelor Arbeit “*Development of a concept for web-online machine controlling based on open source technology*” vor dem kroatischen Bildungsminister, Dekan der Fakultät, Professoren, und Studenten - TU Rijeka (Kroatien)

Fertigkeiten

Sprachen	Kroatisch – Muttersprache Englisch – C2 (Fließend) Deutsch – C1 (Gut) Russisch – A2 (Anfänger)
Computerprogramme	SolidWorks, AutoCAD, ProEngineer, Siemens Tecnomatix Plant Simulation, Simio, Catia, Microsoft Office, Adobe After Effects, Adobe Flash, Adobe Ilustrator
Betriebssysteme	Windows, Macintosh, Linux, iOS, Android
Programmiersprachen	PHP, Python, HTML, AutoHotKey Scripting, Google Scripts, Java Script

Zusätzliche Info.

EU Freizügigkeitsbestätigung	AMS - Bestätigung über das Recht auf freien Arbeitsmarktzugang im gesamten Bundesgebiet Österreichs
Mitgliedschaft	DAAAM International - Verband der Ingenieure und Forscher in einem Gebiet der Automatisierung und Fertigung
Führerschein	AM, B1, B, BE

Referenzen

Dissertationsbetreuer Prof. Dr.sc. Dr.mult. h.c. Prof.h.c. Branko Katalinic, TU Wien, IFT, katalinic@mail.ift.tuwien.ac.at

Publikationen

2017 D Haskovic, B Katalinic, I Zec, I Kukushkin A & Zavrazhina, A: Intelligent Adviser Module: Proposals and Adaptive Learning Capabilities, Proceedings of the 28thDAAAM International Symposium, pp.1726-9679

2016 D Haskovic, B Katalinic, I Zec, I Kukushkin & A Zavrazhina: Structure and Working Modes of the Intelligent Adviser Module, Proceedings of the 27thDAAAM International Symposium, pp.0866-0875

2015 D Haskovic, B Katalinic, A Kildibekov& I Kukushkin: Intelligent Adviser Module for Bionic Assembly Control System: Functions and Structure Concept, Proceedings of the 26th DAAAM International Symposium, pp.1158-1165

2014 D Haskovic, B Katalinic, I Kukushkin: Role of the Adviser Module in the Hybrid Assembly Subordinating Control Structure, Procedia Engineering 100 (2015) pp.1706 – 1713

2014 B Katalinic, I Kukushkin, D Haskovic: Bionic Assembly System Cloud: Functions, Information Flow and Behaviour, 9th International Conference of DAAAM Baltic INDUSTRIAL ENGINEERING