PRODUCT-DRIVEN CONTROL IN MANUFACTURING SYSTEMS
USING IEC 61499 AND RFID TECHNOLOGY

Seno Panjaitan and Georg Frey

University of Kaiserslautern, FB EIT, JPA²
Erwin-Schrödinger-Str. 12, D-67653 Kaiserslautern, Germany
{panjaitan | frey}@eit.uni-kl.de

Abstract: To achieve re-configurability, reusability and flexibility of manufacturing systems, this paper extends a method for re-configuration of control on machine level by the use of auto-identification technology. The scheduling-based control concepts of Functionality Based Control (FBC) and Scheduler, Selector and Synchronizer (S³) are extended for a setting with RFID-technology (for product identification). Using this combination product-driven control can be achieved. Each product brings its own processing schedule on an RFID tag read by a RFID reader at the machines. Using a drill station as example, the calculation of processing schedules for a machine depending on newly arriving products is illustrated. Copyright © 2006 IFAC

Keywords: Intelligent Manufacturing Systems, Distributed Control Systems, IEC 61499, Scheduling, Re-Configuration.

1. INTRODUCTION

The innovation of automation technology goes at a very fast pace. This phenomenon rises since the market, especially in manufacturing, demands automation technology to be responsive, reconfigurable, reusable, cost and time effective, easy to design, easy to use and easy to evaluate. Currently in the supply chain area, techniques using automatic identification (Auto-ID) technology implemented by Radio Frequency Identification (RFID) are rising especially in e-logistic where the chain of parts and packages can be monitored and identified. The current focus of research is how to implement this kind of equipment especially in manufacturing control systems as a part of the supply-chain process. The open question is how to use such technology in an automation system as a ubiquitous item identification network to improve the flexibility of the production process.

Some works related to this field have been published. In (Engels, et al., 2001) the Electronic Product Code (EPC) network has been originally introduced being fully aware or cognizant of a Networked Physical World. The synchronization between physical world and computational world can be achieved thanks to the use of automatic identification and ubiquitous networking. The specification for the concept of an intelligent product has been depicted in (McFarlane, et al., 2003). Such a product should possess unique identification, is capable of communicating to its environment, can retain data about itself, deploys a language to display its features, and is capable of participating in making decision relevant to its destiny. In (Yagi, et al., 2005) the parts and packets unified architecture has been proposed. In this architecture, data or information related to a product are carried by the product itself and can be handled to manage the whole system. Apart from that, in manufacturing automation the trend is to use distributed technology. The Distributed Control Systems (DCSs) in this area currently adopt the IEC-61499 standard which was finalized and documented in January 2005 (IEC, 2005). This standard proposes the use of Function Blocks as a programming framework in Industrial Process, Measurement and Control Systems. It can be seen that, on one hand, there is a strong demand to make products more intelligent by extending the identification technology. On the other hand, DCSs using IEC-61499 Function Blocks seem to be a good paradigm to solve many problems in manufacturing systems. The approach presented in this paper shows a combination of both ideas. The scheduling based re-configurable control approach Functionality Based Control (FBC) using the Sched-
RFID is a technology that refers to the use of radio waves to identify objects. Compared to bar codes which were previously used as an identification technology, RFID offers advantages related not only to speed but also to greater accuracy, distance optimization, flexibility in identification where a product can be recognized although it is moving and not in the line of sight of the reader. Finally a considerably higher amount of data could be stored. Its system consists of two basic components: Tag or Transponder and Reader or Interrogator.

The RFID-tag usually uses a tiny integrated chip to store information about a product or an item in the form of a uniquely numbered code named Electronic Product Code (EPC). The tag has the responsibility to hold an EPC. It may allow the code to be changed in post-manufacture, may hold an immutable code that gives manufacture information and may have additional user data apart from it. The reader has the tasks to read the EPCs of RFID-tags (via a tag protocol), to report the EPCs to a host application (via a reader protocol), and to write the EPC to a tag as commanded if it is allowed in post-manufacture.

RFID actually describes a whole family of technologies based on the different types of tags used. The tags can be classified based on their operation, programmability, and frequency band used. Firstly, the operation of tags can be employed as active, passive, or semi-passive. Active tags contain a battery, while passive tags are driven by using energy from the electromagnetic field of the reader. Semi-passive tags include a battery as well, but the tag lies inactive until receiving a signal from the reader. Secondly, based on their programmability tags can be classified as read-only, write-once-read many, or read/write. The read-only tags cannot be reprogrammed but they are much cheaper to produce than read-write tags. Finally, based on the frequency band used RFID-tags can be classified as low medium or high frequency. The used frequency of cause influences the data transmission rate and the possible distance between reader and tag. For more details on the technological aspects of RFID see (Finkenzeller, 2003).

A question in RFID technology is how to maintain customer’s privacy since tags could be still read from the finished product. Furthermore, the problem of interference of several RFID tags and readers in close proximity, and the cost of this technology are still preventing wide-spread use. These negative phenomena actually can be solved in future implementations since this technology is still in its developmental phase. For example, the privacy could be solved by using tag-deactivation and technology-integrated privacy. The remaining fundamental question about RFID-technology is whether its quality augmentation of the control strategy for the future is better than the current technology. Already existing applications are very promising in this direction.

Applications of RFID appear in many fields. In the manufacturing area, as an example, the Ford Motor company has successfully implemented applications using RFID to improve product quality on the automated assembly production line at its facility in Cuautitlan, Mexico. Here, Ford produces cars and trucks using the just-in-time (JIT) manufacturing model. As a vehicle passes through the different stages of production, different parts of the 22- to 23-digit serial number are referenced, indicating what needs to be done at each station (Johnson, 2002). This is one of the biggest benefits of RFID where the former manual coding system required each identification sheet to be manually updated at every turn in the production line; RFID allows updates to be written to the tag, so it is constantly being updated without risk of operator error. For other reports on RFID implementations see (Brusey, 2003; Imura, 2005). To implement RFID in smart manufacturing, some aspects have been revealed in (Li, et al., 2004) relating to self identification, communication, quality, and concurrent process.

Concerning product-driven control, the work in (Gouyon, et al., 2004) adopting the Holonic manufacturing Systems (HMS) paradigm has been elucidated. In this approach the route of a product through a production system is determined based on a functional description of its processing.

In this paper, some capabilities of RFID are used; i.e., identification of product (name, characteristic, etc) and information about the production schedule and alternative tasks. This information (describing the technological aspects of a products processing) will be used to improve the scheduling of tasks on machine level. Actually the presented approach could be combined with an approach working on a higher level of automation as (Gouyon, et al., 2004). Where the higher level for example determines in which station a given operation like milling should be performed and the presented approach specifies how the operation is performed in detail on different available machines.
The presented work attempts to exploit the use of the FBC and S³ approaches to actually influence the processing in a single machine or workstation by the product.

FBC is an approach which addresses the use of common functionalities in the design process (Panjaitan and Frey, 2005). The detailed description of this approach using Unified Modeling Language (UML) which allows the reusability of both software implementation and modeling level has been explained in (Panjaitan and Frey, 2006). These functionality software components will be used and reused to build a control strategy. The reason behind using this approach is the existing problem in manufacturing control where similar basic functional controls are frequently used but implemented with different software. A functional block encapsulates all functions of a given mechatronic object in a manufacturing system as executable software. Based on FBC, a new scenario of manufacturing control could be implemented effectively by re-instantiation of commonly used function blocks. In addition to the FBC concept, to give re-configurability to the design process by means of an ability to repeatedly arrange and re-arrange the process design in a cost effective way the Scheduler, Selector and Synchronizer (S³) approach has been proposed in (Panjaitan, et al., 2005). This approach presents an architecture to effectively control the execution of FBC components. In case the sequence of processes has to be changed the reconfiguration process is done by just giving a data array describing the new sequence to a Scheduler module. These two approaches are especially useful in the DCS area. An actual implementation controlling a laboratory plant is running of several network enabled controllers, Netmaster (Elisit Srl.), connected by a standard ethernet network. One of the four workstations of this plant will be used as an example in Section 4.

For the implementation of RFID-technology an extension of the FBC-S³ architecture is conceptualized in Fig. 1. In this figure, arrows with solid lines show the event flow and the ones with dashed lines depict the data flow. In this architecture, each product or work-piece will bring its own schedule based on the functional identity in terms of FBC components. When a product comes to a work-station, the RFID-reader will read the product identity which refers to EPC rule and the required tasks from the attached RFID-tag. The tasks (changeable sequence array) related to the current station are obtained by filtering the code and taking the desired information embedded in the product. Later this information will be transferred from the Task Processor as data arrays to the Scheduler input with two options: First In First Out (FIFO) or MERGE. FIFO means the incoming product schedules should wait until the processing of the existing sequence is finished. On the other hand, MERGE means that if the incoming product along with its schedule arrives while the existing task schedule is still processed in a station, the schedules will be joined to run the process for these products concurrently for the next steps.

In the control strategy using FBC and S³ concept, the Scheduler which gets the data array from the Task Processor will send a particular value to the Selector representing one step of the desired task. The Selector will select which functionality units have to be applied. It can execute a single or multiple software components. The value is also sent to the Synchronizer which uses it as reference value to resume the successful operation of all the functional units. After a task is published and the functionalities are running the Synchronizer will evaluate its completion. When all functionalities of a task have been completed successfully, the Synchronizer provides a confirmation to the Scheduler and requests the next task where – as mentioned before – the task is pre-defined by receiving data from the RFID-reader and processed in the Task Processor before it is sent to the Scheduler.

![Image](Fig. 1. The extended FBC and S³ architecture using RFID technology.)
fore they are employed. Intelligent product here means that the product gives a task package (data array as text code) to the controller to process it. In other words, the product memorizes its own tasks. This concept is almost similar with the agent concept where the products are agents which have some control strategies represented as schedule in their body. However, using RFID the system could be implemented with much less demanding hardware (computation and network).

Furthermore, in contrary to traditional flexible manufacturing approaches the use of a central database that relates the ID of a part to the processes to be performed is no longer necessary. The approach can be made even more intelligent, if some optional schedules for the case of machine failures are stored on the RFID-tag.

In the other case where the read-write type of RFID is used, the schedule of tasks can be changed more dynamically since the schedule can be changed during the process using the RFID-reader. This makes the products more intelligent since the dynamic process in the context of interchangeability of the schedules is also considered in real-time situations and done by itself. A possible implementation is in case of failures (breakdown of the machine) or quality problems during processing. For example a workpiece with a surface not milled fine enough could be automatically re-programmed to be re-worked or even to be transformed into another product, where this quality concern is not so important.

4. EXAMPLE OF THE APPROACH

As an example, the automation system shown in Fig. 2 is considered.

Fig. 2. Drilling Station.

In this drilling station there are some FBC components with functionalities as given in Table 1. The four functions in the system representing six basic operations are built by only three different FBC components by instantiation and combination.

In this example, three kinds of products (listed in Table 2) can be specified as follows:

Product A - the process starts by F(1) conveying the product to position 1, then the rotary motor will run F(2)* to bring it to position 2. "*" means that, for safety reasons, this functionality cannot be activated while any another one is active. This functionality is "exclusive". The process continues to drill the product using the composition of three components (F(3)), i.e., Clamp, Drill, and Drill Cylinder, followed by another rotary operation F(2)* to position 3. Now the Tester F(4) will be employed and finally again F(2)* to reach position 4. Here, it is assumed that the product will be taken over by another station.

Product B - the process sequence in this case is the same as for product A without the test process.

Product C – this product is only moved through the stations without drilling and testing processes.

<table>
<thead>
<tr>
<th>Table 1. The Selected FBC Components.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>F(1) Conveyor</td>
</tr>
<tr>
<td>F(2) Rotary motor</td>
</tr>
<tr>
<td>F(3) Drill cylinder</td>
</tr>
<tr>
<td>F(4) Tester</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Specification of Embedded Tasks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prod Spec.</td>
</tr>
<tr>
<td>A Red Plastic</td>
</tr>
<tr>
<td>B Black Plastic</td>
</tr>
<tr>
<td>C Metal</td>
</tr>
</tbody>
</table>

To update the processing sequence of a machine like the drilling station in this example, two options can be used: FIFO and MERGE. The algorithms of FIFO and MERGE can be described as follows (see Fig. 3): both processes determine a new processing sequence $M$ based on the already existing sequence $K$ and the schedule of the newly arriving product $L$.

**FIFO**: The algorithm for FIFO simply appends the processing sequence $L$ given by the new product to
the already existing sequence \( K \) resulting in the new sequence \( M \).

**MERGE:** To merge an already existing sequence \( K \) with a new incoming one \( L \) (resulting the combined sequence \( M \)), at first the updated sequence for the products already under processing has to be derived. Here several cases have to be distinguished:

1. The sequence \( M \) stays of cause identical to \( K \) up to the time when \( L \) arrives.
2. On the other end of the schedule, as soon as all tasks of \( K \) or \( L \) are scheduled into \( M \), the remaining tasks of the not yet finished sequence are simply appended to \( M \).
3. For the remaining steps, the determination of \( M \) depends on whether the two current tasks in \( K \) and \( L \) can be merged.
   - Whenever it is possible, a new merged task is added to \( M \). Merging of tasks means that both original tasks are processed in parallel in \( M \).
   - Merging is not possible if one of the tasks is exclusive and between a task in one sequence and the same task in the other one (e.g. drilling of two products with one drill at the same time).
   - In case the two tasks can not be merged the already existing schedule gets priority.

While the FIFO operation is quite general, the merge operation in the generic structure given above may have to be adapted according to further restrictions in a workstation. In the presented example the rotation operation needs special attention since it applies to all products at the same time. Furthermore it is exclusive in the sense that it can not be merged with any other task. Therefore, the schedules are synchronized on this task.

To achieve parallel processing and to reduce the waiting time for transition from one kind of product to another, the Merge option in the Task Processor is used. For example consider Product A coming into the station first, followed by Product B when Product A is moving to table position 2, then Product C arrives when Product B is shifting to Station 2. The Task Processor will recalculate the task schedules with the Merge option. Therefore, the newly arriving product will change the current schedule array for the next tasks of the Scheduler.

**CASE 1 (arrival times as given above):**

\[
S_{AB} = [F(1), F(2)^*, F(3), F(4), F(2)^*, F(2)^*], \\
S_{ABC} = [F(1), F(2)^*, F(1)&F(3), F(2)^*, F(3)&F(4), F(2)^*, F(2)^*] \\
S_C : \quad \text{---------} \quad [F(1), F(2)^*, F(2)^*, F(2)^*] \\
\]

\[
S_{AB} = [F(1), F(2)^*, F(1)&F(3), F(2)^*, F(3)&F(4), F(2)^*, F(2)^*] \\
S_{ABC} = [F(1), F(2)^*, F(1)&F(3), F(2)^*, F(1)&F(3)& F(4), F(2)^*, F(2)^*, F(2)^*] \\
S_{AB} = [F(1), F(2)^*, F(3), F(4), F(2)^*, F(3), F(2)^*, F(2)^*, F(2)^*] \\
S_{ABC} = [F(1), F(2)^*, F(3), F(4), F(2)^*, F(3), F(2)^*, F(2)^*, F(2)^*, F(2)^*, F(2)^*, F(2)^*] \\
= [1, 2, 4, 9, 2, 4, 2, 2, 2] \\
S_{ABC} = [F(1), F(2)^*, F(3), F(4), F(2)^*, F(3), F(2)^*, F(2)^*, F(2)^*, F(2)^*, F(2)^*] \\
= [1, 2, 4, 9, 2, 5, 2, 2, 2, 2] \\
\]

The Gantt Charts of the task schedules of the three products and the resulting chart for CASE 1 are given in Fig. 4.

**Fig. 4. Gantt Charts for the example: CASE 1.**

For the case of another set of different arrival times see the schedules of CASE 2 below:

**CASE 2 (different arrival times):**

\[
S_{AB} : \quad \text{The updated schedule when product B arrives while product A is doing the third task (F(3)).} \\
S_{ABC} : \quad \text{The updated schedule when product C arrives and product A is in the fourth task and B in the second task (F(2)*).} \\
S_A : \quad [F(1), F(2)^*, F(3), F(2)^*, F(4), F(2)^*], \\
S_B : \quad \text{---------} \quad [F(1), F(2)^*, F(3), F(2)^*, F(2)^*] \\
S_C : \quad \text{---------} \quad [F(1), F(2)^*, F(2)^*, F(2)^*] \\
\]

**With the FIFO option for the Task Processor the schedule for a product in the process is finished be-**
fore the schedule of the incoming product is considered. It is clear that the process (shown in CASE 3 below) is slower than with the Merge operation. However, sometimes this option is necessary when a work station allows processing of only one product.

**CASE 3 (arrival times as in CASE 1):**

\[ S_A : \{F(1), F(2)*, F(3), F(2)*, F(4), F(2)*\} \]
\[ S_B : \{F(1), F(2)*, F(3), F(2)*, F(2)*\} \]
\[ S_C : \{F(1), F(2)*, F(2)*, F(2)*\} \]
\[ S_{AB} = \{F(1), F(2)*, F(3), F(2)*, F(4), F(2)*, F(1), F(2)*, F(3), F(2)*, F(2)*\} = \{1, 2, 4, 2, 8, 2, 1, 2, 2, 2\} \]
\[ S_{ABC} = \{F(1), F(2)*, F(3), F(2)*, F(4), F(2)*, F(1), F(2)*, F(3), F(2)*, F(2)*, F(1), F(2)*, F(2)*\} = \{1, 2, 4, 2, 8, 2, 1, 2, 2, 2\} \]

5. CONCLUSION AND OUTLOOK

In this paper an approach to control manufacturing systems equipped with RFID technology using Functionality Based Control (FBC) and Scheduler, Selector and Synchronizer (S³) has been presented. The proposed approach gives capability to the products to decide their own control strategy by means of task schedules in a system process which paves a way to develop intelligent products. The interface between the RFID-reader reading the processing information from the product and the Scheduler as input-port to the S³ architecture is provided by a Task Processor. The Task Processor will update the task array given to the Scheduler which affects the next operations. An example of a drill station with some cases of arrivals and kinds of products has been given. Using this example the computation process to update the schedule was explained.

It has to be mentioned that the concept was tested on a plant without actual RFID hardware. This was simulated in the FB environment by sending corresponding tasks through an additional block. The simulation was ideal in the sense that it assumed correct functioning of the RFID sub-system. Currently the RFID hardware setup at the presented laboratory plant is under development. Given the hardware setup another component will be developed to check the schedules coming from the RFID reader before giving them to the Scheduler. This check must not only verify the correctness of the data read but also provide a test for schedulability of the newly arriving product into the machine at the given time.

REFERENCES


Elsist S.r.l, Netmaster Official Website (Italy) – http://www.elsist.net


Holobloc, Inc, Website devoted to IEC-61499, FBDDK tool: http://www.holobloc.com


