ISSN 2229-5518

RFID Technology to control manufacturing systems using OPC server

Wassim Mansour, Khaled Jelassi

Abstract— starting from the "Industry 4.0" concept, this work aims to develop an event-condition-action (ECA) based structure to control intelligent manufacturing systems (IMS). RFID technology is used as a main component to gather data from shop floor. These data are used to control product rooting among workstations and to control systems agents (robot's arms for example). The whole system has OPC based architecture. A generic GUI using C-sharp compiler is developed allowing users to edit manufacturing sequences that could be automatically loaded from ERP software.

Index Terms— RFID, OPC server, intelligent manufacturing systems, industry 4.0, event-condition-action structure.

1 Introduction

Manufacturing industry is in the doorsill of the fourth industrial revolution. It is a revolution where automation, flexibility, intelligent autonomous manufacturing systems take more place as well as the integration of customers and business partners in value processes. Such a near-futuristic concept was called 'Industry 4.0". The term was first used in 2011 [7].

In this context, automation plays a major role in the way it prioritizes the introduction of self-optimization and self-configuration methods. These methods require adequate hardware support and convenient software architecture.

This work deals with the conception and implementation of software architecture to control an intelligent manufacturing system where shop-floor low-level decisions are made by the system itself.

To achieve this goal, data collection must be done accurately. RFID technology is used in this work to give the system regular feedback about the status of products (operations achieved, remaining tasks, next workstation …). Then an OPC server is used to deal with client application. In this work, we present an OPC client platform allowing users to control an IMS.

2 Description of the Proposed Architecture

During the last years, several authors have been working on approaches for control architectures of manufacturing systems. Control architectures have evolved from traditional hierarchic architectures [8] to agent-based architectures [9] to holonic manufacturing systems (HMS) where intelligent "holons" (an autonomous and cooperative block of a manufacturing system) interact and negotiate to achieve manufacturing goals [10].

In this work we propose a system to control IMS based on ECA. Major elements on this architecture are RFID technology and OPC server. Fig. 1 shows an outline of the conceived system.

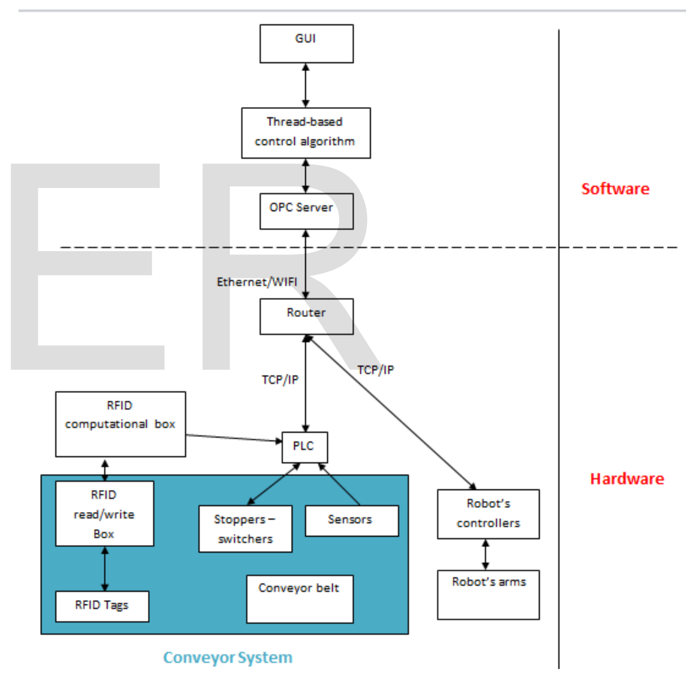

Fig. 1. Outline of the conceived system

The system consists of both software and hardware components. Hardware components are common elements among FMSs. They can be divided into two subcategories: processing elements and handling elements. Processing elements are usually machines used to make an added-value transformation on the manufactured product such as robot's arms or CNC machines. Handling elements are used to move products from a workstation to another. Software elements play a major role in the control of such systems. They are based on read/write logic where usually user's commands are sent to the system while system sends feedback about its status.

In this work, architecture to control FMS is developed (Fig. 1). This architecture is based on an OPC server that communicates data between systems' devices and a thread-based control algorithm. RFID technology plays a major role in this sys-

- *Author Wassim Mansour is currently pursuing PHD program in electrical Engineering in El ManarUniversity, Tunisia. E-mail: w.mansour@outlook.fr*
- *Co-Author Khaled Jelassi is currently a professor in electric engineering department in El Manar University, Tunisia. E-mail: khaled.jelassi@enit.rnu.tn*

ISSN 2229-5518

tem as long it permits identification of products which gives the system the ability to distinguish different products.

2.1 RFID Technology

RFID technology has been widely used for product identification and tracking purposes [2, 3, and 4]. Product identification and tracking is important to achieve accurate control over FMSs especially when it comes to high-variety production schedule [11].

RFID technology consists of a tag attached to the product (or its container) that contain an EPC (Electronic Product Code) with size of 128 bytes generally [12]. RFID reader is able to read and write data when the tag is in the active area of the reader. Written and read data to/from the tag is generally used for product tracking and identification. In fact, read data gives information about the localization of the product in the manufacturing chain. Other data can be carried on the RFID tag such as scheduling data, status data …

In the considered system, RFID tags of 128 bytes are used. RFID Tags used are shown in fig.2. These Tags communicate with RFID computational Box via RFID readers/writers (fig. 3) (each Box is connected to four RFID readers/writers).

Fig. 2. RFID Tag

2.2 OPC server and client

OPC is a family of different standards used to ensure the interoperability of different shop floor devices as well as to provide a standardized way for applications to communicate with such devices including sensors, PLCs…

The OPC standard has been widely used for FMS control purposes [6, 16, and 17]. OPC has superseded other communication technics since its emergence.

Although, the OPC standards are not very common so far for communication with RFID readers [5], some authors have developed FMSs' control systems where an OPC server communicate with RFID readers [2, 3].

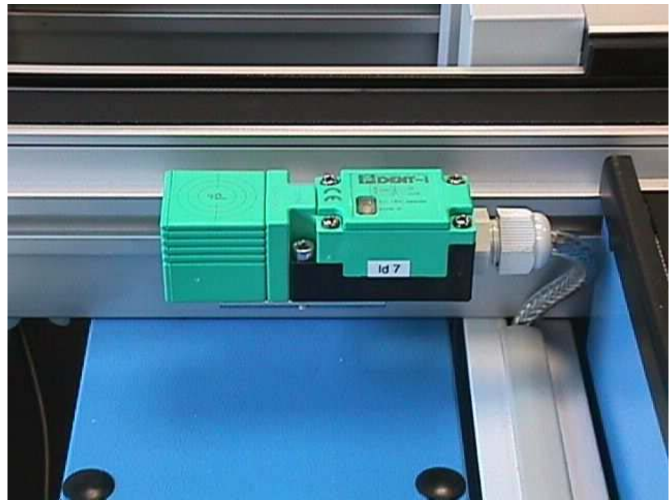

Fig. 3. RFID reader/writer

Developed GUI includes OPC client to ensure data transfer. OPC server consists in a third-party application that communicates with the PLC. In order to achieve communication with OPC server OPCDA and OPCHDA specifications have been used. Data received from OPC server bring information about system's status and changes occurring. These data determine actions to be triggered in a thread-based algorithm.

3 ECA Algorithm to Control Product Rooting and System's Agents

An ECA system consists of a set of rules. The rules are triggered by events (here an event represents a change in OPC data values),check whether certain conditions are satisfied, and, based on these conditions, perform certain actions [13].The ECA paradigm is widely used for different purposes such as manufacturing systems [1, 13, 14 and 15].

In this work, a thread-based algorithm is used to control any change in OPC data values. This algorithm can be considered as a kernel for the client application. It controls any change of OPC data using the *Ondatachange* Method. This method performs periodic check of the OPC data values. Every time data change, this method returns a table listing items of changed data (fig. 4). Then the ECA algorithm performs specific actions for each change. For example, the change of the value of a detector's state from 0 to 1 means the pallet is already in that detector's active area. A relevant action could be, for example, the calling of a robot program to perform adequate manipulations.

ISSN 2229-5518

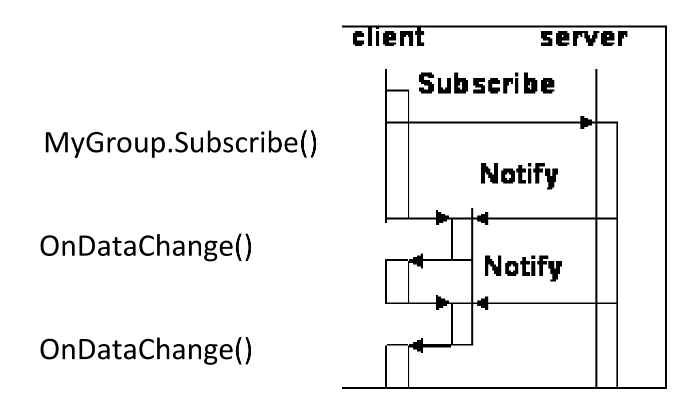

Fig. 4. OnDataChange method – reading synchronization

Depending on a given system's state and on the data change status, convenient actions are performed by the systems. In this work, events represent status change of detectors which means either "the pallet is in the location N°X" or "the pallet has left the location N°X". Actions are robot's programs start (generally when the pallet is in the workstation in the active area of the robot's arm), switchers' actions (to enable/disable the entrance of a pallet into a given workstation) or stoppers actions (to stop/let go a pallet). Conditions can be the occupation of a workstation or parameters related to the scheduling of products among workstations. Relationship between Events, Conditions and Actions are schematized in fig. 5.

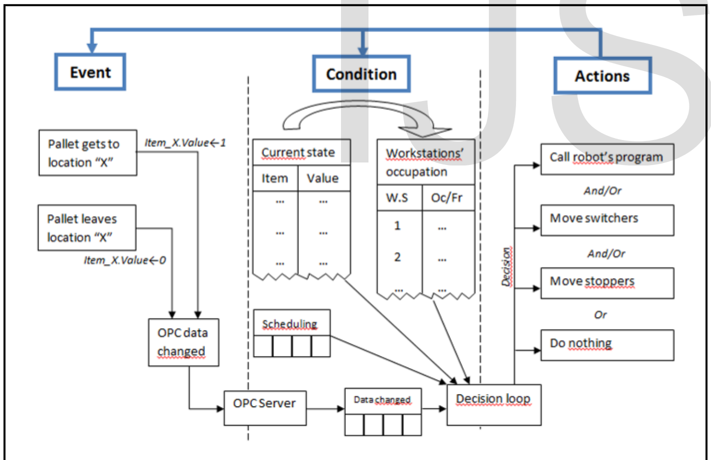

Fig. 5. Relationship between Events, Conditions and Actions

A flowchart of the whole system is given in Fig. 6. As mentioned above, the core of the system consists of a thread-based algorithm checking continuously any event on the system. As several variables might change in a small time lapse, the thread-based algorithm returns each time a table containing items which values changed since last check. Then a loop is run to perform adequate actions for each OPC Item that has a value change. Since actions depend closely on the considered OPC item, a "case of … Break" loop is used with "If … then" sub-loops to check conditions before performing any action. Robot's actions are called from RFID tags.

Since RFID tags used have 128 bytes and as long as we have less than 257 pallets, only one byte is sufficient for pallets identification. In our case, the rest 127 bytes are used to store the scheduling among workstations and robot's programs paths (either in the user's computer or in a remote server). Given that for ***n*** workstations the number of possible schedules is (1), one byte is sufficient for the storage of the schedule on the RFID tag if ***n*<5**.

$$\left(\sum_{i=0}^{n} A_n^i\right) + A_n^n \qquad (1)$$

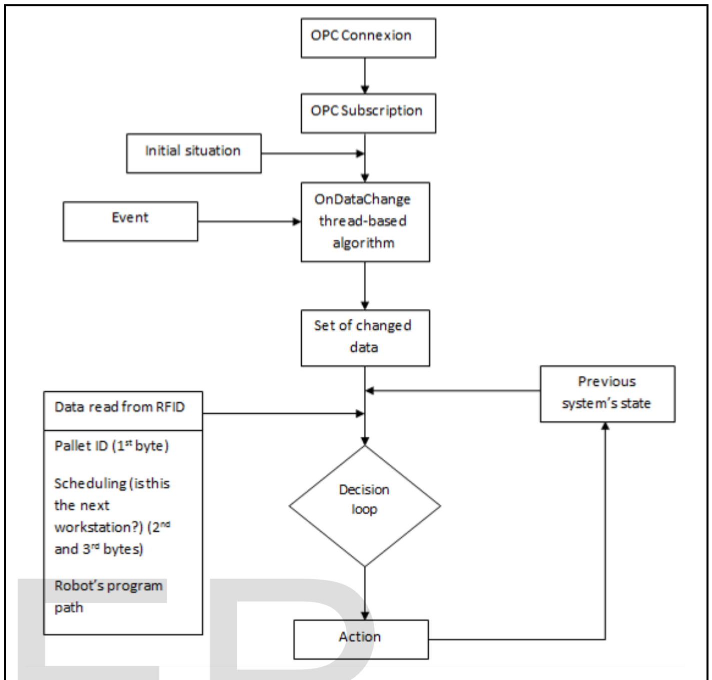

Fig. 6. Flowchart of the developed system

The 2nd byte in the RFID tag includes information about workstations and their order to perform required manufacturing operations. This information is treated in the ECA algorithm to return a binary value: The pallet enters or not to the considered workstation. The relationship between the schedule and the data stored in the 2nd byte of the RFID tag is quite simple: the byte stored is converted to the ASCII code of the corresponding character. The value obtained refers to a given schedule. Besides, additional data can be stored indicating whether the scheduling sequence is strict or flexible. In the 3rd byte, is stored the number of manufacturing operations achieved (0 for a new pallet in the shop floor). An example of a four workstations' manufacturing cell is given in table 1.

In the example shown in table 1, system could easily decide whether a given pallet is entering the concerned workstation or not. Tests on the real system have demonstrated a great accuracy in routing pallets. Extracting schedule (and corresponding decisions) from the RFID tag's 2nd byte was done in a very short time given that related algorithm included optimized and grouped conditions.

In this case, 192 remaining combinations haven't been used since one byte gives 256 possible combinations. Hence, flexible schedules could be introduced. In the case study shown in "IV" paragraph, we have used these remaining combinations as follows:

- ASCII codes from 64 to 127: same schedules shown in table 1 with no strict order among operations.

International Journal of Scientific & Engineering Research, Volume 6, Issue 9, September-2015 392
ISSN 2229-5518

- ASCII codes from 128 to 191: same schedules shown in table 1 with no strict order among operations. Only the first operation has to be the first.
- ASCII codes from 192 to 255: same schedules shown in table 1 with no strict order among operations. Only the last operation has to be the last.

Table 2 shows examples of the considered encoding of schedules.

TABLE 1
POSSIBLE SCHEDULES AND THEIR CORRESPONDING ASCII CODE (4 WORKSTATIONS' CASE)

Schedule	ASCII Code	Schedule	ASCII Code	Schedule	ASCII Code	Schedule	ASCII Code
1	0	2	16	3	32	4	48
1→2	1	2→1	17	3→1	33	4→1	49
1→3	2	2→3	18	3→2	34	4→2	50
1→4	3	2→4	19	3→4	35	4→3	51
1→2→3	4	2→1→3	20	3→1→2	36	4→1→2	52
1→2→4	5	2→1→4	21	3→1→4	37	4→1→3	53
1→3→2	6	2→3→1	22	3→2→1	38	4→2→1	54
1→3→4	7	2→3→4	23	3→2→4	39	4→2→3	55
1→4→2	8	2→4→1	24	3→4→1	40	4→3→1	56
1→4→3	9	2→4→3	25	3→4→2	41	4→3→2	57
1→2→3→4	10	2→1→3→4	26	3→1→2→4	42	4→1→2→3	58
1→2→4→3	11	2→1→4→3	27	3→1→4→2	43	4→1→3→2	59
1→3→2→4	12	2→3→1→4	28	3→2→1→4	44	4→2→1→3	60
1→3→4→2	13	2→3→4→1	29	3→2→4→1	45	4→2→3→1	61
1→4→2→3	14	2→4→1→3	30	3→4→1→2	46	4→3→1→2	62
1→4→3→2	15	2→4→3→1	31	3→4→2→1	47	4→3→2→1	63

In order to optimize RFID's tag storage, the 126 remaining bytes are considered as 1008 ($= 126 \times 8$) consecutive bits. Among which, the 4 first bits are used to indicate which step has the pallet reached in a given moment (every bit "0" means the given operation is still undone '1' means mean it's done). Once all operations are done, these four bits get "1111" as a value to indicate that the pallet has to get off the system. The remaining 1004 (= 1008 – 4) bits are used to store the program path and file name of each robot (251 bits for each robot program).

Every program path name is composed of 251 bits indicating the following:

- 5 first bits: a letter (lower cases are converted to upper cases) indicating the drive (local or remote). The letter indicating the drive is obtained as follows: these five bits are converted to their decimal value (Dx). The letter corresponding to the ASCII code (Dx+65) is the drive.
- Remaining 246 bits indicate filename with a maximum length of 41 characters (drive name, ":\" and file extension ".ciip" are excluded) where 6 bits are allowed for each character (considered characters have ASCII codes from 32 to 95, lower cases are converted to upper cases, characters having ASCII code out of this interval are not allowed, character (") is automatically added to indicate the end of the filename string: e.g. the binary sequence "000010").

Fig.7 explains how data on the RFID tag is decoded. Fig. 8 shows the structure of data in the RFID's tag.

TABLE 2
RANDOM EXAMPLES OF SCHEDULES (4 WORKSTATIONS' STUDY CASE)

2[nd] byte content (character)	ASCII code	Schedule(s)
!	33	3→1
E	69 (1→2→3 with no strict order among operations)	1→2→3 1→3→2 2→1→3 2→3→1 3→1→2 3→2→1
ç	135 (1→3→4 with "1" has to be the first and to strict order for the rest)	1→3→4 1→4→3
Σ	228 (3→1→2 with "2" has to be the last and to strict order for the rest)	3→1→2 1→3→2

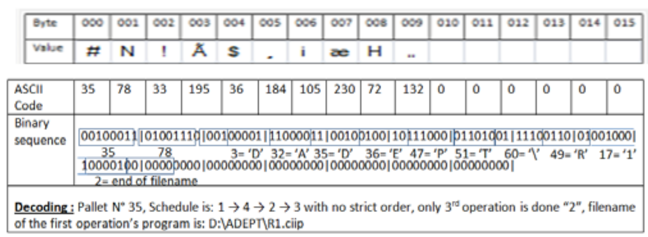

Fig. 7. Sample decoding of data on RFID tag

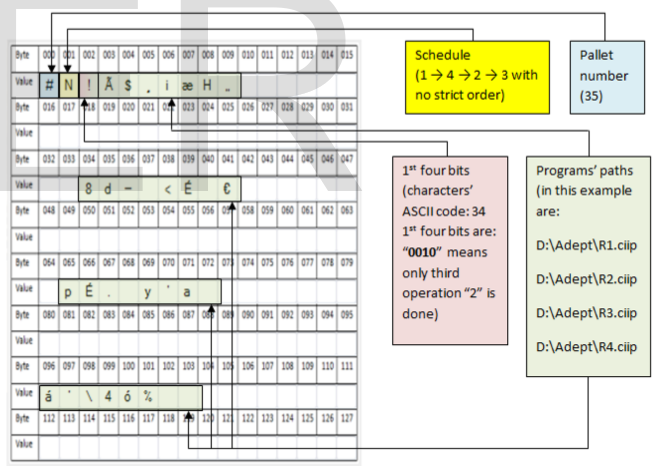

Fig. 8. RFID tag Data structure

4 CASE STUDY SYSTEM

The developed system is brought to application using an experimental real-size flexible manufacturing system (CIIP: Centre Industriel Intégré de Production), located in Tunis' National School of Engineers (ENIT). Fig. 9 shows the layout of the considered manufacturing system. Fig. 10 shows its features.

The considered system is composed of:

- Four robot arms running a V+ OS :
  - Adept Cobra 600 robot
  - Adept One robot
  - Two Stäubli RX60 robots

Each robot is located in a separate workstation what makes the

ISSN 2229-5518

system having four workstations.

- A conveyor system equipped with :
    - 64 position detectors (16 in each workstation)
    - 8 switchers (2 in each workstation): switchers allow pallets to go in a given direction (enter the workstation or just pass beside).
    - 24 stoppers (6 in each station): stoppers are used to block a pallet in a given location.

All these elements are controlled by a Siemens S7-300 PLC.

- 8 RFID read/write Box
- Several pallets equipped with RFID tags.

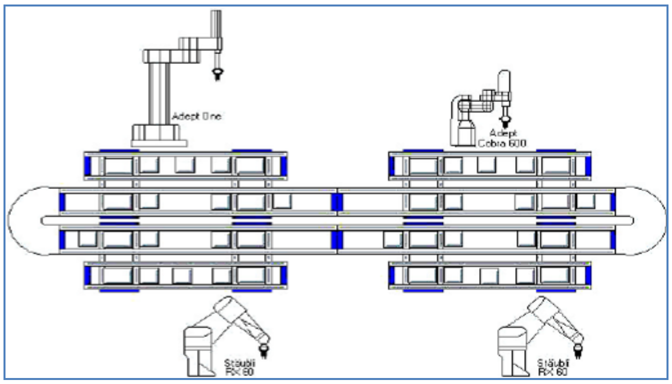

Fig. 9. Layout of the experimental manufacturing system

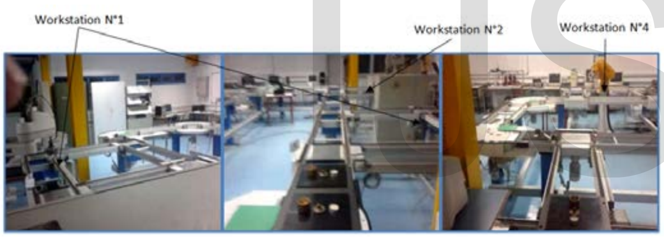

Fig. 10. Photos of the experimental manufacturing system

5 Software solution for IMS

Subsequently, software is developed to allow final users full control of the considered manufacturing system. The developed software is called "FMS-C" and is highly flexible since simple setting allows its use for other flexible manufacturing systems. Also, user is free to choose his proper encoding of the RFID tags and then software considers user's configuration.
Besides, "FMS-C" allows users to:

- Teach robot and save positions in order to make manipulations. Sequences are saved in a special encrypted format (*.ciip) to allow more safety in information exchange (fig. 11).
- Make a production schedule and save it to RFID tags on the pallet. Also save robot's programs paths into the RFID tags so they can be called later during the functioning of the system (fig. 14).
- Direct access (read/write) to raw data in the RFID Tag (fig. 12).
- Have a full control over the manufacturing system by visualizing the location of every pallet and the status of every element of the system (robot's arms, detectors, switchers …) (fig. 13).
- Have a reporting after every production order/set including the value of the makespan (the total completion time of all tasks of a given set), processing time of every pallet, waiting time, …

Fig. 11. Teach robot interface

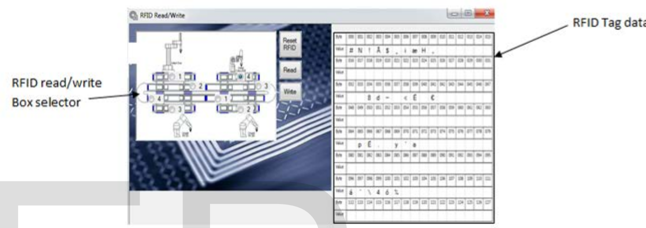

Fig. 12. RFID read/write interface

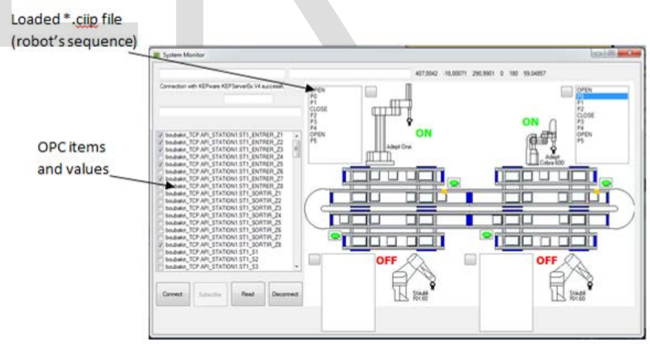

Fig. 13. FMS-C Main control interface

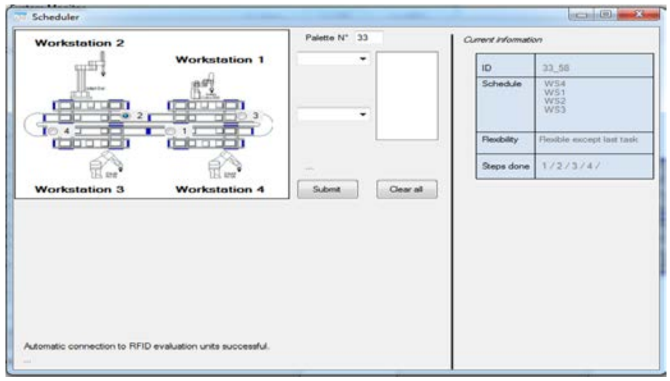

Fig. 14. Schedule and programs' selection interface – storing into RFID's tag

6 Conclusion and Further research

This work had as a main objective the development of an architecture and software solution to allow optimized and full control of a flexible manufacturing system. This aim is reached using RFID technology and OPC server. RFID technology allowed identification of products and then a higher flexibility also it allows the deportation of some data to RFID tags so we have a less centralized system. Data stored in the RFID tags were encoded using a special protocol (explained in paragraph III) which gave us a more secure system. The use of OPC protocol has allowed the integration of different hardware components in the system.

Further research might be conducted towards the use of cameras to allow auto-quality control and avoid manipulation errors to occur. Other research axis could be the integration of an intelligent scheduling where robot's delegate tasks each one to other in case of unbalanced load between workstations.

References

[1] Lindsay V. Allen& al. : Factory-Level Control Consolidation Using Event-Condition-Action: Case Study on the Reconfigurable Factory Testbed. 2008 American Control Conference, Seattle, Washington, USAJune 11-13, 2008.P : 1715-1721.

[2] [2] :A. M. Gaitan& al. : Products Authentication and Traceability using RFID Technology and OPC UA Servers. ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 18, NO. 10, 2012.P : 73-76.

[3] [3] :A. M. Gaitan, C. Turcu, A. Goloca& R. Popa : An RFID and OPC Technology Based Distributed System for Production Control and Monitoring. RFID Eurasia, 2007 1st Annual. P : 1-6.

[4] [4] :Y. Xiao-Wei&D. Xue-Ren : Dynamic monitoring system based on RFID technology for workshop production. International Conference on Management Science and Industrial Engineering (MSIE), 2011.P : 1331-1334.

[5] [5] : L. Ivantysynova& H. Ziekow : RFID in Manufacturing : From Shop Floor to Top Floor. O. Günther, W. Kletti, U. Kubach, RFID in Manufacturing DOI: 10.1007/978-3-540-76454-0, © Springer 2008. P : 1-24

[6] [6] : Y. Yao & al. : The Remote Monitoring System Based on the OPC Technology. International Workshop on Intelligent Systems and Applications, 2009.ISA 2009.P : 1-3.

[7] [7] : H. Kagermann, W.D. Lukas, W. Wahlster : Industrie 4.0: Mitdem Internet der Dinge auf dem Wegzur 4. Industriellen Revolution. Contents of edition No. 13/2011 (in german)

[8] [8] :Dilts DM, Boyd NP & Whorms HH (1991) The evolution of control architectures for automated manufacturing systems. Journal of Manufacturing Systems 10(1): 79–93.

[9] [9] :van Dyke Parunak H, Baker AD & Clark SJ (2001) The AARIA agent architecture: From manufacturing requirements to agent-based system design. Integrated Computer-Aided Engineering 8(1): 45-458.

[10] [10] :Christensen JH (1994) Holonic Manufacturing Systems: Initial Architecture and Standards Directions. Proc of First European Conference on Holonic Manufacturing Systems, Hannover.

[11] [11] : R.V. Barenji, A.V. Barenji, M. Hashemipour : A multi-agent RFID-enabled distributed control system for a flexible manufacturing shop. Int. J. Adv. Manuf. Technol. (2014) 71 : 1773-1791.

[12] [12] : Y. Z. Zhao, O. P. Gan, "Distributed Design of RFID Network for Large-Scale RFID Deployment", in Proc. of the IEEE International Conference on Industrial Informatics 2006, IEEE, 2006, pp. 44–49.

[13] [13] : H. Wijaya, K. Sukerkar, S. Gala, N. Arora, J. Moyne, D. Tilbury& J. Luntz. Reconfigurable Factory-wide Resource-based System Integration for Control. Proceedings of the IEEE International Conference on Electro/Information Technology, May 2006.

[14] [14] E. Almeida, J. Luntz& D. Tilbury. Modular Finite State Machines Implemented As Event-Condition-Action Systems. Proceedings of the16th IFAC World Congress, July 2005.

[15] [15] E. E. Almeida, J. E. Luntz, & D. M.Tilbury. Event-condition action systems for reconfigurable logic control. IEEE Transactions on Automation Science and Engineering, 4: 167-181, 2007.

[16] [16] : A. Girbea, C. Suciu, F. Sisak : Remote monitoring and control of a flexible manufacturing system through a service oriented architecture Subtitle as needed. Roedunet International Conference (RoEduNet), 2011 10th.Pp : 1 – 6.

[17] [17] :A. Gîrbea S. Nechifor F. Sisak L. Perniu : Design and implementation of an OLE for process control unified architecture aggregating server for a group of flexible manufacturing systems. IET Softw., 2011, Vol. 5, Iss. 4, pp. 406–414.