INTELLIGENT JOB SHOP ASSEMBLING WITH HOLONIC ROBOT CONTROL

Theodor Borangiu, Florin Daniel Anton, Nick Ivanescu, Silvia Tunaru, Anamaria Dogar

University Politehnica of Bucharest, Dept. of Automation and Applied Informatics

Abstract: The paper describes a networked fault-tolerant assembly system of job shop type containing multiple robotized stations operating under visual control. During the assembly process, the products are identified, located, qualified, and controlled by analyzing images from virtual cameras associated to stationary- and arm-mounted cameras. The holonic control architecture is formed by the following types of entities: a Global Assembly Scheduler generating Order Holons, a dual layer of Material Handling (Robot) Holons and Sensory Holons, and a System Monitoring & Database entity. These elements are embedded in a 3-layer computing & control structure, and interconnected by a 4-fold fault-tolerant communication network keeping track of assembly job execution. Implementing issues and experimental results are reported.

Keywords: assembly robots, intelligent knowledge-based systems, fault-tolerant systems, scheduling algorithms, robot control, robot vision.

1. INTRODUCTION

A *networked robotized assembly structure* is composed by a number or robotic resources, linked between them by a closed-loop transportation system (conveyor). The final product results by executing a number of mounting, joining and fixing operations by one or several of the networked robots. The set of specific assembling operations is extended to on-line part conditioning (locating, tracking, qualifying) and checking of relative positioning of components and geometry features. These functional extensions may be supported by artificial vision either integrated with motion control (Guiding Vision for Robots - GVR) or as stand alone, computer vision (Automated Visual Inspection – AVI). In both cases, vision is used on line to check for proper geometry features and presentation of assembly components in view of robotized mounting, and for inspecting the assembly in its intermediate and final execution stage – positioning, component alignment, completeness (Borangiu, 2004).

Traditional networked assembly structures have either a *hybrid* or *heterarchical* architecture. The first one, derived from the hierarchical architecture, allows cooperation and sharing of information between lower-level (robot) controllers; a supervisor initiates all the activities and then the subordinates cooperate to perform them. The second is formed by a group of independent entities called *agents* that bid for orders based on their status and future workload. There is no master-slave relationship; all the agents including the manager of a particular order are bidding for it. Due to the decentralized architecture, the agents have full local autonomy and the system can react promptly to any change made to the system.

However, because the behaviour of an order depends on the number and characteristics of other orders, it is impossible to seek global batch optimization and the system's performance is unpredictable. In order to face resource break-downs, networked assembly structures should use robot controllers with multiple-network communication facilities allowing for fault-tolerance: targeted data saving and task redistribution.

2. THE HOLONIC KNOWLEDGE-BASED NETWORKED ASSEMBLY STRUCTURE

To compensate for the deficiencies of both hierarchical and heterarchical manufacturing control systems, the concept of *Holonic Manufacturing Systems* was chosen for designing and implementing a networked robotized assembly system, featuring intelligence, product tracking with integrated quality control, and fault-tolerance.

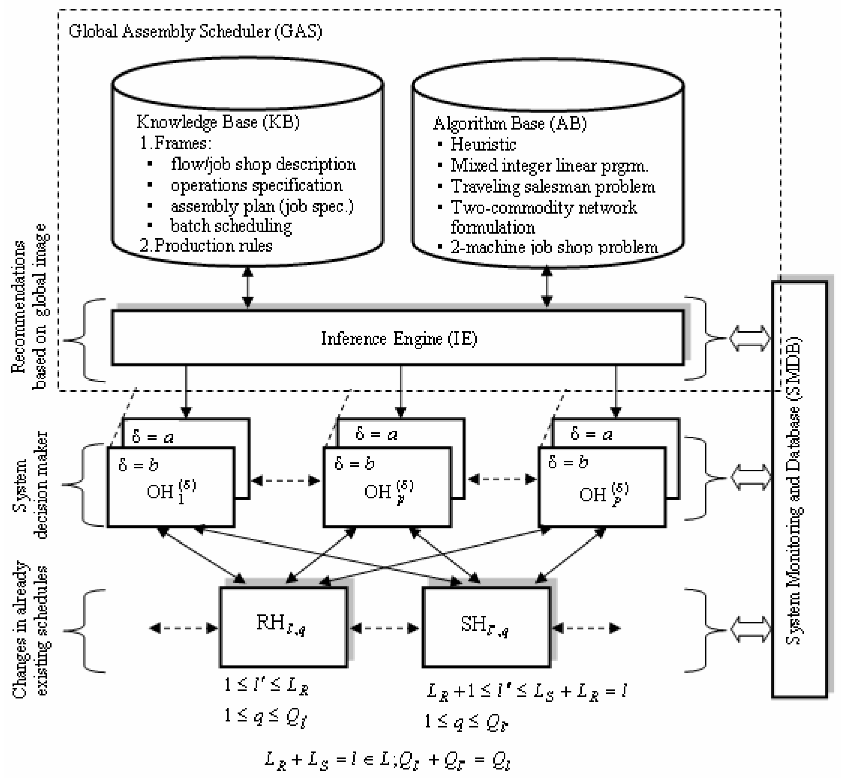
Global Assembly Scheduler (GAS)
Knowledge Base (KB)
1.Frames:
flow/job shop description
operations specification
assembly plan (job spec.)
batch scheduling
2.Production rules
Algorithm Base (AB)
Heuristic
Mixed integer linear prgrm.
Traveling salesman problem
Two-commodity network formulation
2-machine job shop problem
Inference Engine (IE)
Recommendations based on global image
System decision maker
Changes in already existing schedules
System Monitoring and Database (SMDB)
δ = a
δ = b
OH
RH
SH
$1 \le l' \le L_R$
$1 \le q \le Q_l$
$L_R + 1 \le l'' \le L_S + L_R = l$
$1 \le q \le Q_{l''}$
$L_R + L_S = l \in L; Q_{l'} + Q_{l''} = Q_l$

Fig. 1. Knowledge-based holonic architecture for job shop robotized assembly.

The model was taken from automated manufacturing cells containing one or more control units, processing machines, material handling devices (robots) and other pieces of equipment that work together to perform the required operations on the incoming work pieces. By grouping the hardware and software parts of the system based on their functions and applying both the holonic concept and the decentralized control approach, the architecture in Fig. 1 is considered to control the multiple-robot networked assembly system.

The flow of information between control units is always bi-directional due to the decentralized nature of the architecture and the application of the holonic concept of cooperation. This assembly structure is derived from the generic manufacturing one, formed by four types of entities (Babiceanu, *et al.*, 2004):

1. A layer of *Order Holons* ($\mathrm{OH}_p^{(\delta)}, 1 \le p \le P$) of variable depth, corresponding to assembly plans computed off line for the P final products.
2. Two types of Resource Holons:
   - *Robot* (or Material Assembly) *Holons* ($\mathrm{RH}_{l',q}$), formed by all robot manipulators, grippers and tools together with their controllers, responsible for mounting, fixing and fastening assembling components, and for moving their arm-mounted cameras in picture-taking points where the products are visually inspected.
   - *Sensory* (Material Tracking & Checking) *Holons* ($\mathrm{SH}_{l'',q}$), formed by all machine vision systems and magnetic code RD/WR devices respectively used for component / subassembly position and geometry control and product tracking on pallets.
3. A *System Monitoring and Database* (SMDB) entity, responsible for monitoring the number of jobs and availability of resources in the system, and for keeping track of the already executed operations and assembly jobs.
4. A *Global Assembly Scheduler* (GAS), generating basic and alternate assembly plans for all batch products in the form of OH. An algorithm base is embedded into a knowledge-based system (KBS). An inference engine in the KBS controls, according to a forward chaining control strategy, the procedures of: triggering production rules, switching to algorithmic schedule generation, and holonic fault-tolerant task allocation.

In the holonic system, an OH represents an assembly job and all its associated information embedded in a Cell Server (CSv) and replicated in the IBM PC-type Station Computers (SCo) while the RH and SH are represented by the physical assembly and control resources having their own Station Controller (SCn – multitasking Robot-Vision controllers).

In the proposed architecture, the GAS is basically a control unit that delivers (optimal) schedules for the material assembly and conditioning equipment when the system is operating under normal conditions. The main GAS computing occurs off-line: a set of jobs (assemblies) is to be executed with on line control by a set of robot-vision resources (processors), eventually minimizing (maximizing) an imposed performance function. An assembly job (product) may consist of a number of operations. All assembly parameters are assumed to be known a priori; each operation is to be executed by at most one robot or camera at a time. Initially, all robots are operational.

Due to the fact that assemblies are placed on pallets travelling between robot stations on a closed-loop conveyor belt, the assembly problem is of “job shop” type (French, 1982): assemblies may flow in different directions; also, a robot can be visited more than once by the same product. In general, a product does not have to visit all robots or be inspected by all cameras.

Based on its global image of the assembly system and the number of jobs in progress at any given time, the GAS computes the schedules in the form of assembly plans (OHs), but it does not impose its schedule to any of the individual robot-vision RH,SH resources. The schedules delivered by the GAS are treated as *recommendations* by the decision-making entities.
The OHs, created whenever a new job enters the system, are the entities with the authority to award assembly tasks to the RHs and inspecting tasks to the SHs. By comparing the processing offers received from the RHs and SHs, and the schedule received from the GAS, the OHs assign operations to the individual resources. From this point of view, OHs can be viewed as the system decision-making units.

The tasks of the resource holons are to execute all the material conditioning operations (manipulating, inserting, fastening, locating, qualifying, inspecting) in the networked assembly structure and to prepare appropriate offers for every new assembly job issued by the OHs. Having their own control units, the RHs and SHs can decide their course of action based on their status and future workload by assigning different weights to the tasks received from the OHs. Based on *autonomy* and *cooperation* – characteristics specific for the holonic system, the RHs can make changes in already existing schedules when one of them faces unexpected heavy workload or is down.
The SMDB entity has no decision power; rather it works as:

- a database for parameters of the scheduling models;
- a database for already performed jobs and as a monitor for the availability of the manipulating and inspection devices in the system;
- creating the OHs whenever a new order enters the system, deleting the OH when all the tasks associated with this holon are finished, and updating the database with the information related to the finished jobs for further reference.

3. THE GLOBAL ASSEMBLY SCHEDULER

A basic (quasi optimal) process plan and one or more alternative (sub optimal) plans are generated as Order Holons (OH) with each final product (assembly). The mounting and control resources specified by an alternative OH are at least partially different from the resources of the basic plan.
The heuristic Global Assembly Scheduler uses the notations and definitions below:

$O =$ set of all operations (assembly, conditioning)

$P =$ set of all assemblies (final products)

$OA_p =$ set of operations for assembly $A_p,\ p \in P$

$L =$ set of all resource types

$Q_l =$ set of resources of type $l, l \in L$

$f_i =$ completion time of operation $o_i, o_i \in O$

$rt_i =$ remaining processing time of operation $o_i, o_i \in O$

$ns_{ip} =$ number of successive operations of operation o_i in product $A_p, o_i \in O, p \in P$

$ni_{ip} =$ number of immediate successive operations (directly linked by precedence constraints) following operation o_i in $A_p, o_i \in O, p \in P$

$nu_{ip} =$ number of unprocessed operations in product A_p corresponding to operation $o_i \in O, p \in P$

$t =$ current scheduling time

$r_{lq} =$ resource q of type l, $q \in Q_l, l \in L$

For the networked assembly problem, the following types of resources are defined:

- $r_{1q} =$ assembly robot manipulator, $q = 1{,}2$: SCARA, $q = 3{,}4$: vertical articulated;
- $r_{2q} =$ gripper, $q = 1{,}2$: 2(3) –finger number, $q = 3{,}4$: flat / concave-contact profile;
- $r_{3q} =$ end-effector tool, $q = 1{,}2{,}3$: none / bolt / screwdriver;
- $r_{4q} =$ physical-virtual camera duality ($\mathrm{P}_i\mathrm{V}_j$), $q = 1,2,\ldots,\sum nv_i$, $1 \le j \le nv_i$, where $nv_i =$ no. of virtual cameras defined and installed for each physical camera $i, 1 \le i \le 9$;
- $r_{5q} =$ magnetic code R/W device, $q = 1{,}2{,}3{,}4$.

Resource r_{lq} is: *operational* if it can be used after a finite time delay $\Delta_{lq}, q \in Q_l, l \in L, \Delta_{lq} \ge 0$, *available* if $\Delta_{lq} = 0$, and *down* otherwise. The status sr_{lq} of a resource is then:

$$sr_{lq} = \begin{cases} 2, & r_{lq} \quad \text{available} \\ 1, & r_{lq} \quad \text{operational} \\ 0, & r_{lq} \quad \text{down} \end{cases}$$

An *assembly plan* $\mathrm{AP}_p^{(\delta)}$ of a product A_p is embedded in a resulting Order Holon OH as a vector of triplets, each specifying operation number o_i, processing time $t_i^{(\delta)}$ of operation o_i using assembly plan δ , and set of resources $R_i^{(\delta)}$ to process the operation o_i :

$$\mathrm{AP}_p^{(\delta)} = [\ldots,(o_i, t_i^{(\delta)}, R_i^{(\delta)}),\ldots],\ 1 \le i \le f\ ,$$

where $R_i^{(\delta)} = \{r_{1q|i}^{(\delta)},\ldots,r_{5q|i}^{(\delta)}\},\ q \in Q_l, l \in L, 1 \le i \le f$.

The basic assembly plan with o_f as final operation is denoted by $\delta = b$, whereas $\delta = a$ are alternative plans ($b = 0, a = 1,2,\ldots$).
Experiments showed that in most cases $t_i^{(b)} \le t_i^{(a)}, o_i \in O$.

The Global Assembly Scheduler computes off-line the $\mathrm{OH}_p^{(\delta)}, 1 \le p \le P$ at batch level rending assembly plans $\mathrm{AP}_p^{(\delta)}$ *available* for products $A_p, p \in P$. One operation $o_i \in O$ in the p^{th} OH is *executable* if all resources needed to carry it out are defined as <u>operational</u> by at least one $\mathrm{AP}_p^{(\delta)}$. Operation $o_i \in O$ is *schedulable* at time t, if

1. No other operation (mounting, inspecting) upon the same product is being processed at time t.
2. All operations preceding o_i have been completed before time t.
3. All resources needed by the basic or one alternate assembly plans $\mathrm{AP}_p^{(\delta)}$ to process operation o_i are available.

Both the off-line Global Assembly Scheduler and the real time holonic control mechanism need to know the current operation status $os_i, 1 \le i \le f$, retrieved either from GAS computing or from sensory data:

$$os_i = \begin{cases} 0, & \text{op } i \text{ is nonexecutable} \\ 1, & \text{op } i \text{ is executable} \\ 2, & \text{op } i \text{ is nonschedulable} \\ 3, & \text{op } i \text{ is schedulable} \\ 4, & \text{op } i \text{ is being processed (scheduled)} \\ 5, & \text{op } i \text{ has been completed} \\ 6, & \text{op } i \text{ meets schedulability conditions 1,2} \end{cases}$$

$S_k =$ operations set with $os_i = k, k = 0,..,5,\ o_i \in O$

Further notation is used, based on above definitions:

$ne_{ip} =$ number of executable operations in $S_1 \cap OA_p, o_i \in O, p \in P$

$no_{ip} =$ number of schedulable operations in $S_3 \cap OA_p, o_i \in O, p \in P$

It is assumed that in the off line process of assembly schedule generation all operations O are executable, as the necessary resources are *operational*. However, during the GAS computing, an operation might not be processed according to the basic assembly plan due to the *unavailability* of some specified resources. In this situation the GAS exits the heuristic algorithm and enters the inference engine of the knowledge-based (KB) system, where the procedure of selecting alternative plans $\mathrm{AP}_p^{(a)}, p \in P, a \ne b$ is carried out.

The procedural knowledge of the KB_GAS is in the form of *production rules*, from which three sets are used to generate Order Holons $\mathrm{OH}_p^{(\delta)}, 1 \le p \le P$ for all final assemblies: SEL_ALG, ALT_PLAN, and EVAL_RES. Some rules in these sets stop the search of the inference engine and switch the control process back to the heuristic algorithm.

Based on tested performances (Kusiak, 1990), six priority scheduling rules have been incorporated into the *heuristic assembly scheduling algorithm* (SA):

1. Initialize:
   - Set current time $t = 0$ and resource status $sr_{lq} = 1, q \in Q_l, l \in L$
   - Set the four sets $S_0 - S_3$ respectively of non executable (empty), executable, non schedulable (empty) and schedulable operations $os_i = 0,...,3$
2. From the set of schedulable operations, select an operation o_α based on the priority rules below:
   - PR1: $ns_{\alpha p} = \max\{ns_{ip}\}, p \in P, i \in S_3$
   - PR2: $no_{\alpha p} = \min\{no_{ip}\}, p \in P, i \in S_3$
   - PR3: $ni_{\alpha p} = \max\{ni_{ip}\}, p \in P, i \in S_3$
   - PR4: $nu_{\alpha p} = \max\{nu_{ip}\}, p \in P, i \in S_3$
   - PR5: $t_i = \min\{t_i^{(0)}\}, i \in S_3$
   - PR6: select an arbitrary operation o_α (break a tie arbitrarily)
3. Schedule the operation selected in step 2:
   - Operation status $os_\alpha = 2$ for o_α selected
   - Operation status $os_i = 0$ for all the unprocessed operations of the assembly corresponding to o_α Delete operation o_α from S_3. If $S_3 \cup S_2 = \varnothing$, stop (there are no unprocessed operations); otherwise set:
   - Remaining processing time $rt_\alpha = t_\alpha^{(0)}$
   - Resource status $sr_{lq} = 0$ for $r_{lq} \in R_\alpha^{(0)}$, $q \in Q_l$, $l \in L$.

     Update S_3 and S_2. If $S_3 \ne \varnothing$, go to step 2. If $S_3 = \varnothing$ and there are available resources, go to step 4; otherwise, update set S_6, enter the inference engine (rules ALT_PLAN generate an alternate OH triplet), and return.
4. Calculate the completion time of each operation scheduled but not completed at the current time. Set the current time equal to the completion time of the operation with the last remaining time. Add this operation (or operations in case of a tie) to the set of completed operations:
   - $f_i = rt_i + t, i \in S_4$
   - $t = f_{\tilde{i}} = \min\{f_i\}, i \in S_4$
   - Set operation status $os_{\tilde{i}} = 5$ and delete $o_{\tilde{i}}$ from S_4.
   - Set resource status $sr_{lq} = 1, r_{lq} \in R_{\tilde{i}}^{(0)}, q \in Q_l$, $l \in L$ and remaining time $rt_i = f_i - t, i \in S_4$.

     Update S_3 and S_2.
5. If $S_3 \cup S_2 = \varnothing$ (if there are no unprocessed operations), stop; otherwise, go to step 6.
6. If $S_3 \ne \varnothing$, go to step 2. If $S_3 = \varnothing$ and there are available resources go to step 4; otherwise update set S_6, enter the inference engine for alternate resource allocation, and return.

As mentioned above, the inference engine in the GAS triggers in steps 3 and 6 of the algorithm a rule in one of the following rule sets:

- SEL_ALG: selection of the appropriate SA

- ALT_PLAN: starts and controls the mechanism of selecting alternative OHs for a job, triggered by resource unavailability
- EVAL_RES: evaluates the computed assembly plans and decides upon rescheduling to improve the global quality of solution at batch level. The *makespan* performance function is used for this: $C_{\max} = \max\{C_i\}$, with $C_i =$ completion time of operation $o_i \in OA_p, 1 \le i \le f_p, p \in P$.

One example of ALT_PLAN rule is given below:

IF an alternate assembly plan $AP_j^{(a)}$ was defined for an $o_i \in OA_j, 1 \le i \le f_j, j \in P$ with $os_i = 2$ due to a $r_{lq|i}^{(b)}$ in $R_i^{(b)}$ $q \in Q_l, l \in L$, $sr_{lq}^{(b)} \ne 2$

AND $r_{lq|i}^{(a)}$ is such that $sr_{lq}^{(b)} = 2$ for o_i

AND the C_i for operation o_i in the basic assembly plan is longer than the one in the alternate plan

THEN replace $AP_j^{(b)}$ with $AP_j^{(a)}$, generate $\mathrm{OH}_j^{(a)}$, and add o_i to the set of schedulable operations.

4. ORDER PROCESSING IN HOLONIC MODE

Once created the OH layers for individual assemblies a holonic decision-making process for incoming jobs operates in real production time. The entities involved in this process for any incoming job are the OH, R(S)H, and inference engine of the GAS.

In the designed holonic mechanism, the assembling and control operations are executed by the Robot and Sensory resource Holons RSH after an algorithmic evaluation of the requests received from the OHs. The individual assembly schedules $AP_j^{(\delta)}, 1 \le j \le P$ delivered by the GAS are evaluated by the current OH layer and HOLON_MODE (real time) rule set in the inference engine, and awarded by:

- *choosing the lowest cost schedule* of available resources received, or by
- *reallocating resources* if the scheduled ones are not operational (due to break downs), or by
- *re scheduling operations* upon negative result of execution tracking or visual quality control.

The final decision related to an assembly job is made by the OH manager of the job after receiving offers from the R(S)H entities in the system. If necessary, changes in off line computed schedules will be made by R(S) holons based on fault-tolerant cooperation protocols.

To prepare the material conditioning proposals, both the Resource Holons and GAS need to have specific information about incoming jobs and parameters of the assembly structure: transportation times between robotic stations, magnetic code RD/WR locations, visual observation cameras – mobile, stationary, inspection/anchor features, and specification of task-oriented virtual cameras.

Based on the Contract Net Protocol (Smith, 1999), a communication protocol and task allocation process considering information received from the manager OH, networked R(S)Hs, and SMDB was defined. In this process entities have predefined responsibilities (Fig. 2).

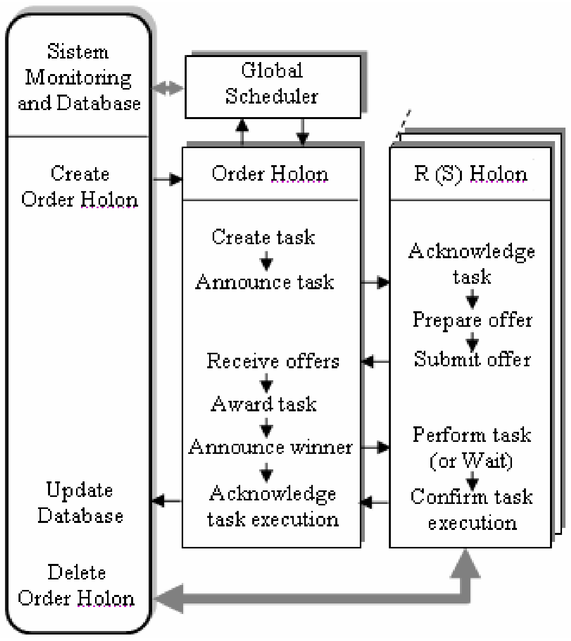

Fig. 2. Task allocation in the HOLONIC_MODE.

For each new job, there is a dedicated holon having specific responsibilities, including the right to make the final decision about which resources will execute the included operations. This decision is made by the manager of the new job - the associated OH based on the proposals received from the RSHs and GAS.

The RSH's offers have their own decision embedded in them, since they are weighted corresponding with their status sr_{lq} and future workload. When a new job enters the system, the first entity to acknowledge it is the SMDB which creates a corresponding OH. This OH announces the new operation tasks to the RSHs and the GAS, sends all the information related to the new assembly type, and sets a deadline for receiving the material conditioning (e.g. transport & manipulation, fastening,…,inspecting) offers.
The decision on which robot, pallet tracking device or virtual camera will perform the operations is made after the deadline to receive the offers has passed, and is based on checking whether the resources off line established by the GAS are operational AND the associated operations have been successfully carried out. If this is the case, the resources a priori selected will be awarded task execution; otherwise, alternate process plans are called from the SMDB and the checking process is repeated. If, during this sequence none of the existing $AP_k^{(\delta)}, k \in P, \delta = \{a,b\}$ can be validated for execution, the HOLON_MODE is switched, and a new job schedule is generated by choosing the best RHS offers for each specific operation (task).

Then, the OH announces the winning robots; the selected resource will have to perform the operations associated with the award, while others, not selected, will finish their existing tasks or stay idle if no other job is in progress. After finishing the task, the selected R(S)H confirms task execution by sending a message to the OH, which acknowledges the task execution and sends all the information to the SMDB for storing for future reference.

Under normal operation execution the holonic system will have the ability to work following a predefined schedule like a hierarchical system while in the presence of *disturbances* the autonomy of the holons allows them working in a decentralized, heterarchical architecture and thus handle the unexpected changes (breakdown, operation failure) much more smoothly.

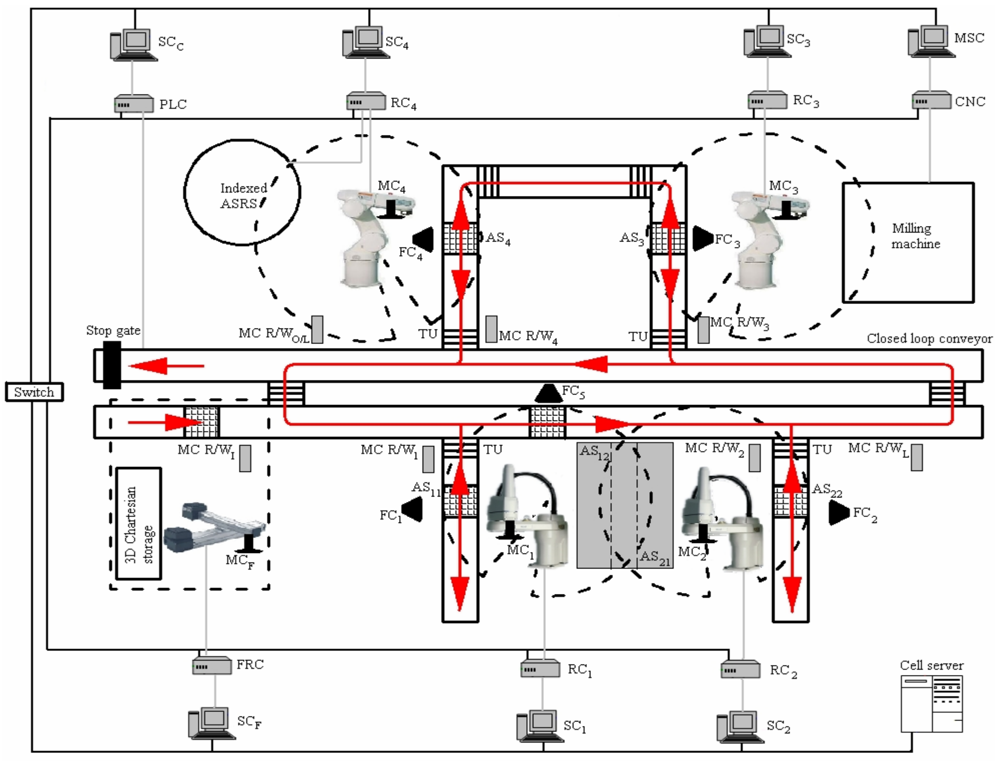

Fig. 3. The assembly cell with networked robots.

5. IMPLEMENTING RESULTS. CONCLUSIONS

The holonic control mechanism in tandem with the knowledge-based global scheduler is implemented on the assembly cell with five networked robot-vision stations, shown in Fig. 3.

The five robotic stations are interconnected by a Bosch Rexroth TS*plus* closed-loop twin-track, pallet-based power-and-free conveyor with four linear bi directional derivations that move subassemblies fixed on pallets in single- or double access AS (Assembly Stations) $AS_{11}, (AS_{12}, AS_{21}), AS_{22}, AS_3, AS_4$. Four robots are used for assembly tasks: two Adept Cobra S600 (SCARA), and two Adept Viper S650 (vertical articulated) robot-vision systems. The fifth robot is a 3D Cartesian Adept Python used to retrieve materials from a 3D storage and place them on pallets which, by the time being, are manually fed on the belt.

All four single-access AS are visualized by stationary down-looking cameras; the four assembly robots are equipped with arm-mounted cameras taking pictures in a priori defined inspection points.

The information associated to a product in embedded in a magnetic chip on the carrier pallet; this information is read in the diverting points of the loop conveyor and updated after completion of the operations performed in local robot-vision stations.

A corresponding number of magnetic code read/write devices $MC\ R/W_i, 1 \le 4$ at the workstations and in cell input (I), output (O) and loop (L) points track thus the products, and send information to the PLC controlling the material transportation system.

Fault-tolerance is provided to the cell communication system (Fig. 4), providing redundancy at both Station Controller level (a break down of a Robot Controller is detectable, the production tasks can be rescheduled to the remaining valid units for graceful degraded behaviour) and at Station Computer level (replication of data bases for the IBM PC-type device terminals, reassignment of computers in case of break downs).

The GAS is implemented by an IBM xSeries 3500 Cell Server, which transfers the recommendations of job scheduling to the OH cluster embedded in IBM PC-type Station Computers SCo_i via a GAS server-SCo client Direct Network (Ethernet). This holonic control layer is *hierarchical*.

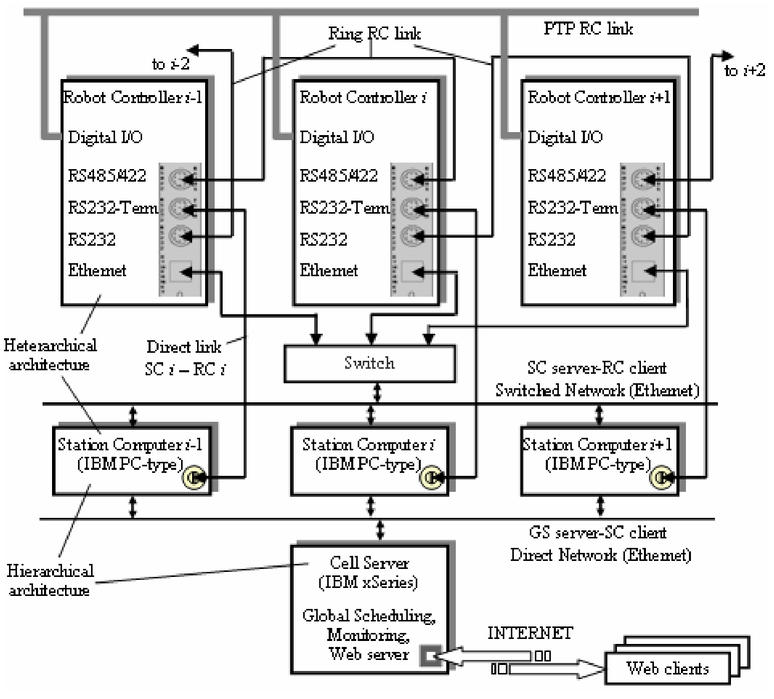
Ring RC link
PTP RC link
to i-2
to i+2
Robot Controller i-1
Robot Controller i
Robot Controller i+1
Digital I/O
RS485/422
RS232-Term
RS232
Ethernet
Heterarchical architecture
Direct link SC i – RC i
Switch
SC server-RC client Switched Network (Ethernet)
Station Computer i-1 (IBM PC-type)
Station Computer i (IBM PC-type)
Station Computer i+1 (IBM PC-type)
GS server-SC client Direct Network (Ethernet)
Hierarchical architecture
Cell Server (IBM xSeries)
Global Scheduling, Monitoring, Web server
INTERNET
Web clients

Fig. 4. Fault-tolerant communication architecture.

The cluster of SCo_i implementing the OH and RSH holonic control layers is connected to the process devices (Robot-Vision Controllers) via a SC server – RC client Switched Ethernet Network, creating a *heterarchical* fault-tolerant architecture:

- the failure of a Station Controller is detected by continuous monitoring via the direct serial links SC_i – RC_i and determines in consequence the rescheduling of jobs for the remaining valid RC;
- if one of the SC is down, its role is taken over by the remaining workstations, as each SC data base is replicated on line in all the other ones;
- if the Switch is down, the heterachical communication between the SC and RC clusters still operates via the direct SC_i – RC_i and the RC – RC Ring links.

Finally, the inter-operational conditioning between device tasks (mutual exclusion, synchronization) is provided by a cross- connection I/O network.

The feature-based description of materials and parts (scene foreground) and working environment: fixtures, feeding devices, storages (scene background), carried out by Automatic Visual Inspection affects robotic tasks by:

- *on line follow-up of products* (type, precedence, changes induced by assembly operations) in the production flow;
- *on line geometry and fastening control* of assembled products, in predefined areas of interest;
- *pose estimation* of assembly components and relative part positioning.

These image-based inspection tasks involve *measurements* which are performed in the holonic system by help of Vision Tools.

Vision Tools represent a powerful category of feature detection and measuring operators. They can be configured by the user to process upon binary or grey level segmented images. Three types of vision tools were used for inspections tasks: rulers, finders and windows region of interest.
Vision tools were used both to detect essential features in a blob's image: *points* (on contour edges or corners) and *edges* (line and arc segments), and to measure (distances, angles) and locate them (point coordinates, offsets, line orientation and arc extremities) relative to other blob features detected at run time:

- from visually located blobs, or
- from other vision tools applied to the visually located blob (Borangiu and Manu, 1999), (Borangiu and Dupas, 2001).

For particular tasks, only a small section of the field-of-view is of interest to the current control task. The computing time was reduced by using a *window processing instrument* (also called Window Region of Interest – WROI) to process only sections of the acquired product image that have critical features.

Rulers were used to get *edge information* or *grey levels* along a line or circular arc in the current image of the component or assembly, and return such data from the start of the ruler.

Finder tools allowed locating lines, points, and arcs within the field-of-view. They operate on binary or raw greyscale data; in the second case, it was possible to look for edges in an unprocessed image.

The example shown in the screen capture (Fig. 5) describes the utilization of arc finder tools to inspect an assembly of three components (modelled with the names "COIN", "LA", and "TE"), which were mounted on a support plate.

The components are fixed together by two screws, in the left-, and centre locations of the median axis of the assembly. The task consists into verifying the completeness of the assembly, as well as the correct placement of components on the support plate.

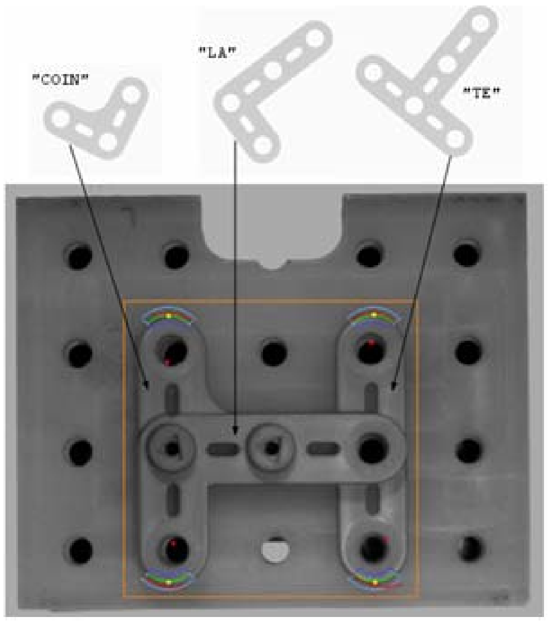

Fig. 5. Greyscale image of the final assembly of "COIN", "LA", and "TE" components, with graphic overlay of the *vision tools* used for inspection of completeness and correct mounting.

An incorrect placement of the assembly may result for example from fixture screws being absent or wrong mounting sequence of the three components.

The graphics overlaid on the acquired image of the pallet carrying the final assembly indicates the solution used for inspection: four *arc finder* tools are placed in the expected positions of the assembly's extremities.

If all four arcs of estimated centre, radius and radius range for search are found, it means that the assembly is complete and properly placed on the support plate.

REFERENCES

Babiceanu, R.F., Chen, F.F. and R.H. Sturges (2004). Framework for control of automated material-handling systems using holonic manufacturing approach, *Int. J. Prod. Res.*, Vol. 42, No. 17, 3551-3564, Taylor & Francis.

Borangiu, Th., M. Manu (1999). Object Oriented Design of Virtual Robot Controllers, *Proc. of the World Multi Conference on Systemics*, Vol. 4, pp. 1049-1055.

Borangiu, Th., M. Dupas (2001). *Vision. Mise en oeuvre an V+*, Ramanian Academy Press & AGIR Press, Bucharest.

Borangiu, Th. (2004). *Intelligent image processing in Robotics & Manufacturing*, Romanian Academy Publish. House, Bucharest.

French, S. (1982). *Sequencing and Scheduling: An Introduction to Mathematics of the Job Shop*, John Wiley, New York.

Kusiak, A. (1990). *Intelligent Manufacturing Systems*, Prentice Hall, Englewood Cliffs, N.J.

Smith, J. (1999). Contract Net Protocol, *IEEE Trans. on Computers*, 29, 1231-1236.