A Multi Agent Architecture to Support Self-organizing Material Handling

Andre Rocha, Luis Ribeiro, and José Barata

CTS, Uninova, Dep. de Eng. Electrotécnica, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
ad.rocha@campus.fct.unl.pt
{ldr,jab}@uninova.pt

Abstract. Emerging market conditions press current shop floors hard. Mass customization implies that manufacturing system have to be extremely dynamic when handling variety and batch size. Hence, the ability to quickly reconfigure the system is paramount. This involves both the stations that carry out the production processes and the transport system. Traditionally system reconfiguration issues have been approached from a optimization point of view. This means allocating a certain batch of work to specific machines/stations in an optimal schedule. Although in a an abstract way these solutions are elegant and sound sometimes the number and nature of their base assumptions are unrealistic. Approaching the problem from a self-organizing perspective offers the advantage of attaining a fair solution in a concrete environment and as a reaction of the current operational conditions. Even if optimality cannot be ensured the solutions attained and the online re-adjustments render the system generally robust. This works extends the IDEAS Agent Development Environment (IADE) developed in the FP7 Instantly Deployable Evolvable Assembly Systems (IDEAS) project which has demonstrated the basic concepts of the proposed approach. The main architectural changes are presented and justified and the prospects for the analysis and self-organizing control are presented.

Keywords: Multi Agent, Transport System, Material Handling, Self-Organization, Load Balancing, Architecture.

1 Introduction

The transport system is usually composed by a network of conveyor belts, AGVs or both. These shop floor components have the responsibility to route material across the system to the different resources.

The load balancing optimization problem, as approached traditionally, normally disregards the role of these elements and perceives the transport system as a more or less passive entity. From an Assembly Line Balancing (ALB) perspective [1] several approaches have been traditionally considered. The Simple Assembly Line Balancing Problem was defined to cover the different manufacturers' goals, with four different variants. The SALBP-E aims to maximize the efficiency of the production line,

L.M. Camarinha-Matos et al. (Eds.): DoCEIS 2014, IFIP AICT 423, pp. 93–100, 2014.

SALBP-1 has as main goal the minimization of the number of stations needed in the system. If the goal is to minimize the execution time, knowing the number of stations in the system, the SALBP-2 formulation is considered. The SALBP-F problem formulation attempts to balance the optimization between number of stations and the desired time of execution[1].

The formulation of these problems only produce results, directly applicable to real systems, in very specific cases.

The ALB approach is therefore very dependent on the system characteristics and on the goals of a specific manufacturer and assembly installation. There is a gap between the academic models and simplifications, and the real scenario.

To cope with the pressing requirements, several approaches have been proposed. These new approaches have resulted in new production paradigms namely: Bionic Manufacturing Systems [2], Holonic Manufacturing Systems [3, 4], Reconfigurable Manufacturing Systems [5] and Evolvable Assembly Systems [6]. The present approach is proposed under the scope of Evolvable Assembly Systems (EAS) and is an improvement of the IADE architecture [7-9] that integrates control and reconfiguration with material handling aspects. Like IADE, the proposed architecture supports a transport system constituted by conveyors and implements the necessary mechanisms that allow, at each moment, the transport system elements to calculate optimal routes and knowing status of the entire transport network. The transport elements restricts their interaction and information exchange to a local scope and rely in self-organization to support the dynamics of the transport system. The main improvements over the IADE architecture is related to the interactions between the agents that have been streamlined to improved performance, plug-ability and, as importantly, the ability to formalize and guide the structure and collective behavior of the transport system.

2 Relation with Collective Awareness Systems

Manufacturing is, worldwide, a strategic sector from a socio-economic perspective. Hence there is a growing body of technologies that serve the purpose of integrating production plants with higher level logistic and management tools.

Although already significantly automated the existing technologies are not yet able to promote to the desired extent multilevel collective awareness in manufacturing.

Improving the self-configuration abilities of current shop-floors is decisive for the production sector and enables new ways of enterprise organization. By making systems more intelligent and aware, making their users aware on why the system is evolving, or proposing to evolve, in a certain direction and enabling the users to understand the impact of designing and managing a system under certain conditions contributes to the creation of more sustainable production practices while open up the door to tackled emerging business opportunities.

Awareness implies instant access to information but not only any assess. It implies an intelligent infrastructure that is able to understand and triggers different actions from its components and users so that the overall system converges to some target functioning points.

The proposed work focus on creating collective awareness at factory level, in particular in the transport elements which are envisioned as a fundamental pillar of the overall logistics, and promoting the generation of knowledge that can be used at other levels to promote collective awareness in manufacturing.

In particular this knowledge can be used to upgrade/update the system increasing its performance and efficiency and hence its sustainability.

3 Related Work

Several approaches and concepts have emerged to circumvent the limitations of traditional ALB. The Automated Material Handling Systems (AMHS) [10, 11] are very used in industry. The AMHS allow an automated routing of materials between stations and along different routes. However, these systems are difficult to modify. Their reduced flexibility, with respect to dynamic route management makes their usage restricted to a few systems and application scenarios. In order to develop approaches capable to tackling these problems, many researchers are focused in the use of multi agent-based architectures [12]. Multi agent-based design although not solving complex problems by default may, in certain cases, simplify some modeling aspects of the problems and enable a self-organizing approach that, not providing optimal solutions, still offers an efficient solution in a suitable time frame. In [12, 13] a multi agent approach is proposed to eliminate the combinatorial explosion associated with traditional scheduling. The test discussed in [12] suggests that a centralized Holonic approach attains scheduling solutions close to the optimal solution in the tested scenarios. A similar approach is found in [14, 15] in the context of FMS. More bio-inspired approaches based on ant foraging and stigmergy to explore in a distributed way alterative paths in an assembly system. are reported in [16] were some conceptual principles that can drive such a system are presented and justified while in [17] an architecture is proposed to use faster than real time simulation to provide continuous adjustments to the physical system. These approaches are not designed to explore and learn from the structure an organization of the transport system. In this context, the long term purpose of the research detailed in this paper is to investigate:

- What is the impact of network topology and distinct self-organization metric in the overall performance of the system?

The work hereby detailed is the first step towards this ambitious goal and relates with the definition and justification of the reference agent based transport architecture.

4 Agent Architecture

4.1 Notion of Skill

The agents in the architecture later detailed expresses their abilities as skills [18]. In this context, each skill has an interface that contains all necessary information for this execution. This information contains among other things the skill type. The type is important for the execution process and can assume one of two values: Atomic Skill or Complex Skill. Atomic Skills are responsible for the low level execution (hardware level i.e. I/O management) associating the skill data to a specific controller. The Complex Skill is responsible for executing processes of high complexity, these skills are constituted by a work-flow. This workflow is constituted by Atomic Skills or Complex Skills.

4.2 Generic Agents

The proposed architecture similarly to the IADE architecture is composed by five type of agents, although two of the agents are different as detailed in Table1. Table 1 describes the functional differences between the agents from the two architectures.

Table 1. Differences between IADE architecture and the new architecture

Agent	IADE architecture	New architecture
Handover (HA)	This entity is responsible to route the products. When a product arrives, this agent consults his routing table and checks what is the destination of the product. It checks, with this information, the next hop (Handover) and which Conveyor Agent or Transport Entity Agent can transport the product to the next hop.	
Source (SoA)	Abstracts the entry point of the system. This agent is used to enable products to enter the system.	It is responsible by the product entry, like in IADE architecture. The SoA is now perceived by the system, as a node (HA) and has the ability to route any product.
Sink (SiA)	Abstracts the products' exit points in the system.	Similarly to the SoA besides taking products from the system also behaves as an HA.
Transport Entity (TEA) / Conveyor (CA)	(TEA) This agent is able to abstract a conveyor belt defined between two HAs. This entity controls the execution of all the stations plugged and associated with a conveyor and all queues between these stations. This agent is also responsible for managing the plug and unplug of stations to and from the docking points inside the conveyor	(CA) It is responsible to control the product transition between two other entities in the transportation system (i.e. HA, SoA, SiA, DPA). This new agent doesn't contain any docking points, so it doesn't manage these operations as in the IADE stack.
Docking Point (DPA)		This entity abstracts the points in the system where it is possible plug stations. These points in the previous architecture were controlled by the TEA. During execution, this entity controls the product execution in this point of the system. In other situations has the same behavior as HA.
Yellow Pages (YPA)	It provides an improved yellow pages service. All agents in the platform can use the service to find other agents or skills, without a complete specification.	

4.3 Conceptual Assessment of the Architectural Changes

The main architectural differences are related to the entities that abstracts conveyor belts. The Conveyor Agent (CA), which is a new entity, is less complex than the previous case. The original specification, detailed in [8] was strongly inspired by the typical mechanical design of conveyors and stations' docking points. In this context, each conveyor contains several points where a station can be plugged. Each one of these docking points has a pre-stopper that prevents a pallet from entering the station when another pallet is already there and an additional stopper that fixes a pallet in place when is operated by the station. This creates several queues that precede each station inside each conveyor. In the IADE stack the user is allowed to add a docking point to an existing conveyor and the TEA autonomously resizes the queues while re-indexing the station's positions. Furthermore the TEA has to manage all the traffic inside all these queues. This also renders the interactions between the routing devices and the TEAs more complex since the routing devices have to index all the stations associated to a conveyor replicating already existing information.

There is an improvement in information management and performance. It may appear that this change prevents the dynamic addition of docking points however, from a mechanical point of view the introduction of a new docking point normally entails down time. In this context the tradeoff is that when a new point is introduced in the IADE architecture the user has to stop the conveyor and connect the docking point and then restart the conveyor reconfiguring it according with the new stations; in the current version the current TEA/CA is dissolved and two new CAs are created before and after the docking point. The overall reconfiguration effort is comparable with the adequate support tools. There is also a better isolation of the agents functions. The CA is mainly responsible for calculating its traversing cost. The docking point manages the stations and the skills therein and the HA is solely responsible for the computation of the best routes.

The new architecture also facilitates the modeling of the system as a network whereby transport entities, which do not have to be restricted to conveyors, behave as links and all the other entities as nodes. This new view if fundamental to study the impact of network topology in the performance of the system as it enables the extraction of configuration patterns that can improve or degrade performance. For instance is a transport system built after a small world model more efficient that the traditional line? Is it more robust? How is if affect by different metrics of transport cost?

The data model, discussed in the next section has also been improved to facilitate scalability and the specialization of new classes that can be seamlessly incorporated in the present architecture. In the IADE architecture a flat data model is followed and specialization of classes is not allowed.

5 Implementation

A simplified representation of the new data model is presented in Figure 1. All the classes are derived from the transport entity concept. A transport entity is something that is able to store products and can receive or dispatch them. A conveyor is, in this

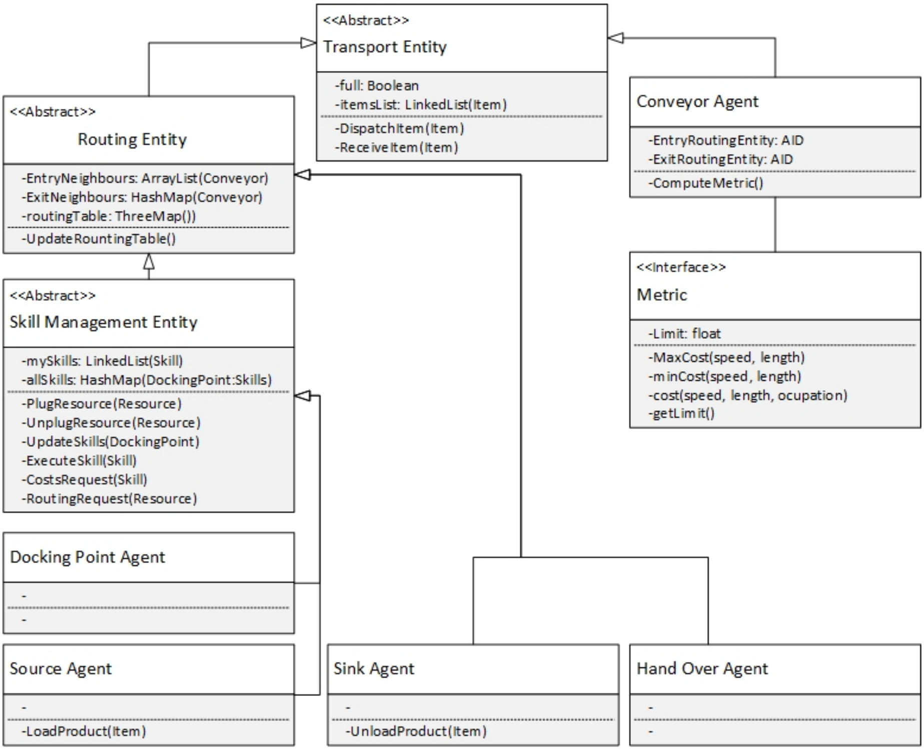

Fig. 1. Class Diagram of new architecture

context a transport entity that can compute its traversing cost using a specific metric. Each conveyor has its specific metric with a formula to calculate its traversing cost and a threshold value that defines when this cost should be updated in the neighbor HAs resend the cost update.

A conveyor is a link in the transport network as would be an AGV (not considered in this initial version of the stack).

The network's nodes are treated as routing entities. These connect several conveyors and can compute the optimal routes according to the current system status. This computation uses the dijkstra algorithm which is computed in steps inside the agent scheduler to ensure a non blocking performance by the JADE based agents.

Some of these nodes are specialized on skill management. For instance docking points are able to handle stations and their skills, This mediates the execution of skills and, plug and unplug actions. The source agent processes the state update from the docking point agents and uses this information for route the products when these enter in the system.

A sink is a specific node where it is possible remove the products when they end their execution. From a technical point of view the implementation uses the Java Agent Development Environment (JADE). Although JADE is cannot fulfill hard real time constraints its performance [19] is still acceptable to handle the dynamics of most transport systems.

6 Conclusion and Further Work

In the IADE architecture it was possible to prove that a self-organized response can be used to control a transport system in a robust manner and with acceptable performance. The architecture presented in this work improves in IADE conceptually and from a performance point of view as the role of the agents has been streamlined and the overall number of interactions was reduced. The main architectural optimization is the addition of DPA. This new agent manages the points of the system where it is possible plug and unplug stations. The CA, previously the TEA, consequently was simplified, and this simplification reduces the complexity and the computation. This reduction increases the performance of the entire system. The stabilization of the architecture and the test currently ongoing are however only the starting point of a far more ambitious work that related with the development of self-organization metrics to regulate the adaptive response of agent-based transport systems.

References

[1] Boysen, N., et al.: Assembly line balancing: Which model to use when? International Journal of Production Economics 111, 509–528 (2008)
[2] Ueda, K.: A concept for bionic manufacturing systems based on DNA-type information. In: Proc.s of the IFIP TC5/WG5. 3 Eight International PROLAMAT Conf Human Aspects in Computer Integrated Manufacturing, pp. 853–863 (1992)
[3] Van Brussel, H., et al.: Reference architecture for holonic manufacturing systems: PROSA. Computers in Industry 37, 255–274 (1998)
[4] Babiceanu, R.F., Chen, F.F.: Development and applications of holonic manufacturing systems: a survey. J. Intelligent Manufacturing 17, 111–131 (2006)
[5] Koren, Y., et al.: Reconfigurable manufacturing systems. CIRP Annals-Manufacturing Technology 48, 527–540 (1999)
[6] Onori, M., et al.: "An architecture development approach for evolvable assembly systems," in The 6th IEEE Int Symp. on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing (ISATP 2005), pp. 19–24 (2005)
[7] Ribeiro, L., et al.: Self-organization in automation-the IDEAS pre-demonstrator. In: IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society, pp. 2752–2757 (2011)
[8] Rocha, A.D.B.D.S.: An agent based architecture for material handling systems (2013)
[9] Ribeiro, L., et al.: A product handling technical architecture for multiagent-based mechatronic systems. In: IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, pp. 4342–4347 (2012)
[10] Haneyah, S., et al.: Generic planning and control of automated material handling systems: Practical requirements versus existing theory. Computers in Industry (2012)
[11] Haneyah, S., et al.: A generic material flow control model applied in two industrial sectors. Computers in Industry (2013)
[12] Shen, W., et al.: Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 36, 563–577 (2006)

[13] Guo, Q.-L., Zhang, M.: An agent-oriented approach to resolve scheduling optimization in intelligent manufacturing. Robotics and Computer-Integrated Manufacturing 26, 39–45 (2010)
[14] Turgay, S.: Agent-based FMS control. Robotics and Computer-Integrated Manufacturing 25, 470–480 (2009)
[15] Rubrico, J.I., et al.: Online rescheduling of multiple picking agents for warehouse management. Robotics & Computer-Integrated Manufacturing 27, 62–71 (2011)
[16] Peeters, P., et al.: Pheromone based emergent shop floor control system for flexible flow shops. Artificial Intelligence in Engineering 15, 343–352 (2001)
[17] Sallez, Y., et al.: A stigmergic approach for dynamic routing of active products in FMS. Computers in Industry 60, 204–216 (2009)
[18] Barata, J.: Coalition based approach for shopfloor agility (2005)
[19] Ribeiro, L., et al.: A study of JADE's messaging RTT performance using distinct message exchange patterns. Presented at the IECON 2013-39th Annual Conference on IEEE Industrial Electronics Society. IEEE (2013)