Application of Intelligent Service Bus in a Ramp-up Production Context

César A. Marín[1], Lars Mönch[2], Liwei Liu[1], Nikolay Mehandjiev[1], Georgios V. Lioudakis[3], Daria Kazanskaia[4], and Vadim Chepegin[5]

[1] Service Systems Group, MBS, The University of Manchester, Manchester M15 6PB, United Kingdom
{*forename.surname*}@manchester.ac.uk
[2] Enterprise-wide Software Systems, University of Hagen, 58097 Hagen, Germany
[3] National Technical University of Athens, Heroon Polytechniou 9, 15773 Athens, Greece
[4] Smart Solutions Ltd, Moskovskoye shosse 17, 443013 Samara, Russia
[5] TIE Nederland B.V, The Netherlands

Abstract This paper reports on our experiences in developing a conceptual architecture based on Intelligent Services within the context of ramp-up systems for aerospace manufacturing: a domain characterised by the conflict between the need for control and rigour and the reality of changes and problems affecting cross-boundary systems. To alleviate these issues, we have decided to focus on combining agent technology with enterprise service bus, providing basis for real-time automatic negotiation, planning, scheduling and optimisation, intra and across factories. With this we push forward a new definition of Intelligent Services.

1 Introduction: Motivating Scenario

Manufacturing has been characterised as a highly dynamic domain [9]. Information systems, in particular service-based systems, haven been used for manufacturing support due to their characteristics for dealing with the dynamism inherent in the domain, e.g. [5, 7]. Yet the level of dynamism supported by those approaches is limited to the capabilities of the particular technologies used, e.g. Web services or software agents.

Our work focuses on small lot production management which is a type of manufacturing where there is a low-volume and high customisation of final products. Examples of this are aerospace and shipbuilding. This production type involves a large number of parts to be assembled by a large number of people, potentially in separate geographical locations and dealing with a large number of suppliers. Problems arise when orders arrive requesting large numbers of products in a short period of time, stressing the limits of the production capacity.

The supplier network has to be planned and controlled in an appropriate way. The production of airplanes is often organised in several flow lines, each of them consisting of several stations. The assembly often takes place in geographically distributed manufacturing sites. Therefore, the information system for manufacturing becomes very important to help in planning and making decisions.

Let us exemplify this with a ramp-up scenario of a new airplane production. This is based on an industrial use case within the context of the ARUM project[6]. The scenario consists of three factories which have to communicate with each other in order to synchronise and optimise a production cycle, avoid delays and stops and so on.

The scenario starts when the factories receive instructions for producing a new aircraft batch. The production plan has to be distributed among the factories and each production stage has a clear deadline. The whole plan is arranged in such a way to meet the last deadline when the customer received the aircraft.

First, an initial scheduling of the production is performed. The operations are allocated to the factories (i.e. to the stations and workers at the shop floor) according to the specified production processes. The factories have to confirm the schedules before starting to work. At some point, factory A identifies that a machine is broken thus it cannot complete the scheduled operations in time. Therefore tasks have to be rescheduled and reallocated to workers in order to accommodate this change. The consequence of such a problem when not solved in a short time is a cascade of delays which require replanning of tasks not just days ahead but perhaps months. This demands for a solution that incorporates changes from the physical world at system runtime.

In either case other factories have to be notified of the reschedule in order to prepare for delays and reschedule their own operations. Such notifications should be timely and involve several levels of the organisation: from administrative people to station managers and workers at the workshop stations. This is to accommodate 1) immediate changes to a worker's tasks, 2) changes to the next shifts and perhaps with a few days in advance, and 3) changes to the long term planning. This demands for a solution that takes into account communication between factories which are geographically distributed.

Reschedule across several factories is not an easy nor automated work. Factories have to negotiate new deadlines, shift tasks and reallocate resources. Sometimes negotiation is one-to-one between factories. Sometimes it is many-to-many. And other times short-term decision making is mandatory. This demands for near real-time automatic negotiation such as that provided by agent technology, cf. [6].

Interoperability is yet another issue. Exchanging information in an electronic way between factories can be complicated due to information being stored in disparate formats from factory to factory, and even from engineering discipline to engineering discipline. To alleviate these issues, an integrated solution is necessary where information is lifted from external systems, e.g. ERP and MES (manufacturing execution system), converted to a standard format and incorporated into the decision support. Moreover, such solution should guarantee information delivery to whoever needs it.

---

[6] http://www.arum-project.eu

Aerospace manufacturing is labour-intensive. Issues mentioned above arise yet people manage to overcome them, but at a high cost: an increased differentiation between original plan versus actual execution which is translated in economic loss.

2 Requirements for a Smarter Production Management

Customer needs have been increasingly diversified in small lot production manufacturing. Thus a large variety of sophisticated products has to be managed in current and future production sites to meet the needs. For example, aerospace companies often develop a large number of different airplane models to satisfy customers' requirements. In particular, small lot production results in low-volume, high customisation manufacturing where an efficient utilisation of labour and machinery creates significant challenges [2, 8]. This type of production also leads to several challenges for engineering, increasing the effort for production planning and control. Frequently occurring errors demonstrate that product design, the production lines, the suppliers, IT, and logistics are not ready by the very first products of a series. Overall, the low-volume, high customisation production leads to the following requirements for a supporting information system in this domain:

1. In contrast to a fully-automated computation of production planning and control instructions in which plans and schedules are changed manually, small lot production is labour-intensive manufacturing. Because of this, decision support has to be offered to the staff at several levels of the production planning and control department: administration, station managers, and workers. This calls for the availability of scheduling and planning tools at the work stations as well as interfaces where the user can design the process;
2. Appropriate integration with external systems (cf. legacy) such as ERP or MES is crucial because these systems provide the data for decision-making during planning and scheduling. This can be fulfilled by a component dedicated to lift information from those systems and making it available to the rest of the system;
3. Provision of production planning, scheduling and rescheduling functionalities able to cope with the frequent changes and other disturbances typical of complex assembly processes. Examples of these changes include changes of products, technologies, equipment, and toolsets at system runtime. Scheduling and planning tools need accommodate changes from the physical world. A knowledge-base where this information is described is then needed;
4. These functionalities have to take into account the challenges of the geographically distributed production. That is, the solution has to be present at a factory yet be able to communicate seamlessly with another instance of the solution in another factory;
5. Negotiation-based approaches for short-term, near real-time, decision making are well known to be beneficial in these type of environments [6]. A

dedicated communication channel is needed to speed up negotiation and thus provide near real-time decision-making support;

6. A scalable supporting software architecture is required to be able to integrate all levels of the information systems, from sensor peripheries to external systems and selected tools for value-added services. Moreover, due to the potential heterogeneity of services, tools and external systems, an efficient communication mechanism which minimises interoperability issues is highly necessary to ensure. This requirement suggests an ESB (enterprise service bus) as the main communication channel between all components and external systems. Yet an ESB by itself is not enough. It needs to be enriched with functionality covering all previous requirements.

Traditional information systems for manufacturing and logistics like ERP and MES tend to be not sufficient in this dynamic, highly complex environment with a large number of disturbances, especially for near real-time decision making. Although the MAS (multi-agent system) and SOA (service-oriented architecture) paradigms support most of the domain requirements, ensuring interoperability between MAS and SOA tends to be restricted by their implementation.

3 Intelligent Service-based Architecture as a Solution

We propose a conceptual architecture for the support of small lot production management – Intelligent Enterprise Service-based Bus (iESB). The architecture is based on **Intelligent Services** defined as independent pieces of software that are expected to provide a particular result; such result is either produced by the Intelligent Service itself or by requesting support from other Intelligent Services. An Intelligent Service is not committed to a single technology which would limit the focus and scope of the solutions to deal with dynamism. Instead it rather focuses on the conceptual features.

An Intelligent Service is different from an intelligent Web service [3]. The latter considers the fusion between Web services and software agents as the means to inject "intelligence" into Web services, i.e. its definition is based on the implementation details rather than on the concept. Here we argue that Intelligent Services are not bound by the underlying technology and can be instantiated by agents, Web services, any of their combinations, or typical software components.

Figure 1 depicts our architecture and internal components as Intelligent Services, namely an ESB as the central communication channel, gateway, process manager, planner, scheduler, peer-2-peer layer, ontology manager, service registry and node manager. We describe these Services below and indicate how they fulfil the requirements in Sect. 2.

Enterprise Service Bus (ESB): An ESB acts as a key Service to alleviate communication and interoperability issues [10] among Intelligent Services, which fulfils requirement 6) from Sect. 2. Via the ESB, messages are sent by Services to each other. Services are also able to register an interest in message types in the ESB to get notified when events of their interest within the system occur.

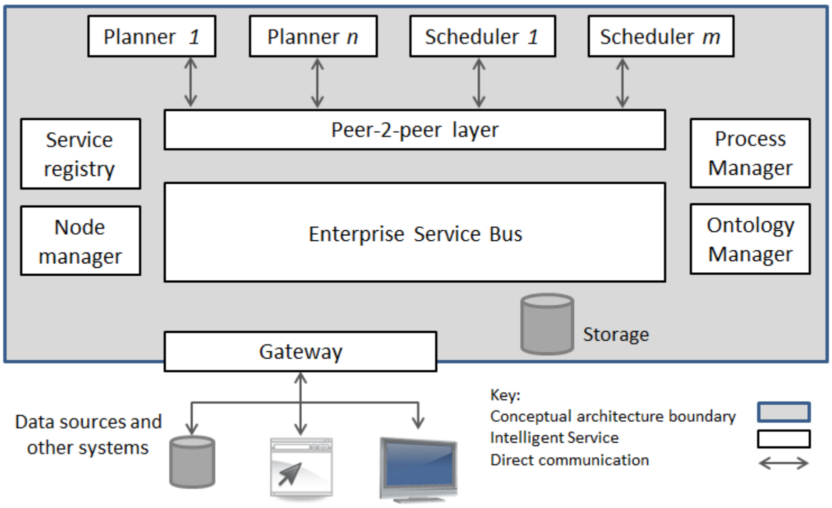

Figure 1. Intelligent Service based conceptual architecture iESB for supporting small lot production manufacturing.

Gateway: It functions as the interface for exchanging data with external data sources and systems such as ERP, MES, SCADA, etc, which satisfies requirement 2). The Gateway not only pulls information but also pushes data into external sources as needed. The availability of information from external sources and systems is notified to registered Services via the ESB for potential consumption.

Process Manager: It satisfies requirement 1), as it deals with the definition and management of manufacturing processes within the factory as well as definition and general allocation of resources such as staff, tools, machines, etc. Process instances are initiated by this Service according to how the user defines the processes themselves.

Planner and Scheduler: These Services take process instances as per the Process Manager and optimise their execution according to available resources, timing, constraints, and unexpected events, as indicated by requirement 3). The difference between the Planner and the Scheduler is the time scope: the Scheduler deals with imminent tasks and next shift tasks, whereas the Planner deals with tasks weeks and months in advance.

In both cases, unexpected events such as sudden increase of process instances, off-sick staff, broken machines, etc., are taken into account for rescheduling and adaptation. There may exist more than one of these Services depending on the factory configuration and the different types of processes to handle. Further details about the internal operation of these Services can be found in [1].

Peer-2-peer Layer: In order to deal with near real-time decision making in requirement 5), the Planners and Schedulers can benefit from a dedicated Peer-2-peer Layer Service for handling negotiation communications without passing through the ESB. The advantage of this is the minimisation of message delivery time and negotiation turnaround. Notice that this Service only supports the Scheduler and Planner Services, however this does not limit them from using other available Services in the architecture.

Ontology Manager: An important characteristic of the proposed architecture is that it relies on ontologies for modelling a variety of concepts. A unified ontological approach for modelling domain knowledge, real-world entities, policies and operational workflows, provides the proposed architecture with several advantages. Two of the main ones, firstly it benefits from formal and machine-interpretable semantics, semantic interoperability and consistency, as well as inference of knowledge not explicitly contained in the models. And secondly, concepts are treated in an integrated manner so that the model can effectively capture their dependencies, while allowing for the enforcement of interdependencies at all phases of their lifecycle. With this Service we contribute to the fulfilment of requirements 3) mainly, but 1) and 6) are also benefited.

Node Manager: It establishes initial connectivity between any two architecture instances, which satisfies requirement 4). That is, Fig. 1 can be instantiated in different geographical locations, e.g. different factories. Then the Node Manager allows different running instances to communicate and share Services from one instance to the other. This procedure effectively converts each instance into a node of a system geographically distributed.

Service Registry: This Service keeps track of available Services at a node in the distributed system. It coordinates with the Node Manager for indicating availability of Services at a node on different, connected nodes. This Service contributes to requirement 6).

4 Discussion and Conclusion

Our architecture supports of small lot production manufacturing, addressing the requirements described in Sect. 2. It deals with the unexpected events in the manufacturing process. To deal with the issues described in Sect. 1, three instances of the conceptual architecture can be used one at each factory. Message delivery is guaranteed thanks to the ESB. Near real-time negotiation can be achieved thanks to the dedicated peer-2-peer communication channel. In addition, the introduction of Intelligent Services without committing to a specific technology allows us to see architecture components in a more abstract way, focusing on functionality rather than implementation constraints.

Committing a technology at design time, for instance Web services, the scenario would have been more reactive in the sense that Web services are stateless and thus they just respond to calls, lacking proactiveness. On the other hand, using MASs only renders the ESB pointless because agents could use their own communication channel. However, interoperability issues arise when agents do not follow a common communication protocol. Combining agents with Web services as an instantiation of the iESB then provides some flexibility and minimises interoperability issues. Nevertheless, it all depends on how such combination is carried out, which might impose a combination 'flavour' on the overall system.

We argue that all the components in the architecture are the minimum necessary to support and improve small lot production such as aerospace manufacturing. The main advantage the ESB brings to the approach is the homogenisation of communication between all components, thus reducing interoperability issues. Adding the rest of the components as Intelligent Services would aid software developers to maintain the focus on functionality as long as the ESB is used.

Currently our approach is conceptual, thus at this stage is difficult to analyse any implementation issues. Within our project ARUM, we are starting the development of the iESB and we are planning test for future evaluations, in particular, performance comparisons with implementations where all components are agents, Web services and their combinations. Thanks to the conceptualisation of our approach we can apply it to other domains with similar requirements (Sect. 2) such as business ecosystems [4].

Acknowledgement

This work is supported by the European Commission funded project ARUM: Adaptive Production Management. Grant agreement No. 314056.

References

1. Goryachev, A., Kozhevnikov, S., Kolbova, E., Kuznetsov, O., Simonova, E., Skobelev, P., Tsarev, A., Shepilov, Y.: Smart Factory: Intelligent System for Workshop Resource Allocation, Scheduling, Optimization and Controlling in Real Time. In: Proceedings of the 2012 International Conference on Manufacturing, Advanced Materials Research, vol. 630, pp. 508–513. Trans Tech Publications, Switzerland (nov 2013)
2. Heike, G., Ramulu, M., Sorenson, E., Shanahan, P., Moinzadeh, K.: Mixed Model Assembly Alternatives for Low-volume Manufacturing: The Case of the Aerospace Industry. International Journal of Production Economics 72, 102–120 (2001)
3. Heuvel, W.J.V.D., Maamar, Z.: Moving Toward a Framework to Compose Intelligent Web Services. Commununications of the ACM 46(10), 103–109 (Oct 2003)
4. Marín, C.A., Stalker, I.D., Mehandjiev, N.: Engineering Business Ecosystems Using Environment-Mediated Interactions. In: International Workshop on Engineering Environment-Mediated Multi-Agent Systems. Lecture Notes in Computer Science, vol. 5049, pp. 240–258 (2007)
5. Popescu, C., de los Angeles Cavia Soto, M., Martinez Lastra, J.L.: Runtime Modeling of Flow for Dynamic Deadlock-free Scheduling in Service-oriented Factory Automation Systems. IETE Technical Review 26(3), 203–212 (2009)
6. Shen, W., Hao, Q., Yoon, H.J., Norrie, D.H.: Applications of agent-based systems in intelligent manufacturing: An updated review. Advanced Engineering Informatics 20(4), 415–431 (2006)
7. Shen, W., Li, Y., Hao, Q., Wang, S., Ghenniwa, H.: Implementing collaborative manufacturing with intelligent Web services. In: The Fifth International Conference on Computer and Information Technology, CIT 2005. pp. 1063–1069. IEEE Computer Society (sep 2005)

8. Storch, R.L., Lim, S.: Improving Flow to Achieve Lean Manufacturing in Shipbuilding. Production Planning and Control 10(2), 127–137 (1999)
9. Verstraete, P., Germain, B.S., Valckenaers, P., Brussel, H.V., Belle, J.V., Hadeli, K.: Engineering manufacturing control systems using PROSA and delegate MAS. International Journal of Agent-Oriented Software Engineering 2(1), 62–89 (2008)
10. Ziyaeva, G., Choi, E., Min, D.: Content-Based Intelligent Routing and Message Processing in Enterprise Service Bus. In: International Conference on Convergence and Hybrid Information Technology (ICHIT '08). pp. 245–249. IEEE (2008)