VIRTUAL FACTORY FRAMEWORK: KEY ENABLER FOR FUTURE MANUFACTURING

P.Pedrazzoli; M.Sacco; A.Jönsson; C.R.Boër
TTS srl - pedrazzoli@ttsnetwork.com; ITIA-CNR - marco.sacco@itia.cnr.it; ETHZ - joensson@imes.mavt.ethz.ch; ICIMSI - claudio.boer@icimsi.ch

The global market with increasing competition calls for new strategies strengthening the future manufacturing systems. This paper presents the underlying models and ideas at the foundation of a new conceptual framework for the next generation Virtual Factory implementation. The approach fosters four pillars: a standard extensible data model; decoupled functional modules, based on an object oriented Virtual Factory paradigm; an event driven paradigm at the core of abstract objects management; the integration of knowledge at different layers for the decoupled functional modules. The implementation of a Virtual Factory based on the presented framework points to the usefulness when developing future manufacturing systems.

1. INTRODUCTION

A primary concept highlighted in the "Manu*future* Strategic Research Agenda" is the innovating production. The idea includes new business models and approaches to industrial engineering that need to be supported by ground-breaking Information and Communication Technologies (ICT). This paper fosters a new framework for the key enabling instrument for the factory of the future: the Virtual Factory (VF).

The implementation of a holistic, integrable, up-gradable, scalable VF carries high cost savings in the implementation of new manufacturing facilities thanks to the effective representation of buildings, resources and products. The ability to simulate dynamic complex behaviour, over the whole life cycle with scalable level of details, is the fundamental cornerstone to achieve time and cost savings while increasing overall performance in the design of new facilities and in the management, evaluation and evolution of existing ones. (Jönsson 03, VDI 04)

This paper presents the underlying models and ideas at the foundation of a new conceptual framework for the next generation VF implementation**,** consistent with the Manu*future* vision, meant to lay the basis for future works and applications in this research area. The approach here presented identifies four pillars:

I - ***Standard Extensible Data Model*** for factories, taking into account the needs of a holistic and scalable modelling, of real time management of manufacturing data and of collaborative engineering networks.

II - ***Decoupled Functional Modules,*** based on an object oriented VF paradigm meant to facilitate modelling of complex behaviour through natural mapping abstractions and modular code development.

III - ***Event Driven Paradigm*** at the core of abstract objects management. This structure enables a centralized synchronization (not control) of the external decoupled modules thanks to the internal transition system of the core, enabling the second pillar and fostering unparalleled performances (see section 5) and increased quality in the environment representation.

IV - ***Integration of Knowledge*** at different layers as engine for the modules. The primary objective is to achieve tools which can model a wider range of complex systems and support greater comprehension of the modelled phenomenon. Moreover, the integration of knowledge throughout the VF has the potential to deliver fundamental advisory capabilities as a companion to product, process, plant development and life-cycle management.

The Virtual Factory Framework (VFF) hereinafter presented lays the basis to simultaneously address all interrelated aspects of product, process and plant life cycle, from design to disposal/recycling, supporting engineers, operating in networks, with knowledge intensive ICT tools based on standardised platform.

2. MANU*FUTURE* & FRAMEWORK PROGRAM 7

The Manu*future* strategic research agenda identifies two fronts of intense and growing competitive pressure on Europe: on one hand, in the high-tech sector we see other developed countries and, on the other hand, in more traditional sectors we face low-wage countries. Moreover, these last countries are rapidly modernising their production methods and enhancing their technological capabilities. To face this challenge, the Manu*future* initiative promotes a response based on an industrial transformation meant to strengthen Europe's ability to compete in terms of high added value, because cost-based competition is not compatible with the goal of maintaining the Community's social and sustainability standards. Manu*future* promotes a response in terms of five main points and their associated new enabling technologies: 1- new, added-value products and services; 2- new business models; 3- new advanced industrial engineering; 4- new emerging manufacturing science and technologies; 5- transformation of existing R&D and education infrastructure to support world-class manufacturing. The hereby proposed approach is meant to empower those points promoting a successful European innovative industry that will be adaptive, digital, networked and knowledge-based (Westkämper 05)

One of the four "applications research", identified in the proposal of the European Parliament and of the Council concerning the seventh framework programme, is the ICT supporting manufacturing industry towards rapid and adaptive design, production and delivery of highly customised goods, digital and virtual production and modelling & simulation, in a dynamic networked collaborative work environment. The development of a knowledge-based Virtual Factory Framework, based on the new paradigm hereinafter presented, is seen as a concrete response to those research needs and as a major strategic priority for European manufacturing enterprises in all sectors.

3. THE R&D FRAMEWORK AND STATE OF THE ART

Thus rapid product, process, enterprise realization and integration through an effective virtual tool have been identified among the imperatives for enabling the next generation manufacturing. This paper proposes a VF approach meant to support these imperatives, that takes into account past results and framework proposals.

Several approaches have been fostered by national and international research programmes and the idea of a reference framework has been introduced by (Krolak 96, Jain 95). Most of the previous projects, ManuFuturing (Boër 97), MPA (Sacco 04, Boër 00), IRMA (http://www.virart.nottingham.ac.uk/Projects_Irma.htm), exploited an existing suite (VEGA Multigen + Arena, or ad hoc application + Quest) for developing VF applications. These project realized very good applications, but they lacked flexibility and reusability. Moreover integration was achieved on ad hoc basis. In the just started projects DiFac and CoSpace, a clear architecture for a seamless, flexible and up-gradable solution for developing the VF is not apparent or emerging. This topic is also highlighted as crucial in the NoE INTUITION. One of the main problems everybody is facing, in the development of a VF, is the availability of proper tools and framework (Mueller 02, Waller 02, Zhai 02)

Most of the previous projects and approaches focus either on the use of commercial tools (Superscape, DigitalMock-up, WorldToolkit) either on more low level software (Vega, Performer, OpenGL directly or customized set of libraries, such as the Unifeye SDK). The first approach faces the problem of excessively rigid tools for developing a complete factory and its functionalities, or too simple to have realistic results. The second, besides the problem of starting from scratches, confronts with the need to offer complex simulation functionality, thus implying the integration of existing tools whose interface has to be studied and adapted.

Lastly, several projects are demonstrating the possibility to democratize the use VR tools, thus making them widespread, such as the IP KoBaS (Pedrazzoli 04).

The VFF represents a fundamental cornerstone for the next generation virtual manufacturing that addresses the issues discussed related to previous approaches: the VFF will be used as an object oriented framework to support single and group work in an immersive and interactive way, for concurrent product design, prototyping and manufacturing (Schuh 03), as well as worker training, providing support for data analysis, visualization, advanced interaction, presence within the virtual environment, and collaborative decision-making. In this context, the VFF must confront with the five technical strands of the DET Framework (Maropoulos 03) to have their relationship evaluated and analysed.

4. VIRTUAL FACTORY FRAMEWORK

The VF is defined as an integrated simulation of major activities and systems of a factory, that considers the factory as a whole and provides an advanced planning, decision support and validation capability (Jain 2001). The VFF implements the framework for a object oriented collaborative virtualized environment, representing a variety of factory activities meant to facilitate the sharing of factory resources, manufacturing information and knowledge and supports the simulation of design, planning, production and management among different participants.

4.1 The event driven Core of the VFF

Figure 1 presents the Core as the funding building block of the proposed paradigm: it is the manager of all the objects of the virtual environment. It is based on an event driven architecture (pillar III), further detailed later, supported by a common data model (pillar I), and can be divided into three major parts: scenegraph, action manager and event manager. These modules deal with the three most important functions of managing information, incoming actions and outgoing events.

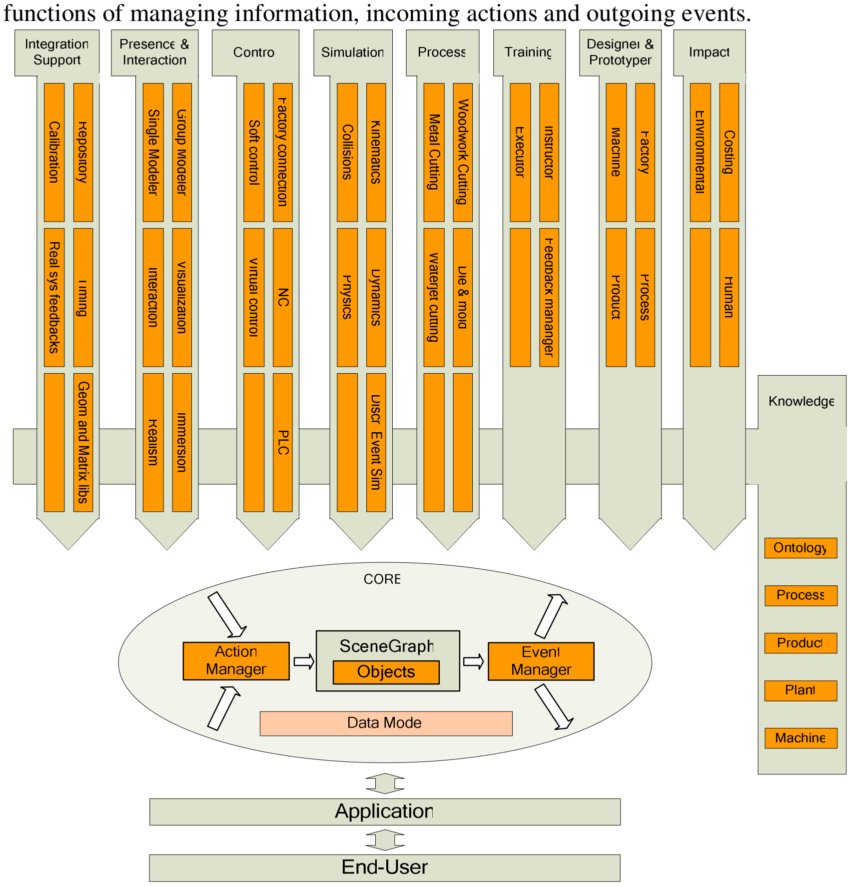

Figure 1– the Virtual Factory Framework – VFF

The scene graph is the repository of the data objects describing each element composing the environment and their relations. This module guarantees data consistency performing internal checks every time an object is created or destroyed. The action manager is taking care of the first phase in the transition-notification paradigm verifying that concurrent actions on shared objects are correctly queued and processed. The event manager handles all the events generated inside the objects that completed a state transition. Each event can be scheduled on two different queues that represent current and next state and that are swapped every time a dispatch occurs. The notification mechanism can be accessed by the core user in order to control the dispatching phase.

This structure enables a centralized complete synchronization (not control) of the external code with the internal transition system of the core (pillar II). In a multi threading context this synchronization reduces the risk of deadlocks or racing situations, by concentrating actions and event responses into defined phases. The common and extensible data model provides a significant impact on the possibility for real time multi-user multi-disciplinary modules to synchronously collaborate.

Starting from this cornerstone and adopting the proposed framework, it is possible to identify several macro-modules (pillar II), fitting into this paradigm, dealing with the various features and potentiality of the virtual factory: Figure 1 presents a non exhaustive scenario of interacting modules in the VFF.

Besides providing a common ontology (VRL-KCiP NoE fosters an effective approach www.vrl-kcip.org), the integration of a horizontal knowledge-based support (pillar IV) offers functionality to model a wider range of complex phenomenon and support greater comprehension of those systems.

The Core and the cooperating modules are to be regarded as a library framework that will be used to create and deploy complex customized VF applications, which will be interacting with the end-user.

4.2 The internal architecture of the event driven Core

Figure 2 represents a more detailed view on the internal architecture of the core and some possible related modules using a standard UML Component Diagram. The three most important parts of the core, the action manager, the scenegraph and the event manager, are described in terms of functional components and dependency.

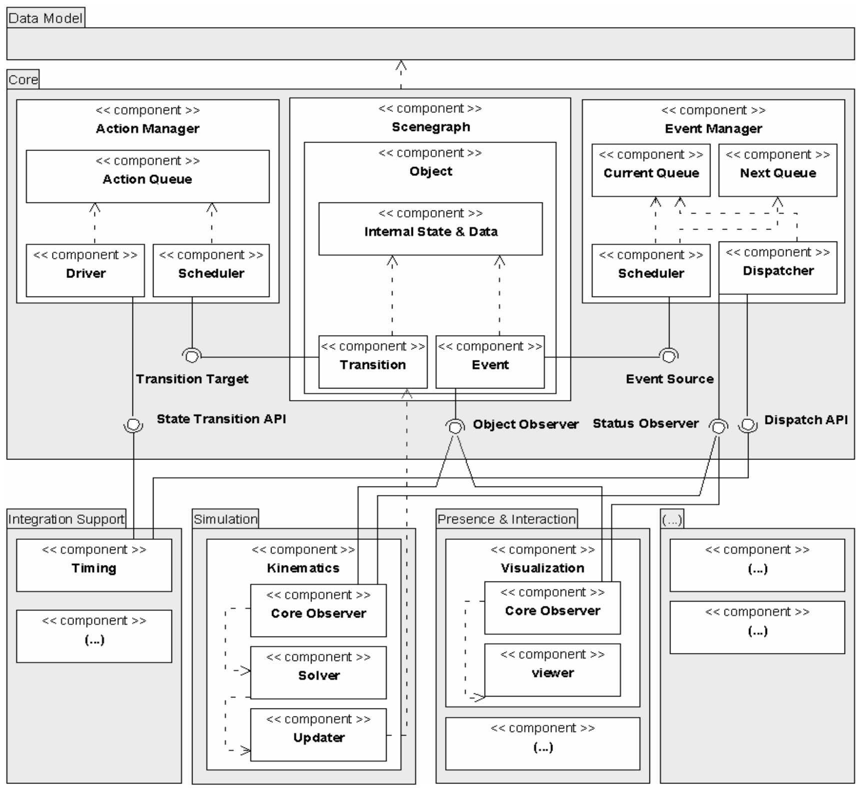

Figure 2 – internal architecture of the event driven core

The scenegraph is composed by a list of object elements representing the logical image of the real objects. Each object contains an internal set of data, which is defined inside the data model, and a related internal state. The transition component is responsible for maintaining the consistency of the internal information. When an external action is requesting a modification of an object, a new transition element manages the request, scheduling it in the action manager as a transition target and preventing the internal state to be immediately modified. The scheduler of the action manager takes care of the transition targets produced by all the scenegraph objects, queuing them inside a dedicated action queue which works as a FIFO buffer. The action queue is managed by the driver component which exposes an API that allows to control when to flush all the queued transitions, applying the new values to the objects data and performing a state transition of the entire scenegraph.

When a new value is applied, a new event is generated inside the object to notify the object observer components that something changed in the internal data. Nevertheless the event is not immediately notified but it is scheduled inside the event manager as an event source. The scheduler of the event manager takes care of queuing the event sources inside the current state queue or the next state queue. The decision about the correct buffer where to store the event source depends on the type of the generated event. The moment the events are notified can be synchronized from outside the core through the API exposed by the dispatcher component of the event manager. During the notification phase, the dispatcher flushes the current state queue releasing all the contained events and swapping it with the next state queue. At the end of this process, the dispatcher notifies the status observers with the end of the notification phase. Figure 2 bottom line shows examples of components interacting with this mechanism, showing how each single module can use the core in different ways, controlling the transition-notification as the timing does, or acting on the objects like the kinematics or just listening for changes like the visualization.

4.3 The Core Data-Model

The common data model can be considered as the shared "language" providing a common understanding and unified definition of the information that will be held by the core and elaborated by the external components.

Thus the modules can exactly know how to perform actions on the object instances available in the scenegraph, being able to access interfaces, method and fields definitions. The data model has to be as complete as possible concerning contained data and easily extensible from a design point of view. The object oriented approach, its principles of class design, package cohesion and package coupling must be extensively used to fit these requirements (Viganò 02) as shown in Figure 3.

In order to support the event driven paradigm, the data model must not only define the VF elements data and interfaces but also the events and observers interfaces used by the scenegraph objects. Figure 3 shows a UML class diagram of an advanced data model suitable for the VF. The schema covers just relational, geometrical, visual and kinematics issues, along with additional information related to basics of collision detection definition and production process simulation. The proposed design is not supposed to be exhaustive but it represents a good and structured example and starting point for the required extension.

The proposed architecture, along with the data-model and the described centralized state control enables an high level of data consistency while granting a total decoupling of the external modules thanks to the event paradigm. From the performance point of view, this architecture fits well in the current trend of multi-core processors: multiple threads can be issued to independently perform intensive calculations and the synchronization points are well defined and exposed.

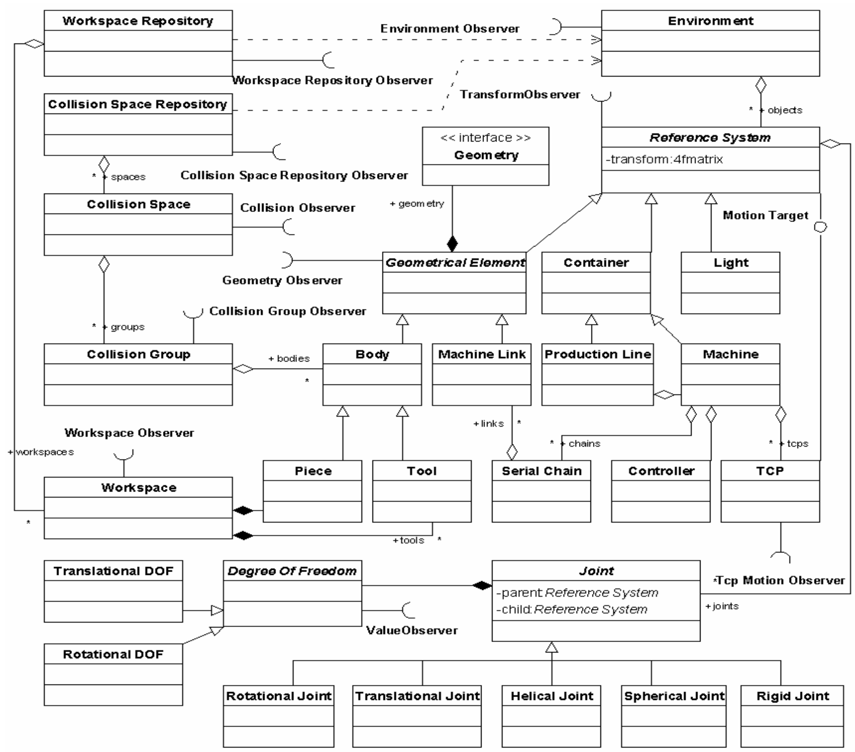

Figure 3 – the Core Data Model (internal information of object definitions have been omitted in order to make the picture more readable)

5. DEMONSTRATION

An example application exploiting the basic concepts and ideas has been developed at TTS. An innovative 3D environment solution based on the VFF principles has been deployed (the scene description is based on (Mancini 04)). The simulation environment can reproduce manufacturing machines and production lines in a virtual environment and mimic the system behaviour. The performances in terms of visualization frame-rates and CPU usage are dramatically better than current applications. In this demonstration, a PC with P4 – 3.0 GHz – NVIDIA 6800 – 256 Mb and a 3D model with 770.000 triangles were used. A comparison is done using Cosmo Player which is an high-performance, cross-platform VRML 2.0 client designed for fast viewing of virtual worlds. Cosmo reaches 13 FPS and 50% usage while the VFF based software reaches 49 FPS and 11% CPU usage. Moreover, both kinematics and visualization engines have been coupled with the core through the event driven approach, demonstrating the feasibility of the overall concept.

6. EXPECTED BENEFITS AND CONCLUSIONS

The implementation of a Virtual Factory, consistent with the VF Framework vision presented, carries high cost savings and increase in manufacturing performances, thanks to the effective support in design, monitoring, management, evaluation, evolution of product, process and plant.

It has been highlighted that, to achieve such objectives, the VF must be holistic, scalable, open, integrable, up-gradable, adherent to reality and must bring high performances. The answer to these challenges is seen by the authors in terms of four pillars, presented within this paper, providing the needed framework.

Finally a software application based on this concept has been created and tested, demonstrating the results of the application of such a framework.

7. REFERENCES

1. Boër CR, Jovane F, Sacco M, Imperio E. Virtual Reality as a Tool for Sustainable Production in the Manu***F***uturing Model. CIRP International Symposium – Advanced Design and Manufacturing in the Global Manufacturing Era 1997
2. Boër CR, Sacco M, Viganò G, Avai A. A Virtual Manufacturing Environment for planning and design of factory layout and equipment. IMCC'2000
3. Jain S, Choong NF, Aye KM, Ming L. Virtual factory: an integrated approach to manufacturing systems modelling. Int Journal of Operations & Production Management 2001; 21: 594 - 608
4. Jain S. Virtual factory framework: a key enabler for agile manufacturing. INRIA/IEEE Symposium on Emerging Technologies and Factory Automation 1995; pp 247-258,
5. Jönsson A, Bathelt J, Broman G. Interacting with real time simulations – virtual reality in industry applications. 9 Eurographics Workshop on Virtual Environments 2003.
6. Krolak P, Annunziato J, Kosta C, Sodhi M. A Framework for Virtual Manufacturing. Proceedings Symposium on Virtual Reality in Manufacturing Research and Education 1996
7. Mancini F, Francese C, Viganò G, Liao Z, Sacco M, Boër CR. The Virtual Factory - A Semi-immersive Interactive 3D Environment. 2004 Summer Computer Simulation Conference, 2004.
8. Maropoulos PG, Rogers BC, Chapman P, McKay KR, Bramall DG, A Novel Digital Enterprise Technology Framework for the Distributed Development and Validation of Complex Products, Cirp Annals 2003
9. Mueller-Wittig W, Jegathese R, Song M, Quick J. Virtual Factory – Highly interactive visualisation for manifacturing. Proceedings of the 2002 Winter Simulation Conference.
10. Pedrazzoli P, Boër CR. Knowledge based services for traditional manufacturing sectors provided by a network of high tech smes. Proceeding of INCOM 2004
11. Sacco M, Mottura S, Greci L, Viganò G, Boër CR. Experiences in Virtual Factory Prototype: Modular Plants Design and Simulation. IFAC-MIM04, Conference on Manufacturing, Modelling, Management and Control. Proceeding of IFAC 2004
12. Schuh G, Bergholz M, Westkämper E. Collaborative Production on the Basis of Object Oriented Software Engineering Principles. Cirp Annals 2003; 52; 1: 393
13. VDI 2206: Design methodology for mechatronic systems, Beuth Verlag, Berlin, June 2004.
14. Viganò G, Liao Z, Sacco M, Mottura S, Boër CR. To build object oriented virtual factory based on virtual reality and simulation tools. DET 2002, 1st CIRP (UK) International Seminar on DET
15. Waller AP, Ladbrook J. Experiencing Virtual factories of the future. Proceedings of the 2002 Winter Simulation Conference; pp 513:517
16. Westkämper E, Gottwald B, Fisser F. Migration of the Digital and Virtual Factory to Reality. CIRP Journal of Manufacturing Systems 2005; vol 34, n.5
17. Zhai, Wenbin, Fan, Xiumin, Yan, Juanqi, Zhu, Pengsheng, An Integrated Simulation Method to Support Virtual Factory Engineering. China International Journal of CAD/CAM 2002; 2; 1: 39 – 44.