15th Triennial World Congress, Barcelona, Spain

OPEN ISSUES IN FIELDBUS BASED SYSTEMS

Jean-Pierre Thomesse

LORIA-INPL
2, Avenue de la forêt de haye
54516 VANDŒUVRE Lès NANCY, France

Abstract: The fieldbuses have been developed for 20 years by different companies or consortia. And the research is always very active, even if the current subjects are more and more dedicated to the application design rather than to the protocols strictly speaking. This paper presents some main open issues in this area as an introduction of the session dedicated on fieldbuses and on fieldbus based systems.
Copyright © 2002 IFAC

Keywords: Fieldbus, Interoperability, Fieldbus based systems, Design, Architecture.

1. INTRODUCTION

Different companies or consortia including companies, end users and laboratories have developed the fieldbuses for 20 years. It has been impossible to find a real consensus at the international standardisation level for several reasons. The first one is the number and the variety of the applications (process control, manufacturing systems, automotive embedded systems, building automation systems, and so on). The second one is related to the different architecture principles to design and to implement the control system. A third reason, and may be the most important, is the strategic aspect of the fieldbus for the quality of service in a distributed system. For more details on this story the interested readers may see (Thomesse, 1998), (Fantoni, 1999), and (Thomesse, 1999).

The research on fieldbus, at its beginning, was focused on the definition of the services and of new protocols to provide the right quality of service for given kinds of application, and for given distributions of the control systems (Decotignie, 1999). For some years, the research has been oriented towards the interoperability problems, the quality of service and towards the design of system architectures. And now two main areas are of interest. The first one is related to the development of applications (cf. section 2). In this part, are considered the validation of the architectures and the interoperability. The interoperability is defined between on one hand, the fieldbus set of equipment and on the other hand, the fieldbus based systems and the external world, (essentially through the internet technologies). The second one (cf. section 3) is dedicated to the definition of new services or new protocols. The associated topics include the development of the use of new transmission media (as wireless) (Decotignie, 2002) but also the definition of new traffic management policies. These two main areas of interest are obviously closely related, including often the use of the same formal methods and tools to prove some properties. But it is interesting to distinguish these two domains because of the usual research objectives. An important characteristic of the research activities is their close relation with the industrial development of new protocols, new integrated circuits, and new solutions in architecture of equipment or of application Four classes of research may be identified in the telecommunications area:

- Specification of services and protocols,
- Validation,
- Performance evaluation,
- Test methodology.

Concerning the design and the modelling of the applications, one may identify the following topics:

— The modelling of operational architectures for validation (functionally, timely, and from performances and dimensioning point of view),
— The definition of software methods and tools for the application programming and implementation, for the configuration and commissioning.

This session addresses these previous topics which will be briefly presented in a synthetic way. In the second section, the application development will be studied, and the services and protocols research is analysed in the third section.

2 APPLICATION DEVELOPMENT

The development of applications uses a lot of techniques and of tools issued from the software engineering activities. But considering the fieldbus based systems, several constraints and characteristics differentiate them from usual applications. Indeed, the set of equipment is very heterogeneous and can never be completely proprietary. Such equipment supports part of the application, and must be configured, managed, or downloaded in a common way. Their built-in functions have to be compatible or interoperable with others. And the applications must be very often, safe, dependable, fault tolerant, and so on. All these differences lead the application designers and the equipment manufacturers to use specific approaches and tools to obtain good quality results.

Ideally, the development may be shown as composed of three main steps:
— the design of the functional architecture, abstracting the set of equipment,
— the choice of a support architecture, including networks, equipment and operating systems,
— and finally the mapping of the functions on the set of equipment to obtain the so-called operational architecture.

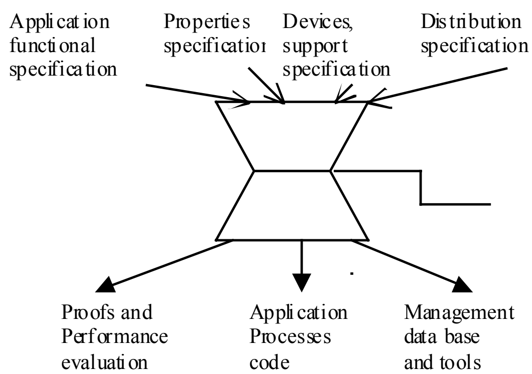

Fig. 1. A "dream" machine for development

More realistically, one may distinguish different parts in a global specification. An application may then be specified through different parts as indicated on the figure 1. Different specifications (functional specification, properties specification, devices specification, and distribution specification) have then to be defined.

These specifications may be considered as inputs of a "dream machine" that provides outputs as some proofs and performance evaluation, as the code of the processes, and as all the management information, for maintenance, evolution, or further modifications.

The "dream machine" does not exist till now, despite several tools and existing processors cover partially its functionalities.

The "dream" machine may be also decomposed into several partial machines, one for proofs one, one for performance evaluation or more for code generation. The current problem is not in the complexity of such machine, but in the specification languages or models that must be compatible to avoid the rewriting of a specification for each partial machine.

2.1 Inputs

Functional specification and modelling; This specification is the description of the functions realised by the application. An application may be functionally specified through different approaches, languages as Java, B, C++, models as Petri nets, tools as function blocks, RT objects (Cornilleau, 1997), architectures description languages as (Simonot-Lion and Elloy, 2002) and so on. All these methods privilege an objective: a language for an implementation, function blocks for a modular, reusable and standardised decomposition, formal methods for properties proofs and so on. See also (Staroswiecki, 1996).

These solutions are obviously used depending on the pursued objective.

Properties specification and modelling; The properties of a solution are numerous, correctness of the code, respect of time constraints, no deadlock, initialisation capabilities, and so on. The properties may be specified through different languages, logic, attributes in formal models.

Devices and support specification; The specification of support is a description of the set of equipment which support the operational application. This description concerns as well the sensors, the actuators, the controllers, the hosts for man machine interfaces, for maintenance, … as the networks that interconnect the equipment. This description may be very hardware oriented through the size of memory, the characteristics of the CPU, the statement set, their runtime, the data rate of a network, … But other specification are necessary, as the protocol stack, the scheduling strategies of tasks and of messages. And finally the application processes located into a device must also be specified. This kind of specification is software or application oriented. It is then necessary to distinguish three types of components in the support specification, the networks, the devices including application processes and the stripped devices (hardware and operating system).

Distribution specification; The distribution specification is the description of the location of a function, or of a datum, in fact, of all the objects introduced in the functional specification. This specification is partially done with the description of the devices that have built-in functions.

Some methods don't separate the different specifications including for example properties specification into functional specification.

2.2 Outputs

Proofs; A lot of proofs may be obtained if the application is formally specified. These proofs are qualitative (proof of algorithm termination, lack of deadlocks, …) or quantitative (termination in a given delay, …). The qualitative proofs are directly issued from the software engineering activities. They are based only on a model of the application processes. The quantitative proofs are more difficult to obtain. They need a formal model of the runtime operating system and of the hardware. Such proofs cannot be obtained formally in complex situations and then, performance evaluation and simulation are often used to obtain some probabilistic guarantee of correctness.

Performance evaluation; The performance evaluation is based on models of the network, of the application and of the operating systems. These models are usually very complex, and a very important open issue is the modelling of a communication stack without having to model all its protocols. An example of performance evaluation and of a modelling technique will be given in the last paper of this session (Cavalieri and Monforte, 2002). See (Navet, 1998) for more explanations on this point.

The combination of performance evaluation and of proofs including the application processes and the communication functions will be presented in the next paper by Juanole (Juanole, 2002).

Code generation; The code generation is the result of compilers and other tools. It is a well known operation for the first computers and it will not be detailed here. But we may mention that with the development of new formal languages and of associated formal methods, this activity is always important but relevant of other competence. We may observe this evolution with the more and more use of UML (Booth, 1999) XML (Bray, 1998), UIML (UIML).

Management system; The management system is composed of all the database and tools which are useful for the configuration of the operational system, for its maintenance, for the supervision by operators, for commissioning, and so on. The automatic generation of such management system is fundamental for the interoperability of heterogeneous components. And it is only possible if all the equipment are coherently defined. Such system has been obtained in the NOAH project (Network Oriented Application Harmonisation) (NOAH, 1999) (Demartini, 1999).

"Dream machine"; The "Dream machine" has different objectives, the production of proofs, the performance evaluation and the proof of timeliness requirements, the production of management tools as data base of the application objects, the production of code for the runtime, the production of man-machine interfaces. Such a dream machine doesn't exist. But a lot of its functions are currently realised by specific tools, compilers, editors, debuggers, or configuration tools which are very often proprietary tools, essentially depending on a given network.
Indeed, the promoters of a network have defined the so-called profiles of equipment, which they impose on a manufacturer to describe its equipment in a predefined manner. These profiles are in a certain sense, a kind of model of the equipment. These models are all oriented towards building the application by logic connection of a device outputs to the inputs of others. The aim of these models is the interoperability of data produced and/or consumed, including syntactic and semantics aspects (essentially by naming standards rather than by a formal semantic). Such solutions will be explained in the paper by Neumann (Neumann, 2002) in this session, see also (Diedrich, 1999).

The future should be the definition of a standard method to describe an equipment rather than the standard definition of an equipment i.e. the definition of a meta-model.

The problem is that only the functional aspects are taken into account. The behaviour of equipment is not described, neither its timing characteristics, operation duration, deadlines, and so on, which may be not compatible with the properties requirements of a given application. At the moment, it is impossible to obtain all the necessary proofs or guarantees of a given architecture.

3. SERVICES AND PROTOCOLS

Regarding the research activity in services and protocols for fieldbus, an important current topic is related to the use of wireless communication. Another topic is the fieldbus "integration" with

Internet. This integration concerns as well the use of Internet protocols as fieldbus protocols as the interconnection of fieldbus with the external world through Internet protocols (Kastner, 1999).

Except for wireless systems where new physical layers and new Medium Access Control layers may be created and studied, it is essential to focus the research on the application layer. The interoperability of equipment goes through the formal definition of co-operation models and of the semantic of the exchanged objects.

From a more theoretical point of view, there is always a very important issue on the modelling of communication stacks in order to obtain tractable models for the evaluation of the distributed applications. It is now very difficult to obtain tractable models taking the models of each protocol layer. It is then necessary to find another approach less detailed, more tractable, more efficient to meet with the requirements explained in section 2.

Another open issue is associated to the static vs dynamic paradigm. A lot of time critical fieldbuses are designed with a static message scheduling in order to meet the main time constraints. But if a dynamic scheduling should be used, a lot of proofs and methods will be obsolete. The required quality of service expressed by the application processes should be derived in terms of other constraints or attributes at the underlying layers in order to manage and to schedule correctly the requests and their associated messages.

4. CONCLUSION

The research activities in fieldbus and in fieldbus based systems are very closely associated. The modelling of fieldbus with the objectives of proof and of performance evaluation is of a major interest for the quality of the application design result.

All the papers of this session are related to this main problem.

The paper presented by Guy Juanole will focus on the combined modelling of the communication stack and of trhe application processes in order to evaluate the quality of service viewed by the end user, in terms of his application.

The paper presented by Françoise Simonot-Lion and Jean-Pierre Elloy is dedicated to a Architecture Description Language for the development of fieldbus based distributed and embedded systems. This language may be considered as an interesting proposal for the application specifications.

The paper presented by Peter Neumann focuses on the Device Description models and languages, their interest for the interoperability of fiedbus based systems and their necessity for the integration in management systems.

The paper presented by Jean-Dominique Decotignie is devoted to a state of the art on wireless fieldbuses.

A last paper, by Salvatore Cavalieri and Salvatore Monforte, will present the modelling of a profile for verification of constraints by performance evaluation.

All the topics cannot be treated in a single session, but I hope these lectures will give a good overview on some important research topics.

REFERENCES

Booch G., Rumbaugh J. and I. Jacobson, (1999) The Unified Modelling Language User Guide, Addison-Wesley Longman.

Bray T., Paoli J., and C. M. Sperberg-(1998) McQueen, eds, Extensible Markup Language (XML) 1.0, W3C Recommendation, 10 February 1998, (http://www.w3.org/TR/REC-xml)

Cavalieri S. and S. Monforte, (2002). Multi-Master Profibus DP Modelling and Worst Case Analysis - based Evaluation. In these proceedings.

Cornilleau T., and E. Gressier-Soudan. (1997). RT-objects based on temporal causal consistency: a new approach for fieldbus systems. Object-Oriented Technology, ECOOP '97, Springer-Verlag, Berlin, Germany, pp.464-7.

Decotignie J.-D. (2002) Wireless Fieldbuses, in this proceedings.

Decotignie J. D. (1999) Some future directions in fieldbus research and development. In Fieldbus Technology, Systems Integration, Networking and Engineering, Proceedings of the fieldbus Conference FeT’99, Springer, pp 308-312.

Demartini C., Iosif R., Raibulet C., and J.P. Thomesse, A DBR based approach for system management. In Fieldbus Technology, Systems Integration, Networking and Engineering, Proceedings of the fieldbus Conference FeT’99, Springer, pp 437-444.

Diedrich C. (1999) Profiles for Fieldbuses – Scope and Description Technologies. In Fieldbus Technology, Systems Integration, Networking and Engineering, Proceedings of the fieldbus Conference FeT’99, Springer, pp 90-97.

Fantoni D. (1999). A never-ending story: the Fieldbus standardisation. Automazione-e-Strumentazione. vol.47, no.11, p.69-75.

Juanole G. (2002). Quality of Service of Local Area Networks and Distributed Automation : Models and Performances, in this proceedings.

Kastner W., Csebits C., and M. Mayer. (1999). Linux in factory automation? Internet controlling of fieldbus systems! 7th IEEE International Conference on Emerging Technologies and Factory Automation.

Proceedings ETFA '99 Fuertes J. M. Ed, IEEE, Piscataway, NJ, USA; 1999; pp.27-31 vol.1.

Navet N. (1999) Evaluation de performances temporelles et optimisation de l'ordonnancement de tâches et de messages. Ph. D. Thesis, Nov 1999, INPL, Nancy, France.

Neumann P. (2002) Engineering of Field Devices using Device Descriptions, in this proceedings.

Neumann P. (1999). Locally distributed automation-but with which fieldbus system? Assembly-Automation. vol.19, no.4, pp.308-12.

NOAH (1999) Network Oriented Application Harmonisation, European Project N°26951, Special session, In Proceedings of the fieldbus Conference FeT'99, Springer, pp 413-452. .

Simonot-Lion F., Elloy J.-P. (2002) An Architecture Description Language supporting Development Process of In-Vehicle Embedded Systems. In these proceedings.

Simonot-Lion F. (1999). Une contribution à la modélisation et à la validation d'architectures temps réel, HDR Thesis, INPL, Nancy, France.

Staroswiecki M., (1996). Models and languages for the interoperability of smarts instruments. Automatica, Vol 32, N°6, pp 859-873.

Thomesse J.-P. (1998). A review of the fieldbuses. Annual reviews in Control 22, pp35-45.

Thomesse J.-P. (1999). Fieldbuses and interoperability. Control Engineering Practice 7, Pergamon, pp 81-94.

UIML UIML website: http://www.uiml.org