Formal description of interoperability

PETR FIEDLER
Department of Control and Instrumentation
Faculty of Electrical Engineering and Communication, Brno University of Technology
Kolejní 4, 612 00 Brno
CZECH REPUBLIC

Abstract: - This paper describes theoretical and formal background for analysis of interoperability issues in autonomous machine based data exchange systems, especially in automation systems. The formalism is based on description of interaction of simple deterministic automatons. The interoperability issues are shown on examples; examples are formally described and classified into four classes.

Key-Words: - Interoperability, data exchange, automation, formal description, interoperability classification

1 Introduction

The Open Group Consortia [1] defines interoperability as follows:
"Interoperability is defined as the ability to both exchange information and to use it."
Such simple definition perfectly describes the crucial fact that exchange of information without proper understanding of meaning of the information is useless. In heterogeneous systems the major effort is focused to exchange of information. However the proper understanding is as important as the exchange itself.

The basic elements that take part in modern automation of processes and technologies are control algorithms, devices and communication interfaces. The control algorithms represent the Automation theory, while the devices and communication interfaces represent the Automation infrastructure. Both theory and infrastructure have to operate in harmony to provide state-of-the-art services. However implementation of control algorithms and design of communication interfaces is vendor depended. It is obvious that the quality of implementation of control algorithms is directly related to optimality of control. However the quality of implementation communication interfaces is not only related to the ability to exchange data, it is related to optimality of control as well. In modern automation applications the quality of data exchange is as important as the control algorithms itself as the optimality of control depends on the quality of input data.

2 Formal description

To formally analyze interoperability issues it is necessary to introduce formalism for description of data exchange among heterogeneous systems. For a convenience we adopt well known formalism used for description of simple deterministic automatons as it may be found in [2].

Let A and B are heterogeneous systems. Both systems A and B are used to represent portions of reality $\rho \in R^{\infty}$. To describe data exchange between A and B, we use three notions specific for each system, the alphabets (Σ^A and Σ^B containing elements used to describe the reality, representation spaces (Q^A and Q^B) containing representations (q^A and q^B) of states of objects, and Universes of Discourse (U^A and U^B) defining the semantics of the objects.

The behavior of a system is described by evolution of states of Q. Object being represented is regarded as a state $q \in Q$.

Definition 1: An automaton α is defined by a set Q of states, by an alphabet Σ (set of all allowed input symbols), alphabet Δ (set of all allowed output symbols) and by a transition function δ. For a deterministic automaton holds that on reception of input $i \in \Sigma$, the automaton shifts from state q into (new, possibly same) state $r \in Q$.

Definition 2: Let Ω be a set of all strings over the alphabetΣ, let Ψ be a set of all strings over the alphabet Δ. Ω (Ψ) and every subset of Ω (Ψ) are called formal languages over Σ (Δ).

Definition 3: Transition function δ, is a function which maps $Q \times \Omega$ onto some subset of Q.

$$\delta:\ Q \times \Omega \rightarrow Q_{SUB} \subseteq Q$$

The transition function is used to derive the effect on the deterministic automaton due to any input string $s \in \Omega$, e.g. $r = \delta(q, i)$.

Let there is a function ξ that maps the state of the reality R^{∞} to the universe of discourse U of the automaton, ξ: $R^{\infty} \rightarrow U$. Let there is a function φ that maps the universe of discourse U into inputs of the automaton:

$$\varphi: U \rightarrow \Omega.$$

The actual state of the reality is at the automaton input continuously reflected as state of the input symbols i.

$$\varphi(\xi): R^{\infty} \rightarrow U \rightarrow \Omega$$

In response to an input symbol $i \in \Omega$ the automaton generates an output symbol o from an alphabet Δ. The generated output symbol(s) depend on both i and actual state of the automaton before reception of i. The output behavior of α is defined by the output function λ: $Q \times \Sigma \rightarrow \Delta$. It is obvious that to bring α from state q_0 into an arbitrary state $q_N \in Q$, it is necessary to find a string $s \in \Omega$ such that $\delta(q_0, s) = q_N$.

Each of the heterogeneous systems A and B is regarded as an independent automaton, hence denoted α^A and α^B, the automaton elements are distinguished by superscripts A and B respectively.

When the function $\varphi^A(\xi^A(\rho_j))$ produces string i^A, this new input string brings a system A from a state q^A_{j-1} to a state $q^A_j = \delta(q^A_{j-1}, i^A)$. The output function λ^A causes the A to output a string $t \in \Psi^A$, $t = \lambda^A(q^A_{j-1}, i^A)$. Hence the string t contains information on both the state q^A_{j-1} and inputs i^A. Moreover it serves as an input string for system B.

The arrival of t causes the B to change its state according to transition function δ^B producing new state of B denoted as r^B_k.

The data exchange was successful iff:

$$r^B_k \equiv r^{B*}_k: r^{B*}_k = \delta^B(r^B_{k-1}, \varphi^B(\xi^B(\rho_j)), \xi^B(\rho_j)) \neq \varnothing$$

In other words, the data transmission between systems A and B was successful if and only if the information gained from the data transmission by system B from the system A is equivalent to information that would the system B obtain by direct observation of the particular portion of reality $\rho_j \in R^{\infty}$ using its own universe of discourse U^B, if the U^B would reflect the ρ_j.

3 Four interoperability classes

Interoperability issues are experienced when heterogeneous systems have to exchange data. For successful communication between two computational agents (devices) each device must understand what the other says. It shows that agreement on syntax and semantics (language they communicate) only, does not guarantee satisfactory results. A common ontology is at least as important as the syntax and semantics.

"…, the reality is that various agents can and often do use different terms to denote elements in a common domain, and this presents a pervasive problem: Words that are not in one agent's ontology will be completely unintelligible when presented by another agent, even if they have agreed on a common language of communication (an *interlingua*) ahead of time, and even if their ontologies are similar, even significantly overlapping." [3]. In the past vast effort has been spent on formal verification of various communication protocols, however the major focus was on verification of syntactical, semantical and timing correctness.

Communication at all levels of ISO/OSI model is performed as exchange of symbols. Syntax concerns the relation that symbols have among themselves and the ways in which they can be manipulated. Semantics concerns the relation between symbols, on one hand, and the things the symbols mean, on the other [5].

However to resolve interoperability issues, it is necessary to became aware that communication is something more than simple exchange of data based on semantical and syntactical rules and relations .

Inspiring analysis of interoperability issues among CAD/CAM systems was done by Vergeest and Horvath [4], however results of their research, after small modifications, are applicable to all heterogeneous systems. According to the Vergeest and Horvath the interoperability issues can be classified into four types (classes). After we extend the original classes to a more abstract level, we can apply this classification of interoperability issues to almost any field of study, where interoperability issues in communication occur. Abstract description of the four classes follows:

3.1 Class I

The first class is a syntactical issue, which occur when the same object is coded using different languages. When the information content of two objects is exactly equal a simple *1:1* mapping can be applied to solve the issue.

The objects are described by data regarded as strings. The first object, object q_A, is described using string s of symbols chosen from language Σ^A. The second object, object q_B, is described using string s' of symbols chosen from language Σ^B. Although the information content of both objects is the same, it holds that $s \neq s'$.

Interoperability issues of Class I represent the most simple challenge when dealing with interoperability issues as it is always possible to find not only unambiguous function f_I that performs mapping from Σ^A to Σ^B, f_I: $\Sigma^A \rightarrow \Sigma^B$, but also an inverse function g_I: $\Sigma^B \rightarrow \Sigma^A$. Solution of such issues is in some cases straightforward and easy. However issues of Class I may lead to issues of Class IV in case that needed dynamic range of symbols of Σ^A does not meet dynamic range of symbols of Σ^B and vice versa.

The issues of Class I can be simply described as issues caused by the fact that identical object is described by different languages. Solution of such issue requires an interpreter who is able to convert between the two languages.

Example of Class I issues is a Boolean value, that can be stored as *single bit* or as *one or more bytes*.

3.2 Class II

The second class issues occur, if we have to face alternative representation of equivalent objects.

Let us suppose that system A uses for description of object q different methodology than system B. Even if the systems A and B use same language Σ for description of q, the description made by A, denoted as s^A, is not usable in system B and vice versa.

Usually it is possible to find a conversion function $f_{II} : f_{II}(s^A) = s^B$ and an inverse function $g_{II} : g_{II}(s^B) = s^A$. However, the existence of f_{II} does not guarantee existence of g_{II}. In other words conversion in one direction might be feasible, whereas the unambiguous conversion in the reverse direction might be not.

Class II issues can be regarded as “Alternative representation of identical objects”. Symbols used for heterogeneous representation of an object in two heterogeneous systems may lead to loseless or lossy conversion functions.

Exapmle of loseless conversion function may be conversion of temperature from Centigrade system to Fahrenheit scale. On the other hand conversion from CMYK to RGB color space is representing lossy conversion function.

3.3 Class III

The Class III issues occur when objects with identical static properties act differently under operation. Usually the main challenge when facing issues of this class is the discovery of the reasons for the unexpected behavior. Typical reason is that incomplete description of states of q in system A is used. It follows that the description of q available to B is not sufficient to sufficiently reconstruct all relevant properties of q and vice versa.

Example of Class III issue may be linear description (approximation) of nonlinear dynamical system.

3.3 Class IV

The Class IV issues occur when objects with incompatible static properties have to be represented in the other system. Such issues occur, when language Σ^A of the system A, does not contain string s that could be used to represent object q, which is in the system B represented by string s' of language Σ^A.

Example of such issue is an attempt to convert device status information between LonWorks technology device and Profibus PA device.

4 Real-world experience

Important interoperability issues have to be faced when interconnecting high-level fieldbuses. It is difficult to achieve flawless interpretation of syntactically more complex data structures that do not have exactly same nor similar counterparts in the semantics of the other protocol.

Gateways are good choice for interconnection between high-level fieldbus and low-level fieldbuses (e.g. Profibus - AS-Interface gateway), where the data representation space of the low-level network is a subset of the more complex data representation space of the high-level network. In such heterogeneous environments there are no attempts to represent the complex data structures of the higher-level network using data structures available on the low-level network, the high-level network operates as a `master' and the low level network operates as a `slave'.

The experience from development of various gateways shows that the interoperability issues concerning interpretation of data between high-level networks are caused by incompatible philosophies used for representation of real-world data (incompatible device profiles) and contradictory philosophies used for transmission of data (pooling vs. event driven transmissions). Moreover in many fieldbus architectures only `Master' has access to the information provided by slaves, which results in the need to combine in one device functionality of the gateway and network master. This makes the gateways very complex devices. On the other hand some network architectures (e.g. LonWorks) do not allow for unlimited of access to all data available on the network. Thus it is not possible to create a gateway that would be capable of unlimited access to all data.

As every philosophy used is suitable for description of process data it is almost always possible to create tailored interpretation algorithm and gateway that will serve for the particular application. However development of universal, generally applicable gateway capable of interpretation of all available data can be impossible due to the complexity of experienced issues. Development of gateway needs either cognitive abilities and deep understanding of the protocols and/or detailed knowledge of meaning of the process data.

5 Conclusion

Interoperability issues in automation are caused by use of heterogeneous systems with heterogeneous data exchange philosophies. It has been shown that 'communication' stands for exchange of information, which consists of both exchange of data (syntax, semantics) and proper understanding of the data (ontology). However, the interpretation of information for successful conversion of language symbols among heterogeneous system is just one issue that has to be faced when dealing with interoperability. Timing issues may also lead to unsolvable problems.

Presented analysis of general interoperability issues and above described method for their formal description forms a background for description of interoperability issues. Formal and abstract description of the issues will enable uniform and systematic analysis of interoperability issues at all levels of ISO/OSI communication model. Such analysis is crucial for successful design of gateways among heterogeneous systems.

6 Acknowledgment

Author of the paper would like to express his thanks to the Grant Agency of the Czech Republic (GACR) for their kind support of this research under the grants GACR GA102/03/1097, GA102/05/0663 and Research Intent MSM 0021630503.

References:

[1] http://www.opengroup.org/

[2] Hopcroft J.E., Motwani R., Ullman J.D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Publishing Company, 1979.

[3] Campbell A.E., Shapiro S.C, *Algorithms for Ontological Mediation,* In: Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems, Montreal, 1998.

[4] Vergeest J.S.M., Horváth I., *A fundamental limit of interoperability*, Symposium on Product Data Technology, Quality Marketing Services, pp. 195-202, 25. April 2001

[5] Rapaport J.W., *Understanding of understanding: Syntactics semantics and computational cognition*. In: James E. Tomberlin, editor, AI, Connectionism, and Philosophical Psychology, vol. 9 of Philosophical Perspectives, pp 49-88. Ridgeview, CA, U.S.A, 1995.