Formal Description Technique SDL for manufacturing systems specification and description

Sousa, R. M.; Putnik G. D.
Production and Systems Engineering Department
University of Minho, School of Engineering
4800 Guimarães, PORTUGAL
Tel: ++351 (0)53 510278, Fax: ++351 (0)53 510268
Email: rms@eng.uminho.pt
Email: putnikgd@eng.uminho.pt

Abstract

This paper addresses the formal specification and description of manufacturing systems. It is considered the use of SDL (Specification and Description Language), a standard FDT (Formal Description Technique), to model the behaviour, data and structure aspects of a manufacturing system. SDL was originally developed for telecommunication systems (protocol specification and data processing). The adequacy of FDTs, namely SDL, for the manufacturing systems domain is investigated by developing the SDL specification of part of a Distributed/Virtual Manufacturing System cell installation (D/V MS Project), and analysing it.

Keywords

Manufacturing Systems, Formal Specification, Formal Description Techniques, SDL

INTRODUCTION

A manufacturing system can be considered as a set of several different, but interrelated, components. These correspond to different aspects that must be considered in order to fulfil the system specification. Each one of these aspects (e.g. physical layout, electrical installation, system behaviour, data architecture, etc) must be modelled, and that is accomplished with probably different languages/techniques, each one of them appropriated to the corresponding aspect. One problem is that the specification/description of almost all components of a manufacturing system, is usually given in natural language or using informal diagrams. These sorts of informal specifications have a strong tendency to conduct to implementations with errors, omissions and incompatibilities. Other problem is the lack of a methodology that enables the complete formal specification of production systems, regarding all the system components. Obviously this is a complex problem and its solution implies the use of FDTs for manufacturing systems. Very few examples (Lewerentz and Lindner, 1995) were found in literature about application of formal methods to the manufacturing systems area. This paper addresses the utilisation of one FDT (Formal Description Technique) for rigorous description/specification of behaviour, data and structure aspects of a manufacturing system. FDT is the common designation of three standard languages: ESTELLE (Extended Finite State Machine Language) (ISO, 1989e), LOTOS (Language of Temporal Ordering Specification) (ISO, 1989l) and SDL (Specification and Description Language) (CCITT, 1988). Accordingly to (Turner, 1993) FDTs were developed to ensure unambiguous, clear and concise specifications, and completeness, consistency and tractability, in order to achieve conformance of implementations to specifications. All these FDTs must be considered as potential candidates to the formal specification of manufacturing systems. ESTELLE was already object of study (Sousa and Putnik, 1998). The present paper focuses on SDL. The paper starts with an overview about the foundations of FDTs. The basic concepts of SDL are presented. FDT SDL is then applied to formally describe part of the Distributed/Virtual Manufacturing Systems Project (D/V MS Project) (Putnik, 1998) (Figure 4).

FORMAL DESCRIPTION TECHNIQUES

In 1988 and 1989 ISO (International Organisation for Standardisation) and CCITT (International Consultative Committee on Telegraphy and Telephony, now ITU-T International Telecommunications Union - Telecommunication Standardisation) standardised three formal languages: ESTELLE, LOTOS and SDL. These Formal Description Techniques (FDTs) were specially developed for the area of telecommunication systems (protocol specification and data processing). The ISO FDT group based the creation of FDTs on two categories of approaches: Finite

State Automata and Algebraic Ideas. ESTELLE was standardised within the first category, and uses the concept of extended finite state machine (EFSM). The EFSM concept allows ESTELLE to deal with a problem of real systems - state space explosion. LOTOS belongs to the second category of approaches and provides means to deal with two different aspects: system behaviour and abstract data typing. System behaviour is modelled using process algebras: Calculus of Communicating Systems (CCS) (Milner, 1989) and Communicating Sequential Process (CSP) (Hoare, 1985). Abstract data typing is based on ACT ONE (Ehrig and Mahr, 1985), an abstract data type language. In collaboration with ISO, CCITT has standardised SDL that includes characteristics of both ESTELLE and LOTOS, namely the EFSM concept and abstract data types (ADT) respectively.

The reason why FDTs emerged is also applicable to other kind of systems (aeronautics, robotics, production systems, etc) - when systems complexity increases only formal approaches allow proper description/specification, verification and implementation (Turner, 1993).

SDL BASIC CONCEPTS

SDL is a formal object oriented language with formal definition of the semantics (SDL 96 is the latest version). SDL provides a hierarchical form of describing systems allowing different levels of abstraction. The system is seen as a set of blocks connected via channels, providing a general overview of the system structure. Each block can then be refined in subblocks or directly in processes. Each process represents an Extended Finite State Machine (EFSM). So at this level SDL describes a system as a set of communicating extended finite state machines (CEFSMs). Communication between processes (CEFSMs), and also between system and the environment is done exchanging messages. Messages are modelled using the signal construct. If necessary a signal can carry parameters.

SDL provides two kinds of representation: Phrase Representation (PR) and Graphical Representation (GR). The automatic translation between GR and PR (and vice versa) is possible. The next example introduces the GR form (Figure 1)

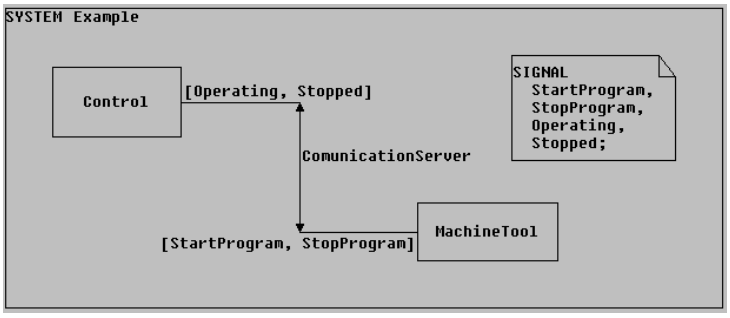

Figure 1 SDL specification at system level in GR form.

This diagram represents the system structure and uses the following constructs:

- **system** Example;
- **block references** Control and MachineTool;
- **channel** CommunicationServer;
- **signals** StartProgram, StopProgram, Operating and Stopped;
- **text symbol** signal definition.

In the next abstraction level each one of the blocks is described. The following diagram (Figure 2) shows the internal structure of the MachineTool block.

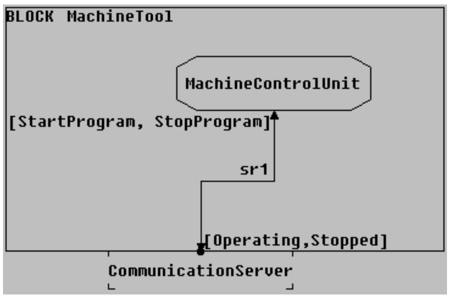

Figure 2 SDL specification of the Control block.

This block contains only the process MachineControlUnit that communicates with the block environment via **signal route** sr1. In the lowest level of the specification processes are described as extended finite state machines (EFSMs). In this simple case the MachineControlUnit process is described (Figure 3) by a regular FSM.

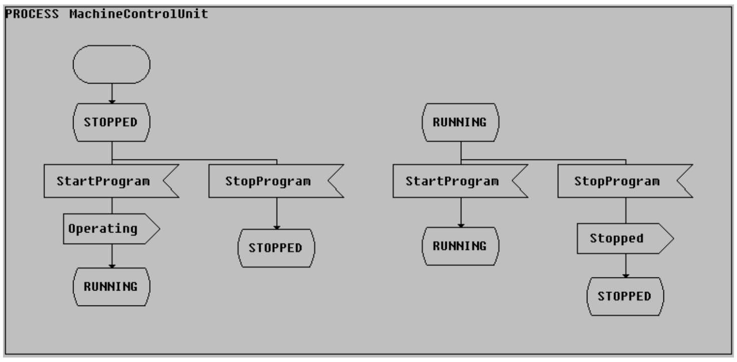

Figure 3 SDL specification of the MachineControlUnit process.

This notation resembles flowcharts and is straightforward. For instance, if the FSM is in the state STOPPED and a StartProgram message arrives, then an Operating output message is sent and the FSM jumps to the state RUNNING. The diagrams of this example were produced with a shareware demo version SDLite V1.0 SDL Graphical Editor for Windows, a tool from Verilog SA.

THE D/V MANUFACTURING SYSTEMS PROJECT

The Distributed/Virtual Manufacturing Systems project (D/V MS project) (Putnik, 1998) is running at the Production and Systems Engineering Department, University of Minho. It investigates the design, and control, of distributed and virtual manufacturing systems. Associated to this project the following installation (Figure 4) was implemented in one of the department's laboratory.

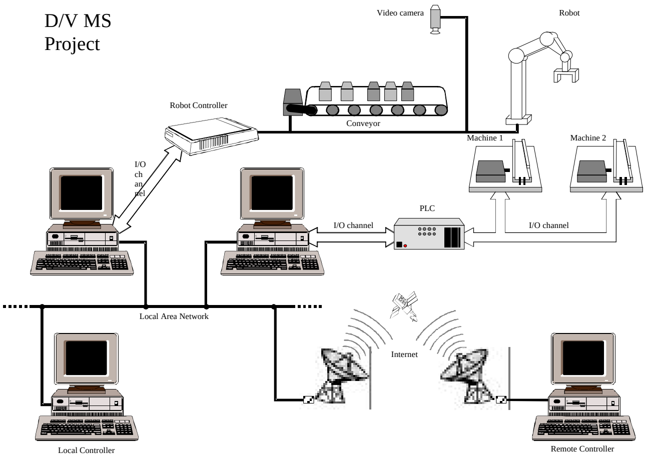

Figure 4 Implemented distributed manufacturing system prototype.

The installation has two cells. The robot cell includes a vision system allowing part recognition, a robot Scorbot ER VII and a conveyor. The machine cell has two simple machines controlled by a programmable logic controller (PLC). Each machine receives parts fed by the robot and simulates an operation by a given operation time. The local controller provides this timeout accordingly to the kind of part. In the early stages of this project the informal specification of messages between system components was based on tables like the following one (Table 1).

Table 1 Informal specification of communication from PLC to PC

Message	*Information*	*Code*
1	Part waiting on machine 1	0000
2	Part waiting on machine 2	0001
…	…	…
10	Operation concluded by machine 1	1001

The code column is clearly implementation oriented and is not necessary at this specification level. This superfluous information will eliminate other valid implementations. To avoid this and other problems referred ahead, the formal SDL description of the system was developed (Figure 5).

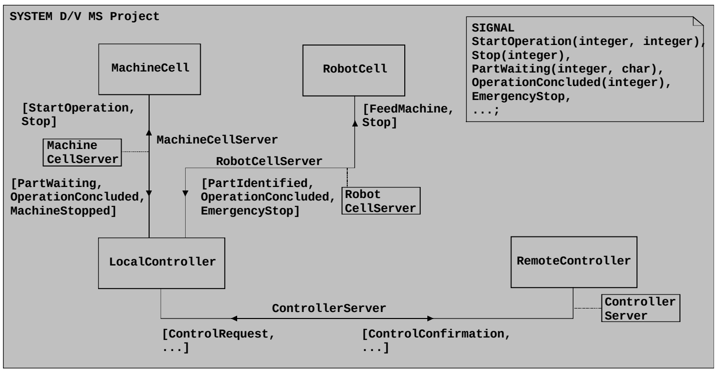

Figure 5 SDL formal description of distributed manufacturing system prototype.

Obviously the entire specification would be very large so only the block LocalController is treated here. This block contains a single process named Controller with 3 signal routes allowing the communication with the 3 channels in the environment of the block. The following diagram (Figure 6) represents part of the formal specification of this process.

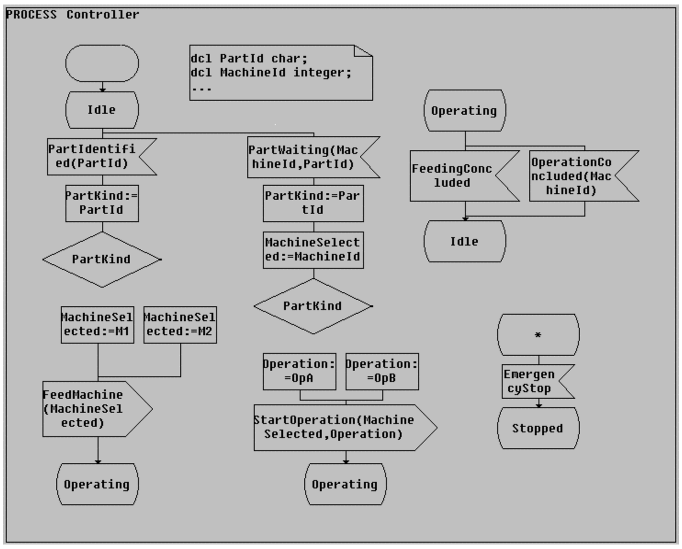

Figure 6 Part of SDL formal specification of Controller process.

Contrarily to the informal specification (Table 1) here unnecessary information about the messages is not present. The messages PartWaiting and OperationConcluded (and all the others too) are specified in a high abstraction level defining only the parameters that they should carry (and obviously the name of the message). Thus all possible implementations can be considered as potential candidates to implement the desired system.

Another problem avoided by formal specification is related to the completeness issue. Often the behaviour of a FSM is described using state diagrams, but these informal diagrams do not specify how the FSM should behave if in a given state arrives an unexpected message. An SDL specification states that unexpected messages are consumed and there is no state change on the FSM. This means that system will ignore those messages. With informal specifications the system behaviour in those situations can be unpredictable.

CONCLUDING REMARKS

The use of formal approaches for system design avoids several problems. The high abstraction level eliminates unnecessarily implementation oriented aspects, that restricts the number of valid implementations. The behaviour of the system under development is specified for all input circumstances, including those usually not considered by informal specifications (e.g. unexpected messages in a given state). The lack of this specification completeness characteristic is one of the main problems of informal approaches leading to unpredictable behaviour if unusual conditions occur. In this context probably the main advantage of FDTs is the possibility of design validation before the implementation phase.

Using adequate tools is possible to produce clear, unambiguous and complete SDL specifications. Even if the development process has already started, the use of SDL (or other FDT) will detect deficiencies before they cause problems. The demo tool used to draw the SDL diagrams in this paper belongs to the tool environment ObjectGeode that uses SDL for performing architectural and detailed design of real time systems. It was possible to conclude that SDL can be used to specify/describe at least some kind of manufacturing systems.

REFERENCES

CCITT (1988) CCITT Z.100: Specification and Description Language. International Consultative Committee on Telegraphy and Telephony, Geneva.

Ehrig, H. and Mahr, B. (1985) Fundamentals of Algebraic Specification, in Monographs on Theoretical Computer Science, 6, Springer-Verlag, Berlin.

Hoare, C.A.R. (1985) Communicating Sequential Processes. Prentice-Hall International, Englewood Cliffs, New Jersey.

ISO (1989e) ISO/IEC 9074: Information Processing Systems - Open Systems Interconnection - ESTELLE - A Formal Description Technique based on an

Extended State Transition Model. International Organisation for Standardisation, Geneva.

ISO (1989l) ISO/IEC 8807: Information Processing Systems - Open Systems Interconnection - LOTOS - A Formal Description Technique based on the Temporal Ordering of Observational Behaviour. International Organisation for Standardisation, Geneva.

Lewerentz, C. and Lindner T. (1995) Formal Development of Reactive Systems - Case Study Production Cell. Springer-Verlag.

Millner, A.J.R.G. (1989) Communication and Concurrency. Addison-Wesley Reading, Massachusetts.

Putnik, G.D.; Sousa, R.M.; Moreira, J.F.; Carvalho, J.D.; Spasic, Z. and Babic, B. (1998) Distributed/Virtual Manufacturing Cell: An Experimental Installation, in 4th International Seminar on Intelligent Manufacturing Systems, Belgrade, Yugoslavia.

Sousa, R.M.; Putnik, G.D.; Moreira, J.F. (1998) Using Formal Description Technique Estelle for Manufacturing Systems Specification or Description, in 4th International Seminar on Intelligent Manufacturing Systems, Belgrade, Yugoslavia.

Turner, K.J. (1993) Using Formal Description Techniques - An Introduction to ESTELLE, LOTOS and SDL. John Wiley & Sons.

BIOGRAPHY

Rui Manuel Alves da Silva e Sousa received his diploma in Electrical Engineering from the Faculty of Science and Technology of the University of Coimbra, in 1989. From 1990 to 1993 he was assistant lecturer at the Polytechnic Institute of Leiria. In 1996 he received his MsC degree in Systems and Automation, branch of industrial automation, from the University of Coimbra. At present he is assistant lecturer at the Production and Systems Engineering Department of the University of Minho and he is working on his PhD thesis entitled 'Contribution to a formal theory for manufacturing systems'.

Dr. Goran D. Putnik received his Dipl. Eng., M.Sci. and Dr.Sci. from the Belgrade University, both M.Sci and Dr.Sci. in domain of Intelligent Manufacturing Systems. His current position is Associated Professor in the Department of Production and Systems Engineering, University of Minho, Portugal, for the subjects CAD/CAPP, CAM, Intelligent Systems for Manufacturing and Design Theory. He is also the Director of the Master and Postgraduate Course on Computer Integrated Manufacturing (CIM) and the Director of the Centre for Production Systems Engineering (CESP) of the University of Minho. His interests are manufacturing system design and control, theory and implementations, and machine learning.