Specifying and Verifying the Steam Boiler Control System with Time Extended LOTOS — Extended Abstract

Andreas Willig[1] and Ina Schieferdecker[2]

[1] Technical University Berlin
tel.: (030) 314 23 831
e-mail: Willig@ftsu00.ee.tu-berlin.de
[2] GMD Fokus
tel: (030) 254 99 241
e-mail: ina@fokus.gmd.de

Introduction

The paper considers the specification of the steam boiler in TE-LOTOS — Time Extended LOTOS [?]. LOTOS [?] has successfully used for the description of communication protocols. Its formal semantics is the basis for a well-founded design methodology ranging from abstract to concrete specifications via stepwise refinement, offering means to formal reasoning, prototyping, and testing [?]. With the introduction of a time concept into LOTOS, it becomes also applicable to the design of real-time systems. This paper presents one attempt to gather practical experiences for this new application area of LOTOS. The specification of a real-time system comprises of

1. the system architecture, i.e. its structuring into components and subsystems with their interfaces and the relation between them and to the environment,
2. the functional behaviour of the components and subsystems, and
3. the timing constraints that have to be met for a correct and safe system execution.

TE-LOTOS offers several features for the specification of real-time systems:

1. the definition of processes and interaction points (the so-called gates),
2. the parallel or sequential composition of processes including the definition of proper synchronisation between them,
3. the definition of behaviour expression that represent the externally visible and internal behaviour of processes as sequences of actions, and
4. the definition of timing constraints that impose time windows for synchronisation or the passage of time.

We used a series of specifications in order to introduce step by step the constraints on the behaviour of the steam boiler system. Firstly, we considered only the functional behaviour and specified the steam boiler system in LOTOS. This functional specification itself resulted also from a number of refined LOTOS

specifications [3]. The functional design step has been formally verified for its correctness by the use of LOTOS tools. A number of tests, which represent safety conditions for the steam boiler system, were applied to the functional specification.

Afterwards, we introduced the timed behaviour by the use of the timed operators of TE-LOTOS. Finally, we specified the data dependencies of the steam boiler system in the functional data type language Gofer [**?**] and integrated them into the time-extended functional specification. Due to the lack of tools for TE-LOTOS, the latter two steps have been manually checked for their correctness.

The resulting specification in TE-LOTOS with Gofer gives an architectural design for the steam boiler system. It defines *how* the requirements of the steam boiler system are fulfilled, i.e. the requirements are implicitly reflected in the specification. The specification is implementation-near and can be used for building a system prototype. In addition, the requirements are explicitly defined in the tests. The tests define *what* requirements have to be met. The tests are applied to the specification in order to verify the correctness of the specification.

The resulting specification is very modular, succinct, and easy to read. Most importantly, it is an implementation-near specification. It offers means for logical reasoning on the system properties, for simulation and prototyping of the system.

The paper mainly adheres to the three design steps. The functional specification and its verification is considered first. The subsection on verification describes the testing approach that was used to check the correctness of the specification. Afterwards, the time-extended specification is discussed. The data-extended specification is considered at last. Each of this three parts is introduced by a short overview on the used specification features and formalisms. Conclusions identify advantages and drawbacks of TE-LOTOS with Gofer for the specification and verification of real-time systems.

Overview on the Specification

The steam boiler system comprises of a number of physical devices — the steam boiler, a device for measuring the quantity of water in the steam boiler, one pump to provide the steam boiler with water[4], a device for measuring the steam coming out of the steam boiler, a valve to empty the steam boiler —, an operator desk containing the control unit, and a message transmission medium.

In our specification, the functionality of the pump controllers is carried over to the pumps themselves. A pump is able to inform the control unit about its current state and it is able to react properly on the instructions of the control

[3] LOTOS supports strongly functional refinement, although that is not explicitly shown here.

[4] We constrained the specification to contain only one pump instead of four pumps in order to restrict the state space and to make the specification automatically analysable with the LOTOS tools within an acceptable amount of time.

unit. Of course, it is a philosophical question whether to separate the mechanical concerns (the pumps) from the controlling concerns (the pump controllers). Since all other devices such as valve and measurers are also considered as being a device without separate controller, we decided to model the pumps in the same way.

In addition, we assume that device failure messages are received by the operating personnel, which could execute repair actions if needed.

The controller communicates with the physical devices by the use of messages via a transmission medium. The transmission medium connects each physical device with the controller. The transmission time is neglected. The controller works periodically with each cycle consisting of:

1. receiving messages from the physical devices,
2. determining appropriate reaction according to the received information, and
3. transmitting messages to the physical devices.

The specification structure is represented in Fig. 1. The external interface of the steam boiler system, which are LOTOS gates written in italics, represents the possible inputs and outputs at the physical system and at the operators desk. The specification contains several processes that are appropriate to the system structure:

1. the physical system (`PS`) that consists of one process for each physical device, i.e. the valve, the pumps, the steam boiler, and the measurers for steam and water, and of two control processes for getting the mode of the control unit and for handling the device failure messages,
2. the transmission medium (`TM`) that consists of two unidirectional unreliable transmission lines for transferring messages from the physical system to the control unit and vice versa, and
3. the control unit (`CU`) that consists of three scanning processes to handle incoming messages from the transmission medium and from outside and of the automaton with separate processes for each mode.

The system is represented by the parallel composition in Fig 2. The physical system and the control unit are defined to be independent of each other, except of their common communication over the transmission medium.

Conclusions

TE-LOTOS is an expressive formalism for the specification of hybrid real-time systems. The steam boiler specification is concise and well understandable. The modular specification is well supported by TE-LOTOS with several structuring mechanisms such as process hierarchies and the definition of action visibility.

TE-LOTOS allows us to specify a system on different abstraction levels and the use of different specification styles. Several methods for formal reasoning exist, from which we selected the testing method to verify the correctness of the steam boiler specification.

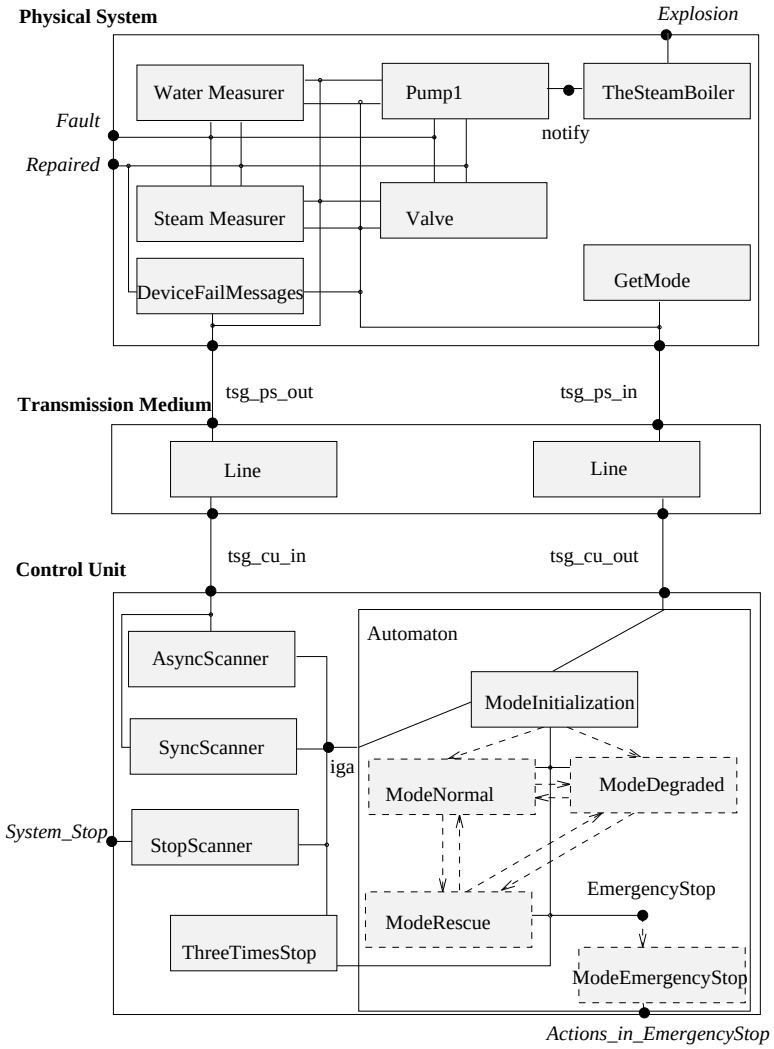

Fig. 1. The specification structure

Besides the advantages of TE-LOTOS, we identified some drawbacks in the features of TE-LOTOS. Most importantly, the lack of tools for TE-LOTOS makes it currently inapplicable to real problems, although their development is currently under work.

Last but not least, we must admit that the learning curve both for developing and understanding a TE-LOTOS specification is relatively large. But this is acceptable when compared with the advantages of TE-LOTOS.

```
specification SteamBoiler_System
[Fault,Repaired,ShowEmergencyStop,Actions_In_EmergencyStop,System_Stop,Explosion]:
noexit
...
behaviour
hide tsg_ps_in,tsg_ps_out,tsg_cu_in,tsg_cu_out in
 ( PS [tsg_ps_in, tsg_ps_out,Fault,Repaired,ShowEmergencyStop,Explosion]
   |||
   CU [tsg_cu_in,tsg_cu_out,Actions_In_EmergencyStop,System_stop] )
 |[tsg_ps_in,tsg_ps_out,tsg_cu_in,tsg_cu_out]|                          10
 TM[tsg_ps_in,tsg_ps_out,tsg_cu_in,tsg_cu_out]
where ... endspec
```

Fig. 2. The System Behaviour

This article was processed using the LaTeX macro package with LLNCS style