# Designing a protocol using knowledge based systems

3H-3

Yao Xue Zhang Norio Shiratori Noguchi Shoichi

Research Institute of Electrical Communication, Tohoku University

1. *Introduction*

Increasing variety and complexity of computer networks has demanded faster production of more complex, higher quality, and more efficient communication protocols. However, the present protocol design and development tools are not adequate for meeting the demand, because they create some problems which obstruct increasing productivity of protocol developments. This paper proposes a new knowledge based protocol design paradigm to solve the problems created by the traditional protocol design models.

2. *Problems created by traditional protocol design and development tools*

To reduce the development complexity, traditional protocol design and development tools break the design and development process into several independent and sequential development steps: *specification step*, *validation step*, *implementation step*, *testing step* and *etc.*. Although this so called "life cycle" design and development paradigm reduces the development complexity, it creates some problems which obstruct the further increase of the productivity of protocol development. Some main problems created by the life cycle paradigm can be given as follows.

1) Long development time.
2) Lack of traceability.
3) Intertwining nature of the design and development steps.
4) Dependency to the level of a protocol designer.

3. *A knowledge based paradigm for protocol design and development*

To solve the problems introduced in Section 2, we propose a new protocol design and development model, *a knowledge based paradigm*, for protocol design and development in this Section. **Fig.1** describes the basic conception of the proposed paradigm. The paradigm is mainly supported by an expert system for protocol design and development. By using the knowledge stored in its knowledge base, this expert system produces a deliverable protocol in a waterfall form, which satisfies the design requirements as shown in **Fig.1**. The *experts* of protocol design and development devote their knowledge to construction of the knowledge base of the expert system for protocol designs and developments. The other two modules of the expert system are: a *human interface* and an *inference engine* for executing the knowledge stored in the knowledge base. We propose this inference engine module is provided by an expert system development shell, and do not show it in **Fig.1**.

Structure of the proposed expert system based on the knowledge based paradigm is shown in **Fig.2**. The knowledge base consists of six modules: a *protocol synthesis module*, a *finer specification module*, an *implementation module*,

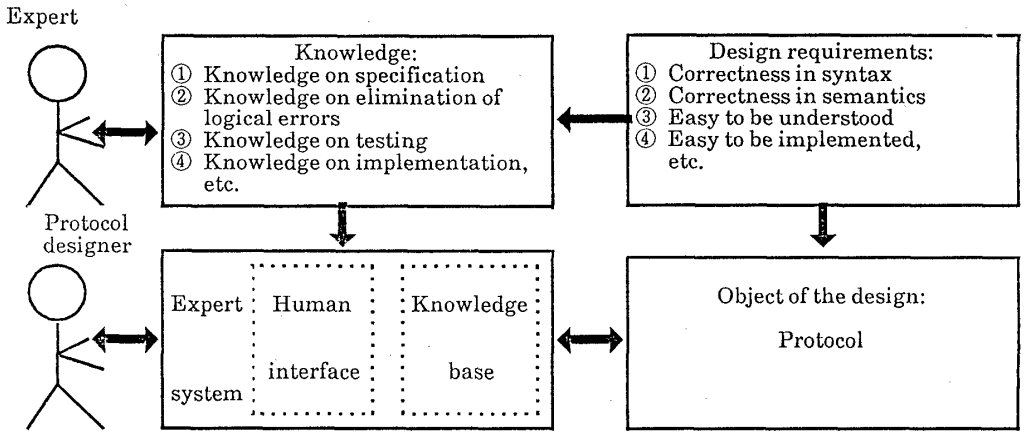


Fig.1 Basic concept of the knowledge based paradigm



a *conformance testing module*, a *protocol conversion module*, and a *regulation module* as shown in **Fig. 2**. These modules have their own subknowledge bases which constitute the

knowledge base of the expert system.

We briefly describe the human interface module and the modules of the knowledge base.

1) *The human interface module*
This module facilitates the design activities by providing a user friendly working environment for the protocol designer to reduce the dependency to the level of the protocol designer.

2) *The protocol synthesis module*
The protocol synthesis module builds up the Finite State Machine (FSM) description of the protocol by using the knowledge for protocol synthesis. The input of this module is the informal description of a protocol to be designed, given by the protocol designer through the human interface. The output of this module is the FSM description of the protocol. These FSM's do not contain the logical errors such as *deadlocks*, *unspecified receptions*.

3) *The finer specification module*
This module creates a high level language such as SDL, ESTELLE or others by using corresponding knowledge. One kinds of input of this module is the FSM descriptions which are the output of the protocol synthesis module. Another kinds of input is the *primitives* which define the syntax and give a detail specification of the semantics of the protocol.

4) *The implementation module*
The implementation module translates the finer specification into an existing programming language such as C or PASCAL, by using the knowledge for automatic implementation.

5) *The conformance testing module*
This module investigates whether the specified protocol provides the asked functions. It derives test sequences from the finer specification of the protocol and outputs tested protocol sequences.

6) *The protocol conversion module*
The protocol conversion module is important in big networks. The protocol conversion module converts a finer specified protocol into a target finer specification. The conversion of the two specifications concerns with the syntax and semantics of the two specification forms.

7) *The regulation module*
This module does the final justify of the protocol. It also has the functions for managing the status of the system. The input of this module is a tested and implemented protocol. The output is the deliverable protocol.

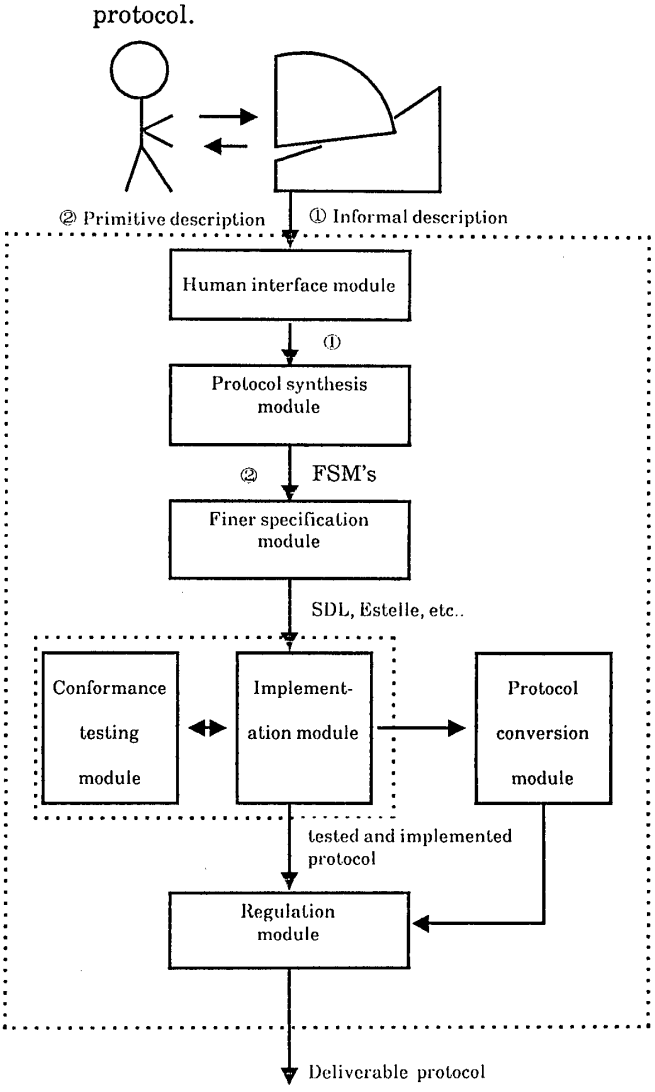


Fig.2 Structure of the expert system for protocol designs

4. *Valuation and conclusion*

The proposed paradigm is used to design a protocol without the problems introduced in Section 2. In this paradigm, some modules such as implementation and testing can be performed in a parallel form, so that the development time becomes shorter. Moreover, dynamical access of the knowledge in each subknowledge base makes the dynamical trace possible and the intertwining nature clear. On the other hand, the human interface reduces the dependency to the level of the protocol designer.

The prototype of the proposed system has been implemented in OPS83 and C with approximate 200 rules and 2000 lines C program. The extension of the prototype is needed to be further researched.

References:

Y.X.Zhang et al., "A knowledge based system for protocol synthesis (KSPS)," IEEE Jour. Selected Areas in Comm., Vol.6, No.5, June 1988.