Reprint

José M. Costa, Ostap Monkewich and Claude Royer, "Use of Formal Description Techniques in the Specification of Baseband Processing for Next Generation Wireless Communication Systems, paper presented at *The International Congress on Dynamic and Control of Systems*, DYCONS'99, 5-7 August 1999, Ottawa, Canada (The Conference Proceedings were to be published by World Scientific).

http://www.control.utoronto.ca/people/profs/dycons99/dycons99.html

USE OF FORMAL DESCRIPTION TECHNIQUES IN THE SPECIFICATION OF BASEBAND PROCESSING FOR NEXT GENERATION WIRELESS COMMUNICATION SYSTEMS

JOSÉ M. COSTA, OSTAP MONKEWICH AND CLAUDE ROYER
Nortel Networks
P.O. Box C-3511, Ottawa, Ontario, CANADA K1Y 4H7
E-mail: *costa@nortelnetworks.com*, *ostapm@nortelnetworks.com*,
croyer2@nortelnetworks.com

1. Introduction

This paper investigates the application of Formal Description Techniques (FDTs) for the specification of the baseband processing for the radio interface in third generation wireless communications systems, such as the standards being developed by the International Telecommunication Union (ITU) for International Mobile Telecommunications – 2000 (IMT-2000). Section 2 refers to the radio systems architecture for IMT-2000 and describes the current approach to the standardization of the radio aspects of IMT-2000, which includes the separation of the radio frequency (RF) processing and the baseband processing. Formal Description Techniques (FDTs) have been standardized in ITU-T and have traditionally been applied to the specification of telecommunication protocols with significant advantages over other techniques, such as the protocols in Layers 2 to 4 of the Open Systems Interconnection (OSI) Reference Model. Section 3 summarizes the key aspects of and the approach in the use of FDTs in telecommunications specifications.

What is new in this paper is the consideration of the application of FDTs in physical layer specifications for radio interfaces. Traditionally, they have been implemented in hardware and are analogue in nature; hence, the application of FDTs does not appear suitable. However, with the advent of digital signal processing techniques in the implementation of digital radiocommunications, FDTs offer particular advantages in the specification of the digital baseband processing for IMT-2000, even if it is implemented in hardware. Section 4 examines these possibilities and the conclusions are summarized in Section 5.

2. Radio System Architecture

Recommendation ITU-R M.1034 (Ref. 1) on Requirements for the Radio Interface(s) for IMT-2000 gives an overview of the radio subsystem for IMT-2000 and provides guidelines for the development of the structure of the radio sub-system for IMT-2000.

The IMT-2000 functional model is defined in Recommendation ITU-R M.1035 (Ref. 2) and subsequent work in the ITU-R has divided the key characteristics of the radio interface into RF characteristics and baseband characteristics. The rationale behind grouping the key characteristics into RF and baseband is to achieve as much commonality as possible in the RF front end, which is where most of the cost resides. For the baseband characteristics, commonality is not as critical an issue but the level of commonality must satisfy market needs.

The baseband characteristics may include: Multiple access technique, Chip rate, Number of time slots in one carrier, Variable spreading factor, Inter base-station asynchronous/synchronous operation, Handover, Channel coding and interleaving, Random access, Modulation, Spreading code, Scrambling code, Pilot structure, Detection, Power control, Variable data rate, Diversity, Adaptive equalizer, and Dynamic channel allocation. It is for the specification of the baseband characteristics that the use of formal description techniques is being investigated in this paper.

3. Formal Description Technique (FDT) Approach

Formal specifications in machine-executable format allow companies to use their facilities for simulation, validation, automatic code generation or other types of automatic processing should they wish to do so. Guidelines on the use of FDTs are contained in ITU-T Recommendation Z.110 (Ref. 5).

The main components involved are formal specifications, computer-aided validation, test generation and testing. The use of FDTs is made practical through commercially available tools. The process is illustrated in Figure 1. Guidance for this process is given in the ITU-T Recommendation A.3.

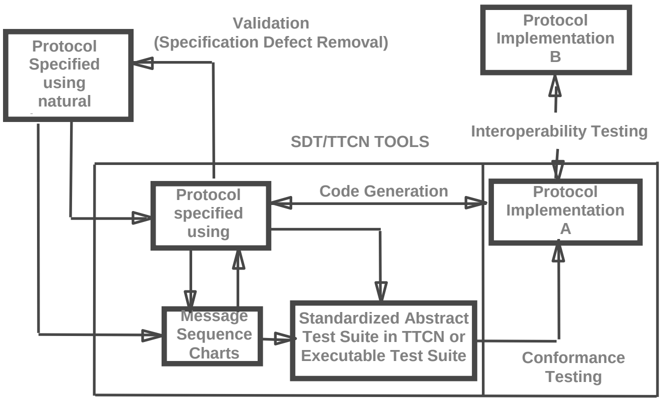

Validation
(Specification Defect Removal)
Protocol Specified using natural
Protocol Implementation B
Interoperability Testing
SDT/TTCN TOOLS
Protocol specified using
Code Generation
Protocol Implementation A
Message Sequence Charts
Standardized Abstract Test Suite in TTCN or Executable Test Suite
Conformance Testing

Figure 1. The formalized process.

An Abstract Test Suite was not produced in the context of this paper; it is mentioned here for the sake of completeness. For each specification, an Abstract Test Suite in the standardized (X.292) Tree and Tabular Combined Notation (TTCN) may be developed in compliance with X.290 series of Recommendations. Each ATS is test platform independent but may be parameterized, compiled and executed to test the implementation for conformance to the standard.

4. Application of FDTs to specifications for baseband processing in radio transmitters and receivers

To illustrate the benefits of the formalized approach, a turbo code module for cdma2000 from baseband processing as specified in the TIA TR-45 standard was chosen to express in FDT. This interleaver is part of a Turbo encoder module which is to be used for low error rate transmission of high speed data over the wireless 3G mobile channel.

The TIA TR-45 natural language description is given in Appendix A. A small sample of the corresponding description in the Specification and Description Language (SDL) is given in Appendix B. SDL is an object-oriented FDT defined in Recommendation Z.100 (Ref. 3). Its graphical form (GR-form) makes possible a very user-friendly design environment. The GR-form is executable and can be easily converted to its textual phrase representation (PR-form). Tools exist to convert either form to C language code, which can be compiled and executed on any computing platform.

SDL is a programming language with features which allow engineering decisions to be made during the specification process. Different design structures, architectural partitioning and modular hierarchy may be tried and, if necessary, changed at a minimal cost. Commercial tools for all major computing platforms provide syntax and semantic checking as well as extensive simulation and testing capability.

The interleaver description in natural language is given in Figure A.1 of Appendix A. Extracts from the corresponding SDL specification are given in Figure B.1 of Appendix B. For comparison, the SDL structure was chosen to resemble the diagram of Figure A.1. The essential difference between the two diagrams is that the diagram of Figure B.1 is executable.

At the higher level of the model, not shown here, the CalcAddress block is interconnected with two other system blocks: Initialize block and SequenceStore block. The Initialize block computes the value of the parameter n for a prticular value of Nturbo, CalcAddress performs the output address calculation while SequenceStore stores blocks of data in sequence and retrieves them according to the calculated addresses.

The blocks are interconnected by channels carrying data as parameters inside signals designated by names inside brackets along side arrow heads indicating direction. The parameters and their types are similar to data and data types in other programming languages.

At the next level of detail, each SDL process is modelled as a Finite State Machine (FSM). Each process describes a behaviour in terms of a set of state transitions. Transition actions include mathematical operations, sending and receiving signals and branching based on Boolean decisions.

In Figure B.2 the process AcceptDiscard is modelled in terms of states named ComputeAddress and ComputeTentative. The circled letters A and D represent connection points to which other pieces of the same SDL process are connected but appear on one or more other pages.

Transitions between states are triggered by the arrival of signals such as nBITS3(nbits3) and 5BITS(5bits) with parameters nbits3 and 5bits. During transition, the model may compute values, make decisions and send signal such as Discard(Addr) and Accept(Addr) via defined channels.

The full specification cannot be given here due to space limitation. However, the SDL specification was shown to be precise and unambiguous specification of the interleaver given in Appendix A. It was validated using Telelogic Tau 3.4 tool set (Ref. 7). The specification was executed using behaviour scenarios as specific paths through the SDL specification. The scenarios were represented using another formal language, the Message Sequence Charts (MSCs). By observing the behaviour under each scenario, it is possible to establish that the specified system has been specified correctly and in accordance with the intended behaviour.

5. Benefits of the Formalized Methodology

From the manufacturer's perspective, the application of the methodology leads to a reduction in product errors, development cost and time to market. From the standardizer's perspective the methodology leads to higher quality standards and improved chances of product interoperability. From the user's and supplier's perspectives, it helps verify product behaviour prior to commitment to costly implementation and leads to rapid agreement with the customer on the correct design specifications.

The specification required about 3 person days to complete using Telelogic Tau 3.4 tools. Simulation and validation required an additional 2 person days. This represents the total effort needed to achieve sufficient confidence to commit the interleaver behaviour to hardware. The benefits are summarized in Table 1.

Table 1 - Benefits of SDL in the Turbo Interleaver Example

Property	SDL Methodology	Current Practice
Time to generate the specification	3 person days	Greater than 3 days
Time to validate/verify the specification	2 person days	Greater than 2 days
Number of errors or ambiguities in the specification	None	3
Implementation C code production	minutes	weeks
Test case traceability to specification	exact	poor
Platform independence	Yes	No

The corresponding numbers to specify the interleaver in natural language were not available but expert opinion indicates that the required effort was considerably greater. Findings from related studies (Ref. 6) support this.

It was found that the interleaver standard is not unambiguous and could be interpreted differently by independent implementors. For this reason, ITU-T has adopted the formalized set of guidelines for its Study Groups to follow in the production of new Recommendations.

6. Conclusions

At the radio baseband level, the SDL methodology with the aid of commercial tools can eliminate ambiguities in the specification. The benefits this approach include: a) the reduced time to develop complex specifications; b) higher quality standards; c) reduced product field faults; d) improved interoperability; e) reduced development cost; f) reduced time to market.

References

1. Recommendation ITU-R M.1034, “Requirements for the radio interface(s) for IMT-2000”
2. Recommendation ITU-R M.1035, “Framework for the radio interface(s) and radio sub-system functionality for IMT-2000.”
3. ITU-T Recommendation Z.100, “Specification and Description Language (SDL).”

4. ITU-T Recommendation Z.105, “Specification and Description Language (SDL) combined with Abstract Syntax Notation One (ASN.1).”
5. ITU-T Recommendation Z.110, “Guidelines on the use of Formal Description Techniques (FDTs).”
6. O. Monkewich, “SDL-based specification and testing strategy for communication network protocols,” SDL Forum’99, Montreal, 21-25 June 1999.
7. Telelogic, "Telelogic Tau Technical Presentation", Telelogic Technical Documentation, Malmo Sweden, June 1998.

Appendix A

Turbo Interleavers

This appendix provides the specification of a turbo code module for cdma2000 given in the TIA TR-45 standard. Input addresses are from 0 to $N_{turbo} - 1$, where N_{turbo} is the number of symbols in the interleaver. The sequence of output addresses is generated as described below.

1. Parameter n is the smallest integer such that $N_{turbo} < 2^{n+5}$.
2. Initialize an (n + 5)-bit counter to 0.
3. Extract the n most significant bits (MSBs) from the counter and add one. Then, discard all except the n least significant bits (LSBs) of this value.
4. Obtain the n-bit output of the table lookup defined in the standard with a read address equal to the five LSBs of the counter
5. Multiply the values in Steps 3 and 4, and discard all except the n LSBs.
6. Bit-reverse the five LSBs of the counter.
7. The tentative output address has its MSBs equal to the value obtained in Step 6 and its LSBs equal to the value obtained in Step 5. Discard if $< N_{turbo}$
8. If less than N_{turbo}, the tentative output address is the output address.
9. Increment the counter and repeat Steps 3 through 8.

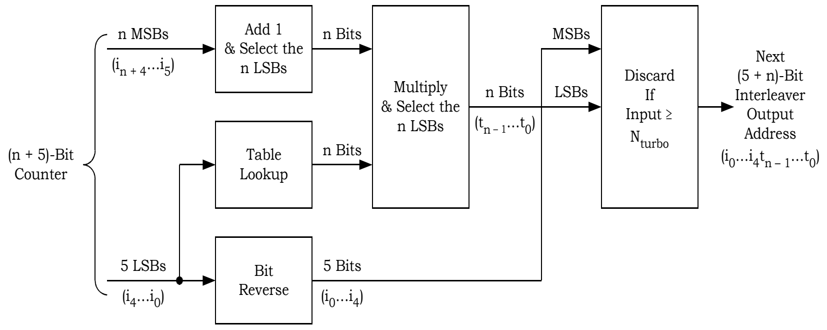
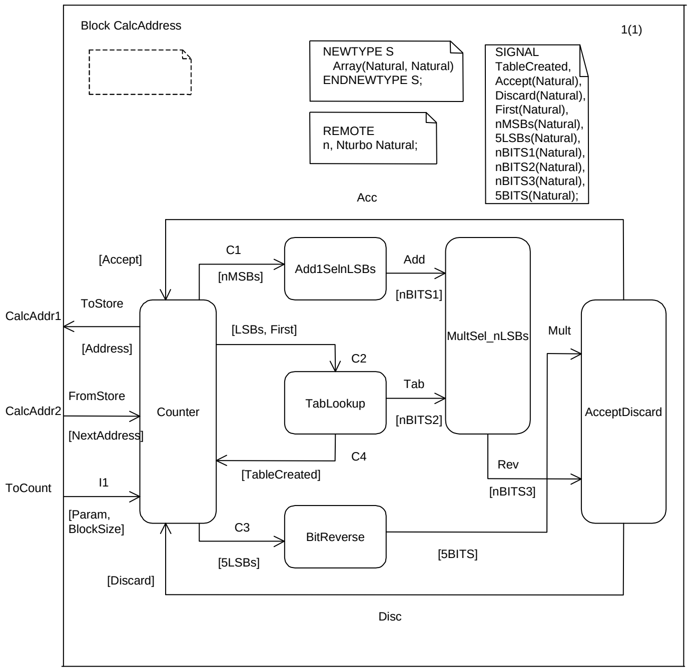

(n + 5)-Bit Counter
n MSBs
$(i_{n+4}\dots i_5)$
5 LSBs
$(i_4\dots i_0)$
Add 1 & Select the n LSBs
Table Lookup
Bit Reverse
n Bits
n Bits
5 Bits
$(i_0\dots i_4)$
Multiply & Select the n LSBs
n Bits
$(t_{n-1}\dots t_0)$
MSBs
LSBs
Discard If Input ≥ N_{turbo}
Next (5 + n)-Bit Interleaver Output Address
$(i_0\dots i_4 t_{n-1}\dots t_0)$

Figure A.1 - Interleaver Address Calculation Procedure

Appendix B

SDL for the Turbo Interleaver



```
NEWTYPE S
  Array(Natural, Natural)
ENDNEWTYPE S;
```

```
REMOTE
n, Nturbo Natural;
```

```
SIGNAL
TableCreated,
Accept(Natural),
Discard(Natural),
First(Natural),
nMSBs(Natural),
5LSBs(Natural),
nBITS1(Natural),
nBITS2(Natural),
nBITS3(Natural),
5BITS(Natural);
```

Figure B.1 – SDL block CalcAddress containing SDL processes.

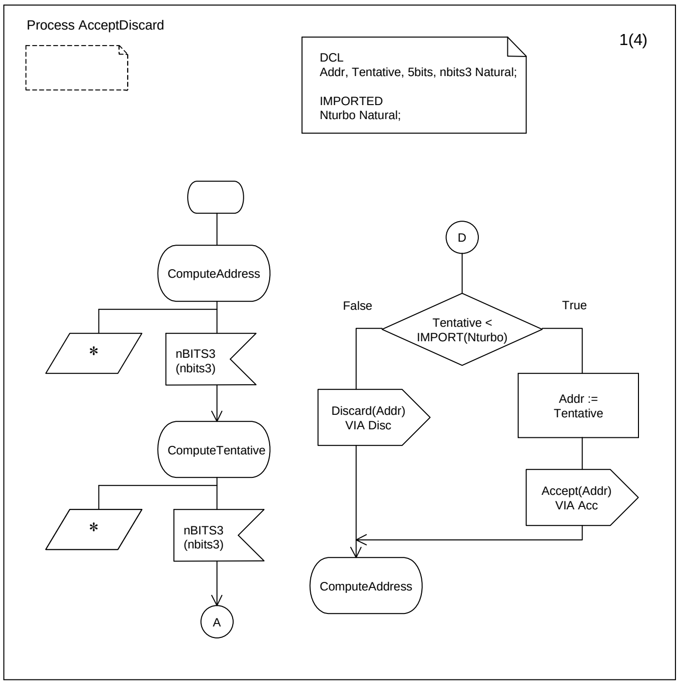


Figure B.2 – SDL process AcceptDiscard at the state transition level.