

# ITU-T SG10 CHILL 전문가 회의개최 보고



이 상 홍 / 한국통신 연구개발원  
기술기준관리실 실장

## 1. SG10 연구활동 개요

SG10은 전기통신용 언어의 국제표준을 연구하는 연구위원회로 연구분야는 크게 MML, SDL, CHILL과 같은 전기통신시스템 운용 또는 개발을 위한 프로그래밍 언어와 전기통신용 소프트웨어의 품질보증 방법 및 프로그래밍 개발환경의 모델링 등이며 93년 3월의 제1차 WTSC회의 이후 10개의 연구과제(Question)를 중심으로 표준화(Z Series)를 추진하고 있다.

이번 회기에는 데이터 지향의 새로운 명세기법의 표준화와 망관리 분야에서의 응용, 소프트웨어 품질 메뉴얼의 구조와 구성에 관한 연구가 주요 표준화대상이 되고 있으며 특히 SDL과 CHILL분야에서는 프로그래밍 언어의 새로운 개념인 객체지향 파라다임의 적용을 통해 보다 강력한 언어표현 능

력구현이 가능한 객체지향 CHILL/SDL의 표준화가 추진되고 있다. 각 연구과제의 추진방법은 기존의 실무위원회(Working Party) 중심의 운영과는 달리 각 연구과제별 라포터를 선정하고 각 라포터의 책임 하에 해당 연구과제의 표준화를 위한 각각의 의견수렴과 전문가회의를 통한 표준초(안)이 개발되도록 하였다.

객체지향 CHILL(Object-Oriented CHILL : OC-CHILL)의 표준화는 연구과제 Q.10으로 선정되었으며, Q.10의 라포터는 독일 Siemens사를 대표하는 Dr. Winkler(Friedrich-Schiller 대학교수)가 맡고 있다.

국내의 경우 TSG10 연구위원회에서 ITU-T의 SG10 표준화에 대응하는 활동을 추진중이며, 연구위원회내의 실무위원회3에서 연구과제 Q.10의 표준화 활동분석 및 국내 개발환경에서의 활용 방안 등이 연구중이다.

## 2. CHILL 전문가회의 서울개최 유치과정

국내에서는 ETRI 등에서 추진중인 ATM 교환기 등의 개발시 객체지향 CHILL 도입을 위한 환경 구축을 위해 '90년 6월 핀란드 Espoo에서의 SG회의이후 외국의 CHILL 구현방향 및 국제표준화 과정에 큰 관심을 갖고 대처해 왔다.

국내 연구위원회에서는 '91년의 객체지향의 개념, 특징, 관련언어에 대한 기초연구에 이어 '92년에는 객체지향 언어의 설계방안, 병행 객체구현 연구 및 객체지향 CHILL을 기개발한 Simens와 NTT의 구현내용을 분석하였다.

또 '93년에는 그동안의 국내 연구결과를 바탕으로 객체지향 CHILL에서의 중복기능(Overloading)과 지연 프로시쥬어(Deferred Procedure)에 대한 2편의 기고문을 SG10 회의에 제안하였다. 이 회의는 제1차 WTSC 회의이후 첫번째 개최된 '93년 10월 SG10 제네바 회의로써 Q.10 회의에서는 독일, 일본, 한국 등에서 총 7명이 참석했으며 검토된 기고서는 국내에서 제안한 2편 뿐이었다.

이 회의에서는 기고서 검토와 함께 이번 회기동안 진행될 객체지향 CHILL 연구일정을 결정하였으며, 다음번 전문가회의 개최지를 서울로 잠정 합의하였다. 또한 전문가 그룹내의 mailing-list 작성 및 요구사항 정리도 한국에서 주관하기로 결정하였다.

이 회의결과는 국내 SG10 연구위원회 5차 Workshop에서 국제회의 참가보고를 통해 보고되었다. 향후 객체지향 CHILL 개발환경 구축을 위해 보다 많은 전문가의 의견과 외국의 구축방안에 대한 검토가 필요하고 객체지향 CHILL 정의단계에서부터 한국의 입장을 최대한 반영할 필요성이

있다는 의견들이 국내 SG10 위원들간에 제시되었으며; 또한 CHILL 전문가 회의는 참석자수나 회의기간, 규모 등을 감안할 때 국내 SG10 연구위원회에서 주최가 가능하다는 결론에 합의를 보았으며 이 내용을 e-mail을 통해 Q.10 의장에게 회신하였다. 그 이후 회의장소, 회의주제, 일정의 결정과 참가자 명단 파악, 사전 의견 수렴, 검토할 기고서 등은 사전에 구성된 e-mail(chill @ ketri. etri. re. kr)을 통해 추진하였다.

## 3. SG10 CHILL 전문가 회의개요

ITU-T SG10 CHILL 전문가 회의는 3월 14일부터 18일까지 5일간의 일정으로 한국통신 연구센터 4세미나실에서 개최되었다. 전문가 회의참석자는 Freidrich Schiller 대학교수인 Dr. Winkler 씨와 독일 Siemens사의 Dr. Sorgenfrei, 노르웨이 KVATRO A/S사의 Dag. H. Wanvik 이 참가하였으며 국내에서는 TTA TSG10 국내 위원회에서 활동중인 위원을 중심으로 ETRI, 삼성전자, 경기대, 한국통신 연구개발원에서 7명이 참가하였다.

3월 14일 시작된 회의는 오전에 회원소개, 회의를 준비한 TTA SG10 국내연구위원회 회장의 인사말에 이어 Q.10 의장 Dr. Winkler의 사회로 5일간의 회의일정, 회의주제, 회의시간 등을 최종 합의함으로써 본격적으로 진행되었다.

회의주제는 각종 객체지향 개념의 구체적인 정의(객체모드, 상속성, 중복기능, 동적/정적 프로시쥬어, 가상/지연 객체모드)와 지네릭, 병렬성, ASN.1과의 연계에 관한 토의, CHILL 문법의 명확화(CHILL Clarification)등으로 설정되었으며, 정식 기고문 등 준비된 문서로서는 CHILL의 객체지향화를 위한 제한, CHILL의 제네릭 모듈제안 등 독일 Siemens사의 2건, CHILL Clarifica-

tion 분야에 대한 브라질의 Mr. Gustavo의 e-mail 제안이 되었으며 ETRI의 중복기능, 자연 프로시듀어에 대한 개정제안, 객체지향 CHILL의 4가지 주요 개념정립, 병행객체에 대한 검토문서가 있었다. 그러나 객체지향 CHILL의 표준화에 큰 관심을 보였던 브라질 Telebras사의 Mr. Gustavo, 일본 NTT의 Mr. Sato 등의 참석이 이루어지지 않아 제안된 기고서 등을 중심으로 한 회의진행은 다음번 회의로 미루고 참석자들을 중심으로 CHILL에 구현할 객체지향 개념들에 대한 토의와 각 속성들에 대한 정의를 추진키로 결정하였다.

회의 최종일인 18일에는 이번 회의결과를 회의 문서 “객체지향 프로그래밍을 위한 개념”(Concepts for Object-Oriented Programming)으로 최종 정리함으로써 5일간의 회의를 종료하였으며 끝으로 한국통신 연구개발원 표준연구단장의 감사 인사와 선물전달이 있었고 서울시내 및 근교의 관광일정이 진행되었다.

## 4. ITU-T SG10 CHILL 전문가 회의 결과 요약

5일간의 회의결과는 Q.10의장인 Dr. Winkler에 의해 TD001-10로 정리되었으며 이 내용은 객체지향 개념으로 총칭되는 다양한 속성들의 정의 및 가능한 전송상태 등 향후 추가 연구되어야 할 방향들을 설정하고 객체지향 CHILL에의 포함여부 등을 결정한 문서로써 96년까지 완료한 객체지향 CHILL 국제표준의 기본자료로 활용될 것이다.

○ 이 문서에 자주 사용될 3가지의 약어(OM, OMD, DOMD)를 우선 정의하였음.

—OM(Object Mode) : 객체모드

—OMD(Object Mode Definition) : 객체 모드

정의

—DOMD(Derived Mode Definition) : 상속된 객체 모드 정의

### (1) 객체 및 객체모드

#### ○ 객체 및 객체 모드의 정의

“객체(object)는 일반적으로 데이터와 동작(data and operations)으로 이루어진다”고 할 수 있다. 이들 객체는 크게 활성/비활성(병행)객체들로 나누어질 수 있고, 하나의 장소(location)를 가진다. 한편 CHILL 언어에서 선언/정의된 모든 요소들이 객체의 구성요소가 되는데 가능한 구성요소로는 SYN(상수정의), SYNMODE/NEWMODE(타입정의), DCL(변수선언), NON-GENERAL-PROC(제너럴 속성이 아닌 프로시듀어), REGION(임계지역), PROCESS(프로세서), SIGNAL(시그널)을 포함토록 하였으며 MODULE은 실행문이므로 제외시켰다. 또한 객체를 값으로 가지는 모드들은 모두 객체 모드로 정의되며 객체 모드들은 복합 모드들이다. 객체지향에 관련된 객체모드 정의방식에는 현재 3가지가 있는데, 첫째 구조체 중심의(Structure-based) 객체모드(Oberon 스타일)이고, 둘째 모듈 중심의(Module based) 모듈클래스(Object CHILL 스타일)이고, 셋째 모듈 중심의 객체모드이다.

#### ○ 객체들의 유형

객체의 유형은 상호작용을 보호하기 위한 큐를 가지지 않은 순서적인 모듈(Module)객체, 상호작용을 보호하기 위한 큐를 가지며 순서적인 리전(Region) 객체 및 상호 작용을 보호하기 위한 큐를 가지며 병행적인 병행(Concurrent)객체등 3가지로 나누었으며 3가지 유형 모두 필요한 것으로 결정하였다.

객체의 구성요소	객체의 유형		
	Module	Region	Concurrent
SYN	가 시	가 시	가 시
SYNMODE/NEWMODE	가 시	가 시	가 시
DCL	가 시	비가시	비가시
PROC NON-GENERAL-PROC	적 용 에 서 제 외		
MODULE	가 시	가 시	가 시
REGION	가 시	가 시	가 시
PROCESS	가 시	가 시	가 시
SIGNAL	가 시	가 시	가 시

객체의 유형별로 각 객체의 구성요소에 대한 가시/비가시성의 정의는 다음과 같다.

○객체의 초기화/종료화(Initialization/Finalization)

한 객체가 생성될 때 이들에 대한 초기화를 할 것인가/아닌가, 또한 객체가 소멸될 때 이들에 대한 종료화를 할 것인가/아닌가에 대한 토의가 이루어졌는데 가능한 상태중 생성자/소멸자(Constructor/Destructor)는 포함키로 하였으나 실행문의 연속(Action-Statement-Sequence) 및 생성자/소멸자와 실행문 연속의 조합은 제외시키기로 했다. 그 이유는 생성자/소멸자는 실행문의 연속보다는 더 유연(flexible)하며 두 경우의 조합은 너무 복잡하기 때문이다. 그의 “LOC과 READ의 장소들에 대한 초기화를 어떻게 처리할 것인가”하는 문제는 차후 연구과제(open item)로 남겨 두기로 했다.

생성자/소멸자들의 전송 문제에서는 생성자와 소멸자가 보통 객체모드내에 제한되므로 C++와 같이 하위 객체모드에 상속하지 않는 것으로 결정했다. 또 생성자/소멸자의 이름은 어떻게 사용할 것인가에 대한 토의가 이루어졌는데 객체모드의 이름과 같은 이중사용과 다른 이름 사용이 검토되었으나 같은 이름을 사용키로 결정하였다.

생성자/소멸자들에 대한 동적인 특성의 부여

에서는 생성자 및 소멸자 각각에 대한 동적, 정적 특성부여의 경우 4가지 조합이 토의되었으나 동적 생성자는 필요성이 없는 것으로 검토 되었으며 동적 소멸자는 객체지향의 다양성과 같은 경우 유용한 것으로 합의되어 정적 생성자/동적 소멸자만 포함키로 결정하였다.

○객체모드의 위치

객체모드 정의를 제약없이 허용할 경우에는 프로그래밍의 오류를 유발시킬 수 있다. 예를 들면, 내부 객체모드가 외부 객체모드의 데이터를 참조할 때 외부 객체의 생성없이 내부객체가 생성되어 외부객체의 원소를 참조하면 존재하지 않게 되는 상황이 발생한다. 이러한 상황을 해결하는 방안으로 다음과 같은 방법들이 토의되었다.

객체모드의 중첩 불허용, 내부객체모드의 외부 객체모드 참조제한, 내부객체모드를 외부에 보여 주지만 객체생성은 외부객체모드로만 허용, 내부 객체모드를 외부에 보여주지 못함 등 4가지가 제안되었으나 두번째와 세번째중 하나를 결정하기로만 합의하였으며, 구체적인 해결방안들에 대해서 generic과의 연관성 등을 고려하여 다음 회기에 좀 더 깊이 토의하여 결정하기로 하였다.

또한 정적 원소들의 정의에서는 만약 한 요소가 계층구조내에 한 곳만 존재한다면 “정적이다”라고 한다고 정의하였으며 정적요소만 허용하기로 결정하였다.

○ 객체들을 위한 연산

객체들을 위한 연산으로는 =, /=, := 등과 같은 것들이 있는데 각 객체 유형(Module, Region, Concurrent)들에 따라 사용 가능한 것들/정의 가능한 것들에 대한 토의가 있었으나 결정을 내리지 못하고 다음 회기에 다시 논의하기로 하였다.

(2) 가시성 제어/보호/정보 은닉

기존의 언어에서 제공되어온 가시성 제어/보호/정보 은닉과 같은 특성들은 객체 지향 언어에서도 계속해서 지원되어야 할 것인가/아닌가에 대한 토의가 있었다. 참고로 C++와 같은 언어에서는 정보 은닉, 접근 제어와 같은 특성은 지원하고, 가시성 제어 특성은 지원하지 않고 있다. 결정된 내용은 다음과 같다. 객체 정의모드는 두가지의 인터페이스를 가지는데, 하나는 외부적인 사용자들에 대한 인터페이스이고, 다른 하나는 내부적으로 상속되는 하위객체들에 대한 인터페이스이다. 또한 객체의 정보은닉 범주는 다음 4가지로 분류하였으며 그중 하위객체를 제외한 자신과 외부에만 가시성을 가지는 상태는 제외하기로 하였다.

- 자신의 객체안에서만 가시성 부여 : Private 상태(C++)
- 자신과 하위객체에서만 가시성 부여 : Private 상태(C++)
- 하위객체를 제외한 자신과 외부에서만 가시성 부여
- 자신, 하위객체 뿐 아니라 외부에도 가시성 부여

: Public 상태(C++)

(3) 상속성(Derivation)/다형성/동적 프로시듀어

- 하위 객체모드는 상위 객체모드의 원소와 가시성 두가지를 상속받는다. 원소의 전송(Transfer)에 있어서 상위 객체모드의 모든 원소를 전송하거나(완전전송, monotone extension), 일부 원소만 전송(선택적 전송)될 수 있는데 선택적 전송은 다형성과 상충되므로 제외하기로 결정하였다. 또한 상위 객체로부터 상속받는 하위 객체들에 대한 가시성 전송이 이루어짐으로 인해 상위객체가 가지고 있던 가시성에 대한 변화를 유발하게 되는데 이때 가시성의 유지, 가시성의 확장(Liberalization), 가시성의 제한(Restriction), 가시성의 제한과 확장의 혼합과 같이 4가지 상태변경이 가능하나, 가시성의 확장은 정보은닉에 위배되고 또한 제한의 경우 다형성과 상충되므로 가시성은 상속시 변화되지 않고 유지되는 것으로 합의 하였다.
- 다형성(Polymorphism)에 대한 검토에서는 “REF OM”의 포인터 자신의 객체 뿐 아니라 자신의 하위 객체들에 대해서도 참조할 수 있다고 정의하였다. 그러나 “REF OM”으로 선언된 변수가 LOC 파라미터 및 aliasing하는 LOC 선언변수에는 적용될 수 없다.
- 추상 객체 모드(Abstract Object Modes)검토에서는 적어도 하나의 프로시듀어가 선언만 되고 정의되지 않았다면 “추상적이다”라고 하며 그런 프로시듀어 역시 “추상 프로시듀어이다”라고 정의하였다.
- 정적/동적 프로시듀어 검토에서는 정적 프로시듀어들은 확장된(상속된) 하위 객체모드내에서 재 정의되지 않는다. 즉 계층구조의 객체모드내에 고정된다. 만약 한 정적 프로시듀어가 하위

객체모드내에 정의된 프로시듀어와 동일한 이름을 가진다면, 이것은 이전의 것과는 전혀 다른 새로운 프로시듀어가 되므로 다형성으로 인해 사용자의 오류를 유발시킬 수 있다. 정적 프로시듀어는 동적 프로시듀어와는 달리 실행성능을 저하시키지 않는다. 그리고 동적 프로시듀어 객체지향 프로그래밍의 전형적인 경우이므로 정적 프로시듀어만 허용할 것인지 두가지 경우 모두를 허용하는 경우에 대한 결정은 차후 연구기로 하고 보류하였다.

- 상속은 상위객체로 부터 여러가지 특성들을 물려 받은 것을 말하며, 크게 단일/다중상속이 있다. 다중상속은 객체지향 모델링에서 유연성을 부여 하지만 구현이 쉽지 않고 프로그램의 이해를 어렵게 할 뿐 아니라 통신시스템 소프트웨어에서는 다중상속이 크게 요구되지 않으므로 단일상속만 포함기로 결정하였다.

**(4) 병행 객체들의 동기화를 위한 방법**

병행 객체들의 동기화를 위한 방법으로는 3가지 유형이 있다. 첫째 결과/대답(result/answer)이 없는 비동기식 호출이고, 둘째 즉각적인 반응을 보이는 동기식 호출이고, 셋째 지연 응답을 가진 비동기식 호출이다. 한편 지연 응답을 위한 기법으로

는 3가지가 있을 수 있는데 요약하면 다음과 같다.

- 시그널 방법
- Region 사용
- Event와 Location 사용

**(5) 차기 회의 계획 및 담당기관**

차기 회의 계획은 지난번 1차 SG 회의에서 결정된 대로 94년 10월 19일부터 28일까지 제네바에서 진행될 2차 SG 회의 때에 개최될 Q.10회의를 진행기로 하였으며 공식적인 CHILL 참고문서로는 92년 2월에 작성된 회의보고서(COM X-R34-E)로 합의하였다. 한편 미해결 문제 5건에 대해서는 서울 회의의 참석자를 중심으로 해결기관을 결정하였으며 차기 회의시 연구결과를 기고기로 합의하였다. 다음은 각 안건에 대한 업무 분담 내용이다.

**5. 맺음말**

이번 CHILL 전문가 서울 회의는 지난번 제네바 회의의 1.5일에 비해 4.5일동안 진행되어 객체지향 CHILL의 기본 골격을 대부분 정의할 수 있었던 중요한 회의였으며 한국에서 개최되었기 때문에 자연스럽게 국내의 주장을 많이 반영할 수 있었다.

미해결 문제	담당 기관	담당 자
Friends 개념	삼성전자	조병영
정적 프로시듀어	(KVATRO A/S)	Wanvik
병행 객체의 어의	KVATRO A/S, 경기대, Siemens	Wanvik, 조창연, Sorgenfrei
LOC/READ 장소의 초기화	(KVATRO A/S)	Wanvik
OMD의 위치	ETRI	이준경, 이동길
객체들의 연산	Siemens	Winkler
객체지향 개념의 구분	ETRI	이준경, 이동길
중복기능	ETRI	이준경, 최 완

객체지향 CHILL은 기존의 CHILL을 확장하는 개념으로 정의되는 것이기 때문에 각국의 환경과 언어의 구조, 시스템의 특성에 따라 달라질 수 있는 요소들이 많았다. 특히 일본의 객체지향 CHILL은 독일의 객체지향 CHILL과 국내에서 추진중인 객체지향 CHILL의 골격과 크게 달라 권고안 초안 작성에 많은 어려움이 예상되었지만 불참하였다. 그리고 독일 지멘스사의 객체지향 CHILL과 국내의 객체지향 CHILL 개발방향에 있어서도 다소 차이가 있었기 때문에 이미 객체지향 CHILL을 개발하여 시스템을 개발하고 있는 지멘스의 입장이 강경하리라 예상되었다.

그러나 의장이 아주 기초적이고 세부적인 사항부터 하나씩 하나씩 체계적으로 분석하고 토론하여 결정하는 과정을 거쳐 전체적인 방향을 결정하는 방식을 채택함으로써 아직 객체지향 CHILL을 개발하지 못한 현실적인 약점에도 불구하고 국내 연구위원회에서 준비한 내용을 대부분 권고안 초안의 내용으로 통과 확정할 수가 있어서 앞으로 ETRI 등 국내의 개발환경에 보다 적합한 객체지향 CHILL과 프로그래밍 환경을 구축할 수 있는 터전을 마련하였다.

그외에도 정식 회의 시간 외 외국 참석자들과 함께 할 수 있었던 많은 시간들을 이용해 외국에서

정리한 객체지향 CHILL의 어의 및 문법, 독일, 노르웨이 등에서 개발한 객체지향 CHILL을 지원하기 위한 프로그래밍 환경, 지멘스에서의 객체지향 CHILL 컴파일러 개발현황, 적용경험 등에 관하여 많은 정보를 얻을 수 있었다. 또한 국내에서 취약한 분야인 병렬객체와 동적 특성에 대한 개념 모델과 이의, 목적시스템의 구조와의 관계 등에 대한 집중적인 질문을 통해 CHILL에서의 병렬객체에 대한 이해를 넓힐 수 있는 좋은 기회가 되었다.

또한 차기 회의시 까지 기초연구에 해당하는 기고서 2편을 삼성의 조병영 위원과 경기대의 조창연 박사가 준비키로 하였으며 실제적인 객체지향 CHILL의 정의와 확장에 관한 내용의 기고서 3편을 ETRI를 중심으로 한 국내 연구위원이 담당하게 됨으로써 독일, 노르웨이, 브라질, 일본을 중심으로 이루어지던 객체지향 CHILL 국제표준 개발에 한국의 입장과 위상을 높이는 좋은 계기가 되었다.

끝으로 이번 회의를 위해 회의개최에 따른 많은 지원을 해주신 TTA 및 한국통신 표준연구단에 깊이 감사드리며 5 일간의 짧지 않은 회의 뿐 아니라 회의준비, 식사, 관광안내까지 맡아 열과 성을 다 해준 ETRI의 최완, 이준경 위원을 포함한 모든 참석자들에게 감사드립니다.

