

# SDL 메트릭 집합의 분석적 검증

홍 의 석<sup>†</sup> · 정 명 희<sup>†</sup>

## 요 약

소프트웨어 개발 초기 단계의 문제점이 개발 후반부 산물의 품질에 심각한 영향을 미치기 때문에 설계 단계를 정량화 하는 설계 메트릭들은 전체 시스템 개발비용을 낮추는 데 중요한 역할을 하고 있으며, 설계 정량되는 결과 산물이 매우 큰 통신 소프트웨어 같은 실시간 시스템 설계에 더욱 필요하나 많은 메트릭들이 제안되어 사용되고 있으나 그중 상당한 부분이 적절한 검증 과정을 거치지 않은 채로 사용되고 있다. 본 논문에서는 선행 연구인 [9]에서 제안한 SDL 메트릭 집합을 수정, 확장한 메트릭 집합을 정의하고 결과 메트릭 집합을 분석적으로 검증한다. 분석적 검증 방법으로는 공리적 검증 방법과 차원 분석을 사용하였으며 기존 메트릭 형태의 수정은 이들 검증 조건을 만족하는 방향으로 이루어졌다.

# Analytical Validation of the SDL Metrics Set

Euy-Seok Hong<sup>†</sup> · Myung-Hee Jung<sup>†</sup>

## ABSTRACT

Design metrics that quantify the design phase play an important role in reducing system development cost because the problems in early phases of software development seriously affect the quality of the late products. Real-time systems such as telecommunication systems are so large that design quantification is more important in real-time system design. Although many metrics have been proposed, few of them are correctly validated. This paper revises the SDL metrics set proposed in earlier study [9] and perform an analytical validation of the metrics set. Axiomatic approach and dimensional analysis are used for metrics validation and the old metrics are revised to satisfy the validation criteria.

## 1. 서 론

소프트웨어 개발 기술이 발달함에 따라 각 개발 단계의 산물들과 프로세스들의 정량화 기술도 매우 중요해지고 있다. 많은 회사나 연구 기관들이 기존의 개발 프로세스를 측정 기반 소프트웨어 개발 프로세스(Measurement-based Software Development Process)로 변화 시켜 가고 있다. 이는 SEI의 CMM(Capability Maturity Model)[1, 2]이나 ISO 9000-3[3], ISO 9001[4]과 같은 개발 프로세스 개선 방법의 표준안들에서도 잘 나타나 있다. 프로세스 평가 기준인 여러 수준들 중 상위

수준으로 올라갈수록 잘 정의된 메트릭 집합을 이용한 개발 단계의 정량화와 이들을 이용한 프로세스 개선을 요구하기 때문이다. 이와 같은 정량화 요구에 부응하기 위해 수많은 메트릭들이 제안되고 사용되어 왔다. 하지만 많은 메트릭들이 연구자의 직관에 의해 만들어 졌으며 적절한 검증 과정을 거치지 않은 채로 사용되고 있다[5].

메트릭의 검증 방법에는 분석적 검증 방법(Analytical evaluation)과 실험적 검증 방법(Experimental evaluation)이 있다. 전자는 측정 이론(Measurement theory)[5, 6]이나 검증 기준(Evaluation criteria)[7, 8] 등을 이용하는 것이며, 후자는 실제 데이터를 이용하여 메트릭들의 유용성을 검증하는 것이다. 본 논문의 선행 연구

<sup>†</sup> 경 회 원 인양대학교 영성처리학과 교수  
논문접수 1999년 9월 3일, 심사완료 2000년 3월 5일

인 [9]에서는 ITU-T Z.100 권고안으로 표준화된 실시간 시스템 명세 언어인 SDL(Specification and Description Language)을 지원하는 설계 메트릭 집합을 제안하고 실험적인 검증을 수행하였다. 본 논문에서는 SDL 메트릭 집합을 분석적으로 검증하고 그 결과 나타난 문제점들을 해결하도록 기존 메트릭들의 부분들을 수정한다.

2장에서는 객체지향 설계 개념이 첨가된 OSDL[10]을 지원하고 분석적 검증에 알맞게 수정된 SDL 메트릭 집합을 제안하고 3장에서는 공리적 검증 방법과 차원 분석을 사용하여 SDL 메트릭 집합을 검증한다. 4장에서는 결론과 향후 연구 과제에 대해 기술한다.

## 2. SDL 메트릭 집합

SDL은 실시간 프로세스들의 동작을 EFSM(Extended Finite State Machine) 개념을 도입하여 표준화된 설계 명세 언어로서 ATM 교환기 시스템과 같은 대형 통신 시스템의 명세서와 설계서를 형식적으로 나타내는데 사용된다. 블록은 프로세스들과 이들의 상호작용으로 이루어진 기능 단위이며, 설계 방법은 EFSM을 작성하고 이를 기초로 하여 블록을 나누는 상향적인 방법과 블록 단위로 시스템을 나누고 이를 EFSM으로 세분화하는 하향적인 방법이 가능하나 후자가 일반적인 방법이므로 일반적인 설계 단계는 구조 설계 단계와 상세 설계 단계로 나뉜다. 각 설계 단계에서 정량화할 대상인 핵심 개체들은 구조 설계 단계의 블록, 상세 설계 단계의 블록, 상세 설계 단계의 프로세스이다. 본 정에서는 각 설계 단계에서 측정할 수 있는 기본 메트릭들을 정의한 후 이를 이용하여 세가지 관심 개체 형의 메트릭 집합을 정의한다. 이는 각 개체에 대한 복잡도 메트릭 매트릭, 내부 및 외부 복잡도 메트릭 그리고 혼성 복잡도 메트릭으로 구성된다.

SDL 메트릭들은 SDL 설계 단계와 각 단계의 산물들을 모델화한 구조화된 SDL 모델[9]로부터 정의되며, FBID(Flat BID)는 구조 설계 단계에서 시스템을 가장 자세한 단위들로 분할하였을 때 전체 시스템을 나타내는 BID(Block Interaction Diagram)이고 FPD(Flat PD)는 상세 설계 단계에서 시스템을 가장 자세하게 나타낸 PD(Process Diagram)이다.<sup>1)</sup>

### 2.1 SDL 기본 메트릭 집합

<표 1>은 두 설계 단계에서 직접적으로 측정 가능한 기본 메트릭 집합을 나타낸다. 상세 설계 단계의 기본 메트릭 집합은 구조 설계 단계의 메트릭 집합을 포함하며, 상세 설계 단계에서 이의 기술은 생략하였다. DIT, NOC 역시 상세 설계 단계에서 프로세스형, 블록형에서도 정의되나 표에서 생략하였다. 이들 메트릭들은 측정 목적 개체가 서로 다른 관심 개체형이 될 수 있기 때문에 같은 이름을 사용한다. 또한 블록의 수준값같이 매우 기본적인 특성을 나타내는 메트릭 역시 생략하였다. 기본 메트릭 집합은 대부분 구성원들의 수로 정의하였다. 이는 설계 결과를 입력하여 자동으로 계산할 수 있는 자동화 가능 메트릭 조건[11]을 만족시키고, 메트릭의 간단성 및 용이성, 그리고 메트릭 검증 부분에서 설명할 차원 분석을 만족시키기 위함이다.

<표 1> SDL 기본 메트릭 집합

단계	기본 메트릭	정 의
구조 설계	SSD, SSB	SubSystem Depth/Breadth 블록 부시스템의 깊이/너비
	IBC, OBC, IBS, OBS	number of Input/Output Channels/ Signals of a Block 블록의 입출력 채널/신호수
	BP, BR, BS	number of Processes/Routes/Signals composing a Block 블록의 FBID 내의 프로세스/라우트/신호수
	DIT, NOC	Depth of Inheritance Tree Number of Children 블록형 상속 계층 구조에서 블록 깊이/기식수
상세 설계	IPR, OPR, IPS, OPS	number of Input/Output Routes/Signals of a Process 프로세스의 입출력 라우트/신호수
	IPSD, OPSD	number of Data within Input/Output Signals of a Process 프로세스의 입출력 신호선에 실린 데이터수
	SDRP	number of Shared Data Declared in a Process 프로세스의 PD에서 사용하는 다른 프로세스에서 선언한 공유 데이터수
	SDDP	number of Shared Data Declared in a Process 프로세스의 PD에서 선언한 공유 데이터수
	IDP, EDP	number of Data Imported/Exported in a Process 유입/유출 연산에 의해 유입/유출되는 데이터수
	PST, PTR, PTK, PDC	number of States/Transitions/Tasks/Decisions of a Process 프로세스의 상태/전이/태스크/결정수
	NTC	number of Time Constraints 프로세스/블록 내에 존재하는 시간 제약수
	BPR, BMC	number of Procedures/Macros defined in a Block 블록 내에 정의된 프로시저/매크로수

1) 본 논문에서 메트릭 정의에 사용한 용어들의 강의를 매경 이 문에 대한 자세한 설명은 [9]를 참조하기 바란다.

SSD, SSB는 부시스템 구조의 최대 깊이와 최대 너비이며, {IPSD, OPSD}, {SDRP, SDDP}, {IDP, EDP}는 설계 개체 사이의 세가지 데이터 통신 형태를 측정한다. DIT는 관심 개체의 메트릭으로 적당하지만 NOC는 형을 대상으로 하는 것이 더 적합하다. SDL에서는 다중 상속을 지원하지 않으므로 DIT 정의는 혼돈의 가능성이 없다.

2.2 SDL 복잡도

위에서 정의한 구조화된 SDL 모델과 기본 메트릭 집합으로부터 SDL 복잡도 메트릭 집합을 정의한다. SDL 복잡도는 설계 단계와 관심 개체의 형에 의해 세가지로 나뉘며 <표 2>는 이들을 위한 복잡도 메트릭들을 정의하고 간단히 설명한 것이다. 범주에서 S는 블록 부시스템 구조 관련 복잡도, E와 I는 관심 개체

의 외부 복잡도와 내부 복잡도 관련 기본 복잡도 메트릭들을 의미한다. EC와 IC는 관심 개체의 외부 복잡도와 내부 복잡도 메트릭을 의미하며 C는 혼성 복잡도 메트릭을 의미한다

SDL 개체의 정량화를 복잡도 메트릭 벡터 형태로 하려면 해당 관심 개체의 기본 복잡도 메트릭들을 구성원으로 하는 벡터를 구성하면 되고, 스칼라 형태로 하려면 해당 관심 개체의 가중합, 가중곱 형태의 혼성 메트릭을 사용하면 된다.

DIT와 NOC는 복잡도에 긍정적인 영향과 부정적인 영향을 동시에 내치며 특히 관심 개체의 NOC는 형에 적용되는 경향이 강하므로 복잡도 집합에서 제외하였다. DIT는 상속 계층 구조에서 얼마나 많은 조상 개체 형들이 해당 개체에 영향을 미치는가에 대한 척도이다. 부정적인 면은 DIT가 커질수록 상속받는 것이 많

<표 2> SDL 복잡도

단계	관심 개체	복잡도 메트릭	범주	정의 및 설명		
구조 설계	블록	기본 복잡도 메트릭	BRS	S	Balance Rate of a Subsystem	
			EBC	E	블록 부시스템의 살던 블록 수준들의 분산	
			EBS	E	number of io(External) Channels of a Block	
			BP, BS, BR	I	IBC+OBC number of io(External) Signals of a Block IBS+OBS	
		조합 메트릭	AIBC	IC	BP+BS+BR	
			AEBC	EC	$\sqrt{IBC \times OBC + IBS \times OBS}$	
	상세 설계	프로세스	기본 복잡도 메트릭	EPR	E	number of io(External) Routes of a Process
				EPS	E	IPR+OPR number of io(External) Signals of a Process
				IPD	E	IPS+OPS number of fan-In Data of a Process
				OPD	E	IPSD+SDRP+IPD number of fan-Out Data of a Process
PST, PTR, NTC				I	OPSD+SDDP+EPD	
PDO				I	number of Data Operations of a Process	
조합 메트릭			DIPC	IC	PTK+PDO+NTC	
			DEPC	EC	$\sqrt{IPR \times OPR + IPS \times OPS + IPD \times OPD}$	
			DPC1	C	$w_1 DIPC + w_2 DEPC$	
			DPC2	C	$DIPC \times DEPC^2$	
블록	기본 복잡도 메트릭	IBD, OBD	E	number of Input/Output Data of a Block		
		BST, BTR, NTC	I	number of States/Transitions composing a Block		
		BDO	I	number of Data Operations composing a Block		
				FPD 내의 디스크의 결빙 수의 합		
	조합 메트릭	DIBC	IC	BTR+BDO+NTC		
		DEBC	EC	$\sqrt{IBC \times OBC + IBS \times OBS + IBD \times OBD}$		
	DBC1	C	$w_1 DIBC + w_2 DEBC$			
	DBC2	C	$DIBC \times DEBC^2$			

아저서 행위를 이해하기 어려워지며, 깊은 상속 구조는 설계 명세 전체를 이해하기 어렵게 만드는 것이다. 그러나 DIT가 커질수록 상속받는 것들에 대한 제사용이 늘어난다는 장점이 있다.

### 3. SDL 매트릭 검증

#### 3.1 Weyuker 조건 집합을 이용한 공리적 검증

Weyuker는 복잡도 매트릭이 가져야 할 아홉개의 필요 조건을 다음과 같이 제시하였다[8]. P, Q는 프로그램을 의미하고 |P|, |Q|는 P, Q의 복잡도 값을 의미한다.

- 1)  $(\exists P)(\exists Q) (|P| \neq |Q|)$
- 2) c를 0보다 크거나 같은 실수라 하면 복잡도가 c가 되는 유한개의 프로그램이 있다.
- 3)  $|P| = |Q|$ 인 서로 다른 P, Q가 존재 가능하다
- 4)  $(\exists P)(\exists Q) (P \equiv Q^2) \ \& \ |P| \neq |Q|)$
- 5)  $(\forall P)(\forall Q) (|P| \leq |P|; |Q|^3) \ \& \ |Q| \leq |P|; |Q|)$
- 6) a :  $(\exists P)(\exists Q)(\exists R) (|P| = |Q| \ \& \ |P; R| \neq |Q; R|)$   
b :  $(\exists P)(\exists Q)(\exists R) (|P| = |Q| \ \& \ |R; P| \neq |R; Q|)$
- 7)  $(\exists P)(\exists Q) (Q \text{는 } P \text{의 문장들을 제순열한 것임} \ \& \ |P| \neq |Q|)$
- 8) 만약 P가 Q의 바깥 이첩이면  $|P| = |Q|$
- 9)  $(\exists P)(\exists Q) (|P| - |Q| < |P; Q|)$

이와 같은 검증 조건들에 대해 많은 비판과 토론이 행해졌다 [5]는 조건5와 조건6이 복잡도 정의에 대한 밀접관성을 가지고 있다고 하였다 즉 조건5는 내부적인 크기를 나타내는 매트릭에는 적합하나, 이해나 심리적 복잡성을 나타내는 매트릭에는 적합치 않다. 또한 조건6은 후자에는 적합하나 전자에는 적합치 않다. [6]은 측정 이론에 기반하여 각 조건이 가정하고 있는 매트릭들의 스케일들에 비밀관성이 있다고 하였다 조건9는 비율 스케일(ratio scale)에는 적합하나 간격 스케일(interval scale)에는 부적당하고, 조건6, 7은 비율 스케일을 무시한 조건이다. 비슷한 이유로 [12]는 소프트웨어 측정 검증의 일반적인 프레임워크에 조건5, 6, 9를 제외시켰다. [13]은 Weyuker의 조건들이 좋은 복잡도 매트릭 검증을 위한 충분 조건이 아니라 필요 조건이라 하였다.

비록 최근에도 공개적으로 토론 대상[14, 15]이 되고 있고, 개선하려는 노력들이 이루어지고 있지만, Weyuker의 조건들은 현재 가장 잘 알려지고 널리 쓰이는 분석적 검증 방법이다[16]. 그러므로 본 논문에서는 SDL 매트릭 집합의 검증 방법으로 Weyuker의 조건 집합을 사용한다 Weyuker의 조건에서 P, Q는 프로그램 또는 부분 프로그램을 의미하지만 SDL 매트릭 검증에서 P, Q는 서로 같은 부류<sup>4)</sup>의 관심 개체 P, Q가 된다. 본질에서 관심 개체들이란 같은 부류에 속하는 관심 개체들을 의미한다.

조건1)의 의미는 복잡도 값이 다른 관심 개체들이 존재해야 한다는 것이며, 이의 부정은  $(\forall P)(\forall Q) (|P| = |Q|)$ 이다. SDL 시스템 구성 요소들의 수에 대한 제약 조건은 없고, SDL 복잡도 매트릭 집합은 BRS를 제외하고는 모두 구성 요소들의 수에 기초를 두고 있다. BRS에 영향을 미치는 시스템 구조도 모든 블록 부시스템의 구조가 같아야 한다는 제약은 없으므로  $(\forall P)(\forall Q) (|P| = |Q|)$ 는 거짓이다 그러므로 SDL 복잡도 매트릭 집합은 조건1)을 만족한다.

조건2)는 일정한 복잡도 값을 갖는 관심 개체가 유한개임을 의미한다. SDL 시스템이 사용할 수 있는 키워드, 데이터, 심볼 등은 모두 유한개이다. 또한 시스템을 구성하는 관심 개체의 수 역시 유한개이다. 그러므로 대부분의 SDL 매트릭들은 이 조건을 만족한다

그러나 문제는 이 조건에 고려되는 관심 개체가 한 시스템을 구성하는 관심 개체의 집합이 아니라 문법적으로 가능한 무한개의 관심 개체 집합이라는 것이다.<sup>5)</sup> 이런 가정을 갖는다면 전이의 연관된 PTR, PST, PDC 등은 이 조건을 만족하지 못한다. 같은 집이에 유한개의 식별자와 연산을 사용하여 무한개의 인산 집합을 만들 수 있기 때문이다. [6]은 Cyclomatic Complexity가 이 조건을 만족하지 못하는 이유가 매트릭이 서수 스케일(ordinal scale)이기 때문이지, 이로 인하여 매트릭의 타당성이 문제되는 것은 아니라고 주장하였다

조건3)은 복잡도 값이 같은 다른 관심 개체가 존재할 수 있다는 것이다. 이를 만족하지 못하는 매트릭은 각각의 관심 개체가 유일한 값을 갖게 하는 것이다. 명

2) P의 Q가 같은 기능을 갖는다 P의 Q는 같은 입력들에 의해 멈추고, 이러한 입력 격자에 대해 같은 출력을 갖는다

3) P와 Q를 문법적으로 합한(순서를 생략하여 연결된) 프로그램

4) 관심 개체 부류는 구조 설계 단계의 블록, 상세 설계 단계의 블록, 프로세스와 같이 세 부류가 있다

5) Weyuker는 이런 의도로 조건2)를 만들었으나, [16]은 유한개의 관심 개체 집합만을 고려 대상으로 하여 조건3)를 검증 조건에서 제외하였다.

백하게 SDL 복잡도 매트릭 집합은 조건3을 만족한다

조건4)는 두 관심 개체가 같은 기능을 수행해도 그들의 복잡도는 그들의 구현 또는 내부 설계에 의존한다는 것이다 여기서 중요한 것은  $P=Q$ 의 의미이다 Weyuker는 이를 같은 입력 집합을 가지고, 같은 입력에 대해 같은 출력을 내는 것이라 정의하였다. SDL에서는 이를 두 관심 개체가 같은 외부 환경을 갖는 것이라 확대 정의할 수 있다. 즉  $P=Q$ 이면 내부 구현에 상관없이 외부에서 볼 때 P, Q가 같은 기능을 수행하는 것이므로 P와 Q는 같은 입력 채널 집합과 입력 신호 집합 그리고 같은 입출력 데이터들을 갖는다

Weyuker의 기능에 대한 관점은 개체의 외부 복잡도에 관련된 것들이다. 그러므로 <표 2>의 SDL 복잡도 매트릭 집합에서 범주 E와 EC에 속하는 매트릭들은 이 조건을 만족하지 못한다. 왜냐하면 기능이 같은 관심 개체들은 외부 환경에 대한 상호작용이 같으므로 외부 복잡도가 같기 때문이다. 따라서  $P=Q$ 의 정의에 이미 P, Q의 외부 복잡도가 서로 같다는 것을 포함하고 있으므로 조건4는 외부 복잡도를 나타내는 E와 EC 범주에는 부적당하다. I와 IC는 관심 개체를 내부적으로 어떻게 설계하느냐에 관한 것이므로 이 조건을 만족한다. 하지만 그녀의 관점은 혼성 복잡도 매트릭 형태나 매트릭 벡터 형태에 내부 복잡도와 외부 복잡도를 함께 고려한 SDL 매트릭 전체 집합의 의도와 일치한다. BRS는 끝단 블록이 아니라면 같은 기능을 갖는 관심 개체들이라 할 지라도 그들의 부시스템 분해는 다를 수 있으므로 이 조건을 만족한다.

조건5)는 단조성(monotonicity) 조건이다. 이는 관심 개체 P, Q의 복잡도보다 그들을 문법적으로 합한 P, Q가 더 복잡하다는 것이다. Weyuker는 P; Q를 프로그림 부분들의 문법적 연결이라 기술하였지만 특별한 정의를 내리지 않았다.

관심 개체 P; Q에서 P와 Q가 서로 주고 받는 입출력 정보들은 P; Q 내부로 들어가고, P, Q의 내부 구성원들은 P의 구성원들과 Q 구성원들의 합집합이 된다 한 관심 개체의 n종류의 입출력 형태가 정의되고 이들의 입출력 개수를 나타내는 매트릭들이  $I_1, O_1 (1 \leq n) \leq n$  그리고 구성원의 개수를 나타내는 매트릭들이  $C_1 (1 \leq i \leq m)$ 라고 가정하면 관심 개체 P; Q와 P, Q의 매트릭값들은 다음과 같은 관계가 성립한다. 관심 개

체 E에 대한 매트릭 M의 값은 M(E) 형태로 나타낸다.  $MAX(x, y)$ 는 x, y 중 크거나 같은 값을 나타낸다.

$$\begin{aligned}
 &(\forall P) (I_1(P) \geq 0, O_1(P) \geq 0, C_1(P) \geq 0) \\
 &(\forall P)(\forall Q) (0 \leq I_1(P; Q) \leq I_1(P)+I_1(Q)) \\
 &(\forall P)(\forall Q) (0 \leq O_1(P, Q) \\
 &\leq O_1(P)+O_1(Q)) \quad (1) \\
 &(\forall P)(\forall Q) (MAX(C_1(P), \\
 &C_1(Q)) \leq C_1(P, Q) \leq C_1(P) \cup C_1(Q)) \quad (2)
 \end{aligned}$$

만약 관심 개체가 구조 설계 단계의 블록이라면  $I_1 = \{IBC, IBS\}$ ,  $O_1 = \{OBC, OBS\}$ ,  $C_1 = \{BP, BS, BR\}$ 이다.  $I_1$ 와  $O_1$ 는 범주 E에 속하는 매트릭들이며  $C_1$ 는 범주 I에 속하는 매트릭들이다 (1)에서 P; Q의 입출력 부분 매트릭 값의 최소값이 0인 경우는 P, Q의 입출력 모두가 P, Q간에만 이루어지는 경우이다.

(1)에 의해 범주 E에 속하는 매트릭들은 조건5를 만족 못하며, (2)에 의해 범주 I는 조건5를 만족한다. 설명의 간편화를 위해 조합 매트릭의 표기법을 일반화시킨다. 관심 개체 P의 내부 복잡도를  $IC(P)$ , 외부 복잡도를  $EC(P)$ , 혼성복잡도를  $C(P)$ ,  $C2(P)$ 이라 한다.

(2)에 의해 다음 식이 성립한다

$$\begin{aligned}
 IC(P) = \sum C_1(P) &\leq \sum MAX(C_1(P), C_1(Q)) \\
 &\leq \sum C_1(P; Q) = IC(P; Q) \\
 IC(Q) = \sum C_1(Q) &\leq \sum MAX(C_1(P), C_1(Q)) \\
 &\leq \sum C_1(P, Q) = IC(P; Q) \quad (3)
 \end{aligned}$$

그러므로 범주 IC는 조건5를 만족한다.

(1)에 의해 다음 식이 성립한다.

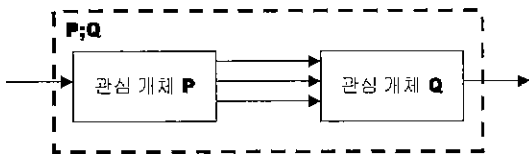
$$\begin{aligned}
 0 \leq EC(P; Q) &= \sqrt{\sum (I_1(P, Q) \times O_1(P; Q))} \\
 &\leq \sqrt{\sum_{x, y \in \{P, Q\}} I_1(x) \times O_1(y)} \quad (4)
 \end{aligned}$$

그리고  $EC(P)$ 와  $EC(Q)$ 는 0보다 크거나 같으므로 범주 EC는 조건5를 만족하지 않는다.

SDL에서 관심 개체들은 동시에 채널이나 신호들을 이용하여 상호 직용한다 일반적인 경우, 즉 여러 개의 개체들이 연결되어 상호 작용하는 경우는 P, Q의 입출력 부분이 늘어나 이 조건을 만족하겠지만 (그림 1)과 같이 P의 출력이 모두 Q의 입력이 되는 극단적인 예에서는 그렇지 않다 그림에서 화살표는 입출력 채널

6)  $C1 = ABC1$  또는  $DI \times 1$  또는  $DBC1$ ,  $C2 = ABC2$  또는  $DPC2$  또는  $DBC2$

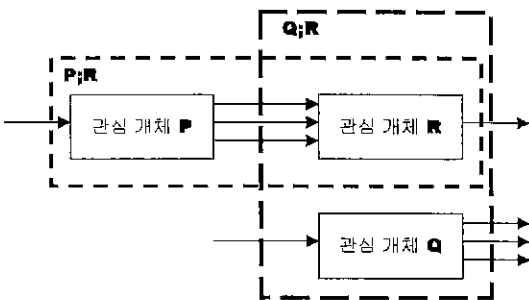
이나 신호의 추상화 형태이다. 만약 화살표가 신호가 없는 채널이고 관심개체가 구조 설계 단계의 블록이라면 EBC(P)와 EBC(Q)는 모두 4인데 EBC(P; Q)는 2가 된다. AEBC(P; Q) 역시 1이므로 AEBC(P), AEBC(Q) 값인 3보다 작다. 이는 C 범주의 ABC1, ABC2의 값도 이 조건을 만족하지 못함을 의미한다. P; Q의 구조는 P나 Q보다 균형을 잃을 가능성이 많지만 항상 그럴 수는 없다. 그러므로 BRS는 조건5를 만족하지 못한다.



(그림 1) 범주 E의 Weyuker 조건5 불만족 예

앞에서 기술한 [5]의 비판과 관계된 부분은 여러 연구에서 나타난다 [17]은 구조적 프로그램에서 루프의 반은 전체 루프보다 의미적인 면에서 미흡하다고 했다 이는 객체지향 시스템에서 완전한 클래스는 불완전한 클래스보다 이해하기 쉽다는 의미로 해석 가능하다[18]. 즉 이는 조건5가 객체지향 시스템에는 적당한 조건이 아니며, 여러 형(클래스)들의 인스턴스(객체)들로 구성되는 SDL 시스템에도 적당치 않다는 것이다

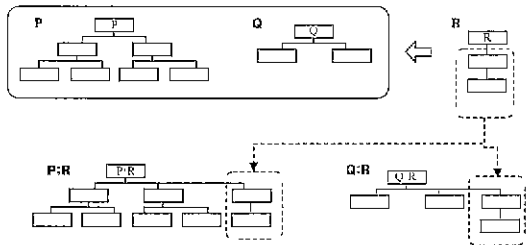
조건6)은 P의 R의 상호작용과 Q와 R의 상호작용이 서로 달라 P; R과 Q; R, 그리고 R; P와 R; Q의 복잡도를 다르게 한다는 것이다. (그림 1)과 같은 가정 하에서 그린 (그림 2)를 보면, EBC(P)와 EBC(Q)는 4이지만 EBC(P; R)은 2이고 EBC(Q; R)은 8이므로 서로 다르다. R; P와 Q; R의 EBC값 역시 같은 이유로 다르다. EBC에 영향을 받는 AEBC의 ABC1, ABC2도



(그림 2) 범주 E의 Weyuker 조건6 만족 예

두 조합 개체에 대해 다른 값을 가지므로 범주 C도 이 조건을 만족한다.

BRS 역시 조건6을 만족한다 구조 설계 단계의 두 블록 P, Q가 있을 때 P, Q를 구성하는 부분블록들은 끝단 블록까지 서로 같은 형의 인스턴스들이 없다고 가정하면 (그림 3)과 같은 블록들의 조합 구조를 생각할 수 있다 그림의 상단부는 P, Q의 구조이며 여기에 각각 R이 합쳐졌을 때 그림 밑부분의 P; R, Q; R의 구조가 생성된다 BRS(P) = BRS(Q) = 0이며, BRS(P; R)은 0이지만 BRS(Q; R)은 약 0.22가 되므로 이는 조건6을 만족하는 예이다 R, P와 R, Q도 마찬가지로 구조가 성립한다.



(그림 3) BRS의 Weyuker 조건6 만족 예

식 (2)의 의미는 관심 개체 P, Q가 P, Q로 합쳐질 때, P, Q의 구성원의 개수가 P와 Q의 구성원 개수의 합보다 줄어들 수 있다는 것이다 예를 들면 두 블록에 같은 프로세스 형의 인스턴스가 들어있다면 이들은 하나로 될 수 있다. SDL 문법 중 NEXTSTATE 상태는 다른 프로세스에 존재하므로 하나로 합쳐질 수 있고, 외부로 나가고 들어오는 채널이나 라우트들도 합쳐질 수 있다 그러므로 범주 I도 조건6을 만족한다. 그에 따라 IC 범주도 이 조건을 만족한다

조건7)을 SDL 관심 개체에 확장시키는 것은 문제점이 있다. 원래 정의에서 프로그램 P의 문장들을 재순열한 것의 의미 때문이다 Weyuker나 다른 연구자들은 이 조건을 만족시키는 예로서 문장을 재순열 했을 때에도 프로그램이 의미 있는(자동하는) 예를 들었다. 그러나 SDL 시스템의 블록을 생각해 보면 그 안의 구성원인 프로세스들을 재순열 한다는 것은 의미 없는 블록을 탄생시킨다. 블록 안의 프로세스들의 순서 자체의 정의가 모호하기 때문이다. 하지만 상세 설계 단계의 프로세스는 태스크나 결정 등의 연산들을 같은

집이 안에서서는 재순열 할 수 있다. 그러나 SDL 매트릭 집합은 관심 개체 내부의 크기에 의존하므로 이 조건을 만족하지 않는다 또한 후진 전이의 수를 하나로 정의하였기 때문에[9] 일반 전이에 있던 연산을 후진 전이로 옮겨도 매트릭 값은 일정하다.

Weyuker의 의도는 if-then-else나 루프들의 네스팅이 많은 경우에는 프로그램 문장들을 이런 구조들의 여러 부분(안과 밖)에 위치 시키는 것이 복잡도에 영향을 미쳐야 된다는 것으로 생각된다 그러므로 조건7은 프로그램 논리가 중요시 되는 과거의 프로그램 방식에 적합하지 객체지향 시스템에는 적합치 않으며[13, 16], SDL 시스템에도 적합치 않다.

**조건8)**은 관심 개체의 이름을 바꿔도 복잡도는 같아야 한다는 의미이다 SDL 매트릭 집합은 개체의 이름에 의존하는 것이 없으므로 이 조건을 만족한다.

**조건9)**는 두 관심 개체의 조합은 구성된 개체들의 상호 작용 때문에 복잡도가 각 구성 개체 복잡도의 합보다 커질 수 있다는 것이다.

(1)에 의해  $(\forall P)(\forall Q) (I_i(P; Q) \leq I_i(P) + I_i(Q))$ 와  $(\forall P)(\forall Q) (O_i(P; Q) \leq O_i(P) + O_i(Q))$ 이 성립한다 이의 부정은  $(\exists P)(\exists Q) (I_i(P; Q) > I_i(P) + I_i(Q))$ 와  $(\exists P)(\exists Q) (O_i(P; Q) > O_i(P) + O_i(Q))$ 이며 이는 조건9이다. 그러므로 범주 I, E는 조건9를 만족 못한다.

(2)와 (3)에 의해 역시 다음 식이 성립한다.

$$(\forall P)(\forall Q) (\sum C_i(P; Q) \leq \sum C_i(P) + \sum C_i(Q))$$

$$(\forall P)(\forall Q) (IC(P; Q) \leq IC(P) + IC(Q))$$

이 역시 조건9의 부정이므로 범주 IC도 이 조건을 만족 못한다.

범주 EC의 조건 만족 여부를 보이기 위해 다음 조건을 만족하는 관심 개체 P', Q'가 있다고 하자.

$$I_1(P') > 0, O_1(P') > 0, I_1(Q') > 0, O_1(Q') > 0 \quad (i=1, 2)$$

$$I_i(P'; Q') = I_i(P') + I_i(Q'), O_i(P'; Q') = O_i(P') + O_i(Q')$$

$$C_i(P'; Q') = C_i(P') - C_i(Q')$$

그러면 EC와 IC에 관한 다음과 같은 식들이 성립한다

$$EC(P') + EC(Q') = \sqrt{\sum(I_i(P') \times O_i(P'))} + \sqrt{\sum(I_i(Q') \times O_i(Q'))} \quad (5)$$

$$EC(P'; Q') = \frac{\sqrt{\sum(I_i(P') \times O_i(P')) + \sum(I_i(Q') \times O_i(Q'))}}{\sqrt{\sum(I_i(P') \times O_i(Q')) - \sum(I_i(Q') \times O_i(P'))}} \quad (6)$$

$$IC(P'; Q') = \sum C_i(P'; Q') = \sum(C_i(P') + C_i(Q')) = \sum C_i(P') + \sum C_i(Q') = IC(P') + IC(Q') \quad (7)$$

$I_i$ 와  $O_i$  결과값들은 모두 독립적인 값들이므로, 이들 매트릭 벡터는 다음과 같은 경우가 가능하다.

$$(I_1(P'), O_1(P'), I_1(Q'), O_1(Q'), I_2(P'), O_2(P'), I_2(Q'), O_2(Q')) = (2, 3, 1, 5, 5, 2, 2, 5)$$

이 경우에  $EC(P') - EC(Q')$ 와  $EC(P'; Q')$ 의 관계는 다음과 같다.

$$EC(P') - EC(Q') = \sqrt{16} + \sqrt{15} < \sqrt{16 + 15 + 35 + 7} = EC(P'; Q') \quad (8)$$

그러므로 범주 EC는 조건9를 만족한다.

C 범주의 만족 여부 판정은 위의 P', Q'를 다시 이용한다 (5), (6), (7), (8)에 의해 다음과 같은 식이 성립한다.

$$CI(P') + CI(Q')$$

$$= w_i IC(P') + w_e EC(P') + w_i IC(Q') + w_e EC(Q')$$

$$= w_i (IC(P') + IC(Q')) + w_e (EC(P') + EC(Q'))$$

$$= w_i IC(P'; Q') - w_e (EC(P') + EC(Q'))$$

$$< w_i IC(P'; Q') + w_e EC(P'; Q') = CI(P'; Q')$$

$$C2(P'; Q')$$

$$= IC(P'; Q') \times EC(P'; Q')^2$$

$$= (IC(P') - IC(Q')) \times (\sum(I_i(P') \times O_i(P')) + \sum(I_i(Q') \times O_i(Q')) + \sum(I_i(P') \times O_i(Q')) + \sum(I_i(Q') \times O_i(P')))$$

$$= (IC(P') - IC(Q')) \times (EC(P')^2 + EC(Q')^2 + \sum(I_i(P') \times O_i(Q')) + \sum(I_i(Q') \times O_i(P')))$$

$$> IC(P') \times EC(P')^2 + IC(Q') \times EC(Q')^2 = C2(P') + C2(Q')$$

그러므로 범주 C는 조건9를 만족한다

BRS는 조건9를 만족한다. 예를 들면 (그림 3)에서  $BRS(Q) + BRS(R)$ 은 0이지만  $BRS(Q; R)$ 은 약 0.22이므로 이보다 크다.

<표 3> Weyker 조건 집합 이용 SDL 매트릭 집합 검증 결과

단계	핀 심 개	범주	조건 1	조건 2	조건 3	조건 4	조건 5	조건 6	조건 7	조건 8	조건 9
구 조 선 계	블록	S	○	○	○	○	×	×	○	○	○
		E	○	○	○	○	×	×	○	○	○
		I	○	○	○	○	○	○	○	○	○
		IC	○	○	○	○	○	○	○	○	○
		EC	○	○	○	○	○	○	○	○	○
		C1	○	○	○	○	○	○	○	○	○
상 세 설 계	프로 세스	E	○	○	○	○	○	○	○	○	○
		I	○	○	○	○	○	○	○	○	○
		IC	○	○	○	○	○	○	○	○	○
		EC	○	○	○	○	○	○	○	○	○
		C1	○	○	○	○	○	○	○	○	○
		C2	○	○	○	○	○	○	○	○	○
	블록	E	○	○	○	○	○	○	○	○	○
		I	○	○	○	○	○	○	○	○	○
		IC	○	○	○	○	○	○	○	○	○
		EC	○	○	○	○	○	○	○	○	○
		C1	○	○	○	○	○	○	○	○	○
		C2	○	○	○	○	○	○	○	○	○

<표 3>은 Weyker의 조건 집합을 이용해 SDL 복잡도 매트릭 집합을 검증한 결과이다. ○는 조건 만족, ×는 조건 불만족, △는 해당 범주에 속하는 매트릭들 중 만족하는 것과 만족하지 않는 것이 있다는 뜻이다. SDL 매트릭 집합 검증에 부적당한 부분은 회색으로 나타내었다. 범주 C는 가중합 형태를 C1, 가중곱 형태를 C2로 분리하여 나타내었다.

조건을 만족시키지 않은 부분은 앞의 기술과 같이 대부분 Weyker의 P, Q와 P=Q 정의의 애매성과 불충분성, 그리고 그들을 SDL 시스템으로 확장할 때의 어려움 때문에 발생하였다. Weyker는 개체의 복잡도를 내부 복잡도와 외부 복잡도의 조합이 아니라 전자만 고려하거나, 혹은 후자만을 고려한 조건도 제시하였으므로 여러 연구자들의 비판과 같이 Weyker의 조건 집합 자체에 비일관성이 존재하고 있기 때문이다.

3.2 차원 분석(Dimensional analysis)

Weyker의 조건 집합을 이용한 공리적 검증과 함께 매트릭의 분석적 검증에서 중요한 부분은 차원 분석에 의한 매트릭 검증이다. 이는 서로 다른 매트릭들 간에 산술적 행위가 이루어질 때 고려되는 문제들을 포함한다. 차원은 매트릭의 단위(unit)의 일반화 형태이다. 아직 소프트웨어 정량화를 위해 널리 알려지고 채택된 기본 단위 집합은 없지만 차원 분석의 몇 가지 규칙은

매트릭 검증에 유용하게 사용할 수 있다[19]

첫번째 규칙은 등식의 우변과 좌변의 차원이 같아야 한다는 것이다. 예를 들면 프로그래밍의 생산성이 한달간 생산한 LOC로 정의된다면 생산성 매트릭의 단위는 LOC/Men-Month가 되는 것이 타당하다. 관심 개체에 관련된 여러 구성원들의 수를 나타내는 측정 단위를 개수라 하면, SDL 복잡도 매트릭 집합에서 기본 복잡도 매트릭 집합의 단위는 BRS를 제외하면 모두 개수이다. IC 범주의 매트릭 단위는 기본 복잡도 매트릭들의 합이므로 역시 개수이며, EC 범주의 매트릭은 제공된 연산자를 사용하여 단위가 개수가 되게 하였다. C 범주의 두가지 매트릭 형태는 가중합 매트릭은 단위가 개수이며, 가중곱 매트릭은 개수<sup>2</sup>이 된다. BRS를 제외한 SDL 복잡도 매트릭은 복잡도의 기본 단위를 개수로 사용하였으므로, 복잡도를 정의하고 나타내는데 일관성이 있는 매트릭 집합이다.

두번째 규칙은 매트릭 간의 산술 연산은 같은 단위를 갖는 매트릭 간에 이루어지는 것이 바람직하다는 것이다 이는 첫번째 규칙과 일맥 상통한다. SDL 매트릭 집합 정의에 사용된 연산은 대부분 같은 단위(개수나 개수<sup>2</sup>)에서 이루어지며, 가중곱 매트릭의 경우만 개수와 개수<sup>2</sup>을 곱한다. 그러나 이 규칙은 곱셈보다는 덧셈, 뺄셈 연산에 더 엄격함을 둔다. BRS를 C 범주의 매트릭 제작에 포함시키지 않은 이유는 BRS가 나타내는 구조 복잡도의 가중치가 알려지지 않았다는 이유도 있지만, BRS의 기본 단위가 개수가 아닌 수준이라는 이유 때문이다.

4. 결론 및 향후 연구

소프트웨어 개발 초기 단계의 문제점이 개발 후반부 산물의 품질에 심각한 영향을 미치기 때문에 설계 단계를 정량화 하는 설계 매트릭들은 전체 시스템 개발 비용을 낮추는 데 중요한 역할을 하고 있으며, 설계 정량화는 결과 산물이 매우 크고 실행 정확성이 요구되는 통신 소프트웨어 같은 실시간 시스템 설계에 더욱 필요하다. 여러 방법론과 그에 따른 명세 방법들을 위한 매트릭들이 제안되었으나 실시간 시스템 명세 언어들 위한 매트릭 연구는 극소수에 불과하며 이들 연구들도 대부분 객체지향 개념을 지원하지 않는 언어들 연구 대상으로 하였다.

본 논문에서는 선행 연구인 [9]에서 제안한 SDL 메



트릭 집합을 OSDL을 지원하는 메트릭 집합으로 확장하고, 조합 메트릭의 형태를 수정하였으며 결과 메트릭 집합을 분석적으로 검증하였다. 분석적 검증 방법으로는 공리적 검증 방법과 차원 분석을 사용하였으며 기존 메트릭 형태의 부분 수정은 이들 검증 조건을 만족하는 방향으로 이루어졌다.

향후 연구는 SDL 메트릭 집합을 복잡도와 관련된 여러 품질 인자를 예측에 사용하여 본 메트릭 집합의 유용성을 보이는 것이다. SDL은 개발 기간이 상당히 긴 수십만 LOC 이상의 통신 소프트웨어 시스템이나 실시간 시스템 설계에 사용되므로 설계 단계에서 구현 단계 산물의 품질을 미리 예측할 수 있는 모델은 시스템 개발비용을 낮추는데 매우 중요한 역할을 한다. 복잡도와 밀접한 관계가 있다고 알려진 유지보수성, 결함경향성 등을 예측하는 예측 모델을 제작하고 실험을 통해 모델을 검증할 것이다

### 참 고 문 헌

- [1] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber. Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-24, 1993
- [2] M. C. Paulk, C. V. Weber, S. Garcia, M. B. Chrissis, and M. Bush, Key Practices of the Capability Maturity Model, Version 1.1, CMU/SEI-93-TR-25, 1993.
- [3] International Organization for Standardization, Quality Management and Quality System Elements (Part 3) Guidelines for Development, Supply and Maintenance of Software, ISO 9000-3.
- [4] International Organization for Standardization, Quality Systems Model for Quality Assurance in Design/Development, Production, Installation, and Servicing, ISO 9001
- [5] N. Fenton, "Software Measurement - A Necessary Scientific Basis," *IEEE Trans Software Eng.*, Vol.20, No.3, pp.199-206, March 1994
- [6] H. Zuse, *Software Complexity Measures and Methods*, Walter de Gruyter, 1991.
- [7] L. O. Ejiogu. "Five principles for the formal validation of models of software metrics," *ACM SIGPLAN Notices*, Vol 28, No 8, pp.67-76, 1993.
- [8] E. J. Weyuker, "Evaluating Software Complexity Measures," *IEEE Trans. Software Eng.*, Vol.14, No.9, pp.1357-1365, Sept 1988.
- [9] 홍의식, 홍성택, 김갑수, 우치수, "SDL 설계 복잡도 메트릭 집합". 정보과학회논문지(B), 제24권 제10호, pp.1053-1062, 1997.
- [10] J. Ellsberger, D. Hogrefe. and A. Sarma, *SDL - formal object-oriented language for communicating systems*, Prentice Hall, 1997.
- [11] N. F. Schneidewind, "Methodology for Validating Software Metrics," *IEEE Trans. Software Eng.*, Vol 18, No 5, pp 410-421, May 1992.
- [12] B. Kitchenham, S. L. Pfleeger, and N. Fenton, "Towards a Framework for Software Measurement Validation," *IEEE Trans. Software Eng.*, Vol.21, No 12, pp.929-944, Dec. 1995.
- [13] J. C. Cherniavsky and C. H. Smith, "On Weyuker's axioms for software complexity measures," *IEEE Trans Software Eng.*, Vol.17, No.6, pp.636-638, June 1991.
- [14] B. Kitchenham, S. L. Pfleeger, and N. Fenton, "Reply to Comments on Towards a Framework for Software Measurement Validation," *IEEE Trans Software Eng.*, Vol.23, No.3, p.189, March 1997.
- [15] S. Morasca, L.C. Briand, V.R. Basili, E.J. Weyuker, and M.V. Zelkowitz, "Comments on Towards a Framework for Software Measurement Validation," *IEEE Trans. Software Eng.*, Vol.23, No 3, pp 187-188, March 1997.
- [16] S. R. Chidamber and C. F. Kemerer, "A Metrics Suite for Object Oriented Design." *IEEE Trans. Software Eng.*, Vol.20, No 6, pp 476-493, June 1994.
- [17] A. V. Melton, D. A. Gustafson, J. M. Bieman, and A. L. Baker. "A mathematical perspective for software measures research," *Software Eng. J.*, pp.246-254, Sept., 1990
- [18] B. Henderson-Sellers, *Object Oriented Metrics-Measures of Complexity*, Prentice Hall, 1996
- [19] B. Henderson-Sellers. "The Mathematical Validity of Software Metrics," *Software Eng. Notes*, Vol.21, No.5, pp.89-94, 1996.



### 홍 의 석

e-mail hes@aycc.anyang.ac.kr

1992년 서울대학교 계산통계학과  
졸업(학사)

1994년 서울대학교 대학원 계산  
통계학과(이학석사)

1999년 서울대학교 대학원 전산  
과학과(이학박사)

1999년~현재 안양대학교 영상처리학과 전임강사  
관심분야: 메트릭 기반 소프트웨어 품질 예측 모델,  
웹 기반 멀티미디어 응용 기술 등



### 정 명 희

e-mail : mhjung@avcc.anyang.ac.kr

1989년 서울대학교 계산통계학과  
졸업(학사)

1991년 University of Texas at  
Austin(공학석사)

1997년 University of Texas at  
Austin(공학박사)

1997년~1998년 삼성 데이터 시스템 근무  
1998년~현재 안양대학교 영상처리학과 전임강사  
관심분야: 원격 탐사 영상 처리, 소프트웨어 품질  
데이터 처리 등