

프로세스 유사도 분석과 메타 저장소를 이용한 서비스 재사용 기법

황치곤¹ · 윤창표² · 정계동^{3*}

A Method of Service Reuse using Analysis of Process Similarity and Meta Repository

Chi-Gon Hwang¹ · Chang-Pyo Yoon² · Kye-Dong Jung^{3*}

¹Department of Computer Science, Kwangwoon University, Seoul 139-701, Korea

²Department of Computer & Mobile Convergence, Gyeonggi College of Science and Technology, Gyeonggi 429-792, Korea

^{3*}Division of General Education Information Engineering, Kwangwoon University, Seoul 139-701, Korea

요 약

클라우드 컴퓨팅에서 SaaS는 소프트웨어를 서비스로서 제공하는 프레임워크이다. 테넌트와 용도의 차이에 따라 서비스 제공자가 서비스를 다시 구축한다면 비용이나 관리적 측면에서 많은 자원을 요구하게 된다. 이에 우리는 기존 소프트웨어를 재사용 할 수 있도록 프로세스 알제브라를 이용하여 분석하는 기법을 제안한다. process algebra는 소프트웨어의 구조를 분석하고, 이를 다른 언어로 표현할 수 있으며, 재사용할 수 있는지 검증할 수 있다. process algebra 중 CCS(Calculus of Communicating Systems)는 비즈니스 프로세스나 XML로 변환하기 유용하므로, 이를 이용하여 프로세스를 프로세스 뷰로 구조화하고, 구조화된 명세를 비교 및 관리를 위한 메타 저장소를 제안한다.

ABSTRACT

SaaS at a cloud computing is a framework to provide a software as a service. Depending on the difference of the tenant and the use, if the service provider re-establish a service, they are required resources in terms of costs and managerial. So we propose a technique for analysis of software structure using the process algebra to reuse existing software. A process algebra analyzes the structure of the software, expresses it in different languages and verifies that it can be reused. CCS in a process algebra is useful to convert the business process or XML, by using this, we structure a process as process view and propose meta storage for comparison and management of a structured document.

키워드 : 클라우드 컴퓨팅, SaaS(software as a service), 프로세스 알제브라, 메타 저장소, 프로세스 재사용

Key word : Cloud Computing, SaaS(software as a service), Process Algebra, Meta Storage, Process Reuse

접수일자 : 2014. 05. 01 심사완료일자 : 2014. 05. 30 게재확정일자 : 2014. 06. 09

* **Corresponding Author** Kye-Dong Jung(E-mail:gdchung@kw.ac.kr, Tel:+82-2-940-5288)

Division of General Education Information Engineering, Kwangwoon University, Seoul 139-701, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2014.18.6.1375>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

클라우드 환경에서는 네트워크 상에 소프트웨어를 게시하여 사용자는 자신의 컴퓨터에 소프트웨어를 설치하지 않고 사용할 수 있도록 서비스로 제공하고 있다. 이러한 클라우드 컴퓨팅 기술을 SaaS(Software As A Service)라고 한다[1]. 이 기술은 사용자가 소프트웨어를 설치하지 않고, 서비스 상에 설치된 소프트웨어를 이용함으로써 소프트웨어의 재사용성을 향상시킨다. 이 소프트웨어는 모듈별로 분할하여 재사용할 수 있으며, 하위 모듈로 분할할 수 있고, 결합하여 상위 모듈로 사용할 수 있다. 이러한 모듈들을 프로세스라 하고, 이 프로세스의 재사용성 여부를 확인하기 위한 프로세스 유사도 분석 방법을 제안한다.

이러한 프로세스들을 표현하고 분석하기 위한 기법으로 프로세스 알제브라가 있다. 프로세스 알제브라는 프로세스를 표현하기 위한 정형 명세 기법으로는 자동차의 AUTOSAR(AUTomotive Open System ARchitecture), 실시간 시스템을 위한 정형 명세를 위한 ACSR(Algebra of Communicating and Shared Resources), 정형 명세 기법의 기반이 되는 CCS과 같은 것들이 있다[2,3]. 본 기법은 프로세스를 분할하고 분석하여, 이를 CCS로 표현하여 저장하고, CWB-NC 툴을 사용하여 다른 프로세스와 비교함으로써 유사도를 분석한다[4]. 적용되는 사례는 온라인 결제 시스템의 일반적인 사례를 기반으로 수행한다. 수행 순서는 프로세스를 모듈로 분할하여 작업 관계를 정의하고, 이를 정형 명세 기법인 CCS로 표현하여 이를 저장할 수 있는 메타 저장소의 형식을 제시한다. 프로세스가 필요할 경우, 메타 저장소에 저장된 정보를 바탕으로 프로세스의 유사성을 비교할 수 있다.

이에 따라 본 논문의 2장은 프로세스의 비교와 분석을 위한 관련 기술들을 기술하고, 3장은 프로세스 유사성 비교를 위한 시스템을 제안하고, 4장은 프로세스를 비교하는 방법과 메타 저장소의 운영 방안을 제시한다. 그리고 5장에서는 결론을 기술한다.

II. 관련연구

CCS 프로세스는 agent로 액션(action)들의 집합으로 다음과 같이 표현된다[4,5]. 이에 따른 action은 명시적

인 액션과 묵시적인 액션으로 나눌 수 있다. 명시적인 액션은 a, b와 같이 기술되고, 묵시적인 액션은 내부적인 액션으로 어떤 액션이라도 할 수 있음을 의미한다. 이 묵시적 액션은 τ 로 표현한다. 이때 입력으로 발생하는 액션 a가 발생하면, 이에 대한 출력으로 발생하는 액션을 'a'이라 한다. 입력 액션이 'a'로 발생하면, 출력 액션은 ''a'를 발생한다.

이 ''a'는 a와 같다. 0은 nil로 표현하고, 프로세스의 끝(terminate of process)을 의미이다. '.'는 sequence로, a.b는 액션 a다음에 b가 수행된다는 것으로 프로그래밍에서 순차 처리에 해당한다. '+'는 choice operation으로, P+Q는 두개의 프로세스(P, Q) 중에서 선택된 액션이 프로세스의 액션이 된다. 이는 선택 구조에 해당한다. '='는 반복으로 $K \stackrel{def}{=} P$ 와 같이 표현되며, 이는 P에서 지정된 액션 def를 수행함으로써 K가 반복됨을 나타낸다. '|'는 병렬구성으로, P|Q는 프로세스(P, Q)가 동시에 수행될 수 있음을 의미한다. '\'는 restriction set으로 P에서는 액션 A를 취할 수 없음을 표현한다. CCS는 이러한 규칙으로 프로세스를 구성할 수 있으므로, 프로세스의 구성 나타내고, 유사성 판단에 이용할 수 있다. 본 논문에서는 이를 이용하여 프로세스의 유사도를 비교한다.

III. 프로세스 유사성 비교를 위한 시스템 구성과 CCS 표현

프로세스 유사성 비교를 위한 시스템 구성과 프로세스의 정형화된 표현에 대해 기술한다. 다음으로 정형화된 프로세스의 비교 기법을 제시한다.

3.1. 시스템 구성

유사성 분석을 위한 시스템은 그림 1과 같이 두 가지 작업으로 구성된다. 첫째, 프로세스 생성 요구에 따라 프로세스를 생성하고 이를 CCS로 표현하여 메타 저장소에 저장한다. 이는 이후 유사 프로세스를 생성할 때 비교를 위해 저장된다. 둘째, 프로세스 생성 요구시 기존에 저장된 프로세스와 비교하여 프로세스의 유사도를 평가한다.

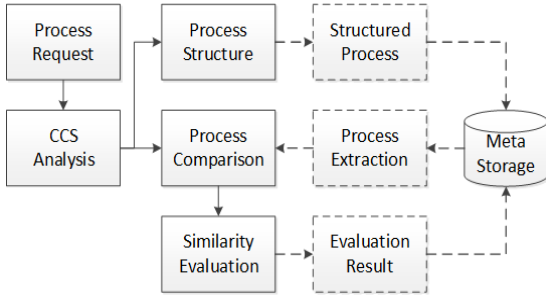


그림 1. 프로세스 유사도 분석을 위한 시스템
Fig. 1 System for Analysis of Process Similarity

그림 1에서 실선은 작업의 흐름이고, 점선은 작업의 결과물을 메타 저장소에 보관하고 참고하는 흐름이다. 시스템의 각 부분에 대한 내용은 다음과 같다.

- Process Request : 프로세스의 분석과 비교를 위한 요구를 받는 곳으로 시스템이 시작되는 요소이다.
- CCS Analysis : 프로세스의 요구에 따라 프로세스를 분할하여 분석한다. Process Comparison을 이용하여 신규 프로세스가 되어야 하는지 재사용할 것인지 여부를 판단한다.
- Process Structure : 분석된 프로세스를 바탕으로 CCS로 구성하여 메타 저장소에 저장한다. 메타 저장소에 등록되지 않은 프로세스를 등록하기 위해 사용된다.
- Process Comparison : 메타 저장소에 저장된 프로세스를 추출하여 요구되는 프로세스와 비교한다.
- Similarity Evaluation : 프로세스의 유사도를 평가하여 평가된 결과를 메타 저장소에 저장한다.

3.2. 프로세스 정형화를 위한 CCS 표현

이 절에서는 관련연구에서 제시한 CCS 표현에 따라 전자 결제 과정을 표현한다. 이를 위한 적용 사례인 결제 시스템은 그림 2와 같다.

그림 2에서 각 사각형은 결제 시스템을 구성하는 에이전트이며, 선들은 에이전트들 사이에 발생하는 액션들이다. 이 에이전트들은 메타 저장소에 저장되는 대상인 프로세스가 된다. 처리과정은 apply 에이전트를 통하여 요구(req)에 따라 시작되며, 정상적인 업무 처리인 지 확인하기 위한 Approve 에이전트에 검증 요구를 한다. Approve 에이전트는 검증 요구에 따라 Certify 에이

전트에서 검증을 수행(req)한다. 그리고 수행된 결과는 'req와 같이 '(Apostrophe)를 붙여서 해당 액션의 결과 액션임을 표현한다. 검증 과정이 완료되면 Apply 에이전트는 Banking 에이전트를 통하여 결제과정을 진행한다. Banking 에이전트는 이전에 진행한 Approve 에이전트의 검증된 결과 조회를 바탕으로 결제 서비스를 수행하고 Apply 에이전트와 Complete 에이전트에 통보한다. 최종적으로 Apply 에이전트는 Complete 에이전트를 통하여 결제 과정을 완료하는 흐름을 가진다.

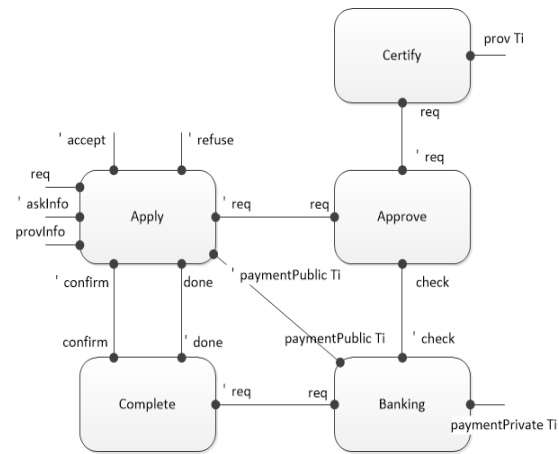


그림 2. 결제 시스템의 업무 관계
Fig. 2 Work Relations of Payment System

```

proc Apply =
  req.'askInfo.provInfo. (('accept.'req
    .'paymentPublic Ti.'confirm.done.Apply) +
    ('refuse.Apply))
proc Approve = req.'req.check.Approve
proc Certify = req.prov Ti.Certify
proc Banking = 'check.Banking +
  paymentPublic Ti
  .paymentPrivate Ti.req.Banking
proc Complete =
  'req.confirm.'done.Complete
    
```

그림 3. 각 프로세스들의 CCS 표현
Fig. 3 CCS Expressions of Each Processes

이러한 과정에서 각 역할을 수행하는 에이전트들을 CCS로 표현하면 그림 3과 같다. Apply 에이전트는 사용자의 요구 req를 받고, askInfo의 응답, provInfo의 액

션을 순차적으로 수행한 후 accept의 응답과 refuse의 응답을 선택적으로 처리하게 되며, accept와 refuse의 내부 액션을 수행한 후 본래의 에이전트인 Apply로 돌아간다. 이러한 과정은 그림 2에서 표현하였고, 다른 에이전트들과 각자의 액션에 따라 수행된다.

IV. 메타 저장소와 프로세스 비교

4.1. 메타 저장소

CCS에 의해 분석된 프로세스는 프로세스 메타 정보, 온톨로지 정보, 프로세스 분석 결과물을 저장하여 관리하고, 프로세스 비교할 수 있도록 메타사전에 저장한다. 프로세스 메타 정보는 프로세스 식별자, 소프트웨어 이름, 에이전트 이름, 액션 정보이다. 온톨로지 정보는 프로세스, 액션의 이름 및 메타 정보에서 발생할 수 있는 이질성 해결을 위해 저장된 연관관계 정보이다. 프로세스 분석 결과물은 CCS에 의해 표현된 구조화된 문장이다. 이러한 메타 정보의 저장은 유사 프로세스의 비교 및 분석을 위해 필요한 정보로 그림 4와 같이 구성된다.

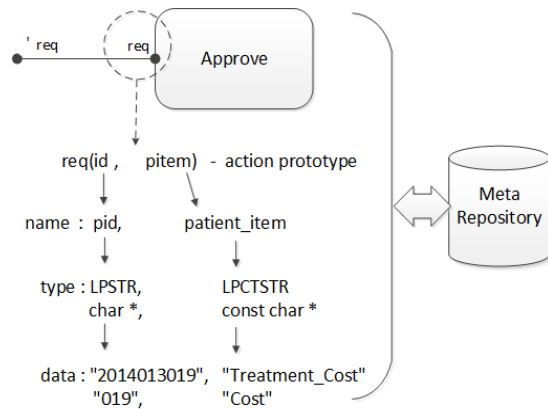


그림 4. 프로세스 재사용을 위한 분석과 이질성
Fig. 4 Analysis for Process Reuse and Heterogeneity

그림 4는 에이전트에서 사용되는 액션들에 대한 이질적인 문제를 표현하고 이러한 문제는 프로세스의 재사용을 위해서 반드시 해결되어야 하는 문제이다. 이를 위해서 메타 저장소에 액션의 호출에 이용되는 데이터들에 대한 이질성 해결 규칙을 저장한다. req(id, pitem)

는 req 액션을 호출하기 위한 기본 프로토 타입을 나타낸다.

이에 데이터의 충돌의 원인이되는 이질적인 문제가 나타날 수 있는 것은 파라미터의 이름인 name이 id, pid로 이용되고, pitem, patient_item으로 이용될 수 있다. 이러한 부분은 <is>연산을 이용하여 id <is> ID, pid <is> ID로 설정하여 메타 저장소에 저장함으로써 모두 ID임을 지정한다. 그리고 어떤 프로세스에서 사용되는지를 위한 프로세스 한정자를 추가함으로써, [액션이름].[파라미터이름]으로 표현하여 설정한다. 이를 통해서 파라미터 이름으로 나타나는 이질성은 메타 저장소에 Approve.id <is> Approve.ID와 같은 방식으로 표현한다. 이 파라미터는 명칭의 이질성 뿐만 아니라 구조적 이질성을 가질 수 있다. 대표적으로 사람의 이름을 표현할 경우 성과 이름을 분리하는 경우와 함께 사용하는 경우의 차이가 있다. 이러한 경우는 <comp>연산을 통해 Apply.patient.name <comp> Apply.patient.firstname + Apply.patient.lastname 와 같이 저장한다.

다음으로 데이터 타입에 의해 발생하는 문제는 Approve.id.LPSTR <is> Approve.ID.char*와 같이 저장함으로써 매핑을 통해서 조정을 거칠 수 있도록 한다. 데이터에 의해 발생하는 문제도 앞에서 언급한 이름 문제와 타입 문제와 같은 표현으로 메타 저장소를 이용한다.

4.2. 프로세스 비교 및 분석

CCS에 의해 기술된 프로세스들은 CWB-NC 틀에서 제공되는 기능을 이용하면 많은 기능을 수행할 수 있다. 그 기능은 문법적 오류 검사, 타입 검사뿐만 아니라, 서비스들 간의 상호작용에서 발생할 수 있는 교착상태 가능성, 기존 버전과 새로운 버전 사이에서 발생하는 행위의 동시성 검증을 확인할 수 있다. 이러한 검증 과정은 SaaS 애플리케이션 개발 과정에 추가함으로써 보다 안정적인 SaaS 애플리케이션을 개발할 수 있다.

위의 틀을 이용하여 제공되는 유사한 프로세스라고 할 수 있는 사례는 다음과 같다.

$$proc\ ap = a.b.(c0 + c1).nil \quad (1)$$

(1)에서 프로세스 ap는 액션 a와 액션 b를 순차적으로 수행하고, 액션 c0와 c1 중에 하나를 선택하여 수행

한 후 완료되는 프로세스이다. 이 프로세스와 유사한 프로세스로 (2)와 같은 프로세스가 있다.

$$proc\ ap1 = a.(b.c0 + b.c1).nil \quad (2)$$

(2)는 액션 a가 수행된 이후 b.c0와 b.c1 중 하나를 선택해서 작업을 수행하는 프로세스이다. 이러한 경우 프로세스 ap와 ap1은 거의 동일한 프로세스라 볼 수 있다.

용할 수 있도록 매핑된 가상의 프로세스이다. 기존의 프로세스가 존재하면 구조분석을 통하여 유사한 프로세스가 발견되면 프로세스를 조정하여 프로세스에서 사용하는 액션의 이름, 타입, 값을 조정하기 위한 부합 검사와 조정 과정을 거쳐 프로세스 뷰를 생성하여 메타 저장소에 저장하여 필요에 따라 사용될 수 있도록 한다. 메타 저장소에 저장된 프로세스 뷰는 프로세스와 같이 이용된다.

V. 결 론

클라우드 컴퓨팅 환경에서 SaaS플랫폼을 위한 서비스 재사용을 위한 방안으로 프로세스를 비교 하여 재사용의 필요성이 제기된다. 이에 따라 본 논문에서는 프로세스를 재사용하기 위한 프로세스의 표현과 비교 기법을 제시하였다. 제시된 방안은 프로세스를 분석하여 CCS로 구조화하고, 이 프로세스에서 사용되는 액션들의 파라미터의 이질적 문제를 해결하기 위해 구조화하여 메타 저장소에 저장하였고, 이렇게 재사용하기 위해 저장된 프로세스를 프로세스 뷰라 하여 이를 호출하여 사용함으로써 재사용의 가능성을 제시했다.

이러한 기법을 활용하면 프로세스를 정형화된 표기법으로 표현하고, 프로세스의 유사성을 확인할 수 있다. 프로세스 뷰라는 형태로 저장함으로써 기존의 프로세스를 재사용할 수 있도록 한다. 향후 본 논문은 유사성 분석의 자동화를 위한 비교 방안과 검증 기법에 대한 부분이 필요로 한다.

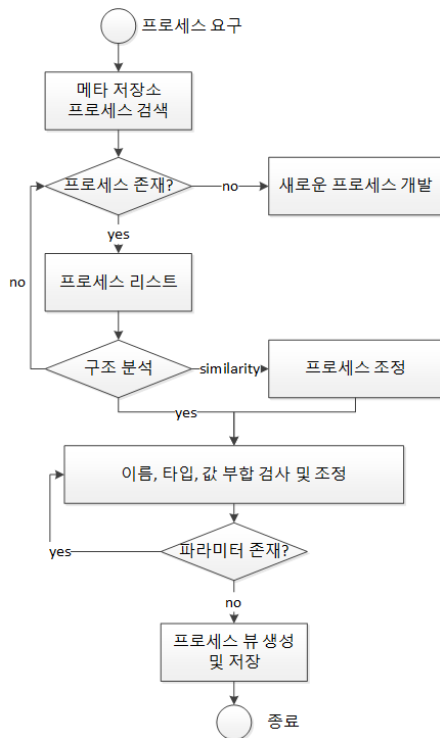


그림 5. 프로세스 재사용을 위한 수행과정
Fig. 5 workflow for process reuses

위와 같은 방법과 메타 저장소를 통한 프로세스 재사용은 그림 5와 같은 알고리즘에 따라 수행된다. 먼저 프로세스의 필요성에 따라 요구가 발생하고, 해당 프로세스의 존재유무를 확인하여 없으면 새로운 프로세스를 개발한다. 그러나 유사 프로세스가 존재한다면 메타 저장소에서 프로세스들과 프로세스 뷰들을 검색한다. 프로세스 뷰란 재사용할 수 있는 프로세스는 신규로 개발하지 않고 기존 프로세스를 메타 저장소를 통하여 재사

REFERENCES

- [1] Wu, L., Garg, S. K., Buyya, R., "SLA-based resource allocation for software as a service provider in cloud computing environments," *CCGrid 2011, 11' IEEE/ACM International Symposium*, pp.195-204, 2011.5.
- [2] Jinhyun Kim, Inhye Kang, Pyoung Soo Kim, Sungwon Kang, "Verification of Timing Properties for Software Behaviors on AUTOSAR Architecture using Process Algebra," *Journal of KISS(B): Software and Applications*, Vol.38, No.1, pp.633-645, 2011.11.
- [3] G. Salaun, L. Bordeaux, M. Schaerf, "Describing and

- Reasoning on Web Services using Process Algebra," *International Journal of Business Process Integration and Management*, Vol.1, No.2, pp.116-128, 2006.6.
- [4] O.Sokolsky, I.Lee, D.Clarke, "Schedulability analysis of AADL models," *Parallel and Distributed Processing Symposium*, p.164, 2006.4.
- [5] Jeong rhan Choi, Young choon Woo, Chang hwan Bae, "A Study on Workflow Verification for Business Process Automation in SaaS Platform," *KCC2012*, Vol.39, No.1(B), pp.220-222, 2012.6.



황치곤(Chi-gon Hwang)

1995년 창원대학교 경영학과 경영학사
2004년 광운대학교 정보통신학과 공학석사
2012년 광운대학교 컴퓨터과학과 공학박사
2006년 ~ 현재 (주)인찬 연구원
※관심분야 : 모바일 클라우드, 멀티미디어 온톨로지, 클라우드 컴퓨팅, 데이터 상호운용



윤창표(Chang-Pyo Yoon)

1998년 광운대학교 전자계산학과 이학사
2001년 광운대학교 컴퓨터과학과 공학석사
2012년 광운대학교 컴퓨터과학과 공학박사
2012년 ~ 현재 경기과학기술대학교 컴퓨터모바일융합과 조교수
※관심분야 : 모바일 클라우드, 안드로이드 보안, 멀티미디어 온톨로지, 네트워크 프로토콜, 무선 네트워크, 네트워크 보안



정계동(Kye-dong Jung)

1985년 광운대학교 전자계산학 이학사
1992년 광운대학교 산업정보학 이학석사
2000년 광운대학교 컴퓨터과학 이학박사
1993년 ~ 2004년 광운대학교 정보과학원 교수
2005년 ~ 현재 광운대학교 교양학부 교수
※관심분야 : XML 분산시스템, 분산 컴퓨팅기술, 이동에이전트