
서비스 재사용을 위한 프로세스 유사도 분석에 관한 연구

황치곤* · 윤창표** · 정계동*

*광운대학교 · **경기과학기술대학교

A Study on Analysis of a Process Similarity for the Service Reuse

Chi-Gon Hwang* · Kye-Dong Jung*

*Kwangwoon University · **GyeongGi College of Science and Technology

E-mail : duck1052@kw.ac.kr

요 약

클라우드 컴퓨팅은 소프트웨어를 서비스로서 사용할 수 있도록 하는 SaaS 프레임워크를 포함한다. 기존의 서비스가 있음에도 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 비즈니스 프로세스나 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 구조화하고, 구조화된 명세를 비교 및 관리를 위한 메타 저장소를 제안한다.

ABSTRACT

A cloud computing include a SaaS frameworks be able to use a software as a service. Despite the existing 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 software structure using the process algebra to reuse existing software. A process algebra analyze the structure of the software, express in business process or different languages and verify that it can be reused. As CCS in a process algebra is useful to convert the business process or XML, by using this, we structure a process and propose meta storage for comparison and management a structured document.

키워드

Cloud Computing, SaaS(software as a service), process algebra, meta storage

1. 서 론

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 하

고, 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 CSS로 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 작업관계를 정의하고, 이를 정형 명세 기법인 CSS로 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 프로세스 분석을 위한 시스템을 제안하고, 3장은 프로세스 정형화를 위해 CCS 구문으로 프로세스를 기술하고, 메타사전에 저장하는 기법을 서술하고, 4장에서는 결론을 기술한다.

II. 관련연구

CSS 프로세스는 agent로 액션(action)들의 집합으로 다음과 같이 표현된다[4][5]. 이에 따른 action은 명시적인 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 'a이라 한다. 입력 액션이 'a로 발생하면, 출력 액션은 ''a를 발생한다. 이 ''a는 a와 같다. 0은 nil로 표현하고, 프로세스의 끝(terminate of process)이다. '.'는 sequence로, 액션 a다음에 b가 수행된다는 것을 의미하며 프로그래밍에서 순차 철리에 해당한다. '+'는 choice operation으로, 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 def을 수행함으로써 K가 반복됨을 의미한다. 이는 반복구조에 해당한다. '|'는 병렬구조으로, 프로세스(P, Q)가 동시에 수행될 수 있음을 의미한다. '\'는 restriction set으로 P에서는 액션 A를 취할 수 없다.

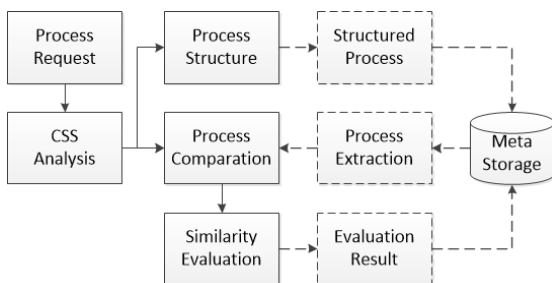


그림 1. 프로세스 유사도 분석을 위한 시스템

III. 프로세스 유사성 비교를 위한 시스템

3.1. 시스템 구성

유사성 분석을 위한 시스템은 그림 1과 같이 두 가지 작업으로 구성된다. 첫째, 프로세스 요구에 따라 CSS분석을 통하여 프로세스의 구조화를 통한 메타 저장소에 저장한다. 둘째, 요구에 따라 저장된 프로세스와 비교를 통한 유사도 평가를 수행한다.

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

- Process Request : 프로세스의 분석과 비교를 위한 요구를 받아들인다.
- ACC Analysis : 받아들인 요구에 따라 CSS의 표기법에 따라 프로세스를 분할하여 분석한다.
- Process Structure : 분석된 프로세스를 바탕으로 CSS로 구성한 프로세스를 메타 저장소에 저장한다.
- 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 CSS 표현

관련연구에서 제시한 CSS 표현에 따라 전자 결제 과정을 CSS로 표현한다. 이를 위한 결제 시스템은 그림 2와 같다.

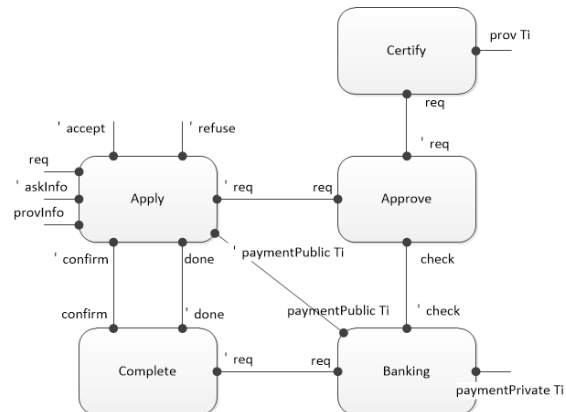


그림 2. 결제 시스템의 작업 관계

그림 2에서 각 사각형은 결제 시스템을 구성하는 에이전트이며, 선들은 에이전트들 사이에 발생하는 액션들이다. 이 에이전트들이 저장의 대상이 되는 프로세스가 되며, 각 에이전트들은 CSS로 표현하면 그림 3과 같고, Apply 에이전트는 사용자의 요구 req를 받고, askInfo의 응답, provInfo의 액션을 순차적으로 수행한 후 accept의 응답과 refuse의 응답을 선택적으로 처리하게 되며, accept와 refuse의 내부 액션은 개별적으로 수행

하게 하고 본래의 에이전트인 Apply로 돌아감을 나타낸다.

```

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

IV. 프로세스의 비교 및 메타 저장소

4.1 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.

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

여기에서 프로세스 ap는 액션 a와 액션 b를 순차적으로 수행하고, 액션 c0와 c1 중에 하나를 선택하여 수행한 후 완료되는 프로세스이다. 이 프로세스와 유사한 프로세스로 $proc\ ap1 = a.(b.c0 + b.c1).nil$ 와 같은 것이 있을 수 있다. 이러한 경우 프로세스 ap와 ap1은 거의 동일한 프로세스라 볼 수 있다.

4.2 메타 저장소

CSS에 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 정보이다.

V. 결 론

본 논문에서는 클라우드 컴퓨팅 환경에서 SaaS 플랫폼을 위한 서비스 재사용을 위한 방안으로 프로세스를 비교 분석을 위한 프로세스의 표현과 프로세서 비교 방법론은 제시했다. 제시된 방안을 활용하면 프로세스를 정형화된 표기법으로 표현할 수 있어 프로세스의 유사성을 확인할 수 있다. 이는 재사용 유무를 결정할 수 있는 중요한 지침이 될 수 있다. 향후 본 논문은 유사성 분석의 자동화를 위한 비교 방안과 검증 기법에 대한 부분이 필요로 한다.

참고문헌

- [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] 김진현, 강인혜, 김평수, 강성원, "프로세스 알레브라를 이용한 AUTOSAR 소프트웨어 행위의 시간 특성 검증 기법," 정보과학회 논문지 소프트웨어 및 응용, Vol.38, No.1, pp.633-645, 2011.11.
- [3] G. Salaun, L. Bordeaux, M. Schaefer, "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] 최정란, 우영춘, 배창환, "SaaS 플랫폼에서 비즈니스 프로세스 자동화를 위한 워크플로우 검증 적용에 관한 연구," 한국컴퓨터종합학술대회 논문집 Vol.39, No.1(B), pp.220-222, 2012.6