

시멘틱 기반의 의료 워크플로우 시스템 모델링 기법 (A Semantics-based Modeling Technique for Healthcare Workflow System)

이상영(Sang-Young Lee)¹⁾ 이윤석(Yoon-Suk Lee)²⁾

요 약

OMG가 UML 액티비티 다이어그램에 대한 텍스트 기반 시멘틱을 제공하지만 정확한 시멘틱의 부족으로 인하여 모호해지는 단점이 있다. OMG의 UML 액티비티 다이어그램이 제공하는 가이드로는 워크플로우 시스템의 정확한 시멘틱을 제공하지는 못한다. 이에 본 논문에서는 대안적인 방법으로 ASM을 사용하여 UML 액티비티 다이어그램의 정형화를 제시한다. 또한 워크플로우 시스템의 특성에 맞춰 ASM 시멘틱을 확장한다.

ABSTRACT

Although OMG gives a textual semantics for UML activity diagrams, some ambiguities can be caused due to the lack of rigorous semantics. That is to say it is difficult to present precise semantics which is taken as important in workflow system with the guide provided by OMG to the UML activity diagram. In this paper, the alternative approach of using Abstract State Machine(ASM) to formalize UML activity diagrams is presented. It suggested that ASM semantics is extended corresponding the workflow system characteristics.

key words : Semantics, ASM, Activity Diagrams, Workflow

논문접수 : 2005. 8. 1.

심사완료 : 2005. 8. 15.

1) 정회원 : 남서울대학교

2) 정회원 : 남서울대학교

본 논문은 2005년도 남서울대학교 학술연구조성비 지원에 의하여 연구되었음.

1. 서 론

현대의 기업 정보시스템에서 나타나는 기술 이슈 중에 기업 환경이 재화와 용역에 관련된 인프라의 디지털 기술을 중심으로 발전하고 있는 경향에 따라 비즈니스 프로세스 리엔지니어링(BPR: Business Process Reengineering)를 통한 조직의 리사이징(resizing)의 필요성이 대두되고 있다[1]. 비즈니스 프로세스란 기업 입장에서 어떤 목표를 달성하기 위하여 수행하는 작업, 제약 조건 및 자원을 포함시키는 일련의 과정을 말하며, 조직의 일련의 활동들에 대해 일관성과 구조를 강요하여 효율성을 높일 수 있기 때문에 매우 중요하다[2,3]. 이러한 비즈니스 프로세스를 바탕으로 하는 비즈니스 프로세스 리엔지니어링은 기존의 프로세스를 가지고 있는 프로세스로 변화시키면서 프로세스 시간의 단축, 서비스 품질의 극대화 및 비용 절감을 주된 목적으로 한다[1,4].

비즈니스 프로세스를 다루는 대표적인 정보 기술인 워크플로우(workflow)는 기업의 IT를 통합하고, 기업 프로세스 단편화를 방지하며, 상호 연동성을 위한 기반을 마련하기 위해 등장하였다. 워크플로우 시스템은 응용 프로그램에서 프로세스를 독립시켜 프로세스 독립성을 제공함으로써 다양한 경영 환경 변화에 부합되는 프로세스의 변혁을 보다 적극적으로 지원한다.

그리고 이러한 워크플로우 시스템에 대한 모델링은 반드시 전문가가 아니더라도 이해할 수 있고 작업하기 쉬운 보편성을 갖고 있도록 개발되어야 한다. 아울러 모델링 기법 자체에는 정형적인 의미를 내포하고 있어야 하며 분석 가능해야 한다[5]. 즉 워크플로우 모델링에 있어 시멘틱으로 표현 한다면 보다 정형적으로 표현할 수 있으며, 분석 가능해진다. 특히 본 논문에서 적용하고자 하는 의료 분야의 경우 관련 비즈니스 프로세스가 매우 복잡하고 다양한 조직들이 연계되어 있는 특성을 가진다. 특히 비즈니스 프로세스 자체가 의료 각 부분의

행위에 따라 역동적으로 변하는 특성을 가진다 [6]. 더욱이 현재 의료정보 네트워크가 모든 의료 정책 및 사업의 최우선 과제로 떠오르며 의료와 관련된 모든 사업이 의료 기관 네트워크를 통한 의료 정보화가 이루어지고 있는 시점에 의료 부문의 워크플로우 시스템에 대한 시멘틱 기반의 구축은 시급한 과제가 될 수밖에 없다.

최근에는 객체지향 기술의 발달에 따라 소프트웨어 모델링에 대한 OMG(Object Management Group)의 표준인 UML(Unified Modeling Language)이 광범위하게 사용되고 있는데 UML 특유의 사용자 편의성 및 통합성 등으로 인해 정보 시스템 개발뿐만 아니라 기업 워크플로우 모델링에도 많이 적용된다[7]. 특히 UML에서 제공하는 다이어그램들 중 액티비티 다이어그램(activity diagram)은 시스템의 동적인 행위를 묘사하는데 유용하게 사용된다. 또한 비즈니스 프로세스를 포함하는 워크플로우를 모델링 하는데 액티비티 다이어그램을 사용하면 효율적이다. 그러나 OMG가 액티비티 다이어그램에 대한 문서의 시멘틱 정보를 제공하고 있지만 정확한 시멘틱의 부족으로 인하여 모호해지는 단점이 있다[5].

본 논문에서는 이러한 단점을 극복하고자 액티비티 다이어그램의 시멘틱 부분을 ASM(Abstract State Machine)[8,9] 시멘틱 방법을 적용하여 제시한다. 또한 이를 워크플로우 시스템 모델링 기법에 적용하여 확장시킨다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2장은 관련연구를 제시하고, 3장에서는 워크플로우 모델링에서의 ASM 시멘틱 표현에 대해서 언급하고, 4장에서는 사례연구를 통해 평가하고, 마지막으로 5장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련연구

워크플로우에 대한 일반적인 명세는 어떻게

워크플로우 시스템이 행동하는가를 묘사한다. 따라서 워크플로우 시스템의 명세와 관련하여 액티비티 다이어그램의 시멘틱을 정의하고 의미를 부여할 필요가 있다. 지금까지 액티비티 다이어그램을 정형화하기 위한 다양한 시멘틱 방법들이 제안되었다. 이와 같은 액티비티 다이어그램의 시멘틱을 표현하기 위한 방법에는 OCL(Object Constraint Language)[10], pi-calculus[11], FSP(Finite State Processes)[12], ASM(Abstract State Machine)[13] 등이 있다.

이 중에 본 논문에서 적용하는 시멘틱 방법인 ASM은 십여 년전에 제안된 이래 많은 소프트웨어 시스템을 명세하고 검증하는데 효율적으로 사용되어 왔다[14]. ASM 이론은 어느 알고리즘이든지 적절한 ASM에 의하여 적합한 추상 레벨에서의 모델링이 가능하다는 장점을 가진다[15]. 특히 ASM은 잘 이해되고 정확한 모델인 수학적 구조를 사용하므로 명확해지는 장점이 있다. 또한 단순한 구문(syntax)의 사용을 통해 읽고 쓰기 쉬운 이해성을 증진시켜 준다. 또한 기존의 방법들 대부분이 특정 도메인에서만 유용하지만, ASM은 여러 도메인에서 모두 유용한 보편성을 갖는다.

이에 본 연구에서는 이와 같은 장점을 가지고 시멘틱 정의에 대해 정형화에 기초를 제공하는 ASM을 사용하여 액티비티 다이어그램 기반의 워크플로우 모델링에 대한 시멘틱을 정의한다.

3. 워크플로우 시스템에서의 ASM 시멘틱 적용

기존의 워크플로우 시스템에 대한 모델링 방법으로는 시기적절성을 수반하는 시스템의 카테고리리를 모델링하는데 적합하지 않다. 특히 실시간 비동기적 시스템을 모델링하려면 다음 사항들을 모델링하기 위한 능력이 필요하다. 즉 환경에서 예기된 변경과 예외, 동적인 변경

사항, 전체 워크플로우 모델을 개발하기 위해 속박된 워크플로우의 양 그리고 시간적 조절 요인 등이 모델링되어야 한다. 또한 액티비티들이 임의의 방법으로 실행될 수 없기 때문에 워크플로우 모델은 조건 또는 제약사항 등을 포함한다. 즉 사전 조건과 사후 조건의 두 가지 형태로 워크플로우 모델에 적용될 수 있다.

또한 워크플로우 모델링에서는 적격의 사용자 또는 애플리케이션 기능에 대한 액티비티의 할당을 통하여 액티비티를 누가 제어 하는가, 액티비티의 입력 및 출력, 작업 수행에 대하여 요구된 데이터와 제어 정보 및 어떤 액티비티들이 액티비티를 완료하기 위해서 요구되는지 등에 대한 정보가 필요하다.

다음 <표 1>은 이와 같은 워크플로우의 특성을 바탕으로 워크플로우 모델링에서의 필요한 정보를 나타낸다.

<표 1> 워크플로우 특성과 정의

요소	정의
데이터	데이터는 액션에서 생산되거나 사용될 데이터를 표현
조건	조건은 액션의 조건을 표현
자원	활동을 수행하기 위해 요구되는 것들을 표현
참여자	작업을 수행하는 참여자는 누가 활동을 수행하는가를 표현
액션	액션은 단 하나의 나눌 수 없는 작업으로 액터에 의하여 수행됨
타이밍	타이밍은 워크플로우의 액션이 수행되는 동안의 시간이나 데드라인 등의 시간적인 요인을 표현

본 논문에서는 이와 같은 워크플로우 특성을 기반으로 하여 액티비티 다이어그램을 확장한다. 먼저 액션 노드의 확장된 부분들을 나타내는 액티비티 다이어그램의 표기법 및 시멘틱 표현은 <표 2>와 같다.

<표 2> 확장된 액션 노드에 대한 표기법 및 시멘틱 표현

Action Node	
이름	액션 노드
기호	
시멘틱 표현	<pre> if currTarget is node (in, in_data, A, out, out_data, isDynamic, dynArgs, dynMult, action_type, time) then if ¬isDynamic then A elseif [dynArgs] = dynMult then for all $L_i \in dynArgs$ A(L_i) elseif time ≥ deadline then active := exit() elseif action_type is manual then active := participaint active := ⇒ out </pre>

일단 액션 타입이 수동인지 자동인지를 구별하기 위하여 수동인 경우에는 색이 채워진 액션 노드로, 자동인 경우에는 색이 채워지지 않은 노드로 표현한다. 그리고 액션 노드에 필요한 입력 데이터와 액션 노드로부터 나오는 출력 데이터는 위아래의 사각형 부분에 각각 입력 데이터와 출력 데이터를 나타낸다. 여기서 파라미터 in과 out은 입력과 출력 전이를 나타내며, A는 원자의 액션 노드를 나타낸다. 이러한 액션 노드는 이외에도 isDynamic, dynArgs 및 dynMult 파라미터를 갖는다. 먼저 만일 A가 dynMult에 해당하는 시간에 실행된다면 원자의 액션 A는 isDynamic하다고 말할 수 있다. 또한 각각의 시간은 L_i 의 독립변수를 가지며, 이는 객체 $\{L_1, \dots, L_n\}$ 의 연속인 dynArgs의 집합으로부터 온 것이다. 출력 데이터는 액션 노드 A가 수행된 후에 생성되어야 한다. action_type의 파라미터는 액션의 수행 형태가 수동으로 수행되는지 자동으로 수행되는지를 나타내는 것으로, 만일 이것이 수동으로 수행된다면 이는 참여자를 반드시 수반하여야 한다. 마지막으로 time 파라미터는 만일 시간이 데드라인을 초과한다면 바로 종료하도록 한다.

다음 <표 3>은 표현된 액티비티의 사전 조건 노드에 해당하는 확장된 ASM 시멘틱을 나타낸다.

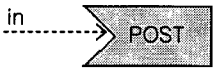
<표 3> 확장된 사전 조건 노드 기호와 시멘틱 표현

Pre-Condition Node	
이름	사전 조건 노드
기호	

시멘틱 표현	<pre> if currTarget is node(PRE, out, isExis then if isExistent then PRE elseif PRE = true then active := → out </pre>
--------	--

사전 조건이 있는 경우에 사전 조건을 수행하며, 사전조건이 참이라면 out 전이를 활성화시키게 된다. 액션을 수행하기 전에 사전 조건을 만족하는 경우-만 액션 노드를 실행할 수 있다. 또한 다음 <표 4>은 사후 조건 노드에 해당하는 확장된 ASM 시멘틱을 나타낸다.

<표 4> 확장된 사후 조건 노드 기호와 시멘틱 표현

Post-Condition Node	
이름	사후 조건 노드
기호	
시멘틱 표현	<pre> if currTarget is node(in, POST, isEx then if isExistent then POST elseif POST = true then active := nextnode(in, in_data, A, out out_data isDynamic, DynArgs, dynMult, action_type, time) </pre>

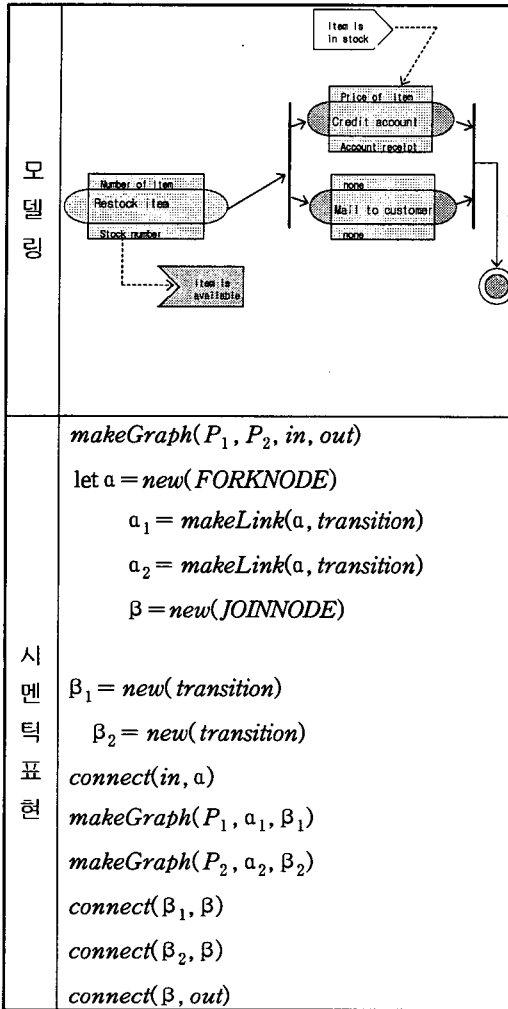
사후 조건은 활동의 종료 시 만족되어야만 하는 조건이다. 액션을 실행한 후에 사후 조건이 있는 경우에 사후 조건을 수행하며, 만일 사후 조건이 참이라면 다음 액션 노드를 실행할 수 있도록 한다.

이렇게 확장된 표기법은 워크플로우를 모델링 하는데 좀더 유연성을 가지도록 도와주며, 워크플로우 시스템의 특성을 좀더 부각시킬 수 있는 장점이 있다.

4. 시멘틱 기반의 의료 워크플로우 시스템 모델링 적용 및 평가

본 논문에서는 의약품을 주문하고 구매하는 프로세스에 대한 워크플로우를 정의한다. 일반적으로 이러한 의약품 구매관리 시스템은 인적 요인 뿐만 아니라 자동화된 시스템 등이 존재한다. 이러한 의료 워크플로우를 제안하는 확장된 액티비티 다이어그램을 이용하여 모델링한다. 프로세스는 병원이 공급사에 품목(의약품 등)을 반품 하는 것을 요구하면서 시작된다. 품목 반품을 요구하기 이전에 품목은 고객이 반품하기를 요구한 것이며 이미 팔린 것이어야만 한다. 이와 같은 조건을 만족한다면 공급사는 품목 반품 요청의 일을 수행하게 된다. 그 다음 회사에서 반품 번호를 얻은 후, 선적 회사에서 품목을 선적하여 반품하게 된다. 여기에서 반품 번호를 얻는 액션에서 반품 번호를 입력 데이터로 넣어야 하며 품목을 선적하는 경우 입력 데이터는 반품 될 품목의 번호이다. 계속적으로 이러한 일련의 프로세스를 진행시킨다.

<표 5> 사례 연구



본 논문에서 제안한 기법은 UML 액티비티 다이어그램을 기반으로 하는 워크플로우 모델링에 대해 ASM 시멘틱으로 표현하는 것이다. 기존의 pi-calculus, FSP 등의 시멘틱을 적용한 액티비티 다이어그램 기법들은 UML 액티비티 다이어그램 자체만을 시멘틱으로 표현했을 뿐 워크플로우 특성을 고려하여 확장된 모델링 기법이 아니다.

먼저 OCL의 경우 액티비티 다이어그램 모델

요소 사이의 제약사항 등을 정의하는데 도움을 준다. 그러나 동적인 행위적인 모델을 표현하기에 너무 다변적이고 비즈니스 프로세스 특성을 모델 하는데 필요한 제어 흐름의 측면을 표현하지 못한다. pi-calculus의 경우 동시에 발생하는 시스템을 표현하는 것에 대한 컴퓨팅 모델의 종류로 프로세스 사이의 상호작용을 표현한다. 그러나 pi-calculus가 적용하기 어려운 까닭은 매우 유용할 수 있는 프로세스의 우선 순위나 예외처리 등이 쉽게 정형화 되지 못한다는 데 있다.

이상의 비교에서와 같이 본 논문에서는 쉽게 적용이 가능하며 보다 보편적인 ASM 시멘틱을 적용한 액티비티 다이어그램의 확장된 기법을 연구하였다. 본 논문의 ASM 시멘틱을 적용한 액티비티 다이어그램의 확장된 기법은 이와 같은 워크플로우 특성을 반영함으로써 보다 효과적이고 효율적인 워크플로우 모델링을 할 수 있도록 도와준다. 또한 기존의 UML 액티비티 다이어그램의 확장으로 기존과 유사한 노드들을 사용함으로써 사용자로 하여금 보다 친근한 인터페이스를 통하여 워크플로우 모델링을 가능하게 한다.

5. 결론

UML의 액티비티 다이어그램은 시스템의 동적인 측면을 표현하는데 적합하기 때문에 비즈니스 프로세스와 워크플로우를 모델링하는데 유용하게 사용될 수 있다. 특히 본 논문의 적용대상인 의료 부문 같이 역동적인 프로세스인 경우에는 보다 효율적으로 적용이 가능하다.

이에 본 논문에서는 워크플로우 시스템의 특성에 맞도록 ASM(Abstract State Machine) 시멘틱을 이용하여 액티비티 다이어그램 기반의 워크플로우 모델링에 대한 시멘틱 표현을 제안하였다. 이와 같은 ASM 시멘틱을 기반으로 하는 정형적인 시멘틱 표현을 통하여 보다 효율적으로 워크플로우를 모델링 할 수 있게 된다. 또한 본 논문의 방법을 사용하면 반드시

전문가가 아니더라도 이해할 수 있고 작업하기 쉬운 보편성을 갖고 있어 더욱 용이하고 유용하게 사용될 수 있을 뿐만 아니라, 워크플로우 수행 중에 발생 가능한 예외사항과 다양한 변화에 대처가능하게 된다.

참 고 문 헌

- [1] Wong S.T.C, Tjandra D, Huili Wang, Weimin Shen, "Workflow-enable distributed computer-based information architecture for digital medical imaging enterprises." *IEEE Transactions on Information Technology in Biomedicine*. Vol. 7(3), pp. 171-183, 2003.
- [2] Jutla D., "Making business sense of electronic commerce." *IEEE Computer*. Vol. 32(5), pp. 67-75, 1999.
- [3] Bussler C., "B2B protocol standards and their role in semantic B2B integration engines." *IEEE Computer Society*. Vol. 24(3), pp. 35-43, 2002.
- [4] Ruth Sara Aguilar-Savlen., "Business process modeling: review and framework." *International Journal of Production Economics*, Vol. 50(5), pp. 235-256, 2003.
- [5] Yang Dong and Zhang ShenSheng, "Using Pi-calculus to Formalize UML Activity Diagram for Business Process Modeling," 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'03), pp. 47-54, 2003.
- [6] Shrivastava S. K, Wheeler S. M., "Architectural support for dynamic reconfiguration of distributed workflow applications." *IEEE Proceedings Software Engineering*, pp. 155-162, 1998.
- [7] Marshall, C., "Enterprise Modeling with UML." Addison Wesley, Reading, MA., 2000.
- [8] Y. Burevich, "Specification and Validation Methods." Oxford University Press, 1995.
- [9] Wuwei Shen, Kevin Compton, and James K. Huggins, "A Toolset for Supporting UML Static and Dynamic Model Checking," 26th International Computer Software and Applications Conference(COMPSAC 2002), IEEE Computer Society, Vol. 3, pp. 147-152, 2002.
- [10] Martin Gogolla and Mark Richters, "Expressing UML Class Diagram Properties with OCL," LNCS 2263, pp. 85-96, 2002.
- [11] Yang Dong and Zhang ShenSheng, "Using pi-calculus to Formalize UML Activity Diagram for Business Process Modeling," *IEEE ECBS*, 2003.
- [12] Roberto W. S. Rodrigues, "formalising UML Activity Diagrams using Finite State Processes," UML2000 Workshop, 2000.
- [13] E. Borger, A. Cavarra, and E. Riccobene, "An ASM Semantics for UML Activity Diagram," In: T. Rust(Ed.), *Proc. AMAST 2000*, LNCS, Vol. 1816, pp. 292-308, 2000.
- [14] Wuwei Shen, Kevin Compton, and James K. Huggins, "A Toolset for Supporting UML Static and Dynamic Model Checking," 26th International Computer Software and Applications Conference(COMPSAC 2002), IEEE Computer Society 2002, pp. 147-152, 2002.
- [15] D. Hammer, K. Hanish, and T. S.

Dillon, "Modeling behavior and dependability of object-oriented real-time systems," Journal of Computer Systems Science and Engineering, Vol. 13(3), pp. 139-150, 1998.