

시맨틱 웹서비스 조합을 위한 Knowledge Preconditions

김상균⁰ 이규철
 충남대학교 컴퓨터공학과
 {skkim⁰, kcleee}@ce.cnu.ac.kr

Toward Knowledge Preconditions for Composition of Semantic Web Services

Sang-Kyun Kim⁰ Kyu-Chul Lee
 Dept. of Computer Engineering, Chungnam National University

요약

그 동안 planning 분야에서는 action 또는 plan이 epistemically feasible한지에 대한 Knowledge Preconditions(KP) 문제 [1]를 다루기 위해 여러 연구들이 제안되었다. 하지만 기존 연구에서는 feasibility에 대한 검사를 design-time에만 수행하며 run-time에서 수행하지 않기 때문에 여러 agent의 트랜잭션들이 발생하는 웹서비스 조합(WSC : Web Services Composition)에서는 문제가 발생하게 된다. 따라서 본 논문에서는 이 문제(Interfering Agent Problem)를 해결하기 위해 transactionally feasible한 WSC를 정의하고 WSC의 atomicity를 보장하기 위한 방법을 제안한다. 뿐만 아니라 WSC를 표현하기 위해 Description Logics기반의 TL-ALCF[2]를 적용하여 시맨틱 웹의 온톨로지(OWL-S)와 성능이 좋은 기존 subsumption 프로시저를 그대로 이용할 수 있도록 한다.

1. 서론

시맨틱 웹서비스의 조합은 전통적으로 Artificial Intelligence(AI)분야에서 planning 문제로 인식되어 왔으며 이를 해결하기 위해 first-order language와 situation calculus에 기반한 다양한 연구가 수행되어 왔다. 그러나 현재 시맨틱 웹은 first-order logics(FOL)의 서브집합인 description logics(DL)에 기반하여 ontology와 individual로 구성되는 Knowledge Base(KB)가 구축되고 있다. 또한 비록 DL이 FOL의 서브셋이라 하더라도 일반적으로 FOL보다 뛰어난 계산 능력(decidability, soundness, completeness)과 표현력을 동시에 가질 수 있는 장점을 가진다.

최근 이와 같은 DL의 장점을 이용하여 시간과 action 그리고 plan을 표현하기 위한 언어로 TL-ALCF가 제안되었으며, 이 논문에서는 기존의 situation calculus와 달리 action과 plan을 Allen의 interval-based temporal logic에 따라 시간 간격에 발생하는 world states들의 집합으로 표현하였다.

하지만 이 논문에서는 WSC 또는 STRIPS-like planning 시스템에서 중요한 action과 plan의 precondition과 effect에 대해서는 논하지 않았다. 특히 이러한 문제는 그동안 시맨틱 웹에서 agent가 action이나 plan을 수행할 수 있는지는 즉, action 또는 plan이 physically feasible한지 그리고 epistemically feasible한지를 결정하는 문제로 알려져 왔다. 하지만 기존의 planning 시스템에서는 이러한 검사를 plan을 수행하기 전에 design-time에만 수행하며 run-time에 그대로 수행될 것이라고 가정하였다.

하지만 WSC는 분산환경에서 여러 agent들이 같이 동작하기 때문에 design-time뿐만 아니라 run-time에도 검사를 수행하여 WSC가 atomic하게 처리될 수 있도록 해야 한다.

따라서 본 논문에서는 이와 같이 WSC의 실행 중에 발생할 수 있는 문제점과 이를 해결하기 위한 WSC의 feasibility를 새로 정의하고 처리하는 방법을 제시한다. 또한 본 논문에서는 action과 WSC를 표현하기 위한 언어로 TL-ALCF를 사용한다.

2. Background

2.1 TL-ALCF

Artale과 Franconi는 interval-based temporal DL 언어의 하나로 TL-ALCF를 제안하였다. 이 언어는 temporal logic(TL)과 propositionally complete DL인 ALCF로 나누고, universal temporal quantification에 제약을 함으로써 decidable할뿐만 아니라 sound하고 complete한 subsumption 프로시저를 제공할 수 있도록 하였다.

Plane-Reservation ≡ ∃(x y u v) (∑ f x) (∑ ni y) (∑ f u) (∑ ni v). ((+SEAT:Vacant) ⊗ x) ⊓ (+SEAT:Reserved) ⊗ y ⊓ (+CARD:Valid) ⊗ u ⊓ (+CARD:Charged) ⊗ v)

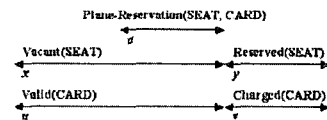


그림 1 Plane-Reservation의 TL-ALCF 표현

그림 1은 비행기 좌석이 비어있고 카드가 유효할 때 좌석을 예약하는 Plane-Reservation action을 TL-ALCF로 표현한 예이다. 이 예제는 4절에서 다시 참조될 것이다.

2.2 Knowledge Preconditions

Agent가 action 또는 plan을 수행할 수 있는지에 대한 문제는 physically feasible한지 (물리적으로 수행가능한지) 그리고 epistemically feasible한지 (agent가 충분히 알고 있는지)에 대한 것으로 논의되어 왔다. 특히 epistemically feasible에 대한 문제는 KP 문제로 알려져 왔으며 이에 대한 여러 연구들이 수행되어 왔다. 특히 Moore[3]는 action에 대해서 그리고 Davis[1]는 plan에 대해서 다음과 같이 정의하였다.

- 만약 agent A가 시간 T에 action을 구성하는 것을 안다면 이 action은 epistemically feasible하다.
- 만약 시간 T에 다음 조건을 만족하면 agent A가 수행하는 plan P는 epistemically feasible하다.
 1. P가 시간 T에 시작해서 끝난다.
 2. 시간 T에 P가 시작한 후에 다음 세가지를 만족한다.
 - a. A는 P가 성공적으로 끝날지를 안다.
 - b. 다음 단계에 수행될 수 있는 모든 action을 안다.
 - c. 모든 다음 단계의 action이 feasible하다.

3. Actions

3.1 Knowledge의 Description Logic 표현

Hintikka[4]는 agent의 knowledge는 "A knows what B is" 로 표현된다고 하였으며 이를 modal theory를 이용하여 $\exists x(\text{know}(A, X = B))$ 와 같이 표현하였다. 따라서 이는 다음과 같이 DL ALC(ALCF의 서브집합)로 바꾸어 표현될 수 있다.

Definition 1. Description Logic ALC를 사용하여 "A knows what B is" 는 다음과 같이 표현된다.

Precondition = $\forall \text{know}.A \cap \exists \text{what}.B$
 (증명) "A knows what B is" 는 "모든 A는 something을 안다" 와 "B는 something(what)이다" 로 나눌 수 있다. 또한 이는 FOL을 사용해서 다음과 같이 표현되며 이는 Definition 1의 식과 같다.

$$\text{Precondition}(x) = \forall z.(\text{know}(x, z) \rightarrow A(x)) \cap \exists y.(\text{what}(x, y) \cap B(y))$$

3.2 Action의 Feasibility

서론에서 언급했듯이 Artale과 Franconi는 precondition과 effect에 대해서는 논의하지 않았지만 일반적인 planning 시스템과 달리 WSC에서는 웹서비스의 effect가 다음에 수행될 웹서비스의 precondition에 영향을 미치며 각각 atomic하게 처리되지 않고 서로 연관성을 가지게 되기 때문에 중요하다. 하지만 비록 Artale과 Franconi가 명시적으로 이를 언급하지 않았더라도 이 둘이 action을 시간 간격에 표현하는 world state들의 집합으로 표현하는 기본 방식에 따라 다음과 같은 정의를 도출할 수 있다.

Definition 2. 시간 [T1,T2]에서 수행되는 action의 precondition은 T1에 agent가 알고 있는 world state들의 집합으로 표현된다.

Definition 3. 시간 [T1,T2]에서 수행되는 action의 effect는 T2에 action을 수행한 결과로 바뀐 world state들의 집합으로 표현된다.

이와 같이 WSC에서는 action의 구성 요소가 world state로 정의되고, WSC의 특성상 finite time interval을 가지는 것으로 제약할 수 있다. 따라서 이를 바탕으로 Moore가 epistemically feasible action에 대해 정의한 것을 다음과 같이 재정의한다. 또한 웹서비스의 실행을 표현하기 위해서 physically feasible action도 정의한다.

Definition 4. 시간 [T1,T2]에서 수행되는 action은 if and only if agent가 시간 T1에 action의 precondition을 안다면 epistemically feasible하다.

Definition 5. 시간 [T1,T2]에서 수행되는 action은 if and only if agent가 [T1,T2]에 수행하여 "succeed"의 effect를 가지면 physically feasible하다.

Proposition 1. Epistemically feasibility는 physically feasibility의 충분조건이다.

Proposition 1은 만약 action이 epistemically feasible하면 physically feasible하다는 관계를 표현한 것이다.

4. 웹서비스 조합

4.1 Interfering Agent 문제

기존의 planning과 WSC간에는 비슷한 점들이 많이 있지만 실제 사용되는 도메인의 성격에 따라 다음과 같은 차이점이 존재한다.

Planning 시스템은 epistemically와 physically feasibility를 오직 plan을 만들고 테스트하는 design-time에만 검사하며 이 plan이 run-time에도 똑같이 수행될 것이라고 가정한다. 하지만 WSC에서는 planning과는 달리 인터넷 상의 분산환경에서 실행되며 많은 agent들이 서비스를 수행하기 위해 제한된 자원들을 공유한다. 따라서 WSC가 design-time에 feasible하다고 하더라도 다른 agent가 실행하는 동안 다른 웹서비스의 precondition을 바꾼다면 run-time에 feasible하지 않을 수 있다. 이러한 문제를 해결하기 위해서 agent는 실행하기 전에 feasible하지 않을 action을 예상하고 이 action에 대한 compensating action을 함께 조합해서 실행해야 한다. 특히, 이 문제는 agent들이 미래를 예상해야 하기 때문에 완벽하게 해결하기는 힘들다. 이에 본 논문에서는 WSC의 feasibility를 정의하고 WSC의 atomicity를 보장하는 방법을 제안한다.

4.2 Web Services Composition의 Feasibility

Feasible plan에 대한 Davis의 정의는 run-time 환경을 고려하지 않았기 때문에 여기에서는 Davis의 정의를 확장하여 WSC의 feasibility를 정의한다.

Definition 6. 시간 [T1,T2]에서 수행되는 Web Services Composition C는 atomic action들의 partial function sequence인 a_1, \dots, a_n 로 정의되며, 이 sequence는 empty이거나 제한된 크기를 가진다.

Definition 7. 시간 [T1,T2]동안 agent A가 Web Services Composition C를 수행할 때, 시간 T1에 다음 조건을 안다면 C는 transactionally feasible하다,

1. C가 시간 T1에 시작해서 T2에 끝난다.
2. 시간 T1에 C가 시작한 후에 다음 5가지를 만족한다.
 - a. A는 C가 성공적으로 끝날지를 안다.
 - b. 다음 단계에 수행될 수 있는 모든 action을 안다.
 - c. 모든 다음 단계의 action이 epistemically and physically feasible하다.
 - d. 한 action이 실패할 때 이미 수행된 action과 연관된 compensating action을 안다.
 - e. 모든 compensating action이 epistemically and physically feasible하다.

WSC가 atomic하게 실행되기 위해서 agent는 WSC가 실행되기 전에 각 action에 대한 compensating action을 알아야 하며 만약 실행 중에 어떤 action이 feasible하지 않으면 대신 compensating action을 수행해야 한다. 하지만 만약 WSC가 다른 agent의 간섭 없이 혼자 수행 가능하다면 위의 정의는 Davis의 정의와 같게 된다.

4.2 Transactionally Feasible WSC의 예제

이 절에서는 agent가 여러 개의 비행기 좌석을 예약하는 Reservation-Plane-Seats WSC를 통해 transactionally feasible WSC에 대해 설명한다. 특히 이 예제에서는 자리의 위치를 지정하기 위해 한번에 하나의 좌석을 예약하는 것을 가정한다.

Reservation-Plane-Seats $\triangleq \circ(i\ j\ j'\ j'')\ (i\ b\ j)\ (j\ b\ j'')\ (i'\ b'\ j')\ (i\ b\ j)\ (i'\ b'\ j'')\ (i\ b\ j)\ (i'\ b'\ j'')$
 $\square\ C\text{-Plane-Reservation}\ \circ\ j\ \square\ \text{Plane-Reservation}\ \circ\ i'\ \square\ C\text{-Plane-Reservation}\ \circ\ j''$

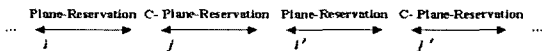


그림 2 Reservation-Plane-Seats WSC

C-Plane-Reservation $\triangleq \circ(i\ l\ m\ n\ p\ q)\ (i\ b\ j)\ (i'\ b'\ j'')\ (i\ b\ j)\ (i'\ b'\ j'')\ (i\ b\ j)\ (i'\ b'\ j'')$
 $\{(*SEAT:Failed)\ \circ\ i\ \square\ (*SEAT':Vacant)\ \circ\ m\ \square\ (*SEAT':Reserved)\ \circ\ n\ \square\ (*CARD:Valid)\ \circ\ p\ \square\ (*CARD:Charged)\ \circ\ q\}$

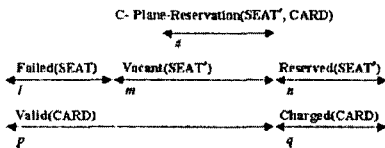


그림 3 C-Plane-Reservation (Compensating action)

그림2은 비행기를 예약하는 Plane-Reservation action(그림 1)과 이 action이 실패할 경우 다른 비행기의 자리를 예약해주는 compensating action(그림 3)으로 구성된다. 만약 Plane-Reservation의 effect가 "Fail"인 경우(physically feasible하지 않은 경우) C-Plane-Reservation이 feasible하게 되어 수행된다. 그림 4는 agent가 N번째 좌석을 예약하고 N+1번째 좌석을 예약하려고 할 때 이미 다른 agent가 남은 좌석을 모두 예약해서 예약하지 못하는 경우 C-Plane-Reservation 실행을 통해 다른 비행기의 좌석 SEAT' 을 예약하는 예제를 보인 것이다.

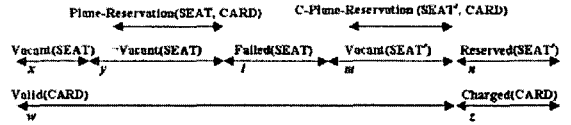


그림 4 Plane-Reservation action의 보상 실행

특히 본 논문에서는 SAGAS[5]의 backward/forward recovery 메커니즘을 적용하여 위와 같은 WSC를 수행하는 알고리즘을 개발하였다.[6]

5. 관련 연구

McIlraith와 Son[7]은 situation calculus에 기반한 논리 프로그래밍 언어인 Golog을 적용하여 자동화된 WSC를 구현하였으며, Davis 논문과 비슷하게 KSSP와 PSSP를 정의하였다. 하지만 이는 DL에 기반하지 않았으며 또한 WSC의 atomicity 보장에 대해서 고려하지 않았다.

Wu et al.[8]은 DAML-S를 HTN planning task로 변환하는 SHOP2 시스템을 제안하였지만 decidable한 reasoning을 수행하지 못하는 단점이 있다.

Morgenstern[9]의 논문은 본 논문과 비슷하게 여러 agent들이 plan을 수행할 때 agent들간의 통신방법에 대해 논하였지만 본 논문에서는 WSC의 atomicity를 보장하는 방법을 제안하지는 않았다.

6. 결론

본 논문에서는 Interfering Agent Problem과 transactionally feasible WSC에 대해서 정의하였다. 또한 이 정의를 기반으로 SAGAS의 메커니즘을 적용하여 WSC가 atomic하게 수행될 수 있도록 알고리즘을 제시하였다. 또한 WSC를 위한 언어로 TL-ALCF를 사용하여 decidable, sound, complete subsumption을 이용할 수 있게 하였다. 향후에는 본 논문을 보다 정형화하고 다양한 WSC의 경우를 고려할 것이다.

7. 참고문헌

- [1] E.Davis. Knowledge Preconditions for Plans. Journal of Logic and Computation, 4(5):721-766 (1994)
- [2] A.Artale and E.Franconi. A Temporal Description Logic for Reasoning about Actions and Plans, Journal of Artificial Intelligence Research, 9:463-506 (1998)
- [3] R.Moore. A Formal Theory of Knowledge and Action. Formal Theories of the Commonsense World, pp.319-358, (1985)
- [4] J.Hintikka. Semantics for Propositional Attitudes. Reference and Modality, pp.145-167 (1969)
- [5] H.Garcia-Molina et al. SAGAS. SIGMOD, pp.249-259 (1987)
- [6] S.-K. Kim and K.-C. Lee. Toward Knowledge Preconditions for Composition of Semantic Web Services. Submitted to DASFAA. <http://www.ce.cnu.ac.kr/~skkim/html/papers/dasfaa2005.ps>
- [7] S.McIlraith and T.C.Son. Adapting Golog for Composition of Semantic Web Services. In Proc. of KR02, pp.482-496 (2002)
- [8] D.Wu et al. Automatic Web Services Composition using SHOP2. In Workshop on Planning for Web Services (2003)
- [9] L.Morgenstern. Foundations of a Logic of Knowledge, Action, and Communication. PhD thesis, New York University (1998)