시각언어의 공동작업 속성

김경덕
위덕대학교 컴퓨터공학과
e-mail: kdkim@mail.uiduk.ac.kr

Collaborative Attributes of a Visual Language

Kyungdeok Kim
Dept. of Computer Engineering, Uiduk University

요 약

본 논문에서는 공동작업을 지원하기 위한 시각언어를 제안하고 생성되는 시각문장에서 공동작업을 위한 속성과 컴파일 방법을 설명한다. 생성되는 시각문장은 객체 아이콘과 연산자로 구성되며, 각각은 공동작업 참여자와 참여자간 상호작용 시점에 따른 공동작업 관계를 의미한다. 생성되는 시각 문장은 객체 아이콘과 연산자의 결합관계에 따라 다양한 공동작업을 지원한다. 또한, 동기 및 비동기 공동작업을 함께 표현함으로써 기존 공동작업을 위한 시각언어보다 효율적으로 공동작업을 지원한다.

1. 서론

시각언어는 사용자 인터페이스나, 멀티미디어 컨텐츠 저작분야 등에서 활발히 연구 및 사용되고 있다. 기존 시각언어는 시각 문장의 확장성과 객체간 산술적인 연산 관계의 표현은 지원하지만, 시간 관계는 지원하기 어렵다[2, 5, 6]. 이러한 문제점을 보완한 동적 시각언어[2]는 객체 아이콘에 시간적 속성을 사용하여, 시간에 따른 동적 변화를 지원함으로써, 다양한 미디어들의 처리와 폭넓은 의미 표현이 가능하다. 대부분의 시각언어는 단일 사용자 환경에서 컴퓨터와의 상호작용을 위한 인터페이스나 시각 프로그래밍은 지원하지만, 인터넷 환경에서 공동작업과 같은 다중 사용자 환경은 지원하기 어렵다[3, 4].

대부분의 기존 시각언어는 객체 아이콘과 연산자들의 조합으로 시각 문장을 생성한다. 객체 아이콘은 작업 또는 수행할 연산을 의미하며, 연산자는 작업 관계 및 연산 결과의 흐름을 의미한다. 본 논문에서 제안하는 시각언어는 사용자간 상호작용 시점에 따른 공동작업을 지원하며, 시각 문장의 의미 속성을 λ수식을 사용하여 효율적으로 공동작업을 지원함을 보인다.

본 논문의 구성은 다음과 같다. 제 2 절에서는 시각언어의 속성을 설명하고, 제 3 절에서는 시각문장의 컴파일 방법을 설명한다. 그리고, 제 5 절에서는 결론 및 연구 방향에 대하여 기술한다

2. 시각언어의 속성

기존 대부분의 시각언어는 일반적으로 위치, 시간, 내용 속성을 가진다[7, 8]. 제안한 시각언어는 위치와 시간 속성을 확장하여, 다중 사용자 환경에서 공동작업을 지원하는 시각언어이다. 즉, 시각언어의 위치 속성은 이차원에 표현되는 객체 아이콘이 평면상의 위치 정보와 인터넷 환경에서 공동작업에 참여하는 사용자 정보를 가진다. 시간 속성에서는 사용자의 공동작업 수행 상태 값을 사용하여 사용자간 상호작용 시점에 따른 공동작업 관계를 더욱 융통성 있게 표현한다.

제안한 시각언어에서 사용하는 객체 아이콘과 연산자를 대수적 관계로 정의하면 정의 1, 2와 같다.

韓國領 독도 Dokdo is Korean Territory

[정의 1] 객체 아이콘 EX는 2 튜플로 $EX = (EX_I, EX_M)$이다. 여기서, EX_I는 객체 아이콘을 표현하는 이미지 집합이고, EX_M는 의미 부분이며 7 튜플로 $EX_M = (A, U, S, IM, OM, IN, OUT)$이다. A는 행위의 집합, U는 행위를 수행하는 사용자의 집합, S는 사용자의 참여 상태 값 집합, $S = \{s_A$(수행 가능), s_b(수행중), s_a(부재)$\}$, IM은 사용자와 컴퓨터간의 상호작용으로부터 입력되는 메시지 집합, OM는 사용자와 컴퓨터간의 상호작용으로부터 출력되는 메시지 집합, 메시지 입력 및 출력 함수 IN, OUT는 $IN : IM^+ \times S \rightarrow ID \times S$이고, ID는 입력 메시지 함수 IN에 의하여 생성된 임시 데이터의 집합, $OUT : ID \times A \times U \times S \rightarrow OM^+ \times S$이다.

[정의 2] 연산자 OP는 2 튜플로 $OP = (OP_I, OP_M)$이다. 여기서, OP_I는 객체 아이콘들의 이미지에 적용되는 연산자이고, OP_M는 객체 아이콘들의 의미 부분에 적용되는 논리적 연산자이다. 연산자 OP_I와 OP_M의 정의는 $OP_I : EX_I \times EX_I \rightarrow EX_I$이고, $OP_M : EX_M \times EX_M \rightarrow EX_M$이다.

정의 2의 연산자 종류로는 공동작업에서 사용자간 상호작용 시점을 기준으로, 비동기(op_a), 부분동기(op_p), 복합동기(op_c), 완전동기(op_s) 연산자가 있다. 비동기 연산자는 사용자간 순차적인 공동작업 관계를 의미하며, 완전동기 연산자는 사용자간 동시적인 공동작업 관계를 의미한다. 부분동기 및 복합동기 연산자는 완전동기 및 비동기 연산자로부터 파생된 연산자로, 부분동기 연산자는 사용자간 상호작용 시점에서 조건에 따른 동기적 공동작업을 지원하는 연산자이다. 복합동기 연산자는 비동기 및 완전동기 연산자의 속성을 모두 가진 연산자로 비동기 연산을 먼저 수행한 후 완전동기 연산을 수행한다.

제안한 시각언어의 문법 VG는 $VG = (UG, R)$이며, UG는 두 개의 객체 아이콘과 한 개의 연산자로 구성되는 단위 시각 문장을 생성하는 문법과 단위 시각 문장을 결합하는 규칙 집합 R로 구성된다. 즉, 단위 시각 문장은 3 튜플로서 $(OP\ EX\ EX)$이다. 문법 UG는 $G(UG) = (N, T, P, S)$이며, N는 비단말 기호의 집합으로 $N = \{S, C, C1, S1, Main, Sub, SubC, OP\}$, T는 단말 기호의 집합으로 $T = \{$ Start, End, Com, User, SStart, Send, ->, *->, 0->, <->$\}$, P는 생성 규칙의 집합으로 $P = \{p_1, ..., p_7\}$, S는 시작 기호이다. 단말 집합의 각 원소 의미는 다음과 같다. Start, End는 공동작업의 시작과 마침을 의미하며, SStart, Send는 부분적인 공동작업의 시작과 마침을 의미하고, 시각 문장의 모듈화를 위하여 사용된다. 기호 Com는 부분적인 공동작업을 내포하는 객체 아이콘을 의미하며, User는 공동작업에 참여하는 사용자를 의미한다. 기호 ->, *->, 0->, <->는 각각 비동기, 복합동기, 부분동기, 완전동기 연산자를 의미한다. 그리고, 생성 규칙은 다음과 같다.

p_1: S ::= $Main$ | Sub
p_2: $Main$ ::= Start->C | User->C|Com->C | $S1$
p_3: C ::= End | $C1$
p_4: $S1$ ::= User OP User
p_5: OP ::= 0-> | *-> | <->
p_6: $C1$ ::= User | Com
p_7: Sub ::= SStart->$SubC$ | User-> $SubC$ |Com->C| $S1$
p_8: $SubC$ ::= Send | $C1$

문법 VG는 다자간 다양한 공동작업 관계를 표현하고 공동작업에서 사용자의 참여 및 탈퇴를 효율적으로 표현한다.

제안한 시각언어는 객체 아이콘과 연산자를 조합함으로써 다양한 공동작업 관계를 효율적으로 표현한다. 연산자에 따른 의미 속성은 다음과 같다.

비동기 연산자 iop_a에서 연산 도메인과, 의미 함수는 수식 (1)과 같다. 여기서, 원시 함수 ***combine***은 두 객체 아이콘의 이미지 결합한 값을 반환한다. 수행환경 Env와 Env'는 객체 아이콘에 관련된 사용자, 사용자의 역할, 초기 상태 값을 설정한다. 원시 함수 ***isnonstrict***는 입력된 수식에서 출력 값이 생성하면, 함수 ***true***를 연산 결과 값으로 반환하고, 그렇지 않으면 함수 ***false***를 연산 결과 값으로 반환한다. 그리고, 집합 $Interaction_a$의 원소 $async.$는 비동기 상호작용을 의미하며, i_{iopa}는 비동기 연산자의 이미지를 의미한다. 함수 ***in***, ***out***은 정의 2의 함수 IN과 OUT을 의미한다. 기호 $\mapsto$는 값의 바인딩을 의미하며, Oi는 객체 아이콘의 집합이다.

$$\mathbf{iop_a} : Oi \times Oi \rightarrow Interaction_a \quad \text{... 식 (1)}$$
$$\mathbf{iop_{ai}} : Oi \times Oi \rightarrow Image$$
$$\mathbf{iop_{am}} : Oi \times Oi \rightarrow ((OM^+ \times S) \times (OM^+ \times S)) \rightarrow Interaction_a)$$
$$Env_1 = \{U \mapsto u,\ S \mapsto s,\ A \mapsto a\}$$
$$Env_2 = \{U \mapsto u',\ S \mapsto s',\ A \mapsto a'\},\ \text{단}, u \neq u'$$
$$Interaction_a = \{async.,\ false\}$$
$$\mathbf{combine} : Image \times Image \times Image \rightarrow Image$$
$$\mathbf{isnonstrict} : expression \rightarrow \{\mathbf{true, false}\}$$

韓國領
독도
Dokdo is Korean Territory

iop_a = λx y. ((iop_{ai} x y) (iop_{am} x y))
iop_{ai} = λx y. ***combine*** x y i_{iopa}
iop_{am} = λx y.((***isnonstrict***(***out*** (***in*** x) u a)) →
(***out*** (***in*** y) u' a') | *false*)

시각 문장 (ex_1 iop_a ex_2)의 의미 연산은 수식 (2)와 같다. 여기서, ex_1, ex_2는 객체 아이콘을 *im*는 입력 메시지를 의미한다.

iop_a ex_1 ex_2 ... 식 (2)
= ((***combine*** ex_{1i} ey_{2i} i_{iopa})
((***manipulate*** ex_2.*im* ex_2.*u* ex_2.*a*) s_{ready}))

여기서, 함수 ***isnonstrict***는 연산 결과로서 ***true***를 생성할 경우이며, 함수 ***manipulate***는 행위의 수행 주체가 데이터를 처리하여 뷰를 변화시킬 수 있는 출력 메시지를 생성하는 원시 함수이다.

완전 동기 연산자 iop_s에서 연산 도메인과 의미 함수는 수식 (3)과 같고, 비동기 연산자의 연산 도메인과 동일하다. 여기서, 원시 함수 ***combine***은 위의 식과 유사하며, 단지 연산자의 이미지를 비동기 연산자의 이미지 대신에 완전 동기 연산자 이미지를 사용한다. 의미 함수 iop_{sm}에서 함수 내에 포함된 연산들의 연산 순서는 내포된 함수들부터 연산을 수행한다. 수식에서 조건문은 두 객체 아이콘의 사용자가 모두 공동작업을 수행함으로써, 조건문의 결과가 참이 되며 공동작업이 계속 진행된다. 그러므로, 두 사용자는 동기적으로 공동작업을 수행한다. 집합 $Interaction_s$의 원소 *sync.*는 동기 상호작용을 의미하며, 함수 ***and***는의 정의는 λx y. (x → y | *false*)이다.

iop_s : Oi × Oi → $Interaction_s$ ... 식 (3)
iop_{si} : Oi × Oi → *Image*
iop_{sm} : Oi × Oi → (((${OM}^+$×S)×(${OM}^+$×S))
→ $Interaction_s$)
$Interaction_s$ = {*sync., true, false*}
iop_{sm_aux} = Oi × Oi → (((${OM}^+$×S)×(${OM}^+$×S))
→ $Interaction_s$)
iop_s = λx y. ((iop_{si} x y) (iop_{sm} x y))
iop_{si} = λx y. **combine** x y i_{iops}
iop_{sm} = λx y.((***and isnonestrict***(***out***(***in*** x)u a)
isnonestrict(***out*** (***in*** y) u' a')) →
(iop_{sm_aux} x y) | *false*)
iop_{sm_aux}=λxy.((***and isnonestrict***(***out***(***in*** x)u a)
isnonestrict(***out*** (***in*** y) u' a')) →
iop_{sm_aux} x y) | *true*)

함수 iop_{sm}의 연산 결과 값이 *false*인 경우는 연산이 중단된 경우를 의미한다. 함수 iop_{sm_aux}은 함수 iop_{sm}의 보조 함수로서 상호작용이 사용자간에 발생하였다면, 동기적 공동작업이 수행되며, 두 작업간에 공동작업이 중단되면, 연산 결과 값으로 *true* 값을 생성한다. 그러므로, 시각 문장 (ex_1 iop_s ex_2)의 의미 연산은 수식 (4)와 같다.

iop_s ex_1 ex_2 ... 식 (4)
= ((***combine*** ex_{1i} ey_{2i} i_{iops})
((***and isnonestrict***((***manipulate*** ex_1.*im* ex_1.*u* ex_1.*a*) s_{ready})
isnonestrict((***manipulate*** ex_2.*im* ex_2.*u* ex_2.*a*) s_{ready}))
→ (iop_{sm_aux} ex_{1m} ex_{2m}) | *true*)

여기서, 함수 ***isnonstrict***는 연산 결과로서 함수 ***true***를 생성한다고 가정할 경우에 함수 iop_{sm_aux}의 재귀적 호출에 의하여 객체 아이콘 ex_1과 ex_2의 사용자들은 서로 동기적 공동작업을 수행한다. 그리고, 두 사용자중에 한 사용자가 공동작업을 종료하면, 공동작업의 수행은 종료한다.

부분 동기 연산에서 시각 문장 (ex_1 iop_p ex_2)의 의미 연산은 동기 연산과 유사하며, 동기 연산과의 차이점은 동기 연산자는 두 객체 아이콘의 사용자 중 어느 한 사용자가 공동작업을 활성화시킴으로써 동기적 공동작업을 수행한다. 부분 동기 연산은 단지 객체 아이콘 ex_1의 사용자가 공동작업을 위한 작업 행위를 활성화될 경우에만 객체 아이콘 ex_1과 ex_2의 사용자가 서로간에 동기적 공동작업을 수행한다. 부분 동기 연산이 종료한 후에는 객체 아이콘 ex_1에 연결된 비동기 연산자나 복합 동기 연산자의 메시지 흐름 방향에 따라 다른 객체 아이콘의 사용자와 공동작업을 활성화시킨다.

복합 동기 연산에서 시각 문장 (ex_1 iop_c ex_2)의 의미 연산은 수식 (5)와 같다. 즉, 완전 동기 연산과 부분 동기 연산으로 구성된다.

iop_c ex_1 ex_2 ...식(5)
= ((***isnonstrict***(iop_{am} ex_1 ey_2)→(iop_{pm} ex_1 ey_2)
| *false*)

3. 시각 문장의 컴파일

생성하는 시각문장은 의미 속성에 의하여 함수들의 결합으로 구성된다. 생성되는 시각 문장의 수행은 연산 시점에 따라 활성화될 부분 시각 문장들만 연산됨으로 시각 문장의 결과 값은 점진적으로 생성된다. 시각 문장의 수행은 다음과 같은 단계로 이루어진다. (1) 시각 문장으로부터 연산자에 따라 비동기 연산 및 동기 연산 관계의 부분 시각 문장들로 분류한다. (2) 각 부분 시각 문장은 의미 분석에 따라 개념 그래프로 변환된다. (3) 개념 그래프의 각 노드는 실행 모듈로 맵핑된다. (4) 개념 그래프의 각 노드는 연산자의 의미 속성에 따라 실행 모듈을 수

韓國領
독도
Dokdo is Korean Territory

행한다. 이러한 단계는 시각 문장에 포함된 부분 시각문장들을 모두 수행할 때까지 반복한다. 여기에서, 실행 모듈은 실제 공동작업에 사용되는 프로시저나 어플리케이션이다.

제안한 시각언어를 사용하여 사무환경에서 다자간 문서 작업과 전달을 지원하는 시각언어를 구현하였다[13]. 구현 환경은 *Solaris* 2.4의 *OPEN WINDOW*, *OSF/Motif, C++*, 미들웨어 *HITE*[12]이다. 제안한 시각언어는 기존에 지원하기 어려운 인터넷 환경에서 공동작업의 수행을 효율적으로 표현한다. 특히, 사용자간 동기 및 비동기 관계를 효율적으로 표현함으로써 공동작업을 효율적으로 수행한다. 기존 공동작업을 지원하는 시각언어 J. C. Grundy의 EVPL[3], K. D. Swenson의 VPL[9], C. Castroianni의 Scarabaeus[1], K. Zheng의 Meteor2 [11], K. Takeda의 AP[10], S. Kim의 COVIL[4] 등과 비교할 때, 제안한 시각언어는 사용자간 공동작업 관계를 1:N의 관계로 지원함으로써 공동작업을 더욱 융통성 있게 지원한다.

5. 결론

제안한 시각언어는 기존의 시각언어가 지원하기 어려운 인터넷 환경에서 다자간 공동작업을 효율적으로 지원한다. 생성되는 시각 문장에서 객체 아이콘은 공동작업에서 작업을 수행하는 사용자를 나타내며, 연산자는 공동작업의 시점에 따라 분류하였다. 또한, 각 연산에 대한 의미 속성을 λ함수로 분석하여 시각 문장의 의미를 기술하였으며, 기존 공동작업 지원 시각언어에서 표현하기 어려운 비동기 및 동기 공동작업의 관계를 모두 지원함을 보였다.

앞으로의 연구 방향은 객체 아이콘의 실시간 추가 및 삭제에 따른 의미 변화와 시각 문장의 최적화 등이다.

참고문헌

[1] C. Castroianni, Development of a Workflow Design Tool for the Scarabaeus Project, Masters thesis, Dept. of Computer Science, Univ. of Twente, 1995.

[2] S. Chang, "Dynamic Visual Languages," Proc. of the IEEE Symposium on Visual Languages, pp. 308-315, 1996.

[3] J. C. Grundy and J. G. Hosking, "Visual Language Support for Planning and Coordination in Cooperative Work Systems," Proc. of the IEEE Symposium on Visual Languages, pp. 324-325, 1996.

[4] S. Kim and M. Jin, "Collaborative Visual Languages," Proc. of the IASTED Int. Conf.: Artificial Intelligence and Soft Computing, pp. 168-171, 1997.

[5] P. Bottoni, S. Chang, M. F. Costabile, S. Levialdi, and P. Musio, "On the Specification of Dynamic Visual Languages," Proc. of the 14th IEEE Symposium on Visual Languages, 1998.

[6] N. Hari Narayanan and R. Hubscher, "Visual Language Theory: Towards a Human-Computer Interaction Perspective," in Visual Language Theory, K. Marriott and B. Meyer (eds.), Springer-Verlag, 1998.

[7] S. Chang, Principles of Visual Programming Systems, Prentice-Hall, 1990.

[8] S. Chang, "Extending Visual Languages for Multimedia," IEEE Multimedia Magazine, Vol. 3, No. 3, pp. 18-26, 1996.

[9] K. D. Swenson, "A Visual Language to Describe Collaborative Work," IEEE Int. Symposium for Visual Languages, pp.298-303, 1993.

[10] K. Takeda, M. Inaba, and K. Sugihara, "User Interface and Agent Prototyping for Flexible Working," IEEE Multimedia Vol.3, No.2, pp.40-50, 1996.

[11] K. Zheng, Designing Workflow Processes in METEOR$_2$ Workflow Management System, Master's thesis, Dept. of Computer Science, Univ. of Georgia, 1997.

[12] 김상욱, 김정미, 김태호, 이정훈, 배유석, "HITE: 객체 지향 멀티미디어 그룹웨어", 정보과학회논문지 (B), 제24권, 제12호, pp. 1513-1521, 1997.

[13] 김경덕, "분산 멀티미디어 환경에서의 협력상호작용 지원 시각언어", 박사학위 논문, 경북대학교 대학원, p.138, 1999.

韓國領 독도 Dokdo is Korean Territory