지능형 협업 환경에서 사용자를 위한 효과적인 공간 인터랙션 제공

Provision of Effective Spatial Interaction for Users in Advanced Collaborative Environment

고수진, Sujin Ko*, 김종원, JongWon Kim**

요약 다양한 센서 네트워크와 유비쿼터스 기술이 제공되는 지능형 협업 환경은 사용자를 위해 확장된 인터랙션을 제공할 수 있다. 기존의 인터랙션이 사용자와 컴퓨터 머신과의 직접적인 인터랙션이 주를 이룬 반면 새로 확장된 인터랙션은 사용자와 공간과의 인터랙션을, 실질적으로 공간을 구성하는, 관리와 제어가 가능한 구성요소와의 인터랙션을 나타낸다. 본 논문은 이러한 공간 인터랙션을 효과적으로 제공할 수 있도록 하기 위해서 공간 오브젝트를 등록, 인식하고, 특히 사용자의 의도에 맞는 태스크를 지원하기 위해 과거의 인터랙션 정보를 이용한 템플릿 기반 맵핑 알고리즘을 설계한다. 제안된 알고리즘을 이용하는 경우, 공간 오브젝트가 증가함에 따라 템플릿을 검색하여 처리하는데 드는 시스템의 비용이 어느 정도 향상되는지 실험을 통해 분석하도록 하며, 진행되는 모든 공간 인터랙션을 시각적으로 보여주기 위한 그래픽 기반의 도시 방법을 소개하고 결론을 맺는다.

Abstract With various sensor network and ubiquitous technologies, we can extend interaction area from a virtual domain to physical space domain. This spatial interaction is differ in that traditional interaction is mainly processed by direct interaction with the computer machine which is a target machine or provides interaction tools and the spatial interaction is performed indirectly between users with smart interaction tools and many distributed components of space. So, this interaction gives methods to users to control whole manageable space components by registering and recognizing objects. Finally, this paper provides an effective spatial interaction method with template-based task mapping algorithm which is sorted by historical interaction data for support of users' intended task. And then, we analyze how much the system performance would be improved with the task mapping algorithm and conclude with an introduction of a GUI method to visualize results of spatial interaction.

해심어: Spatial Interaction, Space Object, Smart Meeting System (SMeet), Template-based task mapping algorithm, HCI

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스 컴퓨팅 및 네트워크 원천기반기술 개발사업의 08B3-O2-10 과제로 지원된 것임.

*주저자: 광주과학기술원 정보통신공학과; e-mail: sjko@nm.gist.ac.kr

**교신저자: 광주과학기술원 정보통신공학과 교수; e-mail: jongwon@nm.gist.ac.kr

1. 서론

일반적으로 협업 공간은 사람, 여러 종류의 컴퓨터 머신, 다양한 콘텐츠 등으로 이루어지며 상세하게는 협업 참가자, 공용/개인용 디스플레이, 오디오/비디오 머신, 인터랙션 디바이스, 제어 머신 그리고 다양한 콘텐츠 등으로 이루어진다 [1]. 상기의 환경에서 사용자에게 제공될 수 있는 인터랙션을 가정해 보면 특정 컴퓨터가 제공하는 사용자 인터페이스를 통해 마우스나 키보드를 이용하여 협업 공간의 공용 디스플레이 상의 콘텐츠 파일을 올리거나 또는 내리는 기능, 또는 해당 콘텐츠 파일을 협업 공간의 다른 디스플레이로 이동시키는 기능 등이라고 할 수 있다. 하지만 이러한 기능은 사용자가 반드시 해당 사용자 인터페이스를 제공하는 컴퓨터 머신 앞에 있을 경우만 가능하며 그렇지 않은 경우 물리적인 공간에서 해당 콘텐츠를 선택하여 다른 디스플레이에 이동시키는 등의 일은 용이하지 않다. 즉, 종래의 인터랙션 기술은 전통적인 방식의 키보드, 마우스 등을 이용하며 이러한 인터랙션 장치와 직접적으로 관계가 있는 특정 부분과의 인터랙션이 주를 이룬다. 이러한 기술적 한계를 극복하고 공간 안의 구성요소를 일반적인 방법으로 제어할 수 있도록 하기 위한 방법론(본 논문에서는 이를 "공간 인터랙션"이라고 정의한다)이 필요하며 해당 정보를 효과적으로 사용자에게 나타낼 수 있는 방안도 요구된다.

이러한 협업 환경에서의 인터랙션을 다루는 연구로는 Fraunhofer IPSI 의 Norbert A. Streitz 와 그의 동료들이 진행한, 인터랙션을 "Human-Computer Interaction (HCI)"에서 " Human-Environment Interaction (HEI)"로 전개해 나가기 위한 연구가 있다[2]. 그들은 Roomware[3] 나 Ambient Agoras[2]와 같은 결과물을 통하여 컴퓨팅 능력이 내재되어 있는 실제 구조물로 구성된 공간과 가상의 정보공간을 통합하는 시도를 해왔으며 이러한 시도는 우리와 유사하다고 할 수 있다. 또한 Stanford 대학의 iRoom [4] 의 경우도 협업 환경에서 공용 디스플레이를 목표로 하는 새로운 타입의 인터랙션 방법을 제공하였다.

그림 1은 기존의 인터랙션과 공간 인터랙션의 운용에 대한 차이를 그림으로 표현한 것으로 종래 기술은 주로 한 명의 사용자가 인터랙션 장치 (그림 1에서는 마우스) 와 연결되어 있는 컴퓨터 머신과의 인터랙션을 수행하는 것을 보여주며, 공간 인터랙션은 동시에 여러 명의 사용자가 공간상에 제공되는 다양한 인터랙션 장치를 이용하여 이형질의 공간 구성 요소 (그림 1 에서는 개인용 컴퓨터 머신, 공용 머신, 캠코더 등)를 제어할 수 있도록 표현되어 있다.

본 논문에서는 이렇게 제어의 범위를 한 개의 컴퓨터가 제공하는 가상공간에서 물리적인 공간으로 확대하기 위한 공간 인터랙션의 방법론을 제시하고 이와 함께 해당 인터랙션의 진행을 원격지의 사용자에게 효과적으로 공유하기 위한 시각적인 방법도 제시한다.

그림 1. 기존 인터랙션과 공간 인터랙션의 차이.

2. 본론

본론에서는 공간 인터랙션을 제공하기 위한 시스템 구성 및 각 요소들을 설명하고 효과적인 인터랙션 제공을 위한 템플릿 기반 맵핑 알고리즘을 설명하며, 템플릿을 구성하는 요소에 따라 시스템의 성능을 고찰하는 실험을 실시하고 결과를 정리한다.

2.1 공간 인터랙션을 위한 시스템 구성

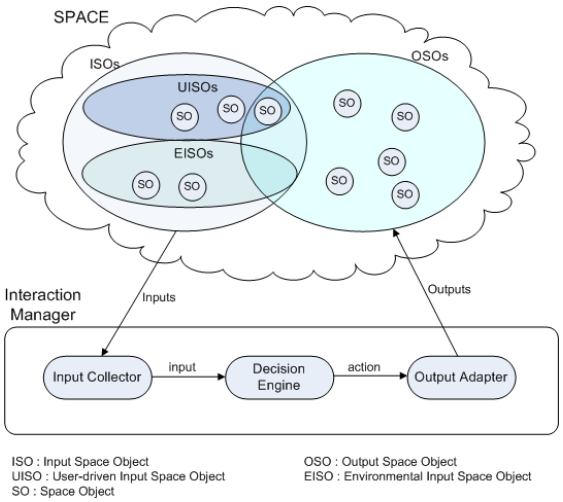

그림 2. 공간 인터랙션을 위한 시스템 구성도.

그림 2는 지능형 협업 환경에서의 공간 인터랙션이 어떤 형태로 이루어지는지의 개념을 보여주는 시스템 구성도를 보여준다. 그림은 크게 실제 물리적인 공간을 나타내는 부분과 해당 물리적인 공간과의 오퍼레이션을 통해 사용자에게 인터랙션을 제공할 수 있는 협업 시스템의 특정 일부인 "인터랙션 매니저 [6]" 파트로 나뉜다. 그림에서 "공간"은 다양한 구성요소들인 "공간 오브젝트" 를 포함하고 있으며, 해당 공간 오브젝트는 사용자의 인터랙션의 대상이 되거나 인터랙션을 발생 시킬 수 있는 요소들로 '컴퓨팅 능력을 가지고 있느냐 없느냐' 와는 상관없이 관리와 제어의 대상이 될

수 있는 모든 구성 요소들은 공간 오브젝트로 간주된다. 관리 대상이 되는 공간 오브젝트는 인터랙션의 주체가 되는 사람을 포함하여 디스플레이, 오디오/비디오 장치, 개인 컴퓨터 머신, 공용 머신, 캠코더, 인터랙션 장치, 다양한 콘텐츠 등으로 역할에 따라 입력용, 출력용으로 나뉠 수 있다.

"인터랙션 매니저" 파트는 다시 공간으로부터 입력 정보를 수집하는 "입력 수집부 "와 수집된 정보를 분석하여 가장 적절한 태스크를 결정할 수 있는 " 결정 엔진부 ", 마지막으로 결정된 태스크의 수행을 위해서, 출력용 공간 오브젝트가 이해할 수 있는 형태로 포맷을 변환해주는 " 출력 변환부 "로 나뉜다. 그러나 실제 협업 공간은 다양한 종류의 오브젝트가 존재하며, 사용자에게 최종적으로 제공되는 태스크의 형태에 따라 복잡한 처리 형태를 갖는다. 이러한 환경에서 인터랙션 매니저가 어떠한 구조와 규칙을 이용하여 공간 인터랙션을 수행하는 지에 대한 내용은 다음 파트에서 기술하도록 한다.

2.2 SMeet 인터랙션 매니저 설계

위에서 언급된 바와 같이 공간 오브젝트는 역할에 따라 "입력용 공간 오브젝트" 와 "출력용 공간 오브젝트" 로 나뉘게 된다. 입력용 공간 오브젝트는 인터랙션을 발생시키는 역할을 하고 그 입력으로 사용되는 정보를 제공하는 공간 오브젝트들이고 출력용 공간 오브젝트는 인터랙션의 대상이 되어 입력 받은 정보에 따라 변화가 발생 되는 공간 오브젝트들을 일컫는다. 결과적으로 공간 인터랙션을 수행함에 있어서, 인터랙션 매니저의 가장 큰 역할은 기능과 역할에 따라 다양하게 구분되어 있는 오브젝트를 관리하며, 입력용 공간 오브젝트로부터 수신된 데이터를 이용하여 가장 일치하는 형태의 "태스크" 를 선정하고 마지막으로 출력용 공간 오브젝트를 제시하도록 하는 것이다. 여기에서, "태스크" 는 공간 인터랙션에 의해서 수행되는 결과를 일컫는 것으로, 본 논문에서는 입력용 공간 오브젝트의 정보에 의해 출력용 공간 오브젝트에 영향을 끼치는 모든 오퍼레이션 단위를 나타낸다. 예를 들어, 디스플레이 상에 콘텐츠 파일을 올리거나 또는 내리는 기능, 또는 해당 콘텐츠 파일을 협업 공간의 다른 디스플레이로 이동시키는 기능 등이 모두 태스크의 일례라 할 수 있다.

인터랙션을 수행하는 사용자의 최종 의도를 파악하여 태스크를 수행하기 위한 연구로써 싱가폴 국립 대학의 SemanticSpace[5]가 사용하는 방법은 Context reasoner 를 이용하는 것으로 이는 forward-chain reasoning 을 사용한 rule-based system 으로 인공 지능 알고리즘을 적용한 추론을 수행한다. 그러나 그들은 각각의 이벤트마다 프로세스를 수행하고 결과를 저장하지 않는 방법을 취함으로써 동일한 입력이 수신되어도 매번 반복해서 알고리즘을 수행시켜야 하는 단점이 있다.

우리는 이러한 단점을 극복하고, 효율적인 태스크 맵핑 알고리즘을 이용하여 공간 인터랙션을 지원하기 위해 그림 3과 같은 인터랙션 매니저 구조를 설계한다. 여기에서 인터랙션 매니저는 다자간 지능형 협업 시스템인 GIST 의 SMeet (***S***mart **Meet**ing Space)의 일부 기능 모듈로서 사용자의 인터랙션을 지원하기 위한 기능을 담당한다. "인터랙션 매니저(Interaction Manager: IM)" 는 SMeet환경에서 Multi-Tracker [7] 나 Space Mouse 와 같은 인터랙션 디바이스와의 직접적인 인터페이스를 가지며, 이 외에도 SMeet Mediator, 그리고 각 서비스 매니저와의 인터페이스를 구성한다. 원격지와의 인터랙션 공유를 위해 Mediator 의 중재를 통한 원격 서비스 매니저와의 인터페이스를 가질 수 있으며 내부적으로 GUI를 통한 정보 제공 기능을 수행한다. IM 의 내부는 인터랙션 디바이스와의 인터페이스를 통해 입력을 처리하고 해당 디바이스의 상태를 관리하는 "인터페이스 관리 파트" 와 Mediator 와의 연계를 통해 필요한 컨텍스트 정보를 저장하고 관리하는 "IM 컨텍스트 관리 파트", 실제 공간 인터랙션을 처리하기 위한 주요 결정 프로세스를 처리하는 "공간 인터랙션 관리 파트" 그리고 요청된 태스크를 처리하기 위해 출력을 적합하게 변환시키는 "출력 관리 파트" 로 구성되며 부가적으로 "SMeet 인터랙션 GUI" 를 관리하고 있다. 이들 간의 세부적인 인터페이스 내용은 그림 3에 각기 기술되어 있다.

그러나 실제 협업 공간은 사용자에게 최종적으로 제공되는 태스크의 형태에 따라 한 개의 오브젝트 정보가 입력으로 사용되거나 또는 두 개 이상의 오브젝트 정보가 동시에 입력으로 사용될 수 있는 등 복잡한 처리 형태를 갖는다. 이러한 관계를 그림으로 도시한 것이 그림 4에 해당하며 이는 입력용 공간 오브젝트와 태스크와의 관계를 보여주고 있다. 하나의 입력용 공간 오브젝트가 한 개의 태스크와 연관관계를 맺는 경우(Case1), 또는 하나의 입력용 공간 오브젝트가 2개 이상의 태스크와 각각 독립적으로 관계를 맺는 경우(Case2), 각기 다른 입력용 공간 오브젝트 여러 개가 독립적으로 동일한 태스크와 연관관계를 맺거나(Case3) 또는 기술된 모든 입력용 공간 오브젝트가 통합되어 하나의 태스크와 연관관계를 맺는 경우(Case4) 등을 보여주고 있다. 이때, 각각의 연관관계가 의미하는 것은 해당 입력용 공간 오브젝트를 수신하면 태스크가 수행될 수 있다는 것이다. 유사한 방식으로 출력용 공간 오브젝트와 태스크와의 관계가 존재하며 사용자에게 적절한 태스크를 수행하기 위해 인터랙션 매니저는 유기적으로 변경되는 태스크와 입출력 공간 오브젝트와의 관계를 관리해야 할 필요성이 생긴다. 본 논문에서는 이러한 입/출력 공간 오브젝트와 태스크와의 관계를 "템플릿" 이라고 명명하고 수신된 입력 정보를 바탕으로 적절한 태스크를 선택하는 규칙으로 활용하도록 한다.

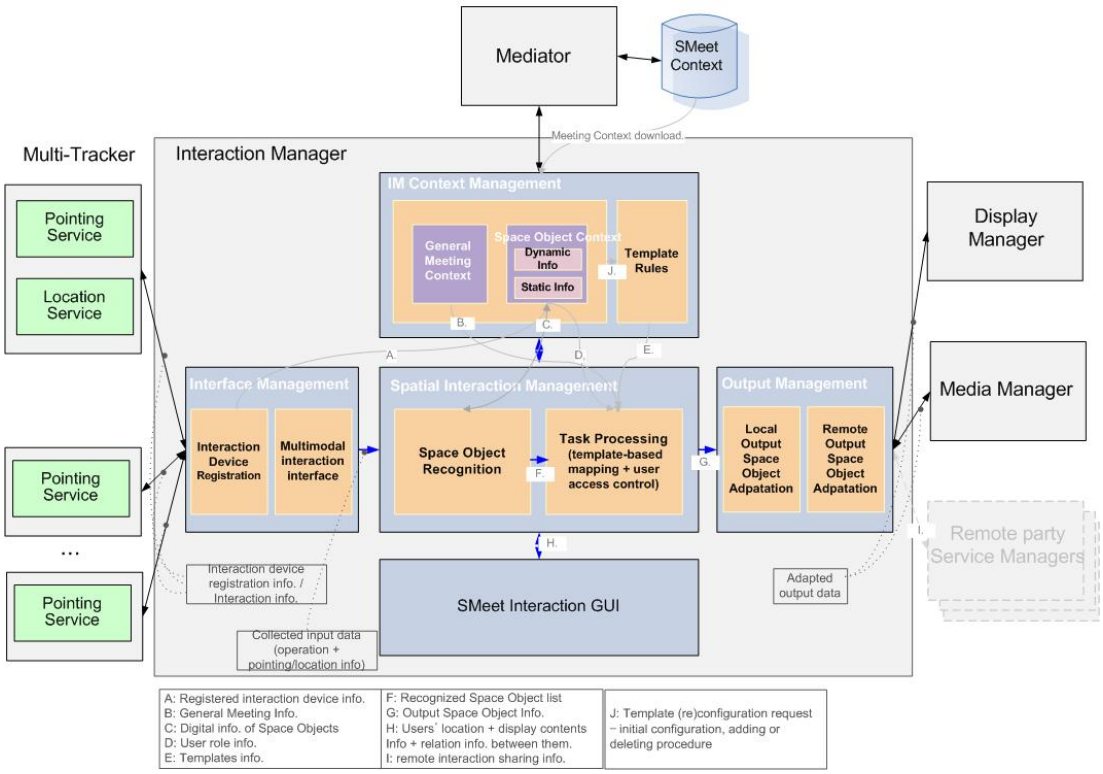

그림 3. 인터랙션 매니저의 Software 구조.

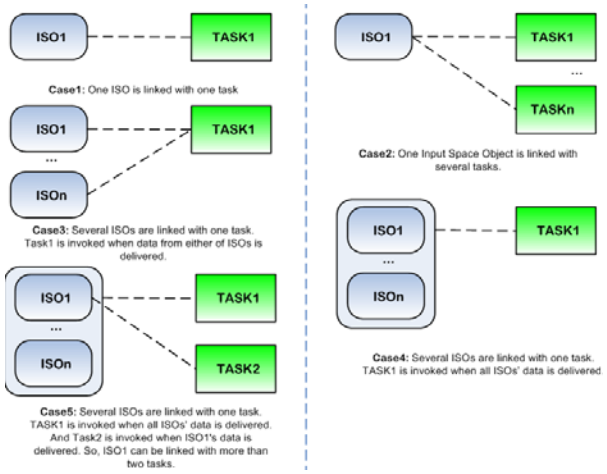

그림 4. 입력용 공간 오브젝트와 태스크와의 관계.

2.3 태스크 추론 알고리즘

본 절에서는 태스크를 선정하기 위한 방법으로 우리가 제안한 템플릿을 구성하기 위한 방법 및 관리, 그리고 실제 어떻게 템플릿을 이용하여 태스크를 찾을 수 있는 지에 대한 상세한 설명을 포함한다.

2.1.1 템플릿의 초기화 및 구성

모든 태스크가 수행되기 위해서는 각자 요구되는 기능들이 있다. 이를 오퍼레이션이라고 하고 태스크를 기술할 때 해당 기능에 대한 명시가 수반되어야 한다. 마찬가지로 입출력용 디바이스로 사용되는 공간 오브젝트들도 사용될 수 있는 오퍼레이션에 대한 정보를 기술하는데 이 둘 간의 맵핑 관계를 이용하여 템플릿을 생성할 수 있다.

T: SMeet 의 모든 태스크 집합,

O_T: SMeet 의 모든 오퍼레이션 집합,

t_i: i 번째 태스크,

o_a: a번째 오퍼레이션 (주로 하나의 기능이 할당되지만 멀티모달 기능이 요구되는 경우 여러 개의 기능이 조합된 형태를 대표한다),

OIt_i: i번째 태스크가 요구하는 입력용 오퍼레이션 집합,

OOt_i: i번째 태스크가 요구하는 출력용 오퍼레이션 집합,

OI_j: j번째 입력용 공간 오브젝트가 소유한 오퍼레이션 집합,

OO_k: k번째 출력용 공간 오브젝트가 소유한 오퍼레이션 집합이라고 정의하면 다음과 같이 표기가 된다.

$T = \{ t_1, t_2, t_3, \ldots, t_n \}$, n = 총 태스크의 개수,

$O_T = \{ o_1, o_2, o_3, \ldots, o_m \}$, m= 총 오퍼레이션의 개수,

$OIt_i = \{ o_1, o_2, o_3, \ldots, o_{nti} \}$, $nt_i = t_i$ 태스크의 입력용 총 오퍼레이션 개수,

OOt_i = { o_1, o_2, o_3, …, o_{lti}}, lt_i = t_i 태스크의 출력용 총 오퍼레이션 개수,

OI_j = { o_1, o_2, o_3, …, o_{nIj}}, nI_j = 입력용 I_j 오브젝트가 소유한 총 오퍼레이션 개수,

OO_k = { o_1, o_2, o_3, …, o_{nOk}}, nO_k = 출력용 O_k 오브젝트가 소유한 총 오퍼레이션 개수.

이때 t_i 태스크와 I_j, O_k 입출력 공간 오브젝트 간에 리스트가 형성되기 위해서는 (1)과 같은 조건을 만족해야 한다.

$$\{ OIt_i \cap OI_j \neq \varnothing \} \text{ AND } \{ OOt_i \cap OO_k \neq \varnothing \} \quad (1)$$

위의 규칙에 의하여 구성되는 템플릿의 예를 그림 5에서 보여준다. 그림 5의 a) 는 오퍼레이션 정보에 의해 관련된 입출력 공간 오브젝트와 태스크의 연관 정보를 그림으로 나타내는 것이고 b), c) 는 이 중 입력용 공간 오브젝트와 태스크와의 관계를 표로 나타내고, 마지막으로 d)는 태스크와 입출력 공간 오브젝트와의 연관 관계를 표로 나타낸다. 이때, 독립적으로 태스크와 연계될 수 있는 오브젝트들의 집합을 표현하기 위해 "()" 를 사용하며 (1) 로 표기되는 것은 오브젝트 ID 1번이 독립적으로 태스크와 연계되는 것이고 (2, 3) 으로 표현되는 것은 오브젝트 2번과 3번이 모두 입력되었을 경우만 태스크를 기동시킬 수 있음을 나타낸다. 최종적으로 생성된 템플릿의 구조는 그림 6에 나타난다.

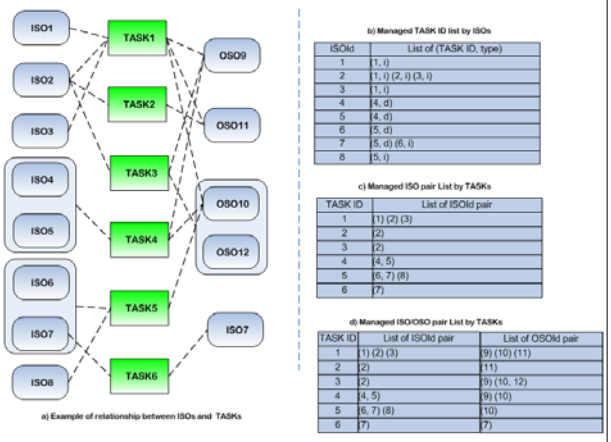

그림 5. 템플릿의 생성 환경 및 태스크와 공간 오브젝트 간의 관계도.

초기에 생성된 템플릿은 공간의 오브젝트 상태 정보가 변경됨에 따라 재구성되는데 예를 들어 새로운 공간 오브젝트가 추가되거나 또는 삭제될 때 해당 정보를 반영하여 다시 템플릿을 생성하게 된다.

- 오브젝트 추가: 먼저, 추가된 오브젝트가 소유하는 오퍼레이션을 확인하여 해당 오퍼레이션이 필요한 태스크와 연관관계를 맺도록 한다. 이때, 해당 오브젝트의 추가로 인해 태스크가 활성화(서비스 제공 가능 상태) 되었다면 해당 태스크와 연관관계를 맺는 모든 입력용 공간 오브젝트에 해당 태스크 정보를 저장하고 활성화된 태스크를 템플릿 리스트에 포함하도록 한다. 만약 오브젝트의 추가로 태스크가 활성화되지 않는다면, 필요한 오퍼레이션 기능 등이 아직 완벽하게 갖춰지지 않은 상태를 의미한다, 단순히 추가된 오브젝트 정보를 태스크 정보에 저장하도록 한다.

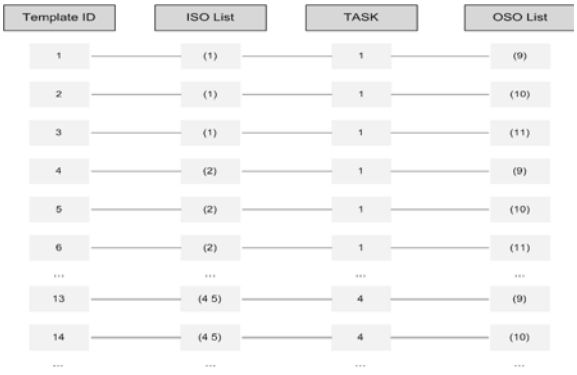

그림 6. 실제 템플릿의 구성.

- 오브젝트 삭제: 삭제되는 오브젝트와 연관관계를 맺고 있는 모든 태스크 정보를 확인하여, 해당 삭제로 인해 비활성화가 되는 태스크가 있다면 해당 태스크 정보를 템플릿에서 삭제하고, 본 태스크와 연관관계를 맺고 있는 모든 입력용 공간 오브젝트에서 태스크 정보를 삭제하도록 한다.

위와 같이 템플릿을 운용하는 경우, 공간 안에 새로운 오브젝트가 유동적으로 추가되거나 삭제되더라도 해당 오브젝트 정보를 포함하는 템플릿을 새로 구성하기만 하면 태스크 수행이 가능하다.

2.1.2 템플릿의 운용

이렇게 관리되는 템플릿을 이용하여 실제 태스크를 추론하는 과정은 다음과 같다. 공간 오브젝트로부터 입력 정보가 수신되면 우리는 수신한 정보를 이용하여 공간 오브젝트 리스트를 생성하고 이와 일치하는 템플릿이 있는 지 확인한다. 템플릿이 있는 경우 해당 태스크를 수행하고 그렇지 않으면 사용자에게 관련된 태스크를 찾을 수 없음을 알린다.

그러나, 협업 환경에서의 공간 오브젝트는 협업 시스템 성능이 향상됨에 따라 유동적으로 증가 될 수 있다. 이런 상황에서는 공간 오브젝트 개수가 증가할수록 태스크 맵핑을 위한 계산에 소용되는 컴퓨터 비용이 기하급수적으로 증가할 것이므로 이에 대한 개선책이 필요하므로 본 논문에서는 이러한 문제를 해결하기 위해 과거에 수행된 템플릿의 히스토리를 저장하고 해당 데이터를 기반으로 템플릿 정보를 새로 구성하는 방법을 제안한다. 협업 공간에서 수행되는 태스크는 주로 사용되는 오퍼레이션이 반복되어 사용되는 특성을 나타내는데 예를 들어 우리의 SMeet 환경 하에서는, 협업 사용자가 인터랙션 디바이스를 이용하여 디스플레이의

콘텐츠를 이동할 수 있는 "인터랙티브 디스플레이 태스크" 가 사용자가 이동함에 따라 기동되는 "follow-me-display" 태스크보다 실제적으로 더 빈번하게 발생됨을 확인할 수 있다. 이러한 전제하에 사용자가 기동시키는 템플릿의 히스토리 정보 (발생 횟수)를 저장한 후 이의 정보를 이용 오름차순으로 템플릿을 정렬하게 되면 템플릿 검색 시 소요되는 시간이 효율적으로 축소될 수 있다. 본 절에서 제안된 알고리즘은 2.5절의 실험 및 결과에서 다시 검증 및 분석되도록 한다.

2.4 공간 인터랙션 GUI

다중 노드 협업 공간을 제어하는 시스템은 원격의 사용자들 간에 진행되는 협업 내용을 쉽게 이해하고 공유할 수 있는 방법이 필요한데, 보편적으로 그래픽 기반의 사용자 인터페이스를 사용하며 특히, 공간 인터랙션이라는 특성을 위해 다음과 같은 사항이 만족되어야 한다. 첫째로, 협업을 구성하는 모든 공간 오브젝트들을 나타낼 수 있어야 한다. 다음으로, 각 오브젝트를 이용한 어떤 인터랙션이 진행되는 경우 해당 인터랙션 정보가 이루어지는 관계를 실감적으로 제공할 수 있어야 한다. 이러한 요구 사항을 만족시키기 위해 본 논문에서는 협업 공간의 공간 인터랙션을 공유할 수 있는 공간 그래픽 사용자 인터페이스를 제안하며 현재 버전의 공간 인터랙션 GUI를 그림 7에 제시한다. 공간 인터랙션 GUI는 먼저 공간 오브젝트 종류 중의 하나인 사용자 정보를 제공 한다. 즉, 협업 환경내의 사용자 위치 정보 등의 데이터를 처리할 수 있도록 하며 인터랙션이 수행되는 경우 종류/태스크 별로 구분할 수 있는 방법을 제공한다. 이와 더불어 사용자가 발생시킨 인터랙션이 수행중인 경우 해당 오브젝트간의 연관 정보를 그림에서와 같이 선으로 표시하도록 한다.

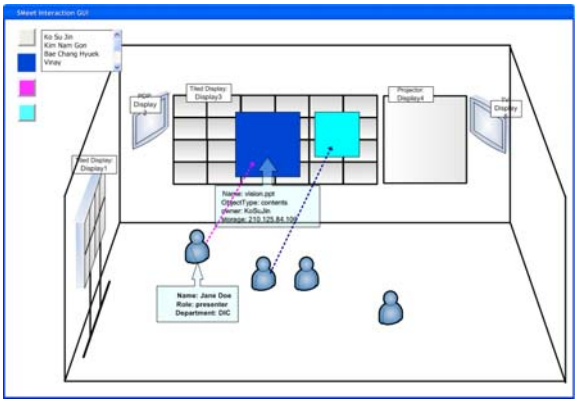

그림 7. 공간 인터랙션 GUI.

2.5 실험 및 결과

본 논문에서 제안하는 공간 인터랙션은 협업 시스템이 관리하는 공간 오브젝트의 종류가 다양할수록 해당 공간 오브젝트를 이용하여 제공할 수 있는 태스크의 개수가 다양하게 되며, 이에 따라 적절한 태스크를 수행할 수 있도록 지원해 주는 일의 복잡도가 증가하게 된다. 예를 들어, 협업 공간 안의 입력용 공간 오브젝트의 개수가 3개인 경우와 10개인 경우 제공할 수 있는 태스크의 개수는 큰 차이가 있게 된다. 특히 하나의 태스크가 한 개의 입력으로 제공되는 것이 아니라 여러 개의 입력이 연결되어 제공되는 경우라면, 태스크의 오브젝트 구성 개수가 늘어날수록 적절한 태스크 템플릿을 찾는데 드는 비용이 늘어날 것이라는 추측이 가능하다. 이러한 가정 하에 적절한 템플릿을 검색하여 처리하는 데 드는 시스템의 비용이 공간 오브젝트의 개수와 템플릿 구성 복잡도에 따라 어떻게 변하는 지에 대한 분석을 위해 다음과 같은 실험 모델을 제시한다.

N : 입력용 공간 오브젝트의 개수

R : 하나의 태스크 안에 최대한 구성될 수 있는 오브젝트의 개수

T_N : 입력용 공간 오브젝트 개수가 N개 일 때, 시스템에서 최대로 제공될 수 있는 템플릿 개수

즉, 하나의 협업 시스템 안에 최대로 생성될 수 있는 템플릿의 개수는 ${}_NC_1 + {}_NC_2 + \ldots + {}_NC_R$ 을 모두 합한 개수로 다음과 같이 표기된다.

$$T_N = \sum_{i=1}^{R} {}_NC_i.$$

그림 8은 입력용 공간 오브젝트와 템플릿 구성 개수에 따른 만들어 질 수 있는 최대 템플릿 개수를 보여준다.

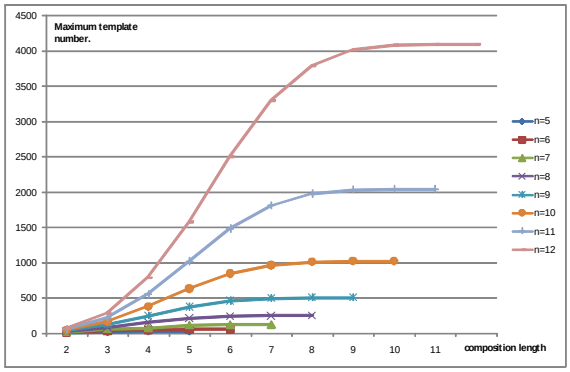

그림 8. 입력용 공간 오브젝트와 템플릿 구성 개수에 따른 최대 템플릿 수 상관 그래프.

그림에서 보이는 것처럼 입력용 공간 오브젝트의 개수는 5개부터 12개까지이고 이 때 각각 템플릿 구성 개수가 2개부터 12개일 경우의 총 템플릿 개수가 표시되어 있다. 예를 들어 입력용 공간 오브젝트가 12개인 경우 구성 개수를 최대 3개까지 하는 경우 만들어질 수 있는 템플릿 개수는 298

개이다.

위의 조건에서 우리가 관심을 갖는 결과는 입력용 공간 오브젝트 개수가 증가하고 템플릿의 복잡도가 증가함에 따라 입력으로 들어온 오브젝트 데이터 셋을 처리하기 위해 템플릿을 검색하는데 어느 정도의 시간이 소요되는지 이다. 이를 위해 한 개의 오브젝트로부터의 입력 또는 2개 이상의 입력 50개로 이루어진 데이터 셋을 입력 오브젝트 개수 별로 각각 생성하고(총 8개의 데이터 셋) 각 데이터 셋을 이용하여 태스크 맵핑을 처리하는데 소요되는 시간을 체크한다. 이 때, 테스트의 편이성을 위해 하나의 특정 포맷의 입력 데이터는 하나의 태스크와만 연관 관계가 있다고 가정하도록 한다.

이 때, 실험이 수행되는 환경은 SMeet 시스템 하의 IM과 준비된 데이터 셋을 이용하여 입력 이벤트를 발생시킬 수 있는 시뮬레이터가 기동되도록 하며 해당 시뮬레이터는 MS Windows XP os 를 탑재한 Dell Precision T3400 제품에서 기동되고 MS visual studio 2005 로 프로그래밍 되어 있는 버전을 이용한다. IM의 경우는 Linux Fedora core 6 버전의 C++ 프로그램으로 되어 있고 태스크 맵핑을 처리하는 데 소요되는 시간은 템플릿 검색 함수의 소요 시간을 체크함으로 나타낸다.

그림 9는 인터랙션 히스토리 정보를 반영하지 않은 템플릿 알고리즘을 이용한 경우의 결과를 나타낸다.

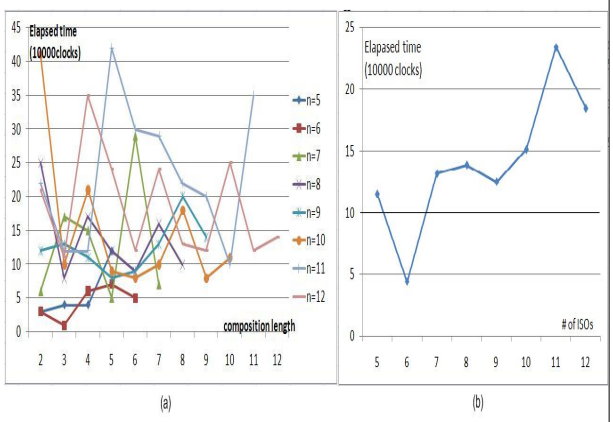

그림 9. 입력용 공간 오브젝트 수와 템플릿 구성 개수에 따른 템플릿 검색 소요시간.

그림 9의 (a)에서 보이는 데이터는 적절한 태스크를 선정하는데 소요되는 시간의 합으로 위에서 생성한 데이터 셋을 주어진 파라미터 값 (입력용 공간 오브젝트 개수와 최대 템플릿 구성 개수)을 변경하면서 총 10번씩 수행한 결과를 clock 단위로 나타낸 것이다. 그림에서의 단위는 10000 clocks를 사용하고 있으며, 각 입력용 공간 오브젝트 별로 소요되는 시간의 평균은 (b)에서 확인할 수 있는 것처럼 입력용 공간 오브젝트의 개수가 증가 할수록 템플릿을 선정하는 데 소요되는 시간이 전체적으로 증가함을 알 수 있다.

이 실험에서 사용된 최대 입력 공간 오브젝트의 개수는 12이나 실제 공간상의 공간 오브젝트 개수는 협업 환경이나 시스템 성능이 향상됨에 따라 유동적으로 증가가 가능하다. 이런 상황에서는 공간 오브젝트 개수가 증가할수록 계산에 소용되는 컴퓨터 비용이 기하급수적으로 증가할 것이므로 이에 대한 개선책이 필요하게 되고, 본 논문에서는 이러한 문제를 해결하기 위해 과거에 수행되는 템플릿의 히스토리를 저장하고 해당 데이터를 기반으로 템플릿 정보를 새로 구성하는 방법을 제안하였다. 이러한 방법을 적용한 후 다시 실험을 한 결과를 나타내면 그림 10과 같다. 실험에서 사용되는 데이터 집합은 그림 9의 결과에서 사용된 것과 동일하며 실제로 시스템 전반에 소요되는 시간은 새로 템플릿을 구성하는 시간이 추가되므로 소요시간이 전체적으로 향상되었다고 언급할 수는 없으나 템플릿을 새로 구성하는 시간은 사용자의 요구에 의해 시스템의 성능에 맞게 횟수를 조정할 수 있으므로 해당 시간은 무시하도록 한다.

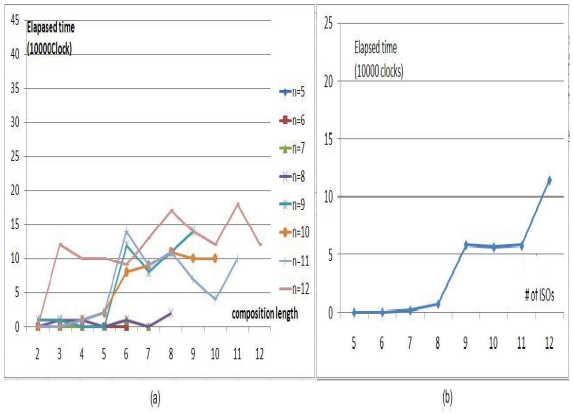

그림 10. 히스토리 정보를 적용한 후 템플릿 검색 소요시간.

본 실험에서 히스토리 정보를 적용하여 템플릿의 재구성은 한 번만 실시하도록 되어 있으며 10번의 반복 시도 중, 첫 번째 시도 때의 데이터를 이용하여 구성하도록 되어 있다. 다른 테스트 절차는 앞과 동일하며 그림 10의 (a)는 총 10번씩 수행한 결과의 합을 보여주고, (b)는 입력용 공간 오브젝트 개수별로 소요되는 시간의 평균을 보여주고 있다. 그림 9의 (b)와 그림 10의 (b) 결과를 서로 비교했을 때, 히스토리 정보를 이용하여 수정된 알고리즘을 사용한 경우 원래의 알고리즘을 사용한 경우의 27% 정도의 시간만을 소요하여 템플릿 검색을 완료함을 확인할 수 있다.

3. 결론

본 논문에서는 효율적인 공간 인터랙션을 제공하기 위한

시스템 및 방법을 제공하며 특히 공간 인터랙션을 수행하기 위한 공간 오브젝트 개념을 도입하고 해당 기능의 효율적인 수행을 위해 인터랙션 히스토리 정보를 이용한 템플릿 기반의 태스크 선정 방법을 보여주었다. 또한 공간 인터랙션의 결과를 다른 지역의 참석자와 공유하기 위한 시각적인 방법인 공간 인터랙션 GUI를 소개하였다. 마지막으로 템플릿 기반의 태스크를 사용하는 경우, 입력용 공간 오브젝트 개수가 증가하고 템플릿의 복잡도가 증가함에 따라 템플릿을 검색하는데 소요되는 시간을 체크하기 위한 모델을 생성하고, 공간 오브젝트 개수의 추가적 증가를 지원하기 위해 과거의 템플릿 히스토리 정보를 반영하여 우선순위를 두는 수정된 템플릿 알고리즘과의 성능 비교를 위한 테스트를 실시하고 제안된 알고리즘이 어느 정도 향상되는지 실험 후 결과를 고찰하였다.

참고문헌

[1] S. Han and et al., "Design of multi-party meeting system for interactive collaboration," Proc. IEEE Int. Conf. On Communication System and Software and Middleware (IEEE COMSWARE 2007), Jan. 2007.

[2] N. A. Streitz, "From Human-Computer Interaction to Human-Environment Interaction," ERCIM UI4ALL Ws 2006, LNCS 4397, pp. 3-13, 2007.

[3] P. Tandler, "The BEACH application model and software framework for synchronous collaboration in ubiquitous computing environments," Journal of Systems and Software (vol. 69/3), Jan. 2004.

[4] B. Johanson et al., "The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms," Pervasive Computing Magazine Special Issue on Systems (vol. 2), pp. 67-74, 2002.

[5] X. Wang et al., "Semantic Space: An Infrastructure for Smart Spaces," IEEE Pervasive Computing (vol. 3/3), pp. 32-39, jul-sept, 2004.

[6] Ko, S. et al., "Design of interaction manager supporting collaborative display and multimodal interaction for advanced collaborative environment." Proc. of SPIE, 6777, 67770S (2007), Boston, MA.

[7] Ko, S. et al., "Multi-Tracker: Interactive Smart Object for Advanced Collaborative Environment (ACE)" Design and Integration Principles for Smart Objects 2008 at Ubicomp2008 (IEEE DIPSO2008), Sep. 2008.