

생활 데이터 분석을 통한 개인화된 상황인식서비스 생성

(Deriving Personalized Context-aware Services from
Activities of Daily Living)

박 정 규 [†] 이 금 해 ^{**}
(Jeongkyu Park) (Keung Hae Lee)

요 약 대부분의 상황인식서비스는 공급자 또는 개발자에 의해 정의되고 있다. 그러나, 자신에게 무엇이 필요한지는 사용자가 가장 잘 알 수 있기 때문에, 사용자가 스스로 필요한 상황인식서비스를 정의할 수 있게 해주는 방법도 중요하다. 우리의 생활공간이 지능공간으로 변화되어 감에 따라 개인화된 상황인식서비스의 정의 수단은 더욱 중요해질 것이다. 본 논문은 사용자의 일상생활을 기록한 데이터로부터 유용한 서비스를 추출해주는 새로운 서비스 정의방법 CASPER를 소개한다. CASPER는 사용자의 생활 기록에 나타난 사건의 인과관계를 분석함으로써 유용한 서비스를 생성해주는 효과가 있다. 본 논문은 CASPER의 알고리즘과 그 유효성 검증 실험에 대해 설명한다.

키워드 : 상황인식서비스, 인간과 컴퓨터 상호작용, 행동분석, 개인화된 서비스

Abstract Currently, most context-aware services are built by developers. Some researchers argued that services should be defined by end users, who understand their own needs best. We view that the significance of enabling the user to define his/her personalized services will multiply as our living spaces grow smarter. This paper introduces a novel method called CASPER, which is capable of deriving personalized services from the log of user's activities of daily living. CASPER can generate useful services that even the user may not perceive, mining causality of events in the log. We present the algorithm of CASPER in detail and discuss the result of an experiment which we conducted as a proof of concept.

Key words : Context-aware Service, HCI, Behavior Analysis, Personalized Service

1. 서 론

상황인식서비스(Context-aware service)는 사용자 주변의 컴퓨팅 기기가 협력하여 능동적으로 사용자에게

유용한 서비스를 제공해주는 서비스제공모델이다. 상황인식서비스가 제대로 효과를 얻기 위해서는 서비스 자체가 사용자에게 유용한 것이어야 한다. 훌륭히 구현된 고가의 상황인식서비스라고 해도 사용자가 불편하거나 불필요하다고 느낀다면 그 가치는 매우 낮을 것이다. 지금까지의 연구나 개발사례를 보면 대부분의 상황인식서비스는 전문지식을 갖춘 공급자(개발자)가 정의하는 것으로 가정하고 있다[1-3]. 그러나 자신에게 필요한 서비스가 무엇인지 가장 잘 알고 있는 사람은 바로 사용자 자신이기 때문에 개인을 위한 상황인식서비스는 해당 사용자가 직접 정의하는 것이 효과적일 수 있다. 본 논문은 개인을 위한 상황인식서비스를 정의해주는 CASPER 방법에 대하여 설명한다. CASPER는 일상생활에서 관찰된 사용자의 활동기록을 보고 이들 간의 인과관계를 인식함으로써 사용자에게 유용한 서비스를 생성해주는 개인용 상황인식서비스 정의방법이다. 본 논문은

· 본 연구는 교육과학기술부/한국연구재단의 2009년도 일반연구지원 기초연구사업에 의해 수행되었음

[†] 학생회원 : 한국항공대학교 컴퓨터공학과
fcopark@kau.ac.kr

^{**} 종신회원 : 한국항공대학교 컴퓨터공학과 교수
khlee@kau.ac.kr

논문접수 : 2010년 1월 20일

심사완료 : 2010년 3월 9일

Copyright©2010 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 받고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제5호(2010.5)

상황인식서비스의 생성에 사용되는 CASPER 알고리즘과 그 효과에 대하여 상세히 설명한다.

2. 관련연구

상황인식서비스에 관련된 연구로 상황인식미들웨어, 상황모델링, 상황인식과 추론, 서비스정의 등 다양한 분야가 있다. 여기서는 본 논문과 직접 관련이 있는 상황인식서비스 정의방법과 사용자 행동패턴 인식 두 그룹으로 나누어 살펴본다.

상황인식서비스 정의방법에 관한 연구는 개발 주체에 따라 개발자정의와 사용자정의로 분류할 수 있다. 개발자정의 방법의 대표적인 연구로는 ECA-DL[3], SOCAM[2]이 있다. ECA-DL과 SOCAM 언어는 표현성이 높지만 언어를 사용하기 위해서는 프로그래밍 수준의 지식이 요구되기 때문에 일반사용자에게는 사용하기가 어렵다는 단점이 있다. 사용자정의 방법의 대표적 연구에는 JIGSAW[4]와 a CAPpella[5]가 있다. JIGSAW는 이해하기 쉬운 그래픽언어로, JIGSAW 사용자는 “대문 열림”과 같이 발생할 수 있는 상황을 그림으로 표현한 퍼즐조각들을 나열함으로써 원하는 서비스를 정의한다. a CAPpella는 앞의 방법들과 달리 “Program by demonstration”을 제안하고 있다. 이 방법에서는 사용자가 새로운 서비스를 정의하기 위하여 센서들이 설치된 개발 시스템 앞에서 행동을 시연한 후, 그 기록을 보고 서비스와 관련된 상태들을 선택함으로써 서비스를 정의한다. 본 논문에서 설명하는 CASPER는 이러한 두 연구처럼 사용자에 의한 서비스 정의를 지향하지만 “서비스를 어떻게 표현할 것인가”라는 물음에 답하려는 것이 아니라 “무엇을 서비스로 할 것인가”라는 물음, 즉, 새로운 서비스의 내용을 만드는 방법이라는 점에서 차이가 있다.

인간의 행동패턴을 인식하기 위한 연구는 패턴인식, 데이터마이닝 등의 다양한 분야에서 이루어지고 있다, CASPER는 인간 행동 기록을 이벤트열(event sequence)로 추상화하고 이를 분석하여 인과관계가 있는 패턴을 인식하는데, 이와 관련 있는 연구로는 데이터마이닝 기법으로 패턴을 인식하기 위한 EPISODE[6-8] 연구가 있다. 이 연구들은 모두 발생 빈도가 높은 이벤트열 패턴을 찾는 것을 목표로 한다. CASPER는 이와 다르게 둘 사이에 인과관계가 있는 이벤트의 쌍을 찾아내어 이를 상황인식서비스로 정의한다. CASPER는 이들 연구와 후보집합 생성, 신뢰도 계산 등의 구체적 구현 방법에 차이가 있고 “알람이 울리면 곧 꺼진다.” 등과 같이 상황인식서비스로 제공하기에 의미적으로 불합리한 규칙들을 제거하기 위한 정제 단계를 가지고 있다는 점에서 구분된다.

3. CASPER

그림 1은 CASPER를 위한 상황인식시스템의 아키텍처 모델이다. 지능공간 내의 각 기기들은 상황인식미들웨어의 기기관리기와 네트워크로 연결되어 있다. 기기관리기는 지능공간 내에 존재하는 기기들을 관리하며 기기에서 발생하는 이벤트가 이벤트저장소에 저장되도록 해준다. 이벤트저장소에는 발생한 이벤트가 순차적으로 기록된다. 기기는 지능공간 내에 존재하는 전자장치들로서 가정의 경우에는 TV, 냉장고, 전자도어, 전자커피트러펍 상황인식시스템과 연동이 가능한 장치들을 의미한다.

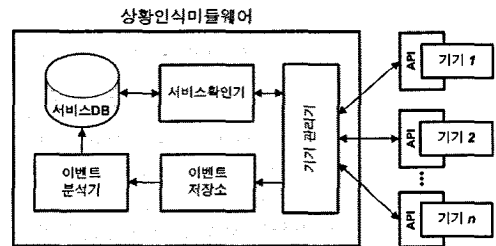


그림 1 CASPER를 위한 상황인식미들웨어 구조

이벤트의 발생은 안방의 전등이 켜진다거나 냉장고의 문이 열리는 것과 같이 사용자가 취하는 행동이 직접 또는 간접적으로 발생시키는 기기의 상태 변화를 의미한다. 본 논문에서 생활데이터라고 부르는 것은 이러한 이벤트들의 발생순서에 따른 모임, 즉, 이벤트열을 가리킨다.

CASPER는 크게 이벤트저장소에 저장된 생활데이터를 분석하여 인과관계가 있는 이벤트들을 생성하는 알고리즘과, 생성된 각 이벤트들을 그에 대응되는 서비스를 상황인식서비스 규칙으로 정의해 서비스DB에 저장하는 두 단계로 구성된다. 여기서 인과관계가 있는 이벤트들은 (A, B)로 표현할 수 있다. 이때 A와 B는 모두 이벤트이고, A가 원인이 되는 사건들, B는 A에 의해 발생된 결과 사건들이다. 원인과 결과는 모두 한 개 이상일 수 있다. 인과관계 (A, B)는 그에 대응된 서비스 규칙, 예를 들면, A→B와 같은 형식으로 표현되어 저장되며, 서비스확인기에 의해 주어진 시간 안에 A에 속한 이벤트가 발생하면 B라는 액션을 수행하게 된다. 상황인식서비스 A→B에서 우리는 A에 속한 이벤트들을 컨디션, B에 속한 이벤트들을 액션이라 부른다. CASPER 알고리즘은 반복된 연속 이벤트열의 패턴인식을 이용하는 다른 방법들과 달리, A와 B 사이에 연관이 없는 다른 사건들이 포함되어 있어도 A와 B의 인과관계를 분석할 수 있다.

기기관리기는 서비스 확인기의 지시를 받아서 기기들

에게 상황인식서비스 실행에 따른 액션 수행을 지시하는 역할도 담당한다. 서비스확인기는 기기에 발생된 이벤트를 보고 서비스DB를 검사하여 이 이벤트를 실행조건으로 가진 서비스가 있다면 기기관리기를 통해 해당 액션을 지시한다. 이벤트분석기는 CASPER를 의미하는 것으로, 이벤트 저장소에 기록된 이벤트들을 분석하여 서비스를 생성하고 생성된 서비스를 지정된 형식으로 서비스DB에 저장한다. 이와 같이 동작하는 CASPER는 사용자의 생활데이터로부터 사용자에게 도움이 될 수 있는 서비스를 자동 생성해준다. 지금부터는 CASPER가 서비스를 생성하는 알고리즘에 중점을 두고 설명하기로 한다.

CASPER는 유사한 시간에 발생한 이벤트를 선별하는 컷오프단계, 검토의 대상이 되는 모든 이벤트 순서쌍들을 생성해주는 분해단계, 이 집합의 각 순서쌍에 대하여 발생빈도를 계산함으로써 후보군을 선정하는 신뢰도계산단계, 그리고, 상황인식 도메인의 의미적 처리를 위한 정제 단계로 구성된다. 지금부터 각 단계를 자세히 살펴본다.

컷오프는 이벤트열에서 유사시간에 발생한 이벤트들을 A→B형태로 선별하는 과정으로, 하나의 이벤트가 컨디션이 되어 액션을 유발하는 1:1 경우를 찾기 위한 액션 컷오프와 두 개 이상의 이벤트가 컨디션이 되어 액션을 유발하는 N:1 경우를 찾기 위한 컨디션컷오프로 나뉜다.

그림 2의 예에서 볼 수 있듯이 액션컷오프는 이벤트열의 모든 이벤트에 대해 순차적으로, 선택된 이벤트(컨디션으로 간주)를 기준으로 일정시간(액션컷오프타임: 예에서는 3초) 후까지 발생한 모든 이벤트(액션후보)를 선택하여 A→B형태의 액션컷오프리스트를 만드는 작업을 반복한다. 반대로 컨디션컷오프는 이벤트열의 모든 이벤트에 대해 각 이벤트(액션으로 간주)를 기준으로 일정시간(컨디션컷오프타임) 전까지 발생한 모든 이벤트(컨디션 후보)를 선택하는 방식으로 컨디션컷오프리스트를 만든다. 앞서 말했듯이 컨디션컷오프는 두 개 이상의 이벤트가 컨디션이 되는 서비스를 생성하기 위함이기 때문에 하나의 이벤트가 컨디션이 되는 경우는 고려하지 않는다. 액션/컨디션컷오프타임은 CASPER 운영자에 의해 결정되는데 일반적으로 N:1관계를 추출하기 위한 컨디션컷오프타임이 1:1관계를 추출하기 위한 액션컷오프타임보다 길다. 그 이유는 다음 과정인 분해를 설명하면서 함께 살펴본다.

컷오프과정에서 추출된 액션/컨디션컷오프리스트들은 다음으로 분해과정을 거친다. 분해는 비슷한 시간에 발생한 이벤트들 중 어떤 것들이 서로 실제 인과관계를 가지는지 계산하기 위한 준비과정이다. 분해는 액션컷오프리스트를 대상으로 하는 액션분해와 컨디션컷오프리스트를 대상으로 하는 컨디션분해로 나뉜다. 그림 2에서 볼 수 있듯이 액션분해는 각 액션컷오프리스트에서 컨

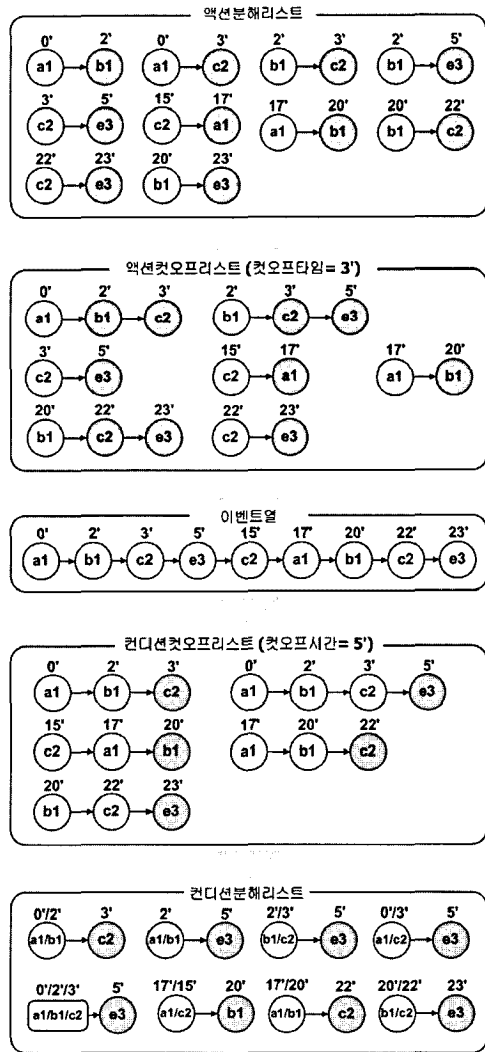


그림 2 CASPER의 컷오프와 분해 예

디션을 기준으로 액션집합의 모든 이벤트와 각각 조합하여 액션분해리스트들을 생성한다. 반대로, 컨디션분해는 각 컨디션컷오프리스트에서 하나의 액션을 기준으로 컨디션집합에서 원소가 두 개 이상을 갖는 모든 부분집합을 조합하여 컨디션분해리스트를 생성한다. 컨디션분해에서 컨디션과 액션의 이벤트 수가 1:1로 연결되는 경우를 제외하는 이유는 컨디션분해가 1:1의 관계를 가진 상황인식서비스를 생성하는데 부적격하기 때문이다. 예를 들어, “TV가 켜지고 전화벨이 울리면 TV가 음소거된다.” 라는 상황인식서비스가 있다고 하자. 이 서비스의 경우 TV가 켜지는 이벤트와 전화벨이 울리는 이벤트가 비교적 긴 시간 간격을 두고 발생할 수도 있다. 반면 “DVD플레이어가 켜지면 TV가 켜진다.”와 같이 1:1

관계를 가진 상황은 컨디션 발생 시 즉시 액션이 발생해야 한다. 만약 컨디션컷오프/분해를 통해 1:1관계를 추출할 경우 긴 시간을 두고 발생하는 이벤트들도 관계가 있는 것으로 계산되기 때문에 부적합하다. 1:1관계를 추출하기 위한 액션컷오프타임은 N:1관계를 추출하기 위한 컨디션컷오프타임보다 작은 것도 같은 이유이다. 생성된 액션분해리스트와 컨디션분해리스트는 컨디션과 액션의 인과관계 분석을 위한 신뢰도계산단계에 이용된다.

신뢰도 계산단계는 컨디션과 액션의 조합으로 표현된 분해리스트에서 해당 컨디션이 발생했을 때 이에 대응하는 액션이 얼마나 빈번하게 발생했는지를 계산한다. A→B에서 A가 발생했을 때 항상 B가 발생한다면 A와 B는 인과관계가 있다고 할 수 있다. 알고리즘1은 신뢰도 계산방법을 보여준다. 계산을 위해서는 최소 발생회수와 연관성임계치가 필요하다. 최소발생회수(m)는 이벤트리스트에서 컨디션이 최소한 m회 이상 일어난 경우만 상황인식서비스로 선택하기 위함이다. 만약 컨디션 C1이 한번 일어나고 이에 대응된 액션 A1이 이어 발생했다면, C1이 발생할 경우 A1이 발생할 확률이 100%이지만 이를 신뢰하기 어렵다. 최소발생회수는 이런 경우를 방지하기 위함이다. 확률을 기반으로 상황인식서비스를 산정하기 위해서는 기준이 되는 확률값이 필요하다. CASPER에서는 서비스산정결과가 연관성임계치(p) 이상이면 상황인식서비스로 선택하고 그렇지 않으면 선택하지 않는다.

알고리즘 1 신뢰도 계산

```

입력: 액션분해리스트의 집합: ADL
      컨디션분해리스트의 집합: CDL
출력: 상황인식서비스 집합: CAS

const int m=m0 //최소발생회수 제한
const float p=p0 //연관성임계값
int countEvent(Event e);
//이벤트리스트에서 이벤트 e의 발생회수 반환함수
int countDL(ADL adl);
//액션분해리스트 adl 발생회수 반환
int countDL(CDL cdl);
//컨디션분해리스트 cdl 발생회수 반환

Service Selection

for x∈ ADL do {
  if countEvent(x.firstelement) >= m
    if (countDL(x)/countEvent(x.firstelement)) >= p
      insert x to CAS
}
for x∈ CDL do {
  if countEvent(x.lastelement) >= m
    if (countDL(x)/countEvent(x.lastelement)) >= p
      insert x to CAS
}

```

정제단계는 선택된 서비스 중 논리적으로 부적합한 인과관계들을 제거하는 단계이다. 정제에는 재귀제거, 충돌제거가 있다. 재귀제거는 동일 장치에서 이벤트와 액션이 동시에 발생하는 경우를 제외하는 것이다(예: “욕실에 전등이 켜지면 곧 전등이 꺼진다.”). 충돌제거는 어떤 실행조건이 같은 개체에 대해 서로 다른 액션을 유발하는 경우를 제외하는 것이다(예: 동시에 일어나는 두 서비스 - “거실등이 꺼지면 현관문 열림” vs. “거실등이 꺼지면 현관문 잠김”). 재귀제거와 충돌제거는 항상 연속되어 발생하지만 인과성이 없는 경우를 제외한다.

4. 실험 및 결과분석

앞에서 설명한 CASPER 알고리즘이 실제로 유용한 서비스를 생성할 수 있는지 그 효과를 확인하기 위하여 실제 데이터에 알고리즘을 적용하여 실험하였다. 실험 수행에 있어서 어려운 점은 개인의 실생활 데이터를 수집하는 것이었다. 아직 가정과 같은 실생활 공간에서 생활데이터를 자동으로 수집할 수 있게 해주는 장치나 방법은 문헌에서 찾기 어렵다. 또한, 지능공간에 존재하는 기기들의 상태변화를 자동으로 기록해줄 수 있는 환경을 실험적으로 구축하거나 이러한 기기에서 발생하는 데이터를 자동으로 수집하여 사용자의 행위를 인식하는 방법 자체도 제한적으로 이루어질 뿐 아직 실용화될 만큼 성숙되어 있지 않다. 현재 센서와 별도의 장치를 부착한 전자기기 등을 이용함으로써 사용자 활동데이터를 수집하기 위한 실험 환경 구축과 이러한 데이터를 이용한 사용자 행위 인식 방법에 대하여 다양한 연구가 진행 중에 있다[7-11]. 이러한 연구가 진전되면, 본 연구에서 필요로 하는 것처럼 지능공간에서의 사용자 활동 데이터를 자동으로 수집할 수 있게 될 것이다.

본 실험의 목적은 알고리즘의 유효성 검증을 위한 것이므로, 이러한 현실적 여건을 고려하여 사용자 활동데이터를 수동으로 수집함으로써 실험하였다. 본 실험을 위하여 연구원 1명을 피실험자로 지정하고, 평일 10일 동안에 걸쳐, 매일 오후 8시부터 다음날 오전 10시까지 집안에서 자신의 활동과 관련하여 일어났던 본 연구와 관련된 모든 이벤트를 기록하도록 하였다. 이러한 방법으로 총 259시간(15574분) 동안 발생한 이벤트가 수집되었다. 그 결과, 348개의 원소로 구성된 이벤트열을 얻게 되었으며, 이 데이터에 CASPER 알고리즘을 적용하여 그 결과를 분석하였다. 운용자에 의해 입력되는 액션컷오프타임(ACT)과 컨디션컷오프타임(CCT)은 각각 5분, 15분, 30분으로 변경하면서 실험을 실시하였고 이와 더불어 연관성임계값(p) 역시 70%, 85%, 100%로 변경하며 실험을 실시하였다. 최소발생회수 m은 10일 동안 최소 이틀에 한 번씩은 발생하는 이벤트들을 고려한다

표 1 CASPER 실험결과

분류	인과관계
적합	거실등이 켜지면, 에어컨이 켜짐
	거실등이 켜지면, TV가 켜짐
	DVD가 켜지면, TV가 켜짐
	거실등이 켜지면, 보일러가 목욕모드로 전환
	에어컨이 켜지면, 보일러가 목욕모드로 전환
	TV가 켜지고 에어컨이 켜지면, 보일러가 목욕모드로 전환
	보일러가 목욕모드로 전환되면, 욕실등이 켜짐
	거실등이 켜지면, 욕실등이 켜짐
	에어컨이 켜지면, 욕실등이 켜짐
	TV가 켜지면, 욕실등이 켜짐
	DVD가 꺼지면, 거실등이 꺼짐
	전화가 걸려오면, TV음소거
	에어컨이 꺼지고 문이 열리면, 차에 시동이 걸림
	에어컨이 꺼지고 TV가 꺼지면, 차에 시동이 걸림
	TV가 꺼지고 문이 열리면, 차에 시동이 걸림
	전화가 걸려오면, DVD가 일시정지
부적합	전화가 걸려오고 전화가 끊기면, DVD 일시정지해제
	전화가 걸리고 전화가 끊기면, TV사운드 정상
	욕실등이 꺼지고 TV가 꺼지면 DVD 켜짐
	보일러가 꺼지고 TV가 꺼지면 DVD 켜짐
	TV가 음소거되면, 전화가 끊김
	DVD가 일시정지되면, 전화가 끊김
	에어컨이 켜지고 욕실등이 켜지면, 보일러가 꺼짐
	에어컨이 켜지고 욕실등이 꺼지면, 보일러가 꺼짐
TV가 켜지고 욕실등이 꺼지면, 보일러가 꺼짐	
TV가 켜지고 욕실등이 켜지면, 보일러가 꺼짐	
욕실등이 켜지고 욕실등이 꺼지면, 보일러가 꺼짐	
욕실등이 꺼지고 보일러가 꺼지면, DVD가 켜짐	

는 의미로 5로 고정하였다. 우리는 이와 같이 파라미터를 변경하면서 총 27번 CASPER를 수행하였다. 그 결과 CASPER는 총 28개의 인과관계를 추출하였다. CASPER가 추출한 인과관계는 유용한 것과 그렇지 않은 것이 섞여 있었다. 우리는 각 서비스를 분석하여 어떤 것이 유용하고 어떤 것이 그렇지 못한지 구분하였다. 그 결과는 표 1과 같다. 표 1에서 볼 수 있듯이 총 28개 중 20개는 인과관계가 있는 것으로 판단되고 나머지 8개는 그렇지 않은 것으로 판단되었다. 이런 결과는 위에서 설명하였듯이 입력 파라미터를 변경하며 27번 CASPER를 수행시킨 결과를 모두 분석한 것이다. 파라미터들이 변경되면 CASPER의 실행결과도 변경된다. 표 2는 파라미터의 변경에 따른 CASPER의 실험결과를 요약한 것이다.

이 표를 설명하기 위해서 예를 들면, 표에 있는 10/1이라는 값은, 액션컷오프타임(ACT)과 컨디션컷오프타임(CCT)이 5분이고 연관성임계값(p)이 60%일 때 CASPER가 10개의 적합한 인과관계와 1개의 부적합 인과관계를 추출하였음을 의미한다. 액션컷오프타임을 15분으로

표 2 실험결과분석

		액션컷오프/분해/서비스산정 (m=5)									
		ACT=5			ACT=15			ACT=30			
		p	60	80	100	60	80	100	60	80	100
컨디션컷오프/분해/서비스산정 (m=5)	CCT=5	60	10/1			13/2			7/2		
		80		7/0			8/1			12/2	
		100			3/0			6/1			7/1
	CCT=15	60	12/1			15/1			9/2		
		80		7/0			8/1			12/2	
		100			3/0			6/1			7/1
	CCT=30	60	14/7			17/7			12/8		
		80		8/1			10/2			13/3	
		100			3/1			6/2			7/2

(단위) m: 회, ACT:액션컷오프타임 & CCT:컨디션컷오프타임; 분, p: %

로 설정하고 컨디션컷오프 타임을 5분으로 설정하며 연관성임계값을 60%로 설정한 후 CASPER를 동작시키면 13개의 적합한 인과관계를 추출하고 2개의 부적합한 인과관계를 추출한다. 표 2에서 볼 수 있듯이 연관성임계값은 일반적으로 80%와 100%일 때 더 나은 결과를 보였다. 액션컷오프타임을 같은 조건에서 값이 5분일 때 더 좋은 결과를 보였다. 컨디션컷오프타임을 5분일 때와 15분일 때 비슷한 결과를 보였지만 30분일 때는 부적합한 인과관계가 많이 추출되었다. 이것은 분할 범위를 지나치게 넓히면 관련성이 상대적으로 낮은 이벤트들이 함께 포함됨으로써 결과를 왜곡시키는 것으로 보인다. CASPER의 성능은 파라미터에 따라 변화되지만 일반적으로 연관성임계값은 높게, 액션컷오프타임과 컨디션컷오프타임을 짧게 설정함으로써 우연한 결과를 배제하는 것이 정확한 결과를 얻는데 유리하다고 볼 수 있다.

5. 결론 및 향후연구

본 논문에서는 생활데이터를 이용하여 개인화된 상황인식서비스를 자동으로 발견할 수 있는 방법인 CASPER를 소개하였다. CASPER는 사용자의 생활 데이터를 분석하여 인과관계가 있는 이벤트들을 상황인식서비스로 추출한다. CASPER를 이용하여 자동으로 개인화된 서비스를 생성하여 사용자에게 제공할 수 있는 모델과 동작 절차에 대하여 설명하였고, 상황인식서비스 생성을 담당하는 CASPER 알고리즘에 중점을 두고 상세히 설명하였다. 또한, 알고리즘의 유효성을 보이기 위하여 실제 생활데이터를 수집하고 알고리즘을 적용 분석한 실험에 대하여 상세히 설명하였다.

실험결과에 의하면 CASPER는 348개 이벤트로 구성된 사용자의 생활 데이터로부터 28개의 서비스를 추출

하였는데 실험에 참여했던 피실험자 자신의 의견으로는 이 중에는 유용한 서비스 20개와 불필요한 서비스 8개가 있는 것으로 나타났다. 본 연구진이 그 내용을 자세히 검토한 결과, 피실험자가 유용한 것으로 판단한 서비스의 30%(6개)는 다른 사람들에게도 유용할 것으로 보이는 서비스였고 나머지 서비스는 "7시에 대문이 열리면 목욕물을 데운다."와 같이 사용자에 따라 그 유용성이 달라질 수 있는 것들이었다. 실험데이터 수집의 어려움 때문에 제한된 데이터를 가지고 수행한 실험이기는 하지만 이 결과로부터 CASPER 알고리즘은 실험에 사용된 피실험자의 생활데이터로부터 개인화된 상황인식서비스를 생성해주는 효과가 있음을 알 수 있었다. 우리는 이러한 알고리즘이 다른 사용자들의 생활데이터에도 적용이 가능할 것으로 본다. 본 연구 결과는 상황인식서비스 추천시스템 등의 방법으로 개인화된 상황인식서비스를 생성하는데 이용될 수 있을 것으로 기대한다.

현재의 CASPER는 서비스도메인에 한 사람이 생활하고 있다고 가정하여 연구된 것이다. 향후에는 여러 가족 구성원이 생활하는 환경에서도 개인화된 서비스를 생성할 수 있도록 확장할 계획이다.

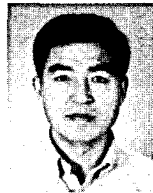
참 고 문 헌

- [1] Luiz Olavo, Bonino da Silva, Remco Poortinga, van Wijnen, and Peter Vink, "A Service-Oriented Middleware for Context-Aware Applications," *5th international workshop on Middleware for pervasive and ad-hoc computing*, pp.37-42, 2007.
- [2] Tao Gu, Hung Keng Pung, and Da Qing Zhang "A service-oriented middleware for building context-aware services," *In Journal of Network and Computer Applications*, vol.28, Issue 1, Jan. 2005.
- [3] Luiz Olavo Bonino da Silva Santos, Fano Ramparany, Patricia Dockhorn Costa, Peter Vink, Richard Etter, and Tom Broens, "A Service Architecture for Context Awareness and Reaction Provisioning," *2007 IEEE Congress on Services*, pp.25-32, 9-13, July, 2007.
- [4] Tom Rodden, Andy Crabtree, Terry Hemmings, Boriana Koleva, Jan Humble, Karl-Petter AKESSON, and Par HANSSON, "Configuring the Ubiquitous Home," *The 6th International Conference on Designing Cooperative Systems*, pp.11-14, May, 2004.
- [5] Anind Dey, Raffay Hamid, Chris Beckmann, Ian Li, Daniel Hsu, "a CAPpella: programming by demonstration of context-aware applications," *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 2004.
- [6] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo, "Discovery of Frequent Episodes in Event Sequences," *Data Mining and Knowledge Discovery*, vol.1, Issue 3, pp.259-289, 1997.
- [7] Lam, K.P. Höyneck, M. Dong, B., Andrews, B. Chiou, Y.S. Zhang, R. and Benitez, D., "Occupancy Detection through An Extensive Environmental Sensor Network in an Open-plan Office Building," *IBPSA Conference*, Glasgow, U.K. 2009.
- [8] Thi V. Duong, Dinh Q. Phung, Hung H. Bui, Svetha Venkatesh, "Human Behavior Recognition with Generic Exponential Family Duration Modeling in the Hidden Semi-Markov Model," *The 18th International Conference on Pattern Recognition (ICPR'06)*, IEEE, 2006.
- [9] Bing Dong and Burton Andrews, "Sensor-based Occupancy Behavioral Pattern Recognition for Energy and Comfort Management in Intelligent Buildings," *The 11th International IBPSA Conference*, pp.1444-1451, July, 2009.
- [10] Sajal K. Das, Diane J. Cook, Amiya Bhattacharya, Edwin O. Heierman III and Tze-Yun Lin, "The Role of Prediction Algorithms in the MavHome Smart Home Architecture," *IEEE Wireless Communications*, vol.9, Issue 6, pp.77-84, Dec. 2002.
- [11] N. T. Nguyen, S. Venkatesh, G. West, and H. H. Bui. Learning, "People movement model from multiple cameras for behaviour recognition," *In Joint IAPR International Workshops on Structural and Syntactical Pattern Recognition and Statistical Techniques in Pattern Recognition*, pp.315-324, Aug. 2004.



박 정 규

2004년 한국항공대학교 컴퓨터공학 학사
2006년 한국항공대학교 컴퓨터공학 석사
2006년~현재 한국항공대학교 컴퓨터공학과 박사과정. 관심분야는 상황인식서비스, HCI, 서비스지향아키텍처, 모바일컴퓨팅



이 공 해

1980년 서강대학교 전자공학과 학사. 1986년 Virginia Tech 전산학과 석사. 1990년 Virginia Tech 전산학과 박사. 1990년~1992년 IBM Development Staff Member. 1992년~1993년 한국 IBM 연구원. 1999년~2000년 IBM Almaden 방문과학자. 1993년~현재 한국항공대학교 컴퓨터공학과 교수. 관심분야는 소프트웨어공학, 객체지향시스템, 모바일컴퓨팅