

# 유저(user)를 대신한 과제 수행과, 소프트웨어 에이전트

## 사용자들에게 자신들이 소유한 불완전한 소프트웨어 에이전트에게 해야 할 일과 위치를 가르치게 하라



소프트웨어 에이전트들은 사용자를 대신해서 특정한 자율적 과제들을 수행하도록 되어있다. 그러나 많은 경우 에이전트의 능력이 원하는 결과를 산출하기에 역부족일 수 있다. 그렇다고 해서 그저 포기하고 과제 전체를 사용자에게 떠맡기기보다는 에이전트 스스로 문제의 원인을 찾아내어 그것을 도움을 줄 수 있고 또 기꺼이 그렇게 할 다른 에이전트나 사용자에게 전하여 그 결과를 원래 목적을 이루는데 이용하는 것이 훨씬 더 나은 대안이 될 것이다.

사용자가 수시로 개입해야 하는 경우일지라도 그렇지 않고도 자동 에이전트가 문제에 봉착할 수 있다면 대부분의 사용자들이 유쾌히 그리 할 것이다. 예를 통한 프로그래밍이라고도 알려져 있는 증명을 통한 프로그래밍(programming by demonstration, PBD)의 패러다임은 이러한 상황들에서 요구되는 대화의 종류에 대한 틀을 제공한다. 사용자와 에이전트는 각자의 능력을 공유하여 현재의 문제를 해결할 뿐만 아니라 에이전트의 기술을 확장시키게 된다. 사용자가 개입하여 에이전트에게 새로운 기술들을 가르치면 미래에 발생할 수 있는 유사한 문제들과 추가적인 훈련을 피할 수 있다.

1999년 독일 교육 과학 연구 과학기술부가 기금을 제공하는 네트 프로젝트에 대한 계획 보조의 일환으로 공개된 훈련가능한 정보 어시스턴트(Trainable Information Assistant, TRIA)라 불리는 우리의 시스템은 웹 사이트들로부터의 정보 수집을 위한 스크립트들을 생성하는 PBD 기술에 대한 새로운 응용안을 도입하

고 있다. 불행히도 많은 웹 사이트들은 수시로 그 외관과 구조를 변경하기 때문에 예기치 않은 상황들을 처리할 수 있을 만큼의 융통성을 갖추지 못한 에이전트들을 격분시킨다. 에이전트가 사용자에게 자신의 한계를 알리고 현 상황에서 그리고 미래의 유사한 상황들에서 어떻게 해야 하는지를 묻는 편이 유익하지 않을까? 여기서 우리는 그러한 에이전트들이 훈련 대화에서 서로 협력하여 사용자로 하여금 자신들에게 특정한 문제를 해결할 수 있는 옳은 지침을 내리도록 인도할 수 있는 방법을 보이고자 한다.

### 주의 깊은 에이전트, 책임있는 교사

웹상의 많은 서비스들은 자신들이 자원(수단)으로 활용하는 다른 웹 페이지들에 대해서는 사용자로 역할한다. 이러한 서비스에는 검색 엔진에 질문하는 메타크롤러(Metacrawler)와 같은 "메타검색 엔진"과 상인 사이트들에의 질문을 위한 엑사이트의 장고와 프리션레스 커머스의 테이테테이트(Tete-a-Tete)와 같은 쇼핑-보조 에이전트들이 포함된다. 이들 "메타에이전트"들은 사용자가 브라우저를 통해 하는 것과 똑같이 표준 HTTP 리퀘스트(청구서)를 발행하여 정보를 찾아본다. 그러면 자원 웹 사이트는 서치 히트, 상품, 가격표 등을 포함한 관련 정보를 포함하는 페이지를 보내주는데 이는 난해한 데이터베이스 포맷을 만족시키기 보다는 사람이 읽기 쉽게 포맷되어있다. 이때 에이전트는 그 페이지로부터 유용한 정보를 뽑아내 처리하여 최종 사용자에게

게 디스플레이한다.

어떤 한 사용자가 여행을 준비하고 있다고 가정해보자. 그 사람은 웹 브라우저를 사용하여 방문할 도시와 예산 한도와 같은 관련 데이터를 입력하고 나머지는 웹상의 여행 에이전트에게 맡기면 이 에이전트는 빠져있는 사항들을 채워서 여행에 대한 여러 제안들을 제공하게 된다. 에이전트는 여행이 계획되고 있는 그 시점에 웹으로부터 가져오는 정보를 사용해야 한다. 그러한 정보의 예로는 기차나 비행기 출발 시간과 호텔 객실 요금, 여행 스케줄에 들어있는 도착지들에서의 오락 행사들을 들 수 있다. 여행 에이전트의 질문에 대한 모든 대답들이 예상된 사이트들에서 실제로 발견된다면 사용자는 자신의 브라우저에서 최종 결과를 제공받는다.

관련 정보가 "그곳에 있어야 하는데" 찾을 수가 없는 경우에는 재미있는 상황이 발생한다. 일반적으로 그러한 실패의 이유는 그 사이트의 레이아웃이나 구조가 일부 변경되었기 때문이다.

오늘날의 메타검색 엔진과 쇼핑 에이전트들에서, 웹 페이지의 포맷을 지정하는 표지나 패턴의 데이터베이스의 유지를 담당하는 불쌍한 사람은 웹 사이트의 새로운 외관을 다루기 위한 또다른 절차를 정기적으로 프로그램해야 한다. 일부 보다 새로운 웹 에이전트들은 또한 시간이 지나면서 새로운 표지들을 "터득"하는 반-자동 기술을 갖추고 있다.

그러나 그들은 우리의 인간 여행자가 웹 사이트의 새로운 레이아웃에 마주쳤을 때 바로 도움을 제공할 수가 없다. 알맞은 대답(다른 방법으로는 에이전트가 작성하는 것이 불가능할 수 있는 여행 제안)에 대한 교환으로 사용자가 특정 시스템을 유지하는 것(에이전트의 지식 기반을 적절한 방법으로 갱신함으로써)은 어떠한가? 다른 사용자들이 이와 동일한 일을 하고 있으며 그룹으로 인해 전체적인 시스템 성능이 개선되고 있다.

훈련 대화를 안내(인도)하기 위해, 사용자의 브라우저에 문서가 열리는데 여기서 사용자는 추출될 텍스트나 그림들의 관련부분을 표시하여 어떻게 해서든지 특정 정보를 확인할 수 있는 방법에 대해 시스템에 힌트를 제공할 수 있다. 결국 새로운 정보-추출 절차가 종합되어 미래의 사용을 위해 데이터베이스에 삽입된다.

웹 질문 언어 HyQL이 정보-추출 절차(2)를 프로그래밍하기 위한 목표 언어로서 사용된다. HyQL은 웹 전체를 항해하는 것

과 추상적인 수준에서 관련 문서 부분들의 특징 부여를 허용하므로 이렇게 되면 많은 문서 수정들이 이들 절차들에 영향을 미치지 않게 된다.

에이전트는 가장 확실한 표적지로 인도할 것을 약속하는 조치들을 제안하고자 한다. 각 단계에서 사용자는 에이전트의 제안들을 무효화할 수 있는 옵션을 제공받는다. 에이전트는 정확한 조치나 힌트(문서의 구조에 따라)를 제안해야 할 책임이 있으며 반면 사용자는 관련 데이터만을 지적할 것으로 기대된다. 에이전트는 표지 디자인 면에서 사용자의 기술 수준을 고려하여, 자신이 최근 사용자를 얼마나 자주 성가시게 했는지를 추적하여 매번 조치를 취해야 할 기회가 생길 때마다 술선해서 수행하는 것이 적합한지 여부를 결정하게 된다.

사용자가 수행할 수도 있는 조치들에는 다음과 같은 것들이 포함된다:

한 HTML 문서의 일부들 선택. 에이전트에게 이 부분들이 관련이 있음을 가르친다.

개념 속성 선택. 후에 에이전트가 표지를 저장할 명칭을 설정한다.

선택된 것의 추가 특징 부여. 관련 특징들을 지적한다. 사용자는 예를 들어 선택된 HTML 부분(에이전트가 제안한)을 특정 스타일이나 포맷을 가진 텍스트로 특성을 부여함으로써 그것의 관련 특성을 지적할 수 있다. 표지 구성을 구조 정보에 덜 종속되게 만든다.

항해 보조도구 규정. 구조적 특징들을 설명한다. 표지가 선택된 것의 특징으로부터 얻게 되는 정보와 함께 문서의 구조적 특징들 또한 설명한다면 표지는 보다 확실할 수도 있다. 예를 들어 선택된 것이 표의 일부라면 "표 환경"이 선택된 텍스트를 둘러싼 주변환경(문맥)의 일부가 된다. 이러한 "주변환경(문맥)"과는 별도로 에이전트는 항해 경계표로 기능하는 그래픽이나 볼드체 표제와 같은 문서의 중요한 특징들을 고려한다.

대화 종료. 현재의 표지를 사용한다. 각 컴퓨터 작동 단계에서 표지는 지금까지 사용자로부터 입수된 정보로 구성된다.

대화 재개. 사용자가 스크래치로부터 표지 구성을 시작할 수 있게 한다.

자동 모드로 전환. 그 이상의 표지 구성은 전적으로 에이전트에게 맡긴다.

이들 조치 각각에서 사용자는 주어진 상황에서 허용되는 허부

조치들만을 보여주는 하부메뉴에 의해 인도된다. 에이전트에 대한 최소한의 입력 자료는 사용자가 에이전트로하여금 표지를 경유하여 웹 페이지로부터 뽑아내기를 원하는 텍스트 부분이다. 에이전트가 표지 구성을 하는 것을 돕느라 시간을 허비하길 원치 않는 사용자들은 추가 정보없이 에이전트에게 표지를 생성하라고 명령할 수도 있다. 그 결과 발생하는 것은 (최소한)주어진 예에 대해 유효한 표지이다.

### 에이전트의 교육

에이전트로 하여금 관련 정보를 가지고 오게 하는 방법에 대한 예로서, 에이전트가 온라인 콘서트 목록에서 한 예술가의 이름을 뽑아내도록 가르칠 수 있는 방법을 생각해보자. 에이전트가 사용자의 여행을 계획하는 일부분으로 특정 도시의 문화 행사들에 대한 목록을 준비하고 있으며 콘서트 정보 목록으로부터 한 예술가의 이름을 뽑아내야 한다고 가정하자. 사용자는 그 목록으로부터 "웨인 한콕"이라는 예술가 이름을 뽑아내야 할 수도 있다(그림1 참조). 그것이 적색으로 되어있든지, 모두 대문자라든지, 그림의 우측에 있든지 하는 식으로 웹 페이지내에서 이 텍스트의 특징을 부여하는 데에는 많은 방법들이 존재한다. 웨인 한콕이라는 예를 증명한 후 사용자는 생성된 패턴이 다음 목록으로부터 "도브테일 조인트"라는 음악가를 뽑아낼 수 있기를 기대할 것이다.

그림에서 대화의 메인 창은 에이전트가 5개 등급의 조치들 즉, 선택된 것에 특성을 부여하고, 경계표를 제안하고, 정황(문맥)을

제안하고, 대화를 종료하고, 상황작용을 개시하는 등의 조치들을 제안하고 있는 것을 보여준다. 이 경우에 권고되는 조치는 시스템으로하여금 선택된 것에 특성을 부여하도록 하는 것이다.

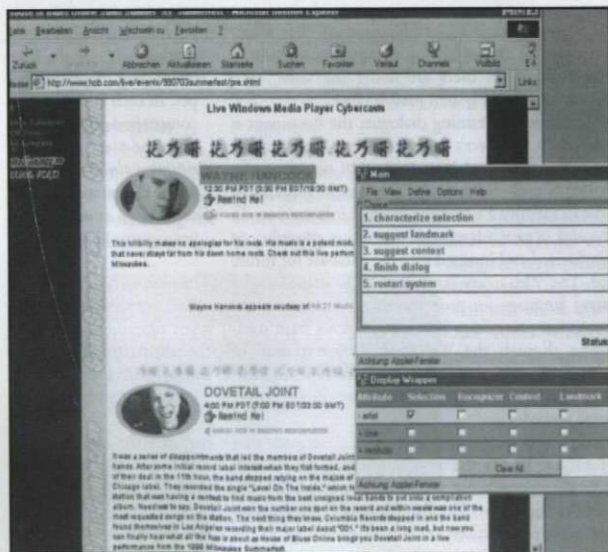
사용자는 시스템으로 하여금 몇가지 경계표를 제안하도록 하여 그로부터 선택할 수도 있다. 사용자는 또한 시스템의 제안들을 무시하고 어떤 한 경계표를 규정할 수도 있다. 이때 시스템은 문맥에 대한 정의가 무효한 것으로 판단하여 해당 사용자 옵션만이 대화에 표시될 수도 있다.

디스플레이 창에는 예술가와 같은 해당 개념들의 관점에서 이전에 정의된 표지들에 대한 개관이 제공되며 각각의 선택된 것, 문맥, HTML 문서에서의 경계표가 표시된다.

### 대화 및 표지 구성

대화 및 표지 구성에는 웹 페이지의 사소한 변경에는 영향을 받지 않은 상태로 남아있는 표지 구축 블록들로부터의 HyQL 스크립트 생성이 포함된다. 특정 표지가 얼마나 확고한지에 대한 평가치를 반영하는 득점 기능이 후보 표지들 각각에 적용된다. 에이전트는 정해진 한계치 이하의 점수를 얻은 표지들은 무시한다. 표지 종류들의 계급이 그 득점에 기여한다. 각각의 종류는 그의 일반적 효용을 반영하는 구체적인 평가치를 가진다. 각각의 종류는 특정 표지에 대해 채워질 파라미터들을 가지는 표지 구축 블럭들을 포함하는 일종의 형판(template)이다. 예를 들어 표제와 같은 어떤 추가적인 구조 정보를 사용하는 표지는 정확한 HTML 스트링에 대한 문서를 스캔하는 것보다 일반적으로 더 확고하다.

코스트 기능(비용 기능)은 각각의 표지 선택에 이르기까지 문서 전체를 향해하는데 필요한 계산 시간을 평가한다. 에이전트는 또한 표지의 위치 측정 및 선택 평가치 모두가 수정되어야 할 때 문서의 주요 특징들을 검색하여 평가치를 변경한다. 문맥과 경계표 또한 고려되어야 한다. 이들은 사용자와 논의하는 여러 단계들을 피하기 위해 대화없이 표지 선택물로서 특성이 부여되고 평가된다. 사용자가 규정한 부분들을 포함하는 표지는 사용자의 전문 기술에 비추어 보다 높은 점수를 받아야 한다. 이 점수는 사용자가 전체적인 지배권을 가지며 에이전트의 제안들을 수용하지 않는 데에는 그만한 이유 - 사용자가 표지-설계 전문가로 간주될 경우에는 보다 큰 가치 - 가 있으리라는 에이전트 설계자의 믿음을 반영한다.



▲ (그림1) 온라인 콘서트 목록에서 예술가의 이름 구하기

에이전트는 경계표, 문맥, 힌트, 선택 특성부여들에 대한 모든 타당한 조합들을 고려해야 한다. 각각의 후보 표지들은 다음의 기준에 비추어 평가된다:

- 선택물 특성부여의 가치 및 규정된 다른 부분들에 대한 그의 위치측정 비용
- 힌트의 가치 및 위치 측정 비용 및 선택물 평가치에 대한 그에 상응한 변경
- 문맥 및 경계표의 가치 및 결과적 위치 측정
- 사용자의 전문 기술에 비추어 사용자에게 의한 수락

그림에서 보여진 시나리오에서, 에이전트는 선택물 특성부여, 힌트, 문맥, 경계표들에 대한 모든 조합을 계산한다. 에이전트는 선택물을, 경계표나 문맥없이 문서 제목부터 시작하여 그것이 그러한 환경에서 10번째 발생된 것일 때 위치측정 비용 10인 볼드체 텍스트 환경으로 HTML에서 표시된 텍스트로 특성 부여한다. 두 번째 표지는 그것이 그 주변의 텍스트와는 달리 적색이며 특별한 활자 크기를 가졌는지를 포함하여 기타 다른 텍스트 특성들을 고찰한다. 그것이 문서에서 그러한 환경으로는 첫 번째 것일 경우 위치측정 비용은 1로 감소하며 선택 특성부여 평가치는 첫 번째 표지에 비례하여 증가한다. 문맥의 정의는 에이전트의 발견적 교수법(heuristics)에 따라 표지 평가를 개선시킬 수 없으므로 해당 사용자 조치만이 가능하다는 점에 유념해야 한다.

일련의 표지 평가치들을 참조하여 에이전트는 사용자에게 어떤 조치들을 제안할 것인지를 결정해야 한다. 현재 선호되는 표지의 평가치 - 처음에는 0 -부터 시작하여 A 조치의 등급이 여러 내재된 평가단계들을 이용하여 계산된다:

- A 조치를 통해 도달된 에이전트의 최상의 표지의 표지 평가치;
- 조치의 예상 용도, 이때에 사용자가 도움이 될 수 있는 힌트들을 제공하는 기술이 고려되어진다;
- 시스템이 사용자로부터의 입력을 요청하는 빈도를 측정하는 "어노이언스 요소(annoyance factor)"

어떤 한 조치의 등급을 매기는데 이러한 단계들을 사용하는 것은 제안된 조치 A가 (현재의) 최상의 평가에 추가되어야 함을 의미하며 에이전트 또한 그 조치를 수행하는데 있어서 개개의 사용자가 겪는 어려움을 고려해야 한다. 그

러나 에이전트가 정보 요청으로 특정 횟수만큼 사용자를 성가시게 했음을 인지하는 경우 사용자에게 의해 좌우되는 어노이언스

레벨에 다달을 수도 있다. 그러므로 추가적인 정의를 필요로 하는 조치들은 보다 낮은 등급을 얻게되어 에이전트는 이용가능한 최상의 표지를 취하거나 피드백없이 자율적으로 표지를 구성하는 자동모드로 전환할 것을 제안하게 된다.

온라인 여행-계획의 예로 돌아가서 에이전트는 대문자 적색 텍스트에 대한 서치를 힌트로 그리고 텍스트 앞의 영상을 경계표로 통합하고 있는 표지에 최고 평가치를 산출한다.

힌트만을 포함하고 있는 표지가 경계표만을 가지고 있는 표지보다 높은 점수를 얻으므로 "선택물 특성부여" 조치는 최고 순위에 도달한다. 에이전트의 목적은 힌트와 경계표를 모두 가지고 있는 표지에 높은 평가치를 줄 수 있도록 사용자를 "경계표 제안" 조치로 인도하는 것임을 주목하라. 그러므로 각각의 제안된 조치들에는 에이전트가 의도하는 후속 조치들이 포함될 수 있다. 시스템은 문맥을 제안하지 않는데 이는 등급이 너무 낮기 때문이나 사용자에게는 그렇게 할 수 있는 기회가 제공된다.

각각의 사용자 조치후, 에이전트는 지금까지 수집된 정보를 통합하는 새로운 등급을 계산한다. 자체의 품질 기준에 따라 최상의 표지 평가치에 도달하면 에이전트는 사용자에게 이 표지를 수용하고 대화를 종료할 것을 제안한다. 에이전트는 표지 형판과 수집된 정보를 이용하여 지정된 HTML 부분을 추출하기 위해 구체적인 HyQL 스크립트를 구성하며 이를 TRIA의 정보 브로커 모듈에 전달한다.

이렇게 하여 PBD 에이전트는 시스템 제안과 그리고 시스템 안내 및 사용자 주도의 그 어떤 조합도 무시한 철저한 사용자 주도의 표지 생성뿐 아니라 표지의 통제된 구성을 가능케 한다.

## 터득된 교훈들

아직 이들 대화 전략들을 엄밀하게 평가하진 못했지만 우리는 TRIAs PBD 환경[2]의 최초의 형태로부터 몇가지 일반적인 교훈을 얻어냈다. 이 최초의 형태는 미래 사용자 조치들에 대한 등급이 매겨진 제안들을 포함하고 있지는 않았으나 모든 결정을 전적으로 사용자에게 맡기는 단순한 그래픽 인터페이스를 포함하고 있었다.

시스템과의 상호작용에서 우리는 다음과 같은 두가지 주목할 만한 질문들에 마주쳤다:

다음에 해야 할 일은 무엇인가? 이를 판단하기 위해 사용자는

학습 에이전트가 그들의 조치로부터 얻게 될 혜택들을 알고 있어야 한다. 대부분의 PBD 시스템들은 내부 처리과정들이 사용자로부터 정교하게 숨겨져 있으므로 사용자가 다음번에 어떤 단계들을 취해야 하는지에 대해 정보에 입각한 판단을 내리는 것이 일반적으로 불가능하다.

중단해도 될까? 어쨌든 사용자는 훈련 과정을 계속해야 한다. 결국 학습 에이전트가 과제에 알맞는 해답을 생성할 수 있을 만큼 충분한 정보를 이미 가지고 있는 경우 그 에이전트에게 계속적으로 추가 정보를 제공하는 것은 무의미하다. 이를 판단하기 위해 사용자는 에이전트의 교육 결과에 대한 추가 조치의 잠재적 효과를 따져보아야 한다.

사용자들이 정보에 입각한 판단을 내릴 수 있을 만큼 충분한 배경 지식을 가지고 있는 경우라면 전적으로 그들 재량에 맞는 것이 유익하다. 우리는 너무 많은 기술적인 세부항목들을 가지고 사용자들을 성가시게 하는 것을 원치 않았기 때문에 훈련 대화는 이러한 질문들을 다음과 같은 세가지 방식으로 제기한다:

- 다음 번에 수행될 가능성이 있는 최적의 조치 제안;
- 생성될 표지의 품질이 어떤 조치에 의해 개선될 가능성이 없는 경우 훈련 과정 종료 제안
- 하부대화들을 직선화하거나 임시 순서가 지정되지 않을 경우에도 특정한 순서로 해당 조치들의 강제 실행

사용자들이 전혀 다른 어떤 것을 실시해야 할 필요가 있다고 느낄 경우에는 언제나 모든 시스템 제안들을 무시할 수 있다는 점 또한 유념해야 한다.

## 기타 응용 시나리오

TRIA 응용 시나리오의 한가지 특색은 사용자와 시스템간의 서비스 제공자와 소비자 역할 교환이다. 우리는 학습 에이전트의 상태에만 전적으로 집중하는 대신 사용자의 "적절한 표현(felicity)" (당장의 문제와 관련있는 정보를 학습할 수 있는 준비성에 대한 커트 반레헨의 용어)을 고려에 넣어야 했다.


그러나 어쩌면 마음내키지 않는 사용자가 훈련 과정에 참여하게 되는 응용이상의 것들이 훈련 가능한 성분들의 철저한 설계로부터 이익을 얻을 수 있다.

우리는 InfoBeans 시스템(1999년에 발매됨)을 실행하는데 TRIA에서의 동일한 PBD 방식을 사용하였는데 이 시스템에서는 경험없는 사용자들도 자신들의 개인적 정보 필요를 충족시키기

위해 자신만의 웹에 근거한 정보 서비스를 구성한다[3]. InfoBeans 응용은 사용자가 자율적으로 처리되는 특정 유형의 상황들을 인식하도록 시스템을 훈련시키는 PBD-시스템 방식만을 고집하기 때문에 우리는 일부 시스템 제안의 예상 효용을 평가할 때의 "어노이언스" 요소를 제거하였다[4]. 이들 형태들중 어떤 것을 사용자들이 더 많이 받아들일 것인가는 앞으로의 실험과 경험을 통해 드러날 것이다.

## 결론

우리는 여기서 초기의 에이전트 훈련 단계뿐만 아니라 에이전트가 그렇게 습득한 절차들을 실행하는 동안에도 PBD 기법들을 사용하는 사례를 들고자 하였다. 이 시나리오는 시스템으로하여금 기존의 PBD 시나리오에서 보다 훨씬 더 세심하게 사용자들을 고려할 것을 필요로 한다. 결국, 사용자는 어떤 유용한 서비스를 해 줄 것으로 기대되는 에이전트들의 불완전한 행위를 수정할 수 있어야 한다.

프로그래머가 아닌 사용자들을 대상으로한 초기의 비공식 테스트들은 우리가 약속한 훈련 메카니즘이 그들 중 많은 이들로 하여금 웹상의 소스들로부터 정보를 추출하는 것과 관련된 문제들을 확인하는 과정들에 내재해 있는 어려움들을 해결할 수 있도록 해주는 것으로 나타나고 있다. 우리가 여기서 약속한 응용 시나리오 외에도, 인텔리전트 통지 서비스에서부터 데이터 저장소에 이르기까지 우리가 "교육가능한" 정보 에이전트라 칭하는 것의 많은 다양한 용도를 상상할 수 있다. 

■ **마씨아스 바우어(bauer@dfki.de)**는 독일 자르부르크에 위치한 독일 인공지능 연구 센터의 인텔리전트 사용자 인터페이스 연구소의 수석 연구원임.

■ **디에트마 덴글러(dengler@dfki.de)**는 독일 자르부르크에 위치한 독일 인공지능 연구 센터의 인텔리전트 사용자 인터페이스 연구소의 수석 연구원임.

■ **가브리엘 폴(gpaul@dfki.de)**은 독일 자르부르크에 위치한 독일 인공지능 연구 센터의 인텔리전트 사용자 인터페이스 연구소의 수석 소프트웨어 공학자임.

■ **마쿠스 메이어(meyer@dfki.de)**는 독일 자르부르크에 위치한 독일 인공지능 연구 센터의 인텔리전트 사용자 인터페이스 연구소의 연구원임.