

# 절차적 정보검색을 위한 서비스팩 개념 도출과 기반 프레임워크 설계

정의석<sup>0</sup> 임수중 왕지현 임명은 윤보현  
언어이해연구팀, 언어공학연구부 한국전자통신연구원  
(teschung, isj, jhwang, melim, vbh)@etri.re.kr

## Deriving Concept of ServicePack and Design of Framework for Procedural Information Retrieval

Euisok Chung<sup>0</sup> Soojong Lim JiHyun Wang Myung-Eun Lim Bo-Hyun Yun  
Dept. of Language Engineering ETRI

### 요 약

기존 인터넷 정보 검색 환경에서 사용자는 특정 목적을 달성하기 위해 비연속적이며 다단계의 정보 검색 과정을 경험해 왔다. 본 논문은 다단계로 구성된 정보 검색 과정을 절차적 정보 검색으로 규정하고 이를 위한 서비스팩(ServicePack) 개념을 정립한다. 또한 모바일 환경에서의 정보검색, 정보 검색 경험들의 축적, 공유, 개인화, 확장 및 재사용의 개념을 서비스팩 개념에 결합한다. 그리고 서비스팩 환경 구축을 위한 프레임워크를 설계하여 제시한다.

### 1. 서론

기존 인터넷 정보 검색 환경에서 사용자는 특정 검색 목적을 달성하기 위해 비연속적이며 다단계의 정보 검색 과정을 경험해 왔다. 본 논문은 각 단계의 정보 검색 과정을 사용자의 특정 정보 검색 목적의 하위 절차라 보고 이를 절차적 정보 검색 과정이라 하겠다. 서비스팩(ServicePack)은 인터넷 정보 검색 환경에서 사용자들의 절차적 정보 검색 경험을 축적하여 공유, 확장 및 재사용을 목적으로 하는 리소스들의 묶음이라 정의한다.

본 논문은 사용자의 절차적 정보검색 과정을 위한 서비스팩 개념을 정립하고 모바일 환경에서의 정보검색, 정보 검색 경험들의 축적, 공유, 개인화, 확장 및 재사용의 개념을 서비스팩 개념에 결합한다. 그리고 서비스팩 환경 구축을 위한 프레임워크를 설계하여 제시한다.

### 2. 관련연구

본 논문은 절차적 정보 검색 처리와 검색 경험들의 축적, 공유, 개인화 확장 및 재사용을 위하여 서비스팩 개념을 제시한다. 이러한 지식 정보의 축적, 공유 및 재사용에 대한 기존연구로는 정보거래 자동 중개 시스템이라는 개념을 도입하여 부가가치성 있는 지식 정보 도출 방안에 대하여 기술한 연구가 있었고 FAQ를 이용하여 질의어와 관련된 정보를 통계적으로 요약하여 제시한다는 개념을 사용한 연구가 있었다[1][2]. 또한 SP의 저장소의 개념과 프레임워크의 개념은 [3]에서의 연구가 유사하다. 여기서는 지식표현, 정보검색 그리고 자연언어 처리 개념을 결합한 하이브리드 QA시스템을 제안하고 있다.

### 3. 서비스팩의 개념 정립

본 장은 서비스팩(이하 SP)의 개념정립을 위해 다양한 관점으로 SP의 특징들을 기술한다. Vertical View는 사용자 입장에서 SP의 확장과 재사용성에 대하여 기술하고 Horizontal View에서는 커뮤니티의 관점에서 SP의 확장, 재사용, 공유 및 기존 IR환경과의 차이점에 대하여 기술한다. 마지막으로 Lifecycle View에서는 SP자체의 관점에서 SP의 확장 및 재사용 개념을 설명한다.

#### ● Vertical View

기존의 인터넷 정보검색 환경은 정보 검색 과정에 세션이라는 개념이 들어가 있지 않다. 사용자는 특정 검색 결과에 대해 일정 시간이 경과 후 관련 세부 질의나 추가적

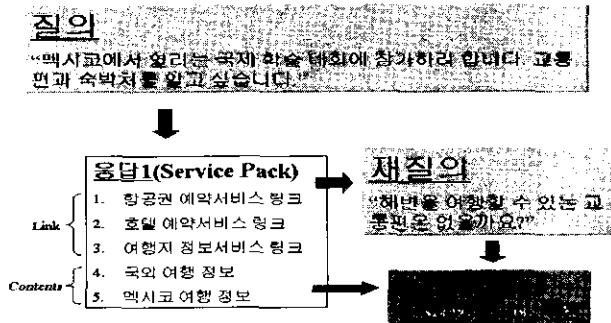


그림1. 서비스팩 기본 개념

서비스팩의 기본 개념은 특정 질의에 대해 밀접히 관련된 정보의 묶음이라 하겠다. 사용자는 서비스팩을 기반으로 재질의를 할 수 있으며 서비스팩은 다시 확장을 하게 된다. 그림1은 이러한 질의와 재질의 과정을 통한 서비스팩의 확장개념을 보여준다. 응답1의 항목들은 질의와 밀접히 관련된 정보들의 묶음으로 여행에 필수적인 서비스들과 관련 지식들의 콘텐츠로 구성된다.

질의를 할 수 있다고 본다. 모바일 환경을 가정할 경우 그림 2와 같이 IR서버로부터 SP결과를 얻고 일정 시간이 지난 후 유지된 세션내에서 재질의를 한다면 추가되는 정보는 기존 세션을 기반으로 필터링되고 또한 기존의 SP는 확장될 수 있는 것이다. 이는 시간의 흐름에 따른 사용자의 경험을 SP로 모델링 할 수 있다고도 볼 수 있다. 만일 유사한 경로의 정보 검색 과정을 거칠 다른 사용자가 존재할 경우 최종 완성된 SP를 다시 이용할 수 있을 것이다. 본 논문은 특정 정보검색 세션의 절차적 검색을 하나의 트랜잭션(transaction)으로 본다. 그림 2는 SP의 확장과 재사용을 설명하고 있다.

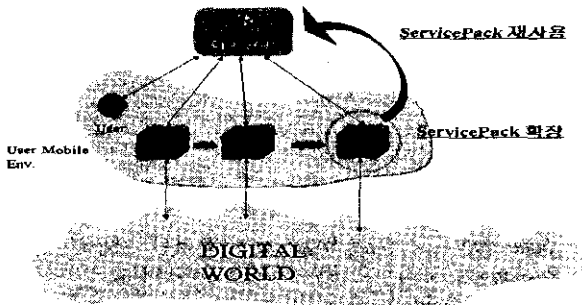


그림2. SP의 개인적 관점

● Horizontal View

기존의 인터넷 정보 검색 환경의 정보 검색 결과는 일회적이며 그 검색 결과가 직접적으로 유사한 검색 질의 사용자에게 영향을 미치지 못하고 있다. 물론 Askgeevies 등의 검색 서비스는 특정 질의 자체에 대한 검색 결과를 학습하여 유사한 질의에 대한 검색 결과로 제시해주는 한다. 본 논문에서 제시하는 SP는 이러한 개념을 보다 확장하여 여러 사용자에게 의해 검색 결과가 공유, 확장될 수 있으며 사용자간 SP의 전달이 가능한 P2P(personal to personal)개념을 도입하고 있다. 이는 SP가 PDA와 같은 모바일 환경에서의 지식 정보 매개체로서의 역할이 가능하다는 것을 말한다. 그림3은 커뮤니티 관점에서 기존 IR 환경과 SP-IR환경의 차이점을 보여주고 있다.

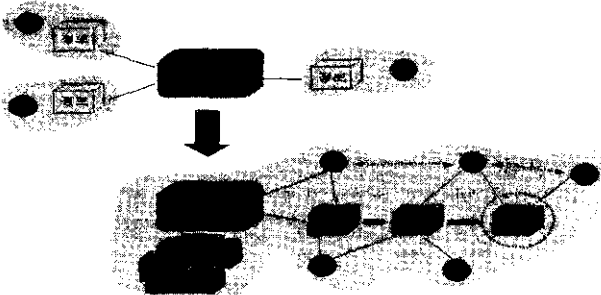


그림3. SP의 커뮤니티 관점

● Lifecycle View

SP의 Lifecycle이란 SP의 생성, 변화, 저장 과정을 말하며 그림4에 나타나 있다. 여기서 지금까지 말해왔던 사용자의 절차적 검색 경험의 집합체는 SP Object(SP)이다. SP Repository란 서버상에 존재하는 저장소로서 특정 주제별로 분류되어 저장되어 있다고 본다. 여기서 SP는 내부적으로 리소스를 포함하고 있다. 따라서 SP Repository

는 사용자의 요구사항에 부합되는 적절한 SP를 사용자에게 제공하는 것이다. SP Instance란 사용자 세션상의 트랜잭션 과정에 유지되고 있는 SP를 지칭한다. 사용자의 트랜잭션이 종료할 경우 SP Object가 생성되어 SP Repository에 저장되는 것이다. SP Template이란 말 그대로 SP의 구조적 정보만을 말한다. 이는 SP Repository의 확장에 따라 새로운 SP Template이 가능하며 SP Template Repository에 저장되는 것이다. SP Template Repository는 SP Repository와 독립적으로 존재하여 사용자 요구에 적절한 SP가 존재하지 않을 경우 SP Template이 사용자에게 제공될 수도 있다고 본다. 또한 사용자의 질의가 일반적이어서 다수의 SPs와 SP Templates이 도출될 경우 사용자의 프로파일이나 검색 히스토리 정보를 이용하여 검색 결과를 축소할 수 있는 개인화 단계가 적용된다고 본다.

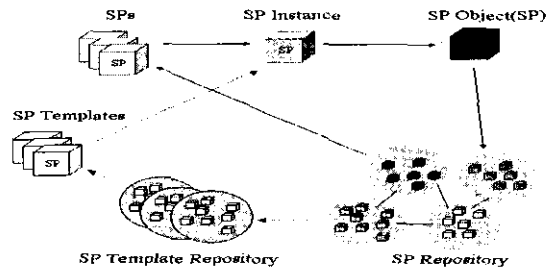


그림4. SP의 Lifecycle

지금까지 다양한 관점으로 SP에 대하여 기술하였다. 본 논문에서의 SP개념은 일반적 인터넷 환경을 대상으로 하고 있으며 자연어 질의 처리를 가정하고 있다. 그리고 SP나 SP Template에 기반한 검색 기법에 대하여서는 구체적으로 다루지는 않는다. 그러나 이에 대한 다양한 접근 방법이 있으리라 본다. SP Template을 예를 들면 결과로 제시된 SP Template은 자체적으로 구조화된 확장 질의어를 내포할 수 있으리라 본다. 그리고 해당 항목별 검색 결과는 SP Template에 채워져 SP를 생성하게 되는 것이다.

사용자 정보 검색은 특정 SP의 검색이라고도 볼 수 있을 것이다. 그러나 관련 SP나 SP Template이 Repository에 존재하지 않을 경우의 문제가 발생하게 된다. 이 경우 동적으로 SP Template을 도출할 수 있는 지식 정보가 필요하게 된다. 이에 대한 구체적인 기술은 본 논문에서는 취급하지 않는다.

4. 서비스팩 기반 프레임워크 구조 설계

SP 기반 프레임워크는 SP의 개념들을 시스템화할 수 있게 다양한 모듈로 구성된다. 기본적으로 SP의 요구자를 Client로 보고 제공자를 Server로 추상화하고 있다. Client와 Server는 ServicePack Request Broker(SRP)를 통해 SP에 대한 요구사항과 대응 결과를 주고 받게 된다. 그리고 Server측은 SP저장소인 SP Repository(SP Template도 포함), 특정 SP를 찾을 수 있는 SP Navigator, SP Repository의 재구성 및 SP Template의 도출을 위한 SP Organizer로 구성된다. Client는 사용자 트랜잭션을 관리하는 SP Transaction Manager, 검색된 SP의 적절한 Browsing과 사용자 인터랙션을 위한 SP User Interface항목으로 구성된다. 추가적으로 사용자의 개인 정보를 이용한 SP검색 결과들의 필터링 역할로 SP Personalizer와 사용자의 인터랙션을 이용한 적절한 SP선

택을 위한 SP Mediator가 프레임워크에 포함되어 있다.

현재 설계 상태의 SP와 SPT의 구조는 SP의 경우 Meta\_Info와 Link & Information Contents(LIC)로 구성되며 SPT의 경우 Meta\_Info만 갖고 있다. SP의 Meta\_Info는 SP식별을 위한 Code, SP검색을 위한 색인정보인 Index, SP의 템플릿 정보인 SPT\_Info, SP사용자 정보인 UserInfo, 그리고 SP의 확장 히스토리 정보인 Update\_History정보, 관련 SP들의 정보인 SP\_Group\_Info, SP의 저장소 정보인 SP\_Rep\_Info등으로 구성되고 SP에 대한 설명 정보인 Description정보로 SP의 메타 정보가 구성된다. SP의 LIC는 해당 SP특징에 적합한 실제 데이터 항목들로 채워지게 된다. SPT의 Meta\_Info는 SPT식별을 위한 Code정보, 검색을 위한 Index정보, 템플릿 항목 기술을 위한 SPT\_Info, 저장소 정보인 SPT\_Rep\_Info, 관련 SPT 정보인 SPT\_Group\_Info 그리고 설명 정보인 Meta\_Description정보로 구성된다.

SP 기반 프레임워크는 각 모듈별 기능 설명과 API항목들로 구성된다. 이는 다양한 시스템 환경으로 구축될 수 있으며 궁극적인 목적은 특정 SP가 이기종 및 이질적 검색 서비스 환경에 공유되기 위함이다.

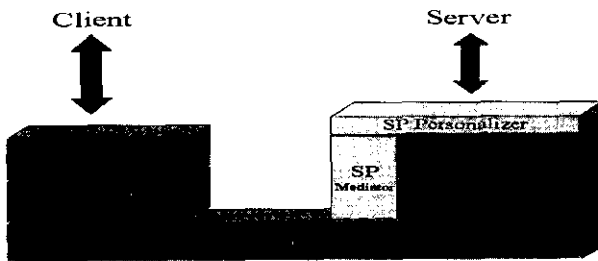


그림 5. SP Framework 개념도

● ServicePack Request Broker(SRP)

Client와 Server간의 연결 모듈로서 Request 관련된 API로는 Client의 SP요구 처리인 *Send\_Request*, Server의 요구 접수 처리인 *Get\_Request*로 구성된다. 그리고 응답 처리 API인 *Send\_Response*, *Get\_Response*를 포함한다. 그림 5는 SRP에 대한 개념적인 표현이다. 실제 구현시 정보 검색 환경에 따라 SRP자체는 서버상에만 존재할 수도 있는 것이다.

● SP Repository

전술한 바와 같이 SP Repository는 SP와 SP Template(이후 SP(T)로 지칭)을 저장하는 저장소이다. 특정SP(T)를 검색하는 *Search\_SP*, SP(T)를 추출하는 *Get\_SP*, SP(T)를 저장하는 *Put\_SP*로 API가 구성된다.

● SP Navigator

특정 SP에 대한 Naming Service를 처리하는 모듈로서 다른 Server들에 존재하는 SP(T)를 검색하기 위한 기능과 자신의 저장소에 존재하는 SP(T)를 알리기 위한 기능을 한다. 구성 API로는 사용자의 SP검색 요구사항을 처리하는 *Get\_SP\_Req*와 검색된 SP(T)를 전송하는 *Send\_SP*, 다양한 서버에 존재하는 SP(T)들의 정보를 수집하는 *Get\_SP\_Info*, 다양한 서버상의 SP Repository정보를 수집하는 *Get\_SP\_Rep\_Info*를 포함하며 자신의 Repository정보를 알리는 *Broadcast\_SP\_Rep\_Info* API로 구성된다.

● SP Organizer

SP Organizer는 SP Repository에 대한 관리 기능과 사용자 요구사항 처리를 담당하는 역할을 한다. 구성 API로는 사용자 요구사항을 처리하는 *Get\_SP\_Req*, 검색된 여러 SP(T)들을 클러스터링하는 *Make\_SP\_Cluster*, 특정 SP(T)들은 필터링하는 *SP\_Filtering*, 적절한 SP(T)의 도출 실패시 동적인 SPT생성을 담당하는 *Generate\_SPT*, 검색되거나 생성된 SP(T)의 내용을 채우는 *Pakaging*, 도출된 SP(T)를 저장하는 *Store\_SP*등이 있다.

● SP Transaction Manager

SP Transaction Manager는 SP Instance의 상태를 관리하는 Session 생성 및 유지를 담당한다. *Generate\_Transaction*과 *Manage\_SP\_Instance*의 API로 구성된다.

● SP User Interface

SP의 검색 요구 처리, 브라우징, 그리고 인터랙션은 SP User Interface 모듈에서 담당한다. 요구사항 처리를 하는 *Send\_Query*, *Send\_ReQuery*, 검색된 SP를 받는 *Get\_Response*, SP의 브라우징 및 인터랙션을 담당하는 *SP\_Browsing*, *Send\_Confirm* API로 구성된다.

지금까지 SP 프레임워크를 구성하는 대표 모듈과 각 모듈별 기능에 대하여 기술하였다. 추가적으로 SP Personalizer는 개인화 처리 기능을 담당하고 SP Mediator는 검색결과에 대한 사용자 인터랙션을 담당한다.

5. 결론

본 논문은 다단계로 구성된 정보 검색 과정을 절차적 정보 검색으로 규정하고 이를 위한 서비스팩(ServicePack) 개념을 도출하고 모바일 환경에서의 정보검색, 정보 검색 경험들의 축적, 공유, 개인화, 확장 및 재사용의 개념들을 서비스팩 개념에 결합하여 제시하였다. 또한 서비스팩 환경 구축을 위한 프레임워크를 설계하여 제시하였다. 본 논문에서의 SP개념은 일반적 인터넷 환경을 대상으로 하고 있으며 차언어 질의 처리를 가정하고 있다. 제시된 서비스팩 개념과 프레임워크는 기존의 정보 검색 관련 연구 결과들과 충분히 결합할 수 있으리라 본다. 향후 우선적으로 SRP, SP Repository, SP Organizer를 구현하여 SP 개념과 프레임워크의 타당성을 검증할 것이다.

참고 문헌

[1] 정의석 외 6명, “정보거래 자동 중개 시스템을 위한 한국어 문형 표준안”, 한글 및 한국어 정보처리 학술발표 논문집, pp.138~145, 2000  
 [2] V. Mittal and A. Berger, “Query-relevant summarization using FAQs”, Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL), Hong Kong, 2000  
 [3] E. Breck, J. Burger, D. House, M. Light and I. Mani, “Question Answering from Large Document Collections”, the AAAI Fall Symposium, North Falmouth, Massachusetts, 1999