

능동문서를 활용한 웹기반 설문 시스템 설계

장선아, 양재군, 배재학
울산대학교 컴퓨터·정보통신 공학부
e-mail:christina.jang@gmail.com, {jgyang, jhjbae}@ulsan.ac.kr

Design of a Web Interview System using Active Documents

Seon-Ah Jang, Jae-Gun Yang, Jae-Hak J. Bae
School of Computer Engineering & Information Technology, University of Ulsan

요 약

최근 고객의 만족도나 선호도와 같은 정보를 수집하기 위해서 컴퓨터를 이용한 설문조사가 다양한 형태로 이루어지고 있다. 현재의 설문시스템은 지필형 설문에 비해서 여러 가지 이점이 있지만 여전히 부족한 점이 있다. 예를 들어 로직, 변수, 구성요소 등과 같은 처리절차 또는 규칙들이 시스템에 하드 코딩되기 때문에 설문이 바뀌면 처리 엔진을 재설계하고 재구현해야 하는 점이 그것이다. 본 논문에서는 설문에 독립적인 시스템을 구축하고자 설문엔진과 로직엔진을 분리 설계하였으며, 각 설문에 설문 처리 로직을 포함시킴으로써 기존 방식의 약점을 보완하고자 하였다. 설문에 로직을 포함시키고 구조화하기 위해서 능동문서(Active Documents)를 이용하였다.

1. 서론

설문조사는 시장조사, 소비자 형태, 사용자 만족도, 광고 효과, 소비자 선호도, 시장 비용 민감도, 브랜드 인지도, 시장 규모 예측 등의 여러 가지 목적을 달성하기 위하여 행해진다[1]. 설문조사는 질문을 진행하는 회견자의 유무에 따라서 두 가지로 구분한다. 회견자가 제시한 질문에 대하여 응답자가 응답하는 방법과 회견자 없이 응답자 혼자서 응답하는 방법이 그것이다. 지필형태의 설문에서 응답에 따라 다음 질문의 순서가 결정되는 등 라우팅이 복잡한 경우, 응답자가 혼자서 응답하다 보면 설문 설계자의 의도와는 다르게 설문이 진행될 수 있으며, 질문유형(선택형, 단답형, 중복 선택형, 순위형 등)에 적합하게 응답하는지를 검사할 수 없다. 또한, 응답한 결과를 파일이나 데이터베이스에 저장하는 과정에서 키 입력 오류가 발생할 수 있는 문제가 있다[2, 3]. 이러한 단점을 보완하기 위해 컴퓨터를 이용한 설문시스템이 활용되고 있다.

설문시스템은 응답자가 편안하게 질문에 답할 수 있는 환경을 제공하여 시간, 비용, 회수율, 응답의 충실성 등[4]에서 많은 이점을 제공하였지만 여전히 부족한 점이 있다. 그 예로, 새로운 설문 진행을 위해 설문지를 작성할 경우 설문 처리 엔진에 로직, 변수, 구성요소 등과 같은 처리절차 또는 규칙들이 직접 코딩되어 설문 구현자(프로그래머)가 설문마다 처리 엔진을 재설계 및 구현해야 하는 문제점과 특정 프로그램 언어로 개발되어 있어 특정 개발자만 수정할 수 있는 문제점이 있다.

본 논문에서는 이러한 문제점을 개선하고자 능동문서(Active Document)를 이용한 웹기반 설문 시스템(WINAD: The Web Interview System with Active

Documents)을 설계하였다.

2. 설문시스템 비교

전통적인 설문형태인 지필형 설문 사례와 컴퓨터를 이용한 설문시스템의 사례를 비교하고 이를 바탕으로 설문 엔진에서 로직을 분리해야 하는 이유를 살펴보고자 한다.

· 지필형 설문 사례

(그림 1)은 영양표시제 도입과 관련하여 사람들의 인식을 조사하기 위한 지필형 설문의 일부이다. 이 설문의 일부에는 두 개의 선택형 질문과 한 개의 단답형 질문이 상호 연관되어 나타난다. 설문 설계자의 의도는, 6번 질문의 응답에 따라서 7번 응답여부를 가리는 것이다. 지필형 설문에서는 7번 응답여부를 주석으로 안내하는 수밖에 없다. 만일 어떤 응답자가 6번 질문에 대해서 4번으로 답하고도 7번 질문에 답을 한다면 해당 문항만 아니라 설문 결과 전체의 신뢰도까지 떨어진다.

알기 쉬운 영양표시 도입 관련 국민 인식 조사
:
6. 귀하의 직업은?
1. 교사 2. 영양사 3. 주부 4. 학생
7. 귀하의 최종학력은?(학생을 제외한 분만 응답)
1. 고졸 2. 초대졸 3. 대졸 4. 대학원
8. 본인을 포함한 가족은 모두 몇 명입니까?()
:

(그림 1) 지필형 설문 사례

• 컴퓨터를 이용한 설문시스템의 사례

컴퓨터를 이용한 설문시스템은 설문문의 내용을 지문, 질문 및 보기 등으로 구분해서 데이터베이스의 테이블에 저장해야 한다(그림 2), (그림 3). 또한, 설문 엔진에서 로직을 처리할 수 있도록 타입, 옵션 등의 항목을 설정해야 한다. (그림 4)는 설문시스템에서 라우팅을 처리하기 위해 하드코딩된 로직의 사례이다. 이 사례에서는 c_type(보기 유형) column과 q_option(옵션) column에 따라 응답자가 요구사항에 맞게 응답했는지를 검사한다. 질문 또는 보기를 데이터베이스에 저장하기 위한 개별적인 로직은 필요 없지만, 보기유형을 새로 추가하거나 기존의 설문 흐름을 변경하려면 설문 엔진을 매번 수정해야 한다.

q_id	p_id	q_no	q_type	c_type	q_option	questioncontent
29	1	28	2	1	4H	4. 귀하는 건강상태가 어떠합니까?
30	1	29	2	1	5H	5. 귀하는 다이어트를 하신 적이 몇 번이나 있습니까?
31	1	30	2	1	7H	6. 귀하의 직업은?
32	1	31	2	4	5H	7. 학생인 경우 구체적으로 적어주세요.
33	1	32	2	1	6H	8. 귀하의 최종학력은? (학생을 제외한 분만 응답)
34	1	33	2	1	6H	9. 본인을 포함한 가족은 모두 몇 명입니까?
35	1	34	2	1	8H	10. 현재 함께 살고 있는 가족 모두의 한달 평균 출수...

(그림 2) 질문 테이블 사례

caseid	questionid	caseno	caseoption	casecontent	pollid
1	1	1	00	매우 필요하다	1
2	1	2	00	필요한 편이다	1
3	1	3	00	별로 필요하지 않다	1
4	1	4	00	전혀 필요하지 않다	1
5	2	1	00	매우 필요하다	1
6	2	2	00	필요한 편이다	1
7	2	3	00	별로 필요하지 않다	1

(그림 3) 보기 테이블 사례

```

m_nTextNum++;
out.println("<input type=text name=textnum"
+m_nTextNum + " size=90 onChange='
setValue(" + m_nQuestionCnt + "," +
m_allQuestionList.get(j).getQuestionno() +", this.value, "+
m_allQuestionList.get(j).getCasetype() +");'>");
}
else
{
m_nTextareaNum++;
out.println("<textarea rows=4
name=textarea"+ m_nTextareaNum + " cols=90
onChange='setValue(" + m_nQuestionCnt + "," +
m_allQuestionList.get(j).getQuestionno() + ", this.value, " +
m_allQuestionList.get(j).getCasetype() + ");'>
</textarea>");
}
:
    
```

(그림 4) JSP로 표현된 로직

설문시스템은 지필형 질문에 비해 다음과 같은 장점이 있다. 응답에 따라 다음 질문을 선택해야하는 복잡한 라우팅 제공시 설문 응답자를 고려하지 않아도 되며, 질문유형에 따른 요구사항이 적합한지를 즉시 검사할 수 있다. 또한, 데이터베이스 또는 데이터 파일에 응답 결과를 오류 없이 바로 저장할 수 있다. 하지만, (그림 4)와 같이 로직을 설문엔진 내부에 하드코딩하면 새로운 설문 진행을 위해 설문마다 처리 엔진을 재설계 또는 구현해야 한다. 또한, 특정 프로그램 언어로 개발되어 있어 특정 개발자만 수정할 수 있는 단점이 있다.

이러한 단점은 컴퓨터를 이용한 설문시스템의 활용과 확장을 제한하고 있다. 따라서 유연한 설문 엔진을 구축하는 방법이 필요하다.

• 로직을 분리한 설문시스템

지필형 설문지에서 지문으로 표기하던 로직을 설문 설계자의 의도대로 동작하게 하려면 로직을 해당 질문에 포함시킬 방법이 필요하다. 이를 위해 서식, 지식베이스, 규칙 그리고, 질의로 구성되는 능동문서 모델을 이용하였다. 이 모델의 각 요소는 문서의 재사용과 상호 운용성을 위해 XML로 일관되게 표현된다. XML 기반의 능동문서는 사용자 인터페이스를 제공하는 수동적인 역할 뿐만 아니라 문서설계자 의도하는 문서처리 절차와 규칙을 기계가 읽고 추론하여 처리할 수 있게 하는 문서이다. 이를 통해 문서와 기계가 상호작용을 할 수 있으며 다른 응용 프로그램과 협력할 수 있다[5].

3. WINAD

온라인 설문을 위한 설문 작성 과정을 살펴보면 첫째, 설문 설계자가 설문조사 목적에 적합하도록 초안을 작성하고, 둘째, 설문 작성자는 작성된 초안을 바탕으로 설문 시스템을 사용해서 온라인 설문지를 작성한다. 셋째, 온라인에서 설문 조사를 실시한다(그림 5).

```

:
if(m_allQuestionList.size() != 0)
for(int j=0; j<m_allQuestionList.size(); j++)
{
%>
<br>
<table border=0 borderColor=aaaaaa cellspacing=1 cellpadding=0
width=800>
<tr>
<td width=780 align=Left><font size=2>
<%=m_allQuestionList.get(j).getQuestioncontent()%>
</font></td></tr>
<%
if(m_allQuestionList.get(j).getQuestiontype() == 3)
{
out.println("</table>");
out.println("<hr align=Left size=1 width=800>");
continue;
}
if(m_allQuestionList.get(j).getCasetype() == 4 ||
m_allQuestionList.get(j).getCasetype() == 5)
{
out.println("<tr>");
out.println("<td width=780 align=center colspan=2>");
if(m_allQuestionList.get(j).getCasetype() == 4)
{
    
```



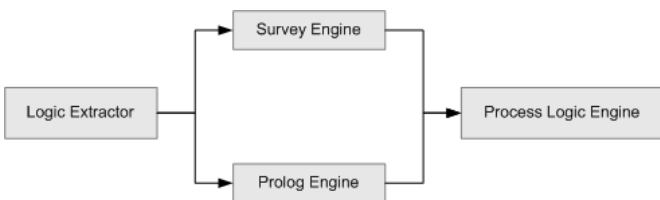
(그림 5) 설문작성 흐름도

WINAD는 설문 작성 과정중 두 번째와 세 번째를 위한 것으로, 설문 설계자가 응답자에게 제공할 설문을 작성하는 작성모드와 응답자가 선택한 질문에 응답하는 응답모드로 구성된다. 본 논문에서는 그 중에서 응답모드를 소개한다.

응답모드 시스템 설계를 위해서 다음과 같은 점을 고려하였다.

- 처리 엔진으로부터 로직을 분리하기 위해 XML을 사용한다.
- 로직 표현이 쉽도록 논리표현 언어인 Prolog를 사용한다.
- Prolog Engine은 상용화된 JPL을 사용한다.

다음 (그림 6)에서는 WINAD의 응답모드를 구성하는 모듈을 보이고 있다. 응답모드는 설문 내부의 로직을 분리하는 Logic Extractor, 설문을 출력하는 Survey Engine, 추출된 Logic을 논리표현 언어로 변환하기 위한 Prolog Engine, Logic을 처리하는 Process Logic Engine으로 구성된다.



(그림 6) 응답모드 구성 모듈

3.1 QE(Questionnaire Expression)

설문 엔진을 수정하지 않고도 설문 설계자의 의도를 설문에 반영하려면 로직을 설문 엔진에서 분리해 해당 설문문에 포함시켜야 한다. 해당설문에는 설문의 내용과 로직이 함께 표현된다. 이러한 설문지 작성을 위해 QE를 고안하였다. QE는 XML Tag를 이용하여 Content와 Logic을

정의하는 표현법으로써, Logic은 <표 1>에서 보는 것과 같이 4가지 유형으로 분류하였다.

<표 1> 설문의 Logic Type

Logic Type	설 명
routing	응답자의 응답에 따라서 분기할 다음 질문을 결정함으로써 설문의 전체 흐름을 결정
check_duplication	복수 선택형 또는 순위형 질문에서 동일한 응답을 할 수 없도록 체크
check_response	질문에 대한 응답여부를 체크
fill_text	질문에 동적으로 이전 응답 내용을 채움

(그림 7)에서는 QE 구조의 예를 보여주고 있다. 점선으로 표시된 영역이 (그림 1)의 6번 질문의 응답에 따라 7번 질문여부를 결정하는 Logic이 표현된 부분이다. name이 Q7인 radiocontrol element의 type이 routing이기 때문에 gate element내 variable element의 name속성 값에 따라 질문 Q6의 응답 결과와 비교하여 num element의 값과 동일하면 다음 질문을 가리키게 된다. 나머지 XML Tag는 Content를 표현한 부분이다.

```

<questionnaire name="survey" title="알기 쉬운 영양표시 도입
관련 국민 인식 조사">
    :
    <radiocontrol name="Q6">
        <question>귀하의 직업은?</question>
        <case value="1">1. 교사</case>
        <case value="2">2. 영양사</case>
        <case value="3">3. 주부</case>
        <case value="4">4. 학생</case>
    </radiocontrol>
    <radiocontrol name="Q7" type="routing">
        <gate>
            <operator name="ne">
                <variable name="Q6">
                    <num>4</num>
                </num>
            </gate>
            <question>귀하의 최종학력은?</question>
            <case value="1">1. 고졸</case>
            <case value="2">2. 초대졸</case>
            <case value="3">3. 대졸</case>
            <case value="4">4. 대학원</case>
        </radiocontrol>
        <textcontrol name="Q8">
            <question>본인을 포함한 가족은 모두 몇 명입니까?
            </question>
        </textcontrol>
    :
</questionnaire>
    
```

(그림 7) QE로 작성한 설문 파일

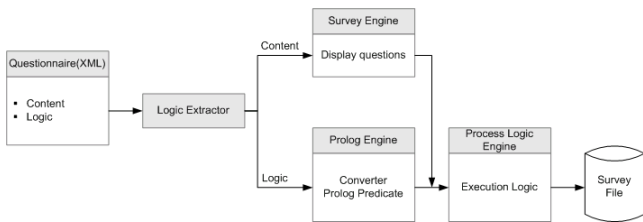
<표 2>는 QE에서 사용하는 XML Tag와 그에 대한 설명이다. Content와 Logic을 포함하는 하나의 설문을 표현하기 위해 root element는 <questionnaire>로 정의하였다. Content 영역은 질문과 보기로 나뉘며 각 질문 element는 <question>으로 각 질문의 보기 element는 <case>로 정의하였다. Logic Type에 따라 조건을 수행해야 하는 Logic element는 <gate>로 정의하였다.

<표 2> Content와 Logic XML Tag 설명

설문요소	element 명	설 명
all	questionnaire	각각의 설문을 구분하는 root element
content	question	질문 또는 지문을 나타내는 질문 element
	case	질문의 보기를 나타내는 보기 element
logic	gate	Logic Type별 동작을 위한 조건 element

3.2 응답모드의 처리 절차

설문 응답은 (그림 8)과 같이 세단계로 처리된다.



(그림 8) 응답 처리 절차

절차 1

QE로 작성된 설문 파일이 Logic Extractor 모듈로 전달된다. 설문 파일에 포함된 Logic을 Content와 분리하여 Content는 Survey Engine로 Logic은 Prolog Engine으로 전달한다.

절차 2

Survey Engine에서는 Logic Extractor로부터 전달받은 Content 데이터를 질문과 보기로 나누어 웹 브라우저에 보여준다. 응답자의 응답이 Logic에 적합한지를 체크하기 위해서는 논리표현 언어로 변환하는 작업이 필요하다. Prolog Engine은 Logic Extractor로부터 전달받은 Logic 데이터를 Prolog 규칙으로 변환한 후 Process Logic Engine으로 전달한다.

절차 3

Process Logic Engine은 전달받은 응답에 대해 Logic 처리가 필요한지를 체크한다. Logic 처리가 필요하다면 Prolog Engine으로부터 전달받은 Prolog 규칙에 따라 처리한다.

4. 결론

지필형 설문의 단점을 보완하기 위해서 개발된 컴퓨터

를 이용한 설문시스템은 응답자가 편안하게 설문에 답할 수 있는 환경을 제공한다. 이러한 설문시스템은 시간, 비용, 회수율, 응답의 충실성 등에서 많은 이점을 제공하지만 여전히 부족한 점이 있다. 새로운 설문지를 작성할 경우 설문 처리 엔진에 처리절차 또는 규칙들이 직접 코딩되어 설문 구현자(프로그래머)가 설문마다 처리 엔진을 재설계 및 구현해야 하는 문제점과 특정 프로그램 언어로 개발되어 있어 특정 개발자만 수정할 수 있는 문제점이 있다.

현재 설문시스템의 문제점을 개선하고자 각각의 설문에 로직을 개별적으로 포함시켰다. 설문 엔진은 설문에 포함된 로직을 능동적으로 수용할 수 있도록 하였으며, 설문에 로직을 포함시키기 위해 능동문서를 이용하였다. WINAD에서 설문을 작성하기 위해 XML을 이용한 표현법 QE를 고안하였다. QE로 작성된 설문지는 Logic Extractor를 통해 Content와 Logic 데이터로 분리된다. Content 데이터는 Survey Engine을 통해 웹 브라우저로 출력되며, Logic 데이터는 Prolog Engine을 통해 Prolog 규칙으로 변환된다. Process Logic Engine에서는 Prolog 규칙에 따라 로직을 처리할 수 있도록 설계하였다.

본 논문에서 설계한 시스템의 장점은 다음과 같다. 설문 구현자는 매 설문마다 설문 엔진을 변경 또는 재구현하지 않아도 되며, 전문 프로그래머가 아니라도 설문이 처리되어야 하는 로직을 표현할 수 있다. 향후 연구에서는 WINAD를 구현하여 본 논문에서 제시했던 이점에 대해 확인하고자 한다.

감사의 글

이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국 학술진흥재단의 지원을 받아 수행된 연구임. (KRF-2008-313-H00009)

참고문헌

- [1] 전찬환, 최황규, "XML을 이용한 응답자 주도형의 실시간 설문조사 시스템 구현", 한국정보과학회 2003년도 봄 학술발표논문집 제30권 제1호(B), pp.648~650, 2003. 4
- [2] Albert D. Bethke, "Representing Procedural Logic in XML", JOURNAL OF SOFTWARE, VOL. 3, NO. 2, pp.33~40, FEBRUARY 2008
- [3] Albert D. Bethke, "Using XML as a Questionnaire Specification Language", SoutheastCon, 2007. Proceedings. IEEE pp.127~131, March 2007
- [4] 천홍말, 김성훈, 변지석, "시스템 활용을 통한 효과적인 설문조사 방안", 한국경영정보학회 추계학술대회, 2002
- [5] 남철기, "지능적인 웹 어플리케이션을 위한 XML 기반의 능동문서", 박사학위논문 울산대학교 대학원 컴퓨터·정보통신공학과, 2003