

# Flex 플랫폼 상에서 사용가능한 DDX 프레임워크 설계 및 구현<sup>☆</sup>

## DDX Framework Design and Implementation Usable in the Flex Platform

김 양 훈\*      정 구 범\*\*      유 갑 상\*\*\*      김 국 보\*\*\*\*  
Yang-Hoon Kim      Gu-Beom Jeong      Gab-Sang Yoo      Guk-Boh Kim

### 요 약

오늘날 컴퓨터 사용 환경은 사용자 지향형의 RIA(Rich Internet Application)로 발전되고 있다. Flex 프레임워크는 RIA의 대표적 개발 방법으로써 메인프레임과 C/S(Client/Server) 구조의 단점을 극복하였다. 그러나 메모리 관리의 난해함, 바인딩 구조의 복잡함, 컴파일 결과물이 큰 용량을 지니고 있다는 점 등 다양한 문제점을 안고 있다.

본 논문에서는 Flex 플랫폼에서 사용자 지향의 웹 어플리케이션을 신속하고 정확하게 개발할 수 있는 프레임워크를 구현하는데 목적을 두고 있다. DDX(Dynamic Data eXchange) 프레임워크는 Flex 플랫폼에서 효율적이고 표준화된 개발 환경을 제시한다. 다양한 업무 분야에서 적용할 수 있는 확장성을 갖춘 라이브러리를 활용하여 성능 향상과 개발 생산성을 증대시켜 안정적인 시스템을 구축할 수 있도록 한다.

### ABSTRACT

Computing environment in these days aim for user-oriented development called RIA (Rich Internet Application). As a representative development method of RIA, Flex Framework overcomes the weaknesses of the Mainframe and C/S (Client/Server). However, the issues, such as, difficulties in memory management, complexity of the binding structure and large capacities of the compile outputs are left to be solved.

The purpose of this paper is to implement the framework which enables the fast and accurate development of user-oriented web application on the Flex platform. DDX (Dynamic Data eXchange) framework proposes standardized and efficient development environment in a Flex platform. And by using scalability-prepared library that is applicable for various job areas, the framework enhances the performance, increase development productivity and help construct stable system.

☞ KeyWords : Dynamic Data eXchange, Flex Framework, Rich Internet Application

## 1. 서 론

컴퓨터 사용 환경은 메인 프레임 중심의 컴퓨터 환경과 개인 PC(Personal Computer), C/S의 형식을 지나 웹의 출현 이후 웹 어플리케이션들로 그 환경이 옮겨져 왔다. 이와 함께 소프트웨어 산업은 메인프레임 중심의 메인프레임 기술, 기술 및 네트워크의 단계를 거쳐 비즈니스 지향형인 웹 2.0의 시대로 옮겨가고 있다[1].

이러한 시멘틱 웹과 웹 2.0의 시대가 진행됨에 따라 소프트웨어 산업에서는 다양한 플랫폼 환경에 적합하고 비즈니스 환경 변화에 융통성 있게

\* 정 회 원 : 대전대학교 컴퓨터공학과 박사과정  
kimyh7902@daejin.ac.kr

\*\* 종신회원 : 경북대학교 상주캠퍼스 컴퓨터공학과 교수  
jgb@knu.ac.kr

\*\*\* 정 회 원 : 동국대학교 정보통신공학과 객원교수  
gsyoo21@empal.com

\*\*\*\* 종신회원 : 대전대학교 컴퓨터공학과 교수  
kgb@daejin.ac.kr(교신저자)

[2010/04/22 투고 - 2010/05/01 심사(2010/07/12 2차) - 2010/07/26 심사완료]

☆ 이 논문은 2010학년도 대전대학교 학술연구비 지원에 의한 것임

대응하고 재사용성과 통합성을 갖춘 SaaS (Software as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service) 등을 대표적 서비스로 하여 소프트웨어 산업의 수익 모델에 대한 관점이 변화하고 있다. 이러한 관점 중 SaaS는 애플리케이션을 서비스 대상으로 하는 서비스 지향의 관점으로 예전의 일반적인 웹 형태에서 웹 2.0으로 진화되면서 UI(User Interface)가 사용자 지향적으로 발전된 형태다[1][2].

기존에는 UI를 단순한 개발 결과물의 환경으로 여겨졌으나 근래에 들어서는 사용자 지향형의 UI를 확보할 수 있는 RIA를 중심으로 웹 애플리케이션 개발이 활성화 중이다. 기존의 웹 애플리케이션은 개인용 애플리케이션에 비해 용도가 한정적이고 기능이 제한되어 있어 대형 애플리케이션으로 제작하기엔 다소 무리가 있었다. 그러나 RIA는 기존의 웹 애플리케이션의 한계성을 뛰어넘어 동적인 서비스를 제공하며 사용자 플랫폼에 독립적인 서비스를 제공할 수 있다.

본 논문에서는 RIA를 개발하기 위한 Flex 플랫폼에서 사용자 지향의 웹 애플리케이션을 정확하고 빠르게 구축될 수 있는 개발방법을 제시하고 다양한 업무 분야에서 적용할 수 있는 확장성을 갖춘 다양한 라이브러리를 제시하여 개발 생산을 극대화하고 안정적인 시스템 구축을 할 수 있는 Flex 플랫폼 프레임워크인 DDX 프레임워크를 제시한다.

## 2. 관련연구

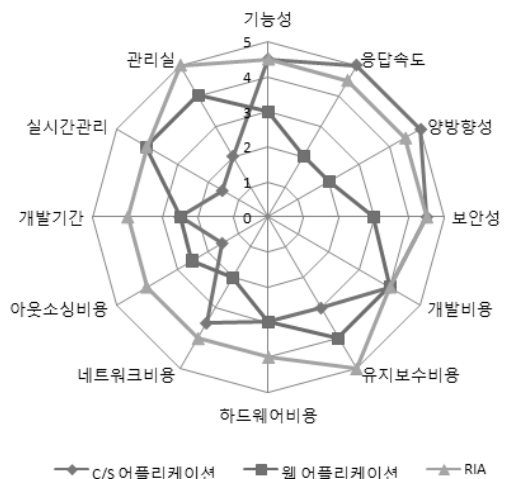
### 2.1 RIA

사용자 지향, 정보 공유, 개방화를 중시하는 웹 2.0의 발전과 함께 2000년 Forrester Research의 CEO(Chief Executive Officer)인 조지 코로니에 의해 'My View : X Internet'이라는 보고서에서 X-Internet이란 용어가 탄생되고, 2002년 Adobe사에 의해 RIA라는 개념이 탄생되어 이듬해에 국내에 도입됨으로써 두 용어는 혼용되어 사용되어져

왔지만, X-Internet의 개념은 RIA라는 내용으로 흡수되고 있다[3].

X-Internet은 C/S의 개발생산성과 Internet을 활용한 Web의 장점만을 가진 프레임워크로 eXecutable Internet과 eXtended Internet의 줄임말로 전자는 자율성·양방향성·상호 보완성을, 후자는 상호 운용성을 내포한다.

RIA는 기존의 웹 애플리케이션 기술이 가진 평면적인 표현과 순차적인 프로세스를 동적 UI와 데이터베이스 연동을 통해 저렴한 비용으로 하나의 인터페이스에서 모든 프로세스가 처리 가능하도록 해주는 기술을 의미한다[2]. 기존의 웹 애플리케이션 모델은 C/S 모델을 따르고 있다. C/S 모델은 과다한 프로세싱이 발생하면 유저는 서버 측에서 해당 프로세스의 순차적인 처리가 종료되어 이벤트를 되돌려줄 때 까지 연속적 작업을 하지 못하는 단점이 있다. C/S 모델은 이러한 단점을 극복하기 위하여 서버 측의 오버로드 해소에 집중을 해왔으나 그 한계가 있었다. 이러한 순차적인 프로세스의 단점을 극복하기 위해 만들어진 기술이 RIA이다. RIA는 표준적인 방식을 사용하여 사용자의 웹 브라우저 종류와 관계없이 동적인 UI의 배치가 가능하여 사용자 지향의 웹 페이지 구현을 목적으로 한다[4][5].



(그림 1) 각 애플리케이션 별 기능성

그림 1은 QTI 인터네셔널에서 제시한 자료로 C/S 어플리케이션, 웹어플리케이션, RIA의 기능을 비교한 그림이다. 응답속도와 양방향성 등 사용적인 측면에서는 C/S 어플리케이션과 RIA가 높은 기능을 보이고 있으며, 유비보수비용, 하드웨어비용, 네트워크 비용, 아웃소싱비용 등 전체적인 부분에서 RIA의 기능이 뛰어난을 알 수 있다[3].

## 2.2 Flex 플랫폼

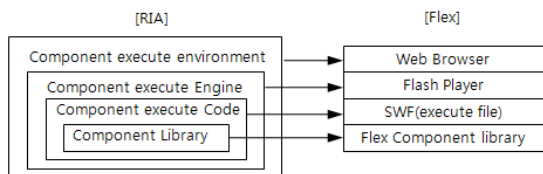
### 2.2.1 Flex 플랫폼의 특징과 장단점

Flex 플랫폼은 RIA를 지향하는 웹 어플리케이션을 개발하기 위하여 RIA상의 컴포넌트 라이브러리, 실행 코드, 실행 엔진, 환경에 대하여 Flex 컴포넌트 라이브러리, SWF 실행파일, 플래시 플레이어, 웹브라우저로 1:1 매핑하는 기술 아키텍처로서 이를 그림으로 표현하면 그림 2와 같다.

Flex 플랫폼은 디자이너들이 플래시로 개발하는 UI를 개발자들이 XML 및 스크립트 언어를 사용하여 UI를 개발할 수 있도록 만든 RIA 개발 방법이다. Flex로 구성된 웹 페이지는 페이지의 전환 없이 해당 화면에서 프로세스가 처리되거나 Data만 송, 수신하기 때문에 C/S 방식에 비해 서버 측의 오버로드가 감소된다.

Flex 플랫폼의 장점을 살펴보면 다음과 같다.

- JSP로 UI를 구성할 경우 CRUD(Create, Retrieval, Update, Delete)시 개개의 페이지를 생성하여야 했지만 Flex에서는 한 화면에서 모두 처리가 가능하다.



(그림 2) Flex의 기술 아키텍처

- 기존의 방식은 각 페이지를 호출하여 화면 전환이 이루어진 반면, Flex의 경우 첫 진입 페이지에 어플리케이션이 존재하여 그 후 페이지 전환이 발생하지 않는다.
- Data의 송, 수신 시 Flex 플랫폼은 XML 형식의 Data 구조를 이용하여 대량의 Data 전송을 자유롭게 하므로 네트워크 자원 및 서버 측의 자원에 오버로드를 감소시킨다.
- Flex 플랫폼에서 컴파일된 파일이 Flash 파일과 동일하므로 화면 내에서 마우스 우클릭을 통한 Source 내용 확인이 불가능하므로 보안상의 이점이 있다.
- Flex 플랫폼은 기존의 웹페이지에서는 표현이 어려웠던 드래그 앤 드랍, 필터링, 소팅 등의 기능을 제공한다.
- Flex 플랫폼은 Flash Player를 기반으로 동작하며, 세계적으로 Flash Player의 보급률은 99%에 달한다.

RIA 개발 방법론으로써 Flex는 다양한 장점을 보이지만, 몇 가지 단점이 있으며 이는 다음과 같다.

- 각 UI에 대한 메모리 관리가 난해하다.
- 기본적인 프레임워크에서 제공되는 컴포넌트를 이용하여 각 데이터들을 바인딩 시키기 위한 구조가 복잡하다.
- 기존의 웹 페이지에 비해 컴파일된 파일의 용량이 크다.

### 2.2.2 Flex의 구성

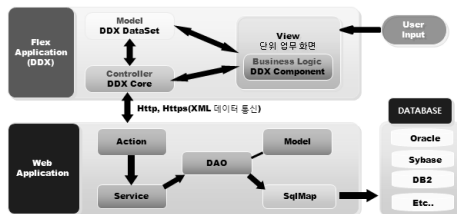
Flex는 Flex SDK, Flex 빌더, Flex 차트, Flex 데이터 서비스의 네 부분으로 구성되어 있다. SDK에는 MXML, 액션 스크립트, Flex 클래스 라이브러리로 구성되어 있고, 빌더는 이클립스 기반의 IDE(Integrated Development Environment) 구성과 자동완성 및 디버깅 코드, 위치워킹 방식의 UI 디자인, MXML-SWF 컴파일을 가지고 있다.

데이터 서비스는 기능 테스트를 위하여 Flex Framework API를 이용한 자동화된 테스트를 한다. 그리고 메시징, 메시지 발행/구독, 데이터 동기화, RTMP(Real Time Messaging Protocol) 게이트웨이를 위하여 JMS(Java Message Service) 메시지 전송 데이터 서비스를 한다. AMF(Action Message Format) 게이트웨이와 SOAP(Simple Object Access Protocol)프록시를 가지고 있다. 그리고 런타임 서비스를 지원한다.

### 3. DDX Framework 설계

본 DDX Framework는 Flex 개발 환경에서 효율적이고 표준화된 개발 방법을 제시하며 다양한 업무 분야에서 적용할 수 있는 확장성을 갖추고 다양한 라이브러리를 활용함으로써 개발 생산성을 극대화 하여 안정적인 시스템 구축을 목적으로 한다.

Web Application은 사용자의 presentation tier, logic tier, data tier의 3-tier 구조를 지니고 있다. 기존의 Flex 역시 동일한 구조를 가지고 있다. 본 연구에서는 MIS의 관점에서 UI와 Data 처리를 기준으로 재구성하여 presentation tier에서 Data 값 변경의 Event 관리를 위한 Data 관점, 데이터베이스의 Data CRUD 업무에 따른 Database 관점, presentation tier에서 단위 업무 화면마다 이루어지는 Binding에 대한 일괄 처리를 위한 Data Binding 관점, presentation tier 상의 UI와 logic tier의 연동을 위한 어플리케이션 연동 관점으로 나뉘었다. 이들 모듈간의 연동 관계를 그림으로 표현 하면 그림 3과 같다.



(그림 3) DDX Framework

### 3.1 Data 관점

조회 시 Model Class를 만들지 않고, 조회 결과 데이터에서 자동 관리하며, 필요 시 화면에서 지정 할 수 있다. DataSet값의 변경에 관련된 Event등을 제공하며 이미 삭제된 레코드의 정보를 얻을 수 있다.

### 3.2 Database 관점

Database 관점은 데이터베이스의 CRUD와 관련된 데이터들에 대해 개발자 지향적으로 편리한 메소드와 프로퍼티를 구현하였다. DataSet과 관련된 메소드는 표 1과 같다.

(표 1) DataSet 메소드

메소드	기능
setDataSet	DataSet의 Column정보, Data 정보 갱신
copy	현 DataSet의 Column정보와 Data 정보 복사
addRow	row 추가
insertRow	지정된 index에 row 추가
deleteRow	지정된 row 삭제
setColumn	지정된 row의 Column에 지정된 값 저장
getColumn	지정된 row의 Column 값을 획득
getRowCount	Data row 수 획득
getColCount	Column 수 획득
getRowType	지정된 row의 Type 획득
setRowType	지정된 row의 Type 지정
findRow	지정된 column을 value 값에 의거해 순차검색
addColumnInfo	Column 추가
saveXml	DataSet 정보를 xml로 획득
filter	넘어온 조건 식에 의해 filter

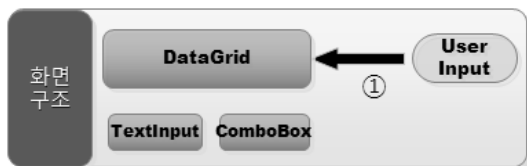
DataSet 프로퍼티는 표 2와 같다.

(표 2) DataSet 프로퍼티

Property Name	Comment
FireEvent	이벤트 흐름을 제어
deleteAction	row 삭제 후 현 index 를 유지
findAction	조회 또는 data 갱신 후 index 를 유지
canColumnChange	column 값 변경 전에 호출되는 function
onColumnChanged	column 값 변경 후에 호출되는 function
canRowPosChange	row 값 변경 전에 호출되는 function
onRowPosChanged	row 값 변경 후에 호출되는 function
onDeleting	row 삭제 전에 호출되는 function
onDeleted	row 삭제 후에 호출되는 function
onRowInserted	row 추가 후에 호출되는 function
row	현재 진행중인 index
selectedIndex	현재 선택된 index
orgBuffer	삭제된 row 정보

### 3.3 Data Binding 관점

UI 개발 시 Component간의 Binding은 사용자의 입력이 발생(①)되면 DataGrid, TextInput, ComboBox 등의 UI에 일괄적인 변형이 1회에 가능하도록 하였으며 이 구조를 그림으로 표현하면 그림 4와 같다.

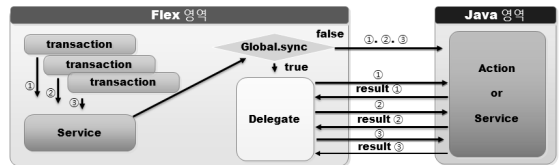


(그림 4) Data Binding 구조

### 3.4 어플리케이션 연동 관점

어플리케이션 연동 관점에서는 여러 개의 서비스를 만들지 않고 한 번의 통신으로 여러 역할을 수행 가능하도록 하였다. 각각의 트랜잭션 ①, ②, ③을 서비스에 연동하고 Global.sync속성을 이용

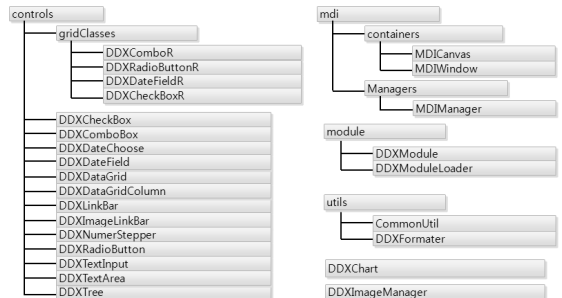
하여 서비스 순서를 조절할 수 있게 하였다. 속성 상태가 false 상태면 동시에 처리하고, true 상태면 각 트랜잭션에 대하여 순차적으로 처리 한 후 지정된 callBack 함수를 이용하여 서비스 후 처리가 가능하도록 하였다. DDX에서 transaction의 흐름을 그림으로 표현하면 그림 5와 같다.



(그림 5) DDX Transaction 프로세스 흐름도

### 3.5 DDX Framework Component 구성

DDX 프레임워크 컴포넌트 구성도는 그림 6과 같다.



(그림 6) DDX Framework Component 구성도

CheckBox	Combo	radioButton	Date	TextInput
<input type="checkbox"/>	선택	<input type="radio"/> 선택 <input type="radio"/> 세울 <input type="radio"/> 경주	2009-05-20	2009-05-20
<input checked="" type="checkbox"/>	세울	<input type="radio"/> 선택 <input checked="" type="radio"/> 세울 <input type="radio"/> 경주	2009-05-24	2009-05-24
<input type="checkbox"/>	경주	<input type="radio"/> 선택 <input type="radio"/> 세울 <input checked="" type="radio"/> 경주	2009-05-22	2009-05-22

(그림 7) DDXDataGrid 화면

## 4. DDX Framework 구현

DDXDataGrid의 화면은 그림 7과 같이 설계하였으며 구현한 내용에 대한 의사코드는 표 3과 같다. 기본적인 DataSet Binding과 속성 정의만으로 쉽게 화면 구현이 가능하며, 직관적인 코딩으로

인해 개발과 유지보수가 용이하다.

(표 3) DDXDataGrid 의사코드

```
<ddx:DdxDataGrid bindDataset="{BindingDataSetName}"
editable="true or false"> // 데이터 바인딩의 정의
  <ddx:columns>
    <ddx:DdxDataGridColumn
      headerText="CheckBox or Combo or radioButton or
Date or TextInput" // 데이터 format
      dataField="if viewValue=Y/N then useYn or
// 각 format에 따른 데이터 필드
if viewValue=inputDataset then campFg or
if viewValue=date then updateDay"
      renderer="each view type{checkBox, comboBox,
radioButton, dataField. updateDay}" // 데이터 표시 속성
      if databinding then innerDataset="{inputDataSet }"
innerColumn="ColumnName" // 데이터 바인딩 속성
innerData="DataFieldName"
      if datatype = "Date" then formatString =
"YYYY-MM-DD" /> // date format의 특
정 표시 속성
    </ddx:columns>
  </ddx:DdxDataGrid>
```



(그림 9) DDXTextArea 화면

DDXTextArea는 그림 9처럼 표시하며, 기존의 Flex의 TextArea 상에서는 일반적인 Data의 표현이나 Data CRUD에 대한 직접적인 접근을 하였으나, 본 DDX Framework에서는 기존의 TextArea에 DataSet을 사용 할 수 있도록 하였다. 이로 인하여 UI상에서 표현되는 Data에 대하여 Binding을 함으로써 내부 DataSet 적용이 가능하다.

(표 5) DDXTextArea 의사코드

```
<ddx:DdxTextArea bindDataset="{BindingDataSetName}"
bindColumn="{BindColumnName}">
  <ddx:htmlText> legacy html syntax
</ddx:htmlText>
</ddx:DdxTextArea>
```



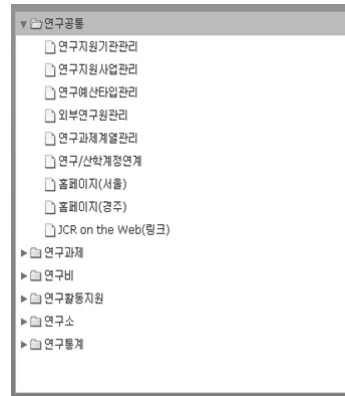
(그림 8) DDXCheckBox 화면

DDXCheckBox는 기존의 CheckBox에 DataSet을 사용 할 수 있으며 지정된 trueValue, falseValue에 의하여 참, 거짓 값을 사용한다. 옵션의 텍스트 label은, 그림 8처럼 DdxCheckBox의 상·하·좌·우에 배치 가능하며 구현한 내용의 의사코드는 표 4와 같다.

(표 4) DDXCheckBox 의사코드

```
<ddx:DdxCheckBox
  bindDataset="{BindingDataSetName}"
  bindColumn="{useYn}"
  trueValue="Y(Default) or N"
  falseValue="N(Default) or Y"
  label="view value"
  labelPlacement="top or bottom or right or left"/>
```

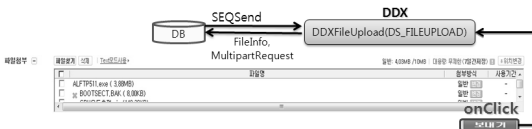
DDXTree는 그림 10처럼 기존의 Tree에 DataSet을 사용 할 수 있으며 depthColumn에 따라 Node의 Depth를 적용 할 수 있다.



(그림 10) DDXTree 화면

DDXFileUpload는 DDX파일 업로드 컴포넌트에서 파일 찾기를 이용하여 파일을 선택하면 해당 파일 정보가 DS\_FILEUPLOAD에 저장된다. 그 정

보를 DDXCore의 FlexFile- UploadAction에 전송한다. DDX UI에서 File을 송신할 경우 DefaultFileUploadInterceptor 통하여 FileInfoMap클래스를 생성한다. FileInfoMap클래스에서 송신한 파일의 정보를 획득 한다. FlexFileUploadAction는 파일을 저장 후 공통 파일Table에 해당파일의 정보를 저장 후 seq를 생성 후 UI(DDX)에 전송한다.



(그림 11) DDXFileUpload 프로세스

(표 6) DDXFileUpload 의사코드

```
@Controller
public class TestAction extends InitialAction{

    @RequestMapping("/common/FileUpload.do")
    protected ModelAndView fileUpload(HttpServletRequest request) {
        FileInfoMap varA = getFileInfo(request);
        logger.debug("each{fileseq/filename/filesize}"
            +
            varA.to{getFileSeq()/getFileName()/getFileSizeInBytes()});
        return getView();
    }

    <bean id="DefaultFileUpload"
class="mjf.web.common.interceptor.DefaultFileUploadIntercept
or">
    <property name="maxMemorySize"
value="value or disk path"/>
</bean>
```

DDXFileDownload는 DDX파일 다운로드 컴포넌트에서 다운하고자 하는 파일번호와 파일순번을 DDXCore의 FlexFileDownloadAction에 전송한다. FlexFileDownloadAction는 Database에서 해당파일의 정보를 검색 후 Http 다운로드 방식으로 DDX에 전송한다.

## 5. DDX Framework 비교분석

### 5.1 Data 관점

기존의 Flex Framework는 조회 시 결과 Data의 객체를 담을 수 있는 Class(Model 또는 Value Object)를 관리함으로써 Model class들이 많아진다. 따라서, 개발 및 관리가 어렵다. Data의 값이 변경 전에 체크를 하기 위해 값이 변경되는 곳마다 별도의 프로그램 Logic을 처리를 해야 하는 단점이 있다.

이와 비교하여 DDX Framework는 조회 시 Model Class를 만들지 않고, 조회 결과 데이터에서 자동 관리하며, 필요 시 화면에서 지정 할 수 있다. Data값의 변경에 관련된 Event등을 제공하며 이미 삭제된 레코드의 정보를 얻을 수 있다.

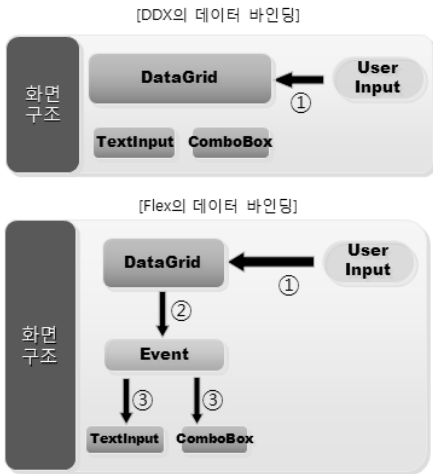
### 5.2 Database 관점

기존의 Flex Framework는 Database변경 시 Model Class를 동일하게 변경하여야 하며 화면상에서 각각의 Model을 사용한 Collection을 추가, 수정, 삭제를 할 경우 각각의 Row들의 상태를 파악하기 어렵다.

하지만 DDX는 Database 변경이 발생하여도, Component 상에 연결된 Column의 변경이 없다면 변경이 사용 가능하다. DataSet 내부적으로 row의 상태 관리할 수 있다. 동적 Column 추가 가능하고 Sort, Filter, Formatter에 대한 지원이 가능하며 모든 Component와 Binding을 지원한다.

### 5.3 Data Binding 관점

Flex는 그림 12의 하단처럼 사용자가 DataGrid의 특정 Column 값을 변경하면 DataGrid에서 값 변경을 인식하여 Event 발생하고 변경된 Column 값에 연관되어 있는 component에 변경된 값 갱신한다. 즉, TextInput 또는 ComboBox의 값이 변경되어도 각각 이 일련의 과정을 거쳐야 하는 단점이 있다.



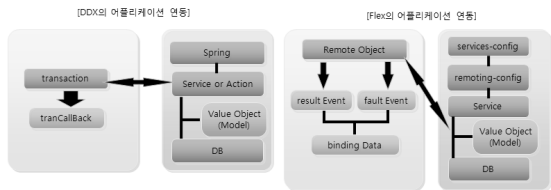
(그림 12) DDX와 Flex의 데이터바인딩

하지만, DDX는 그림 12의 상단처럼 DataGrid, TextInput, ComboBox 중 어떠한 값을 변경해도 모든 값이 자동으로 일치 된다.

### 5.4 어플리케이션 연동 관점

Flex는 그림 13의 우측처럼 서버와 통신하는 경우(CRUD 시) 각각 해당 작업 필요하며 조회 시에는 해당 model class가 필요한 복잡함이 있다.

하지만, DDX는 그림 13의 좌측처럼 공통 transaction으로 처리하여 tranCallBack에서 후처리 조회 시 model class 필요하지 않다.



(그림 13) DDX와 Flex의 어플리케이션 연동

### 5.5 타 프레임워크와 비교분석

표 7은 본 DDX Framework와 현재 Flex 플랫폼 상에서 사용되고 있는 Cairngorm 프레임워크 및 Pure-MVC 프레임워크와의 특징들을 비교하였다.

(표 7) 타 프레임워크와의 비교분석표

	DDX	Cairngorm	Pure-MVC
주요기능	MIS 관점에서 데이터 바인딩을 간략화	MVC 패턴을 적용하는 액션 스크립트 라이브러리 제공	MVC(Model, View, controller) 개념 적용, 느슨한연결성을 지원.
적용규모	중간 혹은 대규모 프로젝트	중간 혹은 대규모 프로젝트	개별 프로젝트
단점	-	수많은 클래스 생성 필요 하나의 이벤트당 하나의 커맨드 객체 매핑	MXML 지원하지 않음. 수많은 레이아웃, 추가 클래스 필요
제공	가이드라인, 라이브러리	가이드라인, Tool, 라이브러리	기본 템플릿
컴파일후 객체 수	2~3	30~40	10~15
바인딩 방법	라이브러리에서 처리	ModelAndLocator를 싱글톤 개념으로 바인딩 하여 mxml에서 사용	각각의 뷰 컴포넌트를 담당할 Mediator 상속받은 클래스를 이용하여 View 쪽 작업을 처리
바인딩 횟수	1회	각 이벤트 마다	View 작업 처리 마다
유지보수	쉬움	난해함	보통

Flex 상에서 사용 가능한 프레임워크는 많이 다양하게 있으나, MIS 관점으로 데이터 바인딩에 초점을 맞추어 처리하는 프레임워크는 부족하다. 컴파일 후 객체 수는 동일한 내용의 프로그램을 프로그래밍 한 후 컴파일 한 후에 생성되는 객체들을 비교하였다. Cairngorm 프레임워크는 각 이벤트당 객체를 생성함으로써 많은 파일을 생성하게되어 유지보수 시 난해함을 보이며 Pure-MVC 프레임워크는 클래스 단위로 파일들을 생성함으



로써 Cairngorm 프레임워크에 비하여 수월한 유지보수 성향을 보였다. 본 논문에서 제시한 DDX 프레임워크는 두 프레임워크에 비하여 데이터의 바인딩에 좀 더 초점이 맞추어져 있어, 컴파일 후 파일 객체수와 바인딩 타이밍, 바인딩 횟수를 최소화시켜 유지보수를 더욱 수월하게 하였다.

## 6. 결 론

컴퓨터 사용 환경의 변화와 사용자 지향, 정보 공유, 개방화를 중시하는 웹 2.0 시대로 발돋움함에 따라 동적 UI와 데이터베이스 연동을 통해 저렴한 비용으로 하나의 인터페이스에서 모든 프로세스가 처리 가능하도록 해주는 RIA가 대두되고 있다. 이에, 효율적이고 표준화되어있으며 확장성을 갖춘 프레임워크 및 컴포넌트들이 필요하다.

본 논문에서는 Flex 플랫폼 개발 환경에서 효율적이고 표준화된 개발 프레임워크를 설계 및 구현하였다. 이 DDX 프레임워크는 다양한 업무 분야에 적용할 수 있는 확장성이 있으며 다양한 라이브러리를 활용하여 개발 생산성 증대 및 어플리케이션 시스템을 안정적으로 구축할 수 있도록 도와준다. 본 논문에서 구현한 DDX 프레임워크는 다음과 같이 4가지 관점과 컴포넌트들로 정리할 수 있다.

첫째, Data 관점은 조회 시 Model Class를 만들지 않고, 조회 결과 데이터에서 자동 관리하며, 필요 시 화면에서 지정 할 수 있다.

둘째, Database 관점은 개발자 지향적으로 편리한 메소드와 프로퍼티를 제공하며 동적 Column제공과 컴포넌트 바인딩을 가능하게 하였다.

셋째, Data Binding 관점은 UI 개발 시 Component 간의 Binding에 관하여 나타내었으며 동적 프로세스를 통한 데이터 바인딩을 가능하게 하였다.

넷째, 어플리케이션 연동 관점은 서버 어플리케이션과 연동 시 다중 작업이 아닌 일회의 통신으로 다수의 CRUD 작업을 가능하게 하였다.

본 연구실에서 향후 연구 과제로는 DDX 프레임워크를 사용함에 있어 동적 UI에 대한 다양한 컴포넌트를 개발하여 사용자 편의성을 보다 더 증대시킬 수 있는 프레임워크를 개발할 예정이다.

## 참 고 문 헌

- [1] 민옥기, 김학영, 남궁한, 클라우드 컴퓨팅 기술 동향, 전자통신동향분석, 제24권 제4호, 2009년 8월, pp. 1-4.
- [2] 삼성 SDS SOA 연구회, SOA : 서비스 지향 아키텍처, 성안당, 2007
- [3] 노규남, Flex 3 핵심노트, 영진닷컴, 2009
- [4] 조경모, Dynamic UI Design을 고려한 Web Application에 관한 연구, 커뮤니케이션디자인학회, Vol27, No 0, 2008. pp. 51-58
- [5] Joshua Duhl, WHITE PAPER : Rich Internet Application, IDC, 2003.
- [6] Jiang Tian, Richard Foley and Huaglory Tianfield, 'A New Agent-Oriented Development Methodology', International Conference on Intelligent Agent Technology (IAT'04), pp 373-376, 2004.
- [7] Wang Linlin Hu Dehua, 'Research and Realization of RIA WebGIS Based on Flex', Intelligent Systems and Applications, ISA 2009 International Workshop on , pp 1-4, 2009.

## ● 저 자 소 개 ●



### 김 양 훈

2005년 대전대학교 컴퓨터공학과(공학사)  
2007년 대전대학교 대학원 컴퓨터공학과(공학석사)  
2007~현재 대전대학교 대학원 컴퓨터공학과 박사과정

관심분야 : 소프트웨어공학, 시스템분석 및 설계, 프레임워크  
E-mail : kimyh7902@daejin.ac.kr



### 정 구 범

1999년 : 대구 가톨릭대학교 박사  
1997년 ~ 2008 : 상주대학교 컴퓨터공학과 교수  
2009년 ~ 현재 : 경북대학교 상주캠퍼스 컴퓨터공학과 교수

관심분야 : 지능정보시스템, 소프트웨어 공학, 엔터프라이즈 아키텍처  
E-mail : jgb@knu.ac.kr



### 유 갑 상

1987년 동국대학교 대학원 전자계산학과(공학사)  
1992년 연세대학교 대학원 전자계산학과(공학석사)  
2001년 Ural State University 응용수학학과(이학박사)  
1996~현재 주성시스템(주) CEO  
2002~현재 동국대학교 정보통신공학과 객원교수  
관심분야 : 운영체제, 소프트웨어공학, 임베디드시스템.

E-mail : gsyoo21@empal.com



### 김 국 보

1984년 서울산업대학교 전자계산학과 졸업(학사)  
1986년 연세대학교 공학대학원 전자계산학과 졸업(석사)  
1997년 대구가톨릭대학교 전산통계학과 졸업(박사)  
1988년~1990년 : 해군 중앙전산소장  
1990년~1993년 : 부경대학교 교수  
1993년~현재 : 대전대학교 컴퓨터공학과 교수  
관심분야 : 소프트웨어공학, 시스템 분석 및 설계, e-Biz 시스템  
E-mail : kgb@daejin.ac.kr