

AMGA GUI Client 툴킷 구현 : AMGA Manager

Implementation of AMGA GUI Client Toolkit : AMGA Manager

허태상, 황순욱, 박근철
한국과학기술정보연구원

Taesang Huh(tshuh@kisti.re.kr), Soonwook Hwang(hwang@kisti.re.kr),
Guenchul Park(gcpark@kisti.re.kr)

요약

EMI gLite 미들웨어 컴포넌트인 AMGA 서비스는 과학기술 연구자들에 의해 메타데이터 저장소로 분산환경의 대규모 실험 데이터 분석에 넓게 활용되고 있고 그 사용은 메타데이터 카탈로그를 필요로 하는 일반산업에까지 확대되고 있는 추세이다. 하지만, 리눅스와 그리드 사용자 인터페이스 기반의 AMGA는 다른 상용 데이터베이스 시스템과 비교해서 범용성 사용자 인터페이스의 부재라는 단점이 있으며, 기능의 우수함에도 불구하고 활용 및 확산하는데 어려움이 있다. 본 논문에서는, AMGA 사용의 제약을 최소화하고 사용자 친화적 인터페이스를 제공하기 위해 객체지향적 모델링 언어(UML)를 이용하여 AMGA GUI 툴킷을 개발하였다. 현재 Belle II, WISDOM, MDM 등과 같이 많은 사용자 커뮤니티에서 AMGA는 주요 컴포넌트로 사용되고 있지만 본 개발을 통해 AMGA 신규 사용자의 진입장벽을 낮춰줄 수 있을 뿐만 아니라 보다 많은 커뮤니티로 확대가 기대된다.

■ 중심어 : | 그리드 | 메타데이터 카탈로그 | AMGA | UML | gLite | EMI | 이클립스 | RCP | 보안 |

Abstract

AMGA service, which is one of the EMI gLite middleware components, is widely used for analysis of distributed large scale experiments data as metadata repository by scientific and technological researchers and the use of AMGA is extended farther to include general industries needing metadata Catalogue as well. However AMGA, based unix and Grid UI, has the weakness of being absence of general-purpose user interfaces in comparison to other commercial database systems and that's why it's difficult to use and diffuse it although it has the superiority of the functionality. In this paper, we developed AMGA GUI toolkit to provide work convenience using object-oriented modeling language(UML). Currently, AMGA has been used as the main component among many user communities such as Belle II, WISDOM, MDM, and so on, but we expect that this development can not only lower the barrier to entry for AMGA beginners to use it, but lead to expand the use of AMGA service over more communities.

■ keyword : | Grid | Metadata Catalogue | AMGA | UML | gLite | EMI | Eclipse | RCP | Security |

1. 서론

EMI(European Middleware Initiative)/gLite (Lightweight

Middleware for Grid Computing) 미들웨어 공식 메타데이터 카탈로그 서비스인 AMGA(Arda Metadata Grid Application)는 세계 각국의 다양한 분야에 걸쳐

접수번호 : #111118-006

접수일자 : 2011년 11월 18일

심사완료일 : 2012년 02월 29일

교신저자 : 박근철, e-mail : gcpark@kisti.re.kr

그리드 작업에 대한 메타데이터 저장소로 활용되고 있다[1][2]. 특히, 개인정보의 보안을 중요시 하는 바이오 메디컬 연구에 활용되고 있으며, 특히 스위스 CERN(European Nuclear Research Centre)의 ATLAS(A Toroidal LHC Apparatus), LHCb(Large Hardron Collider beauty) 실험과 같이 대용량 파일기반의 실험 데이터 분석과 같은 고에너지 물리분야, 독일 DKRZ(Deutsche Klimarechenzentrum)의 기상연구, EGEE(Enabling Grids for E-Science in Europe)의 Health e-Child, 프랑스 IN2P3의 신약후보물질 탐색 및 MDM(Medical Data Manager), 터키 국립지진학센터의 지진 데이터 연구, 흑해 주변국 공동프로젝트 envioGRIDS의 위성이미지 저장 및 처리 등에도 활용되고 있다. 또한 2년 전부터는 우주의 기원을 밝히는 거대실험인 일본의 Belle-II 실험에서 데이터 핸들링 시스템에 활용되기 시작했다[3-8]. 현재 AMGA 컴포넌트는 스위스 CERN, 프랑스 IN2P3, 이탈리아 INFN와 국제협력을 통해 한국 KISTI(Korea Institute of Science and Technology Information)가 주도적으로 개발 중에 있다. AMGA 미들웨어 컴포넌트는 세계 각국에서 기술성을 인정받아 EGEE 인프라에 이어 DCI(Distributed Computing Infrastructure) 인프라에 이르기까지 10만개 이상의 컴퓨팅 노드에서 활용되고 있고 있어 사실상 국제적 표준(de facto standard)이라고 말할 수 있다.

그리드 사용자는 전문적인 그리드 질의, 유닉스 명령어, AMGA 명령어 등을 통해 데이터의 접근이 가능하다. 초기 사용자가 이러한 환경에 익숙해지기까지는 많은 시간이 걸리고, 메타데이터 카탈로그 이용에 많은 고충을 겪어왔다. AMGA 서비스가 보다 많은 사용자를 확보하기 위해서는 다소 생소한 사용자 인터페이스를 범용적으로 쉽게 이용할 수 있도록 사용자 친화적이고 독립적인 인터페이스가 개발이 필요하다.

본 논문에서는 사용자가 별도의 그리드 사용자 인터페이스 사용 없이 AMGA 서버에 접근하여 강력한 보안인증 하에 자신의 메타데이터를 쉽게 관리, 편집하고 개발에 활용할 수 있는 그래픽 유저 인터페이스 구현에 대해서 논의한다.

소프트웨어의 국제적 경쟁력 확보와 복잡한 구현을

위해 객체지향 프로그래밍을 기반으로 하는 객체지향 모델링 언어를 사용하여 요구사항 분석과 설계에 활용하면 다양한 문제점 해결은 물론, 재사용도가 높은 프로그램 개발이 용이해진다. 따라서 본 개발에서는 UML 모델링 기법을 AMGA GUI(Graphical User Interface) toolkit인 AMGA Manager의 요구사항분석 및 설계에 적용하였고, Eclipse Platform인 RCP(Rich Client Platform) 기반으로 구현하여, 윈도우, 리눅스, 맥과 같이 이기종 플랫폼에서 동작할 수 있도록 하였다.

본 연구의 2장에서는 AMGA Manager 개발과 관련된 연구를 서술하고 3장에서는 요구사항을 토대로 한 설계를 기술한다. 4장에서는 AMGA Manager 구현을 보여주고, 5장에서는 결론 및 기대효과로 마무리 한다.

II. 관련 연구

1. UML(Unified Modeling Language)

UML을 활용한 소프트웨어 개발은 생산성 향상을 위해 개발 초기부터 지속적으로 이루어지는 통합 및 검증 작업, 코드생성 자동화, 반복적인 개발은 물론 프로젝트 위험 요소를 감소시켜, 개발팀원들 간의 의사소통을 통한 빠른 개발이 가능하게 한다. 또한 비주얼 설계 사항들로 가시성과 예측성을 증가시켜 개발 소프트웨어의 높은 완성도를 기대할 수 있다.

AMGA GUI Client 개발에서 요구사항 분석단계부터 설계에 이르는 프로젝트 주요 작업을 통합 모델링 언어(UML) 툴인 IBM사의 Rational Rose Real time을 사용하였다. 자사의 RUP(Rational Unified Process) 기반인 본 소프트웨어는 포괄적인 비주얼 개발환경으로 실시간 개발의 어려움을 해소시킬 수 있는 표기법, 프로세스, 개발 도구를 통합시킴으로써 본 개발에 적합한 범용적인 도구로 판단된다.

2. 이클립스 RCP

이클립스(eclipse)는 자바 기반의 개방적이고 확장성을 지닌 IDE 개발도구으로써 다양한 기능의 플러그인 사용으로 확장이 가능하다[22]. C++, 델파이, 파스칼 같은

다양한 언어도 플러그인 형태로 개발 툴로써 사용 가능하여 개발 언어 환경의 확장도 가능하다. Java IDE 개발 도구는 코드 편집기, 컴파일러, 프로그램 수정기, 그래픽 사용자 인터페이스 설치기 등으로 구성되어 사용자에게 강력한 개발 프레임워크를 제공하고 설계에서 빠른 프로토타입의 개발 및, CVS(Concurrent version system)를 활용하여 개발자 상호간의 협업을 지원하는 플랫폼으로 빠르고 효율적인 개발이 가능하다.

이클립스 3.0에서는 이클립스 사용자 인터페이스 공간이 되는 핵심 기능들을 비 IDE 어플리케이션에서도 쉽게 사용할 수 있도록 리팩토링 한 리치클라이언트 플랫폼을 제공한다. RCP(Rich Client Platform)의 주요 장점으로는 사용자 편의성을 높여 풍부한 사용자 경험을 제공하고 다양한 OS 플랫폼으로 독립적 사용을 가능하게 한다. 또한 하부 운영체제의 네이티브 위젯(widget)을 사용함으로써 네이티브 룩앤필(look-and-feel)을 제공한다 할 수 있다. 이클립스 RCP의 기본 구성 컴포넌트는 에디터, 뷰, 퍼스펙티브 등으로 구성된 워크벤치와 OS 플랫폼 환경과 통합된 컴포넌트 및 플랫폼 독립적인 API를 제공하는 SWT(Standard Widget Toolkit), 데이터 뷰(Data view), 위자드(Wizard), 다이얼로그 컴포넌트(Dialog Component), 텍스트 관리, 이미지 및 폰트 컴포넌트 등을 제공하기 위해 SWT를 확장하는 많은 유틸리티 클래스들을 포함하는 JFace, 플러그인 라이프사이클 관리 등을 위한 프레임워크인 OSGi(Open Services Gateway Initiative), 이클립스 구동 시 설치된 로컬 플러그인을 찾아서 각 설정 파일을 읽어 확장과 확장점을 일치시키는 Platform Runtime으로 구성된다[9][10].

3. AMGA

AMGA는 EGEE의 gLite 3.1의 메타데이터 카탈로그이고, 데이터베이스에 저장되는 그리드 잡(jobs)을 위한 데이터로 설계되었다. 그리드 개념을 RDB(relational DB)로 이식한 구조로 서로 다른 이기종의 DB간에도 액세스 및 복제(replication)가 가능한 분산 환경(distributed environment)을 포함하고 있다. 고유 TCP 스트리밍 프로토콜과 고유의 AMGA SQL 쿼리를

가지고 있고, 최근 AMGA 버전에서는 다른 벤더의 DBMS에 대해 범용으로 적용 가능한 네이티브 SQL을 채택하고 있다.

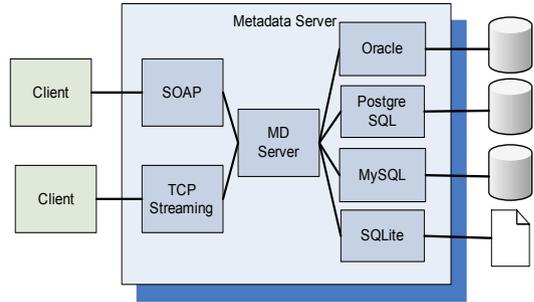


그림 1. AMGA Implementation[5]

또한 AMGA는 그리드 컴포넌트이지만 독립적으로 구성이 가능한 구조로 활용 영역이 다양하다[23][24]. [그림 1]과 같이 gLite AMGA 서비스는 기존의 PostgreSQL, MySQL, Oracle, SQLite과 같은 데이터베이스 백엔드에 보안 프론트엔드 서비스로써 gLite 표준 인터페이스 저장소 역할을 수행한다. 또한 DN 기반의 액세스 컨트롤 리스트와 그리드 인증서를 통해 세밀한 액세스 컨트롤이 가능하다[11][12]. AMGA는 디렉토리-파일 접근을 따르는 트리와 같은 구조로 그룹화된 메타데이터를 계층적으로 표현할 수 있다[13]. AMGA에서 제공하는 보안은 유닉스 스타일의 승인과 컬렉션, 엔트리 제어에 사용되는 액세스 컨트롤 리스트가 있다. 또한 보안 접속의 SSL 방식을 따르며, 클라이언트 인증은 ID/PW, 인증서, 그리드 프로시 인증 방식을 따르고 가상조직관리(Virtual Organization Management)시스템이 있어 가상조직마다 액세스 컨트롤을 관리할 수 있다[14][15]. 메타데이터 카탈로그인 AMGA는 RDBMS와 파일시스템 관점으로 설명될 수 있다. 스키마는 테이블 스키마, 컬렉션은 데이터베이스 테이블, 애트리뷰트는 스키마 컬럼, 엔트리는 테이블 레코드로 RDBMS와 개념적 매핑이 되고, 컬렉션은 디렉토리, 엔트리는 파일로 파일시스템과 매핑이 된다.

표 1. 메타데이터 카탈로그 비교

Function	SDSC MCAT	GriPhyN MCS	KISTI AMGA
Collection	○	○	○
Replication	○	×	○
Fault Tolerance	○	×	△
Federation	○	×	○
GSI Authentication	○	○	○
Standard Interface Support	○	○	○
User defined metadata attribdata	○	○	○
Native SQL Query Interface	×	×	○
API for storing and querying metadata	○	○	○
Platform Independency	×	○	○

메타데이터 카탈로그 서비스 기술은 그리드 기반의 e-Infrastructure 구축에 있어 대용량의 과학데이터 영역으로 갈 수록 핵심기술로써 부각되고 있다.

현재 미국 SDSC(San Diego Supercomputer Center)가 SRB(Storage Resource Broker) 전용 메타데이터 카탈로그로써 개발한 MCAT과 미국 GPN(Grid Physics Network), NVO(National Virtual Observatory) 프로젝트로 개발된 MCS(Metadata Catalogue Service), 한국 KISTI에서 개발하고 있는 gLite 공식 미들웨어 컴포넌트인 AMGA가 대표적인 메타데이터 카탈로그 기술이며, 기능적 측면에서는 AMGA가 타 기술과 동등하거나 보다 우수하다고 말할 수 있다[표 1]. 활용 동향도 10만이 넘는 그리드 인프라에 설치 및 활용되고 있어 사실상 표준화된 기술이라고 할 수 있다.

4. AMGA Issue

AMGA에서 사용되는 쿼리는 카탈로그 구조를 수용하는 AMGA 쿼리와 Native SQL 쿼리를 가진다. AMGA에 사용되는 쿼리는 디렉토리별로 메타데이터 스키마를 가지기 때문에 일반적인 SQL구문과는 달리 독자적인 질의형태를 가지고 있어 AMGA 쿼리 사용이 익숙하기 전까지는 사용자가 문법에 어려움을 겪는다. 또한, 텍스트 기반의 리눅스 인터페이스를 수용하고 있

는 가장 일반적인 mdclient 유틸리티를 사용할 경우, [표 2]와 같이 결과형태도 스키마와 데이터가 반복적으로 표현되어 한눈에 데이터 구조를 판단하기 어렵고 사용자가 모니터링하는데 직관적이지 않다. 물론 타 응용(Application)에서 쉽게 사용할 수 있는 API가 JAVA, Python, C++ 등으로 다양하게 제공되지만 일반적인 사용자가 활용하기에는 쉽지 않다.

표 2. 예제 : AMGA select 쿼리에 대한 결과

```

Example ; AMGA select-query

Query> selectattr /tshuh/weather:FILE /tshuh/weather:city
/tshuh/weather:temp_lo /tshuh/weather:temp_hi
/tshuh/weather:taken /tshuh/weather:precip 'precip=0'
>> imging
>> Tokyo2
>> 4
>> 22
>> 2010-07-22 17:08:01
>> 0
>> img1.jpg
>> Hamburg
>> 15
>> 23
>> 2008-07-17 11:03:17
>> 0
>> img2.jpg
>> Hamburg
>> 13
>> 23
>> 2008-07-18 11:03:17
>> 0
>> . . .
Query>
    
```

III. 요구사항 분석 및 설계

메타데이터 카탈로그 GUI 툴 개발을 위한 요구사항은 많은 과학기술연구자들이 겪었던 사용자 인터페이스의 불편함에서 출발한다. Window OS 어플리케이션 사용자들과 Linux 환경에 익숙하지 않는 사용자들은 AMGA CLI를 이용하는 데 많은 어려움이 따른다. 본 요구사항 분석은 일반사용자가 AMGA 서비스를 이용하기 위해 필요한 CLI를 최대한 GUI로 전환하는 것을

개발의 주요 목표로 하고 아래와 같이 AMGA가 보유한 주요 기능을 중심으로 요구사항을 구성했다.

1. 기본 요구사항

AMGA는 AMGA Metadata Catalogue Support Forum¹을 통해 미들웨어 사용자의 다양한 요구사항을 반영해 왔으며, 사용자들은 불편한 사용자 인터페이스 부분에 대한 개선부분의 의견을 게재해 왔다. 또한 그리드 미들웨어 관련 각종 컨퍼런스, 시연 및 튜토리얼 시 응용개발에 앞서 데이터 관리 및 개발에 편하게 활용할 수 있는 GUI 기능에 대한 요구도 있었다.

표 3. 요구사항 분석 요약

액터	<ul style="list-style-type: none"> - 사용자그룹 · 데이터 제공자 또는 관리자 · 데이터 수요자
엑세스 컨트롤 지원	<ul style="list-style-type: none"> - 외부 시스템 · AMGA 서버, LFC, VOMS 서버, 스토리지
컬렉션 관리	<ul style="list-style-type: none"> - 컬렉션 · 상하위 컬렉션에 대한 액세스 컨트롤 위임 - 엔트리 · 엔트리별 액세스 컨트롤 - 사용자/그룹 · ACL 관리(그룹 리스팅, 그룹추가/삭제) · 사용자 관리
메타데이터 관리	<ul style="list-style-type: none"> - 컬렉션 관리(생성, 수정, 삭제) - 속성 관리(permission, owner) - 사용자 편의의 컬렉션 탐색기 - 엔트리 연계 대규모 매서드 처리 - 사용자 편의의 메타데이터 관리기 - 메타데이터 관리(생성, 수정, 삭제)
보안/접속	<ul style="list-style-type: none"> - 메타데이터 스키마 관리 (Data, Attributes, ACL Contraints, Index) - 사용자 편의의 메타데이터 관리기 - 메타데이터 관리(생성, 수정, 삭제) - 속성참조 - 파일을 통한 대규모 데이터 입출력 - 접속 · ID/PW방식, 인증서 접속방식, Globus 프록시 접속 방식, VOMS 프록시 - VO 세팅

사용자 요구사항을 반영하고, 일반 DBMS 관리 툴과 유사한 GUI 인터페이스를 제공하는 범용적 AMGA 툴킷이 필요하였다. 그리드 사용자들의 데스크탑, 랩탑 OS는 윈도우즈, 리눅스, 맥의 사용 비율이 비슷하여 이

를 모두 수용할 수 있게 세가지 형태의 툴킷으로 배포가 요구되었다. 또한 구현에 앞서 AMGA 버전별 기능을 어떻게 지원할 것인지를 고려하는 것이 중요하다. 주요 기능들이 AMGA 버전 1.9 이후에 보완 되었지만 그 이전의 버전들도 많이 배포되어 사용되고 있기에 이를 반영한 개발이 필요하다. 현재 AMGA의 버전은 2.1로써 사용자 편의성을 위해 1.9버전 이후는 Native SQL을 수용하고 있지만, 과거 버전을 이용하는 사용자들을 위해 AMGA Manager의 주요 기능을 AMGA Query 중심으로 설계하는 것도 필요하다. [표 3]은 요구사항 분석을 요약한 표로 내용과 같은 기능위주의 카테고리 분류할 수 있다. [그림 2]는 시스템 사용 주체 및 연계 시스템간의 액터를 정의한 그림이며 외부 시스템(External Systems)중 LFC(LCG File Catalogue)와 스토리지 요소는 연계성이 적은 관계로 향후 확장 시 개발을 고려하기로 하였다.

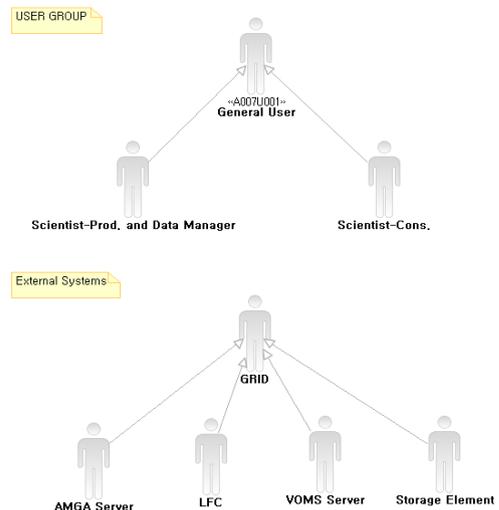


그림 2. 액터정의(사용자 그룹/외부 시스템)

2. 개념설계

AMGA에서 메타데이터는 데이터를 위한 데이터로써 실제 실험 데이터는 파일형태로 이루어지고 본 틀은 메타데이터 카탈로그를 사용자 편의 중심으로 설계했다. 사용자는 자신의 PC환경(MAC/Window/Linux)에

1 <http://amga.ct.infn.it/support/index.php>

구에 받지 않는 클라이언트용으로 설계되었고, 그 아키텍처는 아래 [그림 3]과 같다. 메타데이터 카탈로그 GUI 툴은 그리드 API를 이용하여 AMGA의 프론트 엔드 중 TCP/IP와 연동하여 데이터에 접근한다[16].

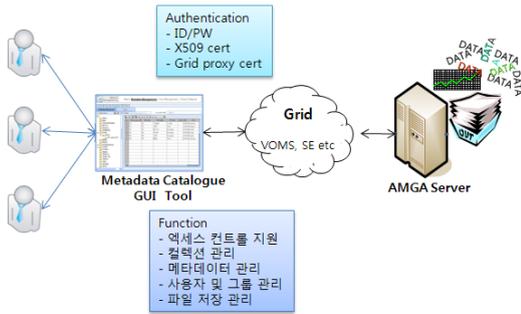


그림 3. 메타데이터 카탈로그 관리 개념도

3. 유스케이스 다이어그램

AMGA Manager의 핵심적인 작업은 컬렉션, 메타데이터, AMGA 커맨드를 사용자가 쉽게 관리하고 직관적으로 AMGA 결과를 확인하는 것을 들 수 있다. 유스케이스 다이어그램은 주요 기능을 컴포넌트 중심으로 이용하고 이후의 모델링에 대한 기본적인 정보와 방향을 표시하는 데 활용할 수 있다.

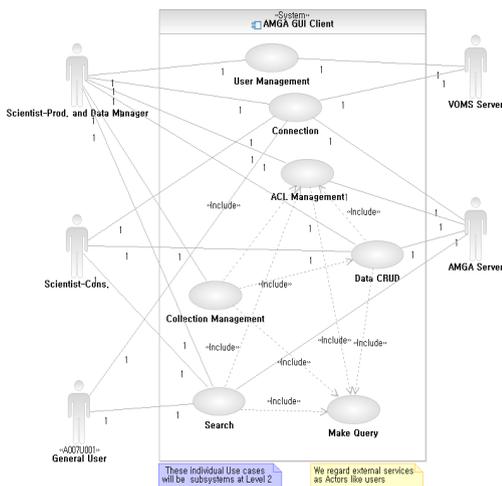


그림 4. Usecase Diagram; AMGA GUI Client (Level 1)

[그림 4]의 유스케이스 다이어그램은 하향식 모델(Top-down model)로 1레벨(1L) 유스케이스를 정의하고 2레벨(2L) 유스케이스로 검증하는 방식으로 명세화하였다. AMGA Manager 기능을 중심으로 크게 사용자 관리(User Management), 접속(Connection), 액세스 컨트롤 관리(ACL Management), 데이터 관리(Data CRUD), 컬렉션 관리(Collection Management), 검색(Search), 그리고 쿼리 생성(Make Query) 1L 유스케이스로 구분된다. 유스케이스 다이어그램은 유스케이스들 간 또는 액터와 유스케이스간의 상호작용을 표현하고 유스케이스간의 선처리 또는 포함되는 기능을 다이어그램에 시각적으로 표현할 수 있다. 또한 개발에 필요한 간단한 명료한 정보만을 유지하고, 추가적인 정보는 별도로 기술하여 유스케이스 다이어그램의 표현을 보완했다. 그리고 [그림 5]의 2L 유스케이스 다이어그램은 1L 유스케이스 중 Data CRUD (Create/Read/Update/Delete)를 세부 컴포넌트로 상세화한 예제 그림이다.

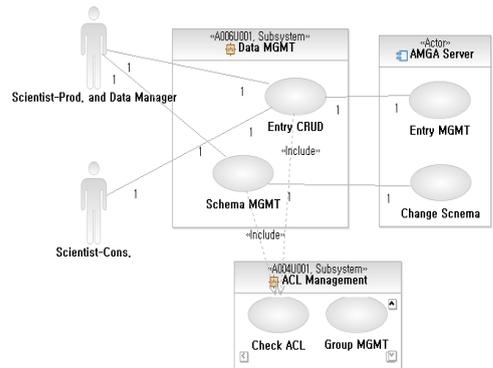


그림 5. Usecase Diagram; Data CRUD (Level 2)

4. 시퀀스 다이어그램

앞서 언급한 유스케이스 7개에 대해 세부 동작 원리를 타임도메인에서 시퀀스 다이어그램으로 기술하였다. [그림 6]은 ACL Management에 대한 시퀀스 다이어그램으로 그리드 사용자가 자신의 AMGA에서 관리되는 액세스 컨트롤에 의해 메타데이터에 대한 권한을 부여 받는 상호작용에 대한 설명하고 있다.

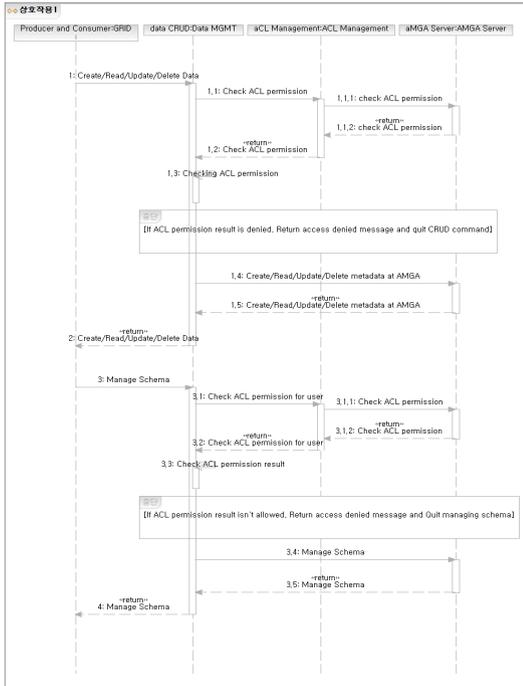


그림 6. Sequence Diagram; ACL Management

5. 클래스 다이어그램

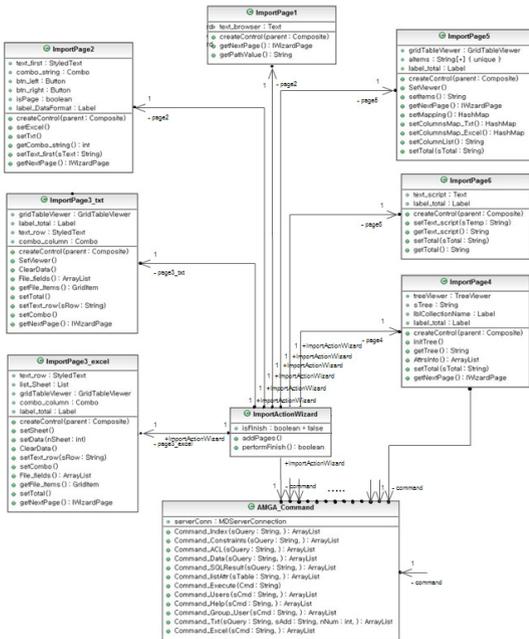


그림 7. 클래스 다이어그램 : Data Import Wizard

[그림 7]은 AMGA Manager의 대량의 메타데이터 파일 입력 인터페이스인 Data Import 마법사의 클래스 다이어그램을 보여준다. Data Import 마법사 단계별로 클래스를 구성하고 있으며 각 클래스들은 자신의 속성과 메소드를 포함하고 있으며 상호 연관관계(Relationship)를 정의한다. 클래스명 ImportPage3은 입력파일 형태에 따라 엑셀과 텍스트 특성을 고려하여 데이터 파싱을 준비한다.

6. 프로세스 보완

효과적인 모니터링 인터페이스를 구현하기 위해 기본 설계에서 도출하지 못했던 사안에 대한 보완으로 플로우 차트를 통해 인터페이스를 고도화 하는 작업을 병행하였다. [그림 8]은 SQL Editor와 연동되어 결과보기를 수행하는 프로세스로서 다양한 질의에 대한 결과 패턴을 분석하여 결과 컬럼의 개수별 데이터 그리드를 제공하는 플로우 차트이다. 총 140여개의 커맨드에 대한 결과를 다음과 같이 세가지 형태로 분류하여 처리하였다.

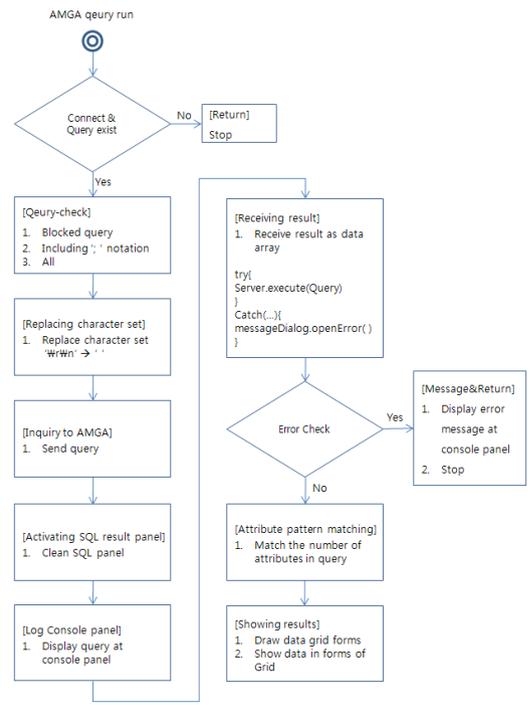


그림 8. AMGA 쿼리 실행 및 결과보기 Flow Chart

단순 질의 결과를 나타내는 1컬럼 데이터 그리드, Is-P 또는 Is-N과 같은 리스트별 정보를 제공하는 2컬럼 데이터 그리드 그리고 select 형식의 다이나믹 어트리뷰트 필드를 포함하는 데이터 그리드 형식으로 AMGA 질의 결과는 분류하여, AMGA 서버와의 접속상태 확인하고 쿼리를 전송하고 매칭된 컬럼 형태로 데이터 그리드 결과뷰를 구성하여 결과값을 보여준다.

7. 아키텍처

[그림 9]은 AMGA GUI Client 아키텍처에서 주요 클라이언트 컴포넌트로 인증 관리(Authentication Manager), 응용프로그램 관리(Application Manager), 스키마 관리(Schema Manager), 권한관리(Privilege Manager), SQL 에디터(SQL Editor), 결과 보기(Result View) 모듈로 구성된다. 인증 관리는 AMGA 서비스 접속을 위해 username/password 접속/관리와 인증서(Certificate)의 등록/관리, 그리드 프록시(VOMS Proxy) 인증 등록/관리를 위한 모듈이고 응용프로그램 관리는 GUI의 공통 컴포넌트를 관리하기위한 모듈로써 SQL 에디터의 쿼리 생성 지원, 대용량의 데이터 입/출력 지원, 코드 생성 등을 수행한다. 스키마 관리는 메타데이터, 스키마 속성, 테이블별 액세스 컨트롤, 제약조건 및 인덱스의 생성, 변경, 삭제 및 조회를 할 수 있는 모듈이고, 권한 관리는 디렉토리 및 파일에 대한 권한 관리 및 사용자 그룹/사용자의 접근을 제어한다.

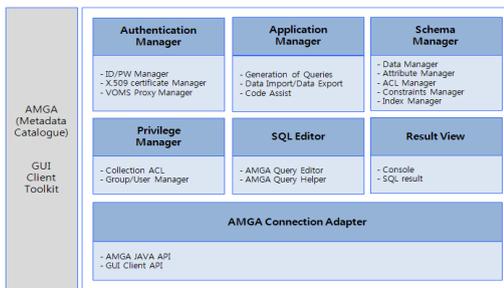


그림 9. AMGA GUI Client 아키텍처

SQL 에디터는 AMGA 쿼리의 생성 및 실행을 수행하고 AMGA 쿼리를 생성하도록 도와준다. 결과보기는

사용자 GUI에서 생성되는 모든 쿼리 및 에러 모니터링 기능을 제공하고 SQL 에디터에서 실행되는 쿼리들의 결과를 볼 수 있는 모듈이다. AMGA 접속 어댑터는 AMGA 서비스의 이용을 위해 AMGA 사이트에서 제공하는 AMGA JAVA API를 기본으로 다양한 질의 및 결과를 제공받을 수 있는 API를 가지고 있다.

8. 화면설계

AMGA Manager GUI 디자인 룰을 정의하기에 앞서 일반적인 디자인 가이드라인을 살펴보면, 화면배치는 시선이나 신체를 크게 변동시키지 않고 화면을 볼 수 있도록 배치하고, 자주 사용하고 비중이 높은 화면일수록 정면에 배치한다. 또 정보의 시간, 공간적 흐름은 왼쪽 위로부터 오른쪽 밑을 향하게 하고, 표현 형식의 열의 숫자가 많지 않도록 하며, 메뉴 버튼은 고른 분포가 되도록 하여 선택시간이 적게 소요되도록 한다[18]. AMGA Manager도 앞서 언급한 디자인 룰을 모두 고려하여 화면설계를 하였고, 조작이 많은 메뉴항목은 틀바에 별도로 배치하도록 하였다. AMGA Manager의 특성상 데이터를 조작하는 GUI는 일반 RDBMS의 GUI 툴(SQLGate, Toad, phpMyAdmin)을 참조하였지만, 컬렉션 트리 관리와 스키마 관리를 같이 하는 측면에서 윈도우 탐색기 특성을 반영하여 화면설계를 하였다. 또한 사용빈도가 높은 조작은 단축키(shortcut key)를 두어 사용자가 보다 편리하게 GUI를 사용할 수 있도록 하였다.

IV. 구현

AMGA 사용자가 쉽고 편리하게 접근하고 다양한 요구사항을 빠르게 반영하도록 AMGA Manager 개발에 컴포넌트 재사용성이 높은 이클립스 RCP 툴을 이용하였다. 본 개발 툴은 AMGA Manager 구현을 위한 기본 모델과 워크스페이스 개념의 빠른 시각적 표현에 대한 기본 에디터와 같은 많은 핵심 기능을 가지고 있으며, 이클립스 플러그인 활용으로 쉽게 프로그램을 구성하고 확장할 수 있어 양질의 사용자 친화적 GUI 제공이

가능하고 개발 주기를 단축시킬 수 있다. [표 4]는 AMGA GUI 클라이언트 개발에 앞서 서버 환경, 클라이언트 개발 환경, 클라이언트 테스트 환경을 설명하고 있다. AMGA의 백엔드 데이터베이스는 Postgresql 뿐만 아니라 Oracle, MySql, SQLite로 구성될 수 있다. 개발 툴킷인 이클립스는 Java 기반이기 때문에 JDK(Java Development Kit)를 설치 사용하였다. 또한 개발되는 AMGA Manager는 다양한 OS에서 범용으로 사용되기 때문에 Mac, Linux, Windows로 배포된다. 따라서 테스트도 각각의 OS별로 이루어졌다.

표 4. 개발 환경구성표

서버환경	OS	SLC4
	Backend DB	Postgresql
	ODBC	psqlodbc-08.03.0200
	AMGA	AMGA 2.0 (서버/클라이언트)
클라이언트 개발 환경	OS	Windows 7
	Development toolkit	Eclipse for RCP and RAP Developers
	Java	JDK 1.6 for Windows 64bits
클라이언트 테스트 환경	OS	Mac/Debian/Windows XP/7
	Java	JDK 1.6 for Linux(32bit/64bits)/Windows(32bit/64bits)
	AMGA Client	AMGA Manager for Mac/Linux(32bit/64bits)/Windows(32bit/64bits)

AMGA API는 glite-amga-api-java-v1.3.03.jar, bcprov-jdk15-135.jar, commons-cli-1.0.jar를 개발 소스에 임포트하여 클래스로 드러나는 high level 인터페이스 및 명령어를 직접 서버로 전송하는 low level 인터페이스로 사용하였다.

메인 화면은 AMGA의 이용 시 가장 활용도가 높은 컬렉션과 애트리뷰트를 포함한 메타데이터를 조작할 수 있도록 구현하였고, AMGA와의 상호작용에 대한 인터페이스로써 콘솔, SQL 결과 판넬을 구성하여 진행 및 결과를 확인할 수 있게 하였다.

AMGA 서비스에 접속하기 위해서는 앞서 설명한대로 4가지 방식의 접속 지원이 필요하다. ID/PW방식, 인증서 접속방식, Globus Proxy 접속 방식, VOMS Proxy

를 수용해야 하는데, 본 구현에서는 gLite 공식 인증방식을 수용하고 있고, 그리드 인프라에 접근하는 통합 워크벤치 프레임워크를 가지는 g-Eclipse의 인증 플러그인(Plug-in) 소스를 g-Eclipse 사이트²에서 제공받아 AMGA 접속 모듈을 개발하였다[19]. 설계에서부터 g-Eclipse의 재사용을 고려하여, 중복 개발을 배제한 그리드 컴포넌트의 재사용을 지향하였다.

표 5. 메뉴구성표

File	New Connection		
	End Connection		
	Change Password		
	Exit		
Tools	Collection	Make Collection Drop Collection	
	Federation Manager		
	Schema Browser	Data	Attributes ACL Constraints Index
		SQL Editor	
		Data Import	
		Data Export	
		Data Rollback	
	Group/User management		
	Site management		
	Help	Program help	
		AMGA Homepage	
About AMGA Manager			

[표 5]는 AMGA Manager의 메뉴구성표를 보여준다. 주요 컨트롤 브라우저라고 할 수 있는 스키마 브라우저는 데이터, 액세스 컨트롤, 컨스트레인츠, 인덱스를 생성, 변경, 삭제할 수 있도록 고안된 인터페이스로 메타데이터 카탈로그 특성 상 컬렉션 뷰의 컬렉션을 트리 구조로 연동하여 조작할 수 있고, 해당 애트리뷰트의 속성을 항상 확인하게 함으로써 메타데이터 카탈로그의 활용성을 높이도록 구현하였다. 또한 상세한 메타데이터 카탈로그를 조작할 수 있도록 SQL Editor 브라우저의 기능 중 다양한 help기능을 제공하여 사용자가 쉽게 다양한 AMGA 쿼리를 생성할 수 있도록 하였다. [그림 10]은 g-Eclipse plug-in 오픈소스를 활용해서 개발한 AMGA 액세스 프로세스로써 그리드 인증 기능을

² <http://dev.eclipse.org/svnroot/technology/eu.geclipse>

제공하고 이 모듈로 AMGA 접속이 완료되면 [그림 11]과 같은 브라우저 사용이 가능하다.

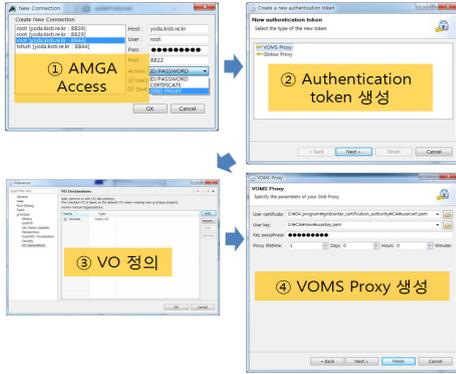


그림 10. g-Eclipse VOMS 프록시 인증 모듈을 활용한 AMGA 액세스 프로세스

있다. 데이터 스키마 브라우저에는 반복적 사용에 강력한 기능인 필터 기능과 데이터의 정렬 기능으로 편의성을 높였고, 대량의 데이터를 스트리밍 형태로 보여줄 수 있도록 구현했다.

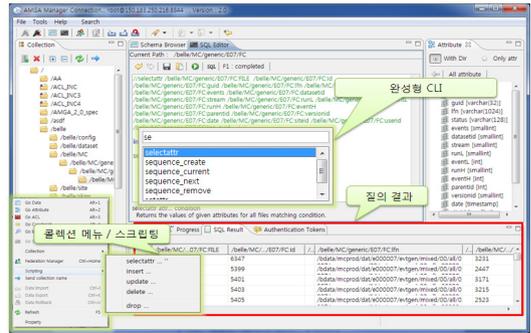


그림 12. AMGA GUI Client toolkit : SQL Editor

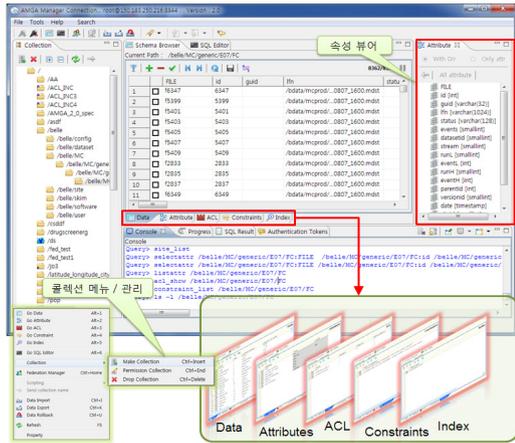


그림 11. AMGA GUI Client toolkit : Schema Browser

[그림 11]는 스키마브라우저에 메타데이터를 컬렉션 뷰와 연계하여 관리 및 모니터링할 수 있는 AMGA Manager의 메인화면으로 콘솔을 통해 서버와의 상호작용을 모니터링할 수 있다. 콜렉션 메뉴를 통해 사용자는 쉽게 콜렉션 생성, 수정이 가능하며, 대량의 데이터에 대해서는 Data Import/Export 마법사를 통해 쉽게 처리가 가능하다. 스키마 브라우저는 5개의 탭으로 구성되어 데이터, 어트리뷰트, 액세스 컨트롤, 컨스트레이크인즈, 인덱스의 관리를 유사한 인터페이스로 조작할 수

[그림 12]은 AMGA 서비스를 사용자가 다양한 도움 기능 이용으로 쉽게 쿼리를 만들고 실행할 수 있는 SQL Editor 인터페이스로서 결과 뷰(Result view)와 연계하여 효과적으로 사용할 수 있다. 사용자는 에디터 옆에 있는 속성을 참고하여 질의를 만들며, 완성형 커맨드라인 인터페이스(Completed CLI)를 통해 쉽게 명령어를 만들 수 있다. 이때 프레임 아래쪽에는 해당 명령어에 대한 설명과 문법을 제공하여 사용자가 명령어를 완성하는데 도움을 준다. 또한 메뉴에 있는 스크립팅 기능으로 기본적인 관리 명령어는 값만 넣으면 완성할 수 있도록 쿼리를 제공하여 준다.

V. 결론 및 기대효과

1. AMGA Manager 결론

본 논문에서는 UML 모델링 도구를 이용하여 AMGA GUI client인 AMGA Manager를 분석/설계하고 구현하였다. UML 기반의 CASE 도구를 이용함으로써 체계적으로 분석하고 설계를 할 수 있었으며, UML 모델을 기반으로 부가적인 명세화 및 알고리즘 보안을 병행하여 소프트웨어를 구현하였고 주요기능은 [표 6]

와 같다. AMGA 명령어 체계는 리눅스의 명령어를 포함하고 있고 AMGA 고유의 SQL 및 Native SQL를 동시에 사용하고 있어, AMGA Manager에서는 이런 다양한 명령어를 쉽게 GUI로 사용할 수 있도록 하였다.

본 소프트웨어는 상용 RDBMS 관리 툴인 Toad, SQLGate 등과 데이터 관리 측면에서 보여지는 화면은 유사하나 AMGA의 특성이 반영되어 그리드 인증방식을 지원, 복제(Replication) 및 병합(Federation) 수행, 계층적 데이터 컬렉션 구조(Directory-like structure)를 직관적인 인터페이스로 제공했다는 점에서 차이를 보인다.

표 6. 주요 기능

구분	내용
접속	- ID/PW, Globus 프록시, VOMS 프록시, 인증서 인증
컬렉션 관리	- 컬렉션 생성 마법사(Wizard), 삭제, 속성보기 - Permission/Owner 관리
스키마 브라우저	- Data, Attribute, Constraints, Index, ACL 관리(탭으로 구성) - Data 필터, 정렬 - 데이터 스트림 형식의 뷰어
SQL 에디터	- 명령어 도움말(설명, 사용법) - 기본 스크립트 자동생성 - 완성형 CLI(Command Line Interface) - 명령어 실행, Undo/Redo - 멀티쿼리 지원 및 비활성 지원
뷰어	- 모니터링 콘솔 - 결과 뷰어 - Progress - Authentication 토큰
메뉴	- 사이트 관리 - 사용자 관리 - Federation 마운트 - 데이터 Import/Export/Rollback 마법사

2. 기대효과

현재 한국 KISTI 주도로 개발되고 있는 AMGA는 그리드에서 사실상 표준(de facto standard) 메타데이터 카탈로그로 인정받고 있다[16][17]. 세계 유수 그리드 환경 기반의 프로젝트 및 IT환경에서 대량의 데이터 관리에 활용되고 있지만 사용자 인터페이스 사용이 다소 어려운 점도 있었다. 하지만 AMGA Manager의 사용자 친화적 인터페이스를 제공함으로써 사용자는 메타데이터 카탈로그 관리가 보다 용이할 것이다. 또한

AMGA Manager는 아래와 같이 몇 가지 기대효과를 갖는다.

첫째, 시스템 명령어, 그리드 명령어, 쿼리 등에 비전문적인 과학기술연구자에게 데이터 활용을 극대화하여 연구 생산력을 높일 수 있도록 지원해준다. 둘째, 사용빈도가 높은 데이터 입/출력 및 검색을 별도의 SQL 명령어 없이 빠르고 쉽게 할 수 있도록 지원해 준다. 셋째, AMGA 메타데이터 카탈로그 서비스 이용을 활성화 할 수 있다. 향후 사용자의 요구사항에 대한 지속적이고 즉각적인 유지보수로 EMI 데이터 영역에서 주요 컴포넌트로 자리 매김할 수 있을 것이다.

참고 문헌

- [1] <http://public.eu-egee.org/>
- [2] <http://glite.web.cern.ch/glite/>
- [3] Ignacio Blanquer, Vicente Hernández, José Salavert, Damià Segrelles, "Integrating TRENCADIS Components in gLite to Share DICOM Medical Images and Structured Reports," Stud Health Technol Inform 2010, Vol.159, pp.64-75, 2010.
- [4] S. Ahn, N. Kim, and S. Lee, "Performance analysis and optimization of AMGA for the large-scale virtual screening," Software, practice and experience, Vol.12, pp.1055-1072, 2009.
- [5] M. Diarena, S. Nowak, and J. Y. Boire, "HOPE, an Open Platform for Medical Data Management on the Grid," Stud Health Technol Inform, Vol.138, pp.34-38, 2008.
- [6] J. P. Baud and S. Lemaitre, "The LHC File Catalogue (LFC)," HEPIX, Karlsruhe, Germany, 2005.
- [7] S. Ahn, K. Cho, S. Hwang, J. Kim, H. Jang, B. K. Kim, H. Yoon, and J. Yu, "Design of the Advanced Metadata Service System with AMGA for the Belle II Experiment," Journal of the

Korean Physical Society, Vol.57, No.4, pp.715-724, 2010.

[8] J. H. Kim, S. Ahn, K. Cho, M. Bracko, Z. Drasal, T. Fifield, R. Frühwirth, R. Grzymkowski, T. Hara, M. Heck, S. Hwang, Y. Iida, Ryosuke Itoh, G. Iwai, H. Jang, Nobu Katayama, Y. Kawai, Christian Kiesling, B. K. Kim, and T. Kuhr, "The advanced data searching system with AMGA at the Belle II experiment," Computer Physics Communications, Vol.182, pp.270-273, 2011.

[9] <http://www.eclipse.com>

[10] H. Heinecke, M. Rudorfer, P. Hoser, C. Ainhauser, and O. Scheickl, "Enabling of AUTOSAR system design using Eclipse-based tooling", ERTS2008, Toulouse, 2008.

[11] <http://amga.web.cern.ch>

[12] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, "Simple object access protocol (SOAP) 1.1," World Wide Web Consortium, Note NOTE-SOAP-20000508, 2000(5).

[13] K. G. Begeman, A. N. Belikov, D. R. Boxhoorn, F. Dijkstra, E. A. Valentijn, W. J. Vriend, and Z. Zhao, "Merging Grid Technologies Astro-WISE and EGEE," J Grid Computing, Vol.8, No.2, pp.199-221, 2010.

[14] <http://ganga.web.cern.ch/ganga>

[15] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, Á. Frohner, A. Gianoli, K. Lörentey, and F. Spataro, "VOMS, an Authorization System for Virtual Organizations," Lecture Notes in Computer Science, Vol.2970, Berlin, 2004.

[16] 허태상, 안선일, 김한기, 이상도, 유상수, 황순욱, "메타데이터 카탈로그 GUI 툴 설계", 2010 한국 컴퓨터종합학술대회 논문집, 제37권, 제1(D)호, pp.155-158, 2010.

[17] N. Santos and B. Koblitz, "Metadata services on the Grid," Advanced Computing and Analysis Techniques, Vol.559, pp.53-56, 2005.

[18] 봉종수, 임현석, 김세영, 김대진, "HDTV 수신 모듈용 GUI 설계 및 구현", 한국콘텐츠학회논문지, 제9권, 제4호, pp.72-80, 2011.

[19] T. S. Huh, G. C. Park, and S. W. Hwang, "Implementation of AMGA Manager Using an Authentication Module on the g-Eclipse," ICC2011, Vol.9, No.2, pp.227-228, 2011.

저 자 소 개

허 태 상(Taesang Huh)

정회원



- 2000년 : 성균관대학교 전기전자컴퓨터공학부(공학사)
- 2002년 : 광주과학기술원 정보통신공학(공학석사)
- 2002년 ~ 현재 : 한국과학기술정보연구원 선임연구원

<관심분야> : 메타데이터 카탈로그, 분산 컴퓨팅, e-Science, 정보검색 등

황 순 욱(Soonwook Hwang)

정회원



- 1990년 : 서울대학교 수학과(이학학사)
- 1995년 : 서울대학교 계산통계학과(이학석사)
- 2003년 : 미국 남가주대학교 전산과학과(이학박사)

• 2003년 ~ 2006년 : 일본 국립정보학연구소 연구원
 • 2006년 ~ 현재 : 한국과학기술정보연구원 책임연구원
 <관심분야> : 그리드 컴퓨팅, 클라우드 컴퓨팅, 분산 컴퓨팅 등

박 근 철(Geunchul Park)

정회원



- 1998년 : 중앙대학교 컴퓨터공학과(공학사)
- 2000년 : 중앙대학교 컴퓨터공학과(공학석사)
- 2006년 ~ 현재 : 한국과학기술정보연구원 선임연구원

<관심분야> : 메타데이터 카탈로그, 그리드 컴퓨팅, e-Science, 데이터베이스 등