ISBN:
978-89-6211-302-0

WSRF 기반 그리드 작업관리서비스 기술 지원 보고서

2008. 12.

한국과학기술정보연구원

차 례

1. 배경 및 목표

2. 기술 지원 내용

2.1 KMI-R2 소개

2.2 서비스 개발 프레임워크 소개

2.3 작업관리서비스 구현 설계

2.4 핵심 모듈별 세부 서비스 구현

2.4.1 작업 제출

2.4.2 작업 모니터링

3. 결론

첨부1: DB 테이블

1. 배경 및 목표

1.1 배경

o 씽크테크, 이노그리드 등의 국내 그리드 기업은 사용자들이 시간과 장소에 제약없이 편리하게 분산된 이기종의 컴퓨팅 자원을 공동 활용하기 위한 그리드 서비스 포털의 개발을 계획하고 있음

o 한국과학기술정보연구원(이하 KISTI)는 컴퓨터, 네트워크, DB 등 이기종 IT 자원의 통합 관련 신기술인 그리드 기술의 선도 연구 기관으로서 수년간 그리드 미들웨어 기술 연구 및 개발을 수행하고 있음

o 본 기술 지원 보고서는 국내 그리드 기업이 그리드 서비스 포털을 개발하는데 있어서 핵심 미들웨어 요소인 작업관리서비스와 관련하여 KISTI가 보유하고 있는 “WSRF 기반 그리드 작업관리서비스” 기술의 유용성을 검토하여 기술 연구 및 개발에 참고할 수 있도록 하기 위해서 작성됨

1.2 목표

o TIGRIS 그리드 인프라를 위한 KMI-R2 그리드 미들웨어 서비스 패키지 소개

o WSRF 기반 그리드 작업관리서비스의 서비스 구현 설계 소개

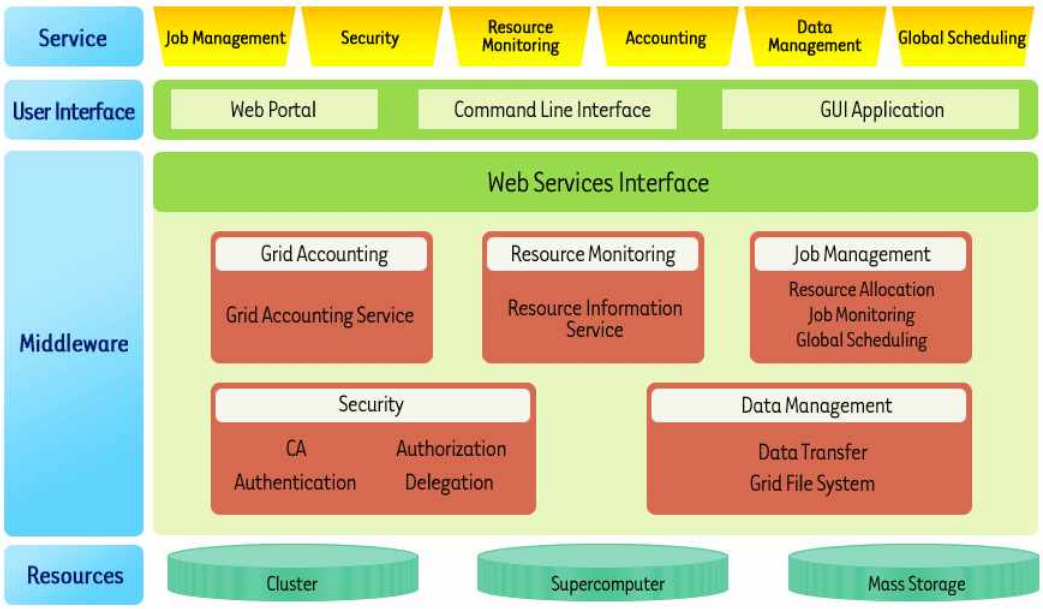
o WSRF 기반 그리드 작업관리서비스의 핵심 모듈별 세부 서비스 구현 소개

2. 기술 지원 내용

2.1. KMI-R2 소개

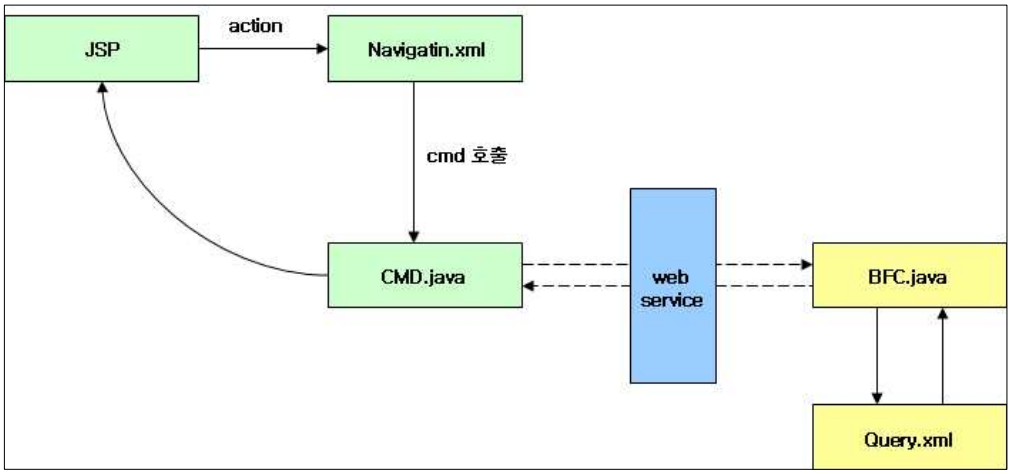
o 국가 그리드 인프라인 TIGRIS(Tera-scale Infrastructure for k*GRId Services)를 위해 개발된 그리드 미들웨어 서비스 패키지
- Globus Toolkit 4 기반의 WSRF(Web Service Resource Framework) 서비스로서 개발

o Globus Toolkit 4의 프리미티브 서비스를 확장하여 핵심 미들웨어 서비스 개발
- 작업 관리 서비스: 단순 작업, HTC(High-Throughput Computing) 작업, MPI 작업 처리 및 모니터링 서비스
- 파일 관리 서비스: 원격 사이트간 파일 전송 및 삼자 전송(Third-party transfer) 지원
- 정보 서비스: 다양한 형태의 그리드 자원에 대한 자원 정보 제공
- 어카운팅 서비스: OGF UR-WG기반의 큐잉 시스템 사용량 어카운팅 정보 제공
- 그리드MPI 서비스: 그리드 환경에서의 MPI 작업 실행 서비스

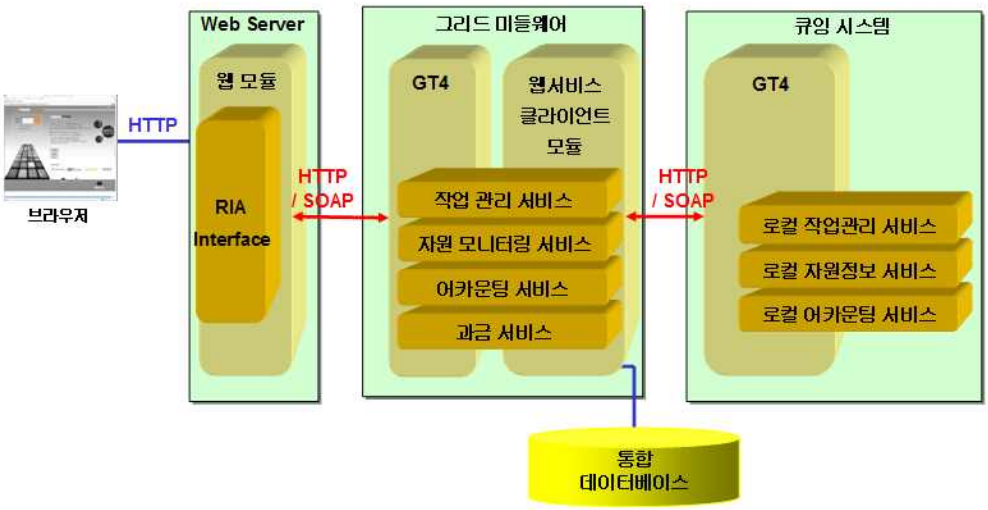


2.2 서비스 개발 프레임워크 소개

o 서비스 개발 프레임워크는 다음과 같은 흐름으로 구성된 처리 과정을 거침
- JSP: 서비스 요청 및 요청된 결과가 반영되는 페이지
- Navigation: 서비스 요청에 대한 주소정보 보관, 요청에 따라서 해당 서비스로 이동
- cmd: 서비스 요청을 실행, 데이터 관련한 BFC 호출
- BFC: 서비스 요청에 해당하는 내용을 DB에 접속해서 실제적인 데이터 입출력 실행
- Query: 서비스 실행을 위한 질의 보관

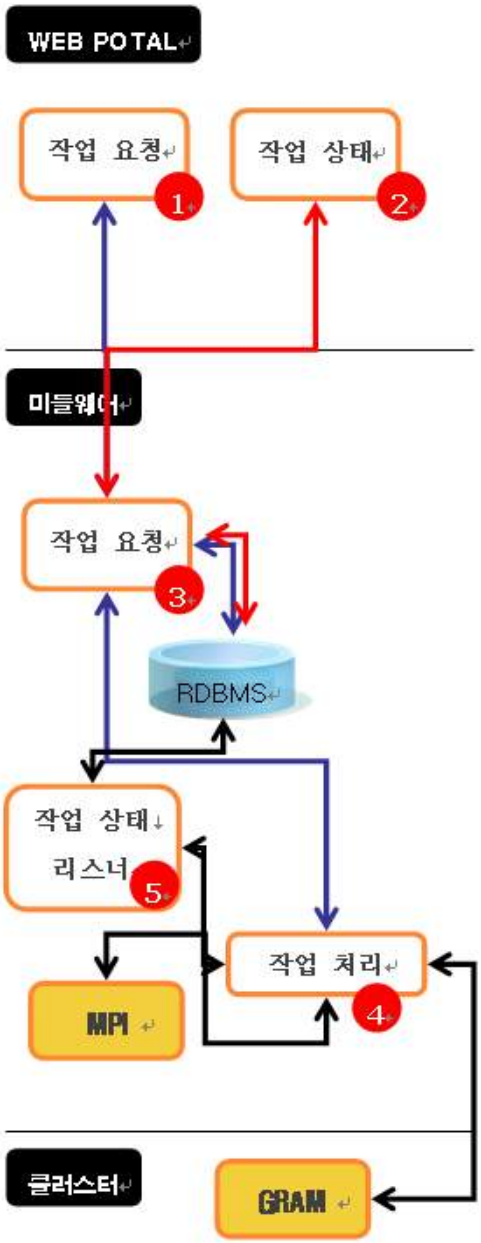


o 작업관리서비스 개발은 서비스 개발 프레임워크 내의 WSRF 기반 웹 서비스로 구현



2.3 작업관리서비스 구현 설계

o 작업관리서비스는 작업을 요청하고 작업 상태를 모니터링하는 웹포털의 호출을 통해서 미들웨어 단에서 실행되며, 실제적인 작업 실행은 클러스터에서 수행



① 작업 요청
- 작업 프로파일을 읽어서 작업 제출 가능
- 작업 내용을 입력받아 JSDL 형태로 변경하여 미들웨어의 작업 요청 접수에 접수

② 작업 상태 조회
- 작업 내역 조회 및 모니터링 확인

③ 작업 요청 접수
- 웹포털의 작업 요청을 접수 받아서 처리, 즉시 작업인 경우 작업 처리 프로세스에 요청

④ 작업 처리
- 파일 스테이징-인 작업 처리
- 작업 타입 확인: MPI 작업인 경우, 그리드 MPI 서비스 호출, 나머지의 경우, JSDL을 RSL로 변환하여 클러스터 GRAM 서비스 호출
- 작업 리스너 실행
- 작업 처리 내용을 DB에 저장

⑤ 작업 상태 리스너
- 작업 상태 변경시마다 DB 상태 변경 저장
- 작업 상태 완료; 파일 스테이징-아웃 처리 (작업 처리가 담당), 작업 상태 리스너 종료

2.4 핵심 모듈별 세부 서비스 구현

o 작업관리서비스는 기능별로는 작업 제출과 작업 모니터링으로 나뉘며, 서비스별로는 단순 작업 서비스, HTC 작업 서비스, 그리드 MPI 작업 서비스로 나뉨

2.4.1 작업 제출

1) 단순 작업 제출

(1) 기능 흐름

- 사용자는 프로젝트 목록을 가져올 수 있고, 단순 작업을 제출할 수 있는 화면을 볼 수 있다.
- 자원직접선택 버튼을 누르면 자원을 선택할 수 있는 화면을 팝업으로 보여준다.
- 실행파일에 파일설정 버튼을 누르면 파일을 선택 할 수 있는 화면을 팝업으로 보여준다.
- 추가정보 작성을 누르면 추가정보를 입력할 수 있는 화면을 보여준다.
- 입력 파일에서 추가를 누르면 입력 파일을 추가할 수 있는 박스가 추가로 생성된다.
- 입력 파일에서 파일설정을 누르면, 파일을 선택할 수 있는 화면을 팝업으로 보여준다.
- 출력 파일에서 추가를 누르면 출력 파일을 추가할 수 있는 박스가 추가로 생성된다.
- 출력 파일에서 파일설정을 누르면 파일을 선택할 수 있는 화면을 팝업으로 보여준다.

(2) 관련DB 테이블

[PROJECT_MT] [JOB_MT] [JOB_CONTENTS] [JOB_STATUS_HISTORY] [ARGUMENT_INFO] [INPUT_FILE_INFO] [OUTPUT_FILE_INFO] [SCHEDULE_INFO] [JOB_ACCT_HT]

(3) 프레임워크 경로

`com.kisti.web.jobMng.JobManagerCMD.requestForJob()`

com.kisti.web.jobMng.JobManagerCMD.selectForCluster()
com.kisti.web.fileMng.FileMngCMD.startApplet()

다음의 웹서비스를 통해 프로젝트목록을 조회하여 가져온다.

bfcService = "com.kisti.middle.bfc.projectmanager.ProjectManagerBFC, selectForProject()"
bfcService = "com.kisti.middle.bfc.monitoring.MonitoringBFC, selectForClusterInfo()"

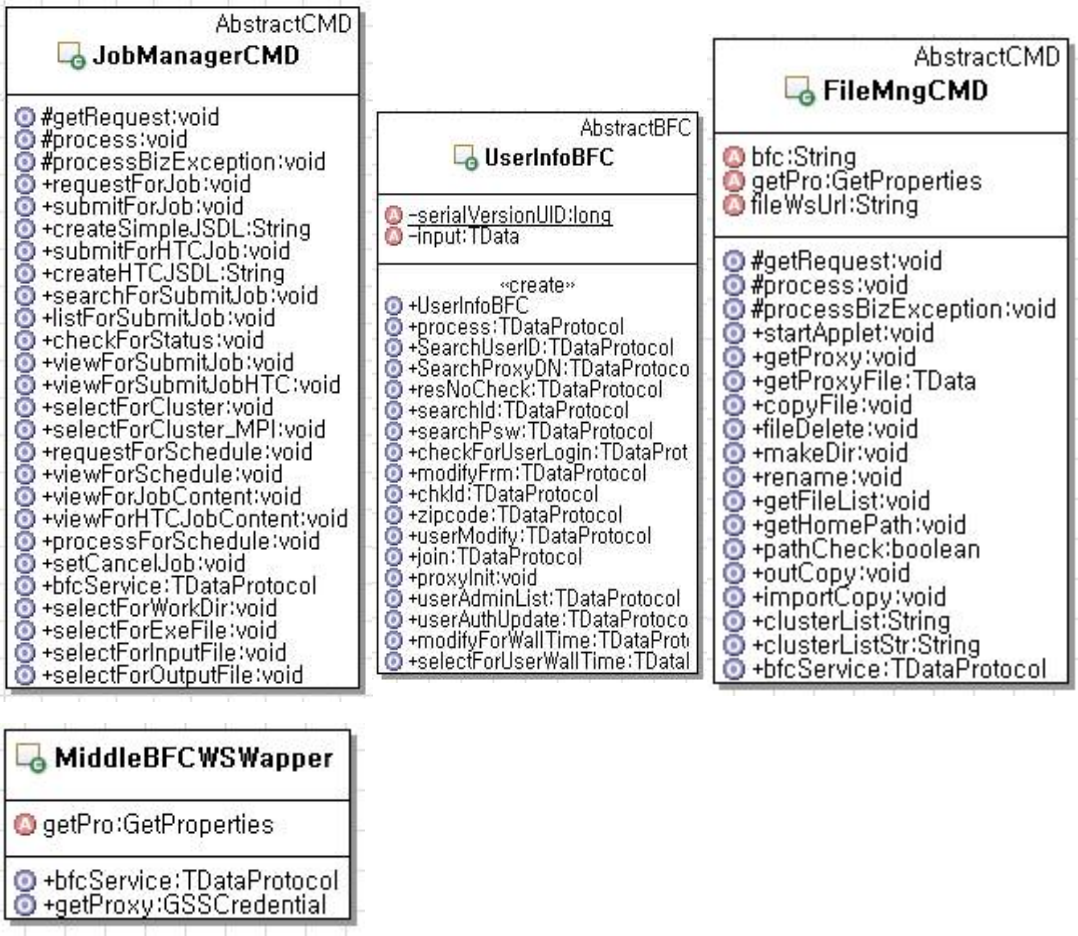
(4) 관련 JSP

- /jobManager/jobRequest/simple/requestForJob.jsp
- /jobManager/exeFile/FlexExeFile.jsp
- /jobManager/input/FlexJobFile.jsp
- /jobManager/output/FlexOutFile.jsp
- /jobManager/workDir/FlexDir.jsp
- /jobManager/jobRequest/common/selectForCluster.jsp, requestForSchedule.jsp

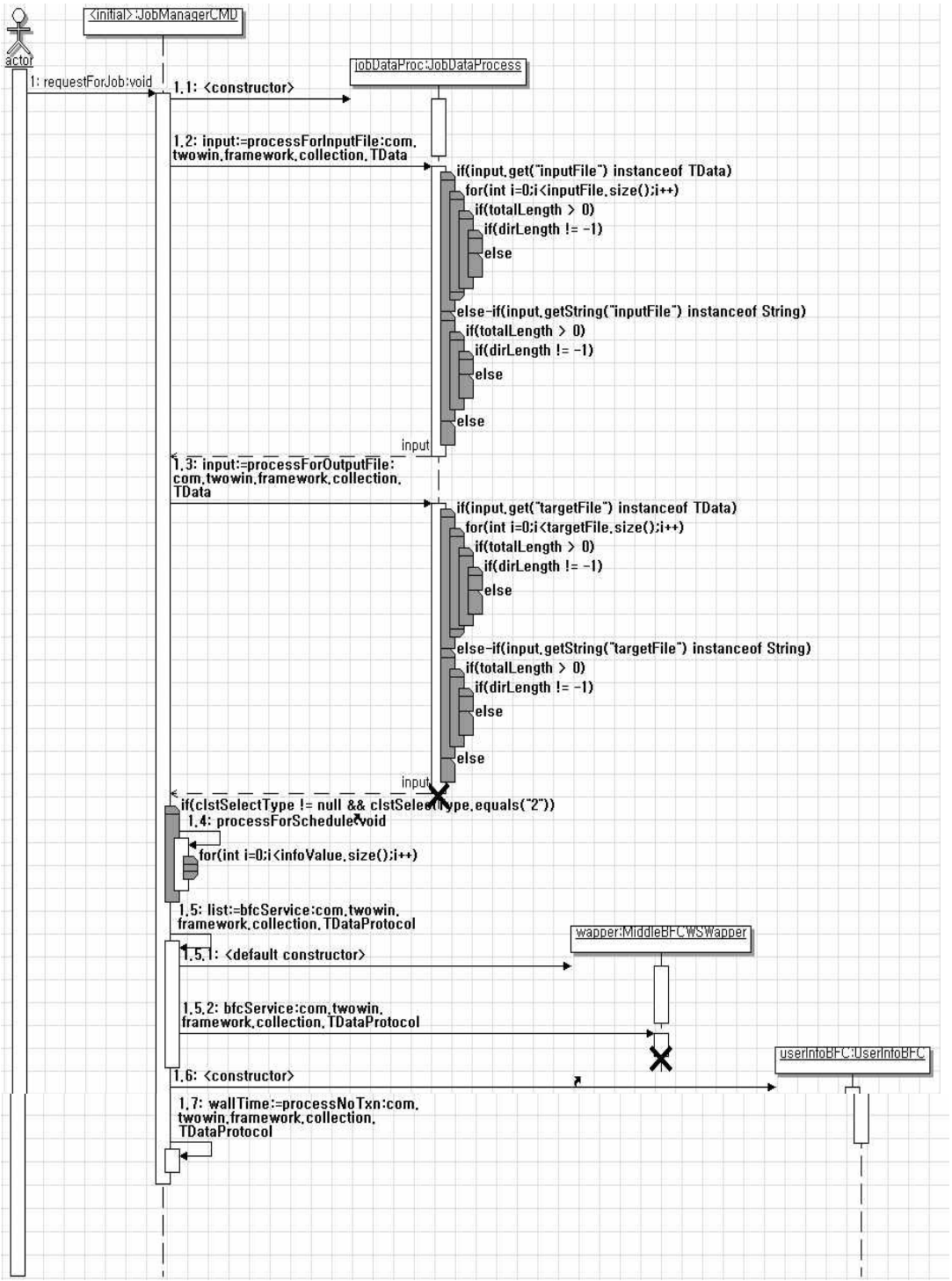
(5) Control 정보(forward.cfg)

WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" requestForSimpleJob"
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" selectForCluster"
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" selectForQueueInfo"
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" selectForQueueInfo"

(6) 클래스 다이어그램


AbstractCMD
JobManagerCMD
#getRequest:void
#process:void
#processBizException:void
+requestForJob:void
+submitForJob:void
+createSimpleJSDL:String
+submitForHTCJob:void
+createHTCJSDL:String
+searchForSubmitJob:void
+listForSubmitJob:void
+checkForStatus:void
+viewForSubmitJob:void
+viewForSubmitJobHTC:void
+selectForCluster:void
+selectForCluster_MPI:void
+requestForSchedule:void
+viewForSchedule:void
+viewForJobContent:void
+viewForHTCJobContent:void
+processForSchedule:void
+setCancelJob:void
+bfcService:TDataProtocol
+selectForWorkDir:void
+selectForExeFile:void
+selectForInputFile:void
+selectForOutputFile:void
AbstractBFC
UserInfoBFC
-serialVersionUID:long
-input:TData
«create»
+UserInfoBFC
+process:TDataProtocol
+SearchUserID:TDataProtocol
+SearchProxyDN:TDataProtoco
+resNoCheck:TDataProtocol
+searchId:TDataProtocol
+searchPsw:TDataProtocol
+checkForUserLogin:TDataProt
+modifyFrm:TDataProtocol
+chkId:TDataProtocol
+zipcode:TDataProtocol
+userModify:TDataProtocol
+join:TDataProtocol
+proxyInit:void
+userAdminList:TDataProtocol
+userAuthUpdate:TDataProtoco
+modifyForWallTime:TDataProt
+selectForUserWallTime:TDatal
AbstractCMD
FileMngCMD
bfc:String
getPro:GetProperties
fileWsUrl:String
#getRequest:void
#process:void
#processBizException:void
+startApplet:void
+getProxy:void
+getProxyFile:TData
+copyFile:void
+fileDelete:void
+makeDir:void
+rename:void
+getFileList:void
+getHomePath:void
+pathCheck:boolean
+outCopy:void
+importCopy:void
+clusterList:String
+clusterListStr:String
+bfcService:TDataProtocol
MiddleBFCWSWapper
getPro:GetProperties
+bfcService:TDataProtocol
+getProxy:GSSCredential

(7) 시퀀스 다이어그램


actor
<initial>:JobManagerCMD
jobDataProc:JobDataProcess
1: requestForJob:void
1.1: <constructor>
1.2: input:=processForInputFile:com.twowin.framework.collection.TData
if(input.get("inputFile") instanceof TData)
for(int i=0;i<inputFile.size();i++)
if(totalLength > 0)
if(dirLength != -1)
else
else-if(input.getString("inputFile") instanceof String)
if(totalLength > 0)
if(dirLength != -1)
else
else
input
1.3: input:=processForOutputFile:com.twowin.framework.collection.TData
if(input.get("targetFile") instanceof TData)
for(int i=0;i<targetFile.size();i++)
if(totalLength > 0)
if(dirLength != -1)
else
else-if(input.getString("targetFile") instanceof String)
if(totalLength > 0)
if(dirLength != -1)
else
else
input
if(clstSelectType != null && clstSelectType.equals("2"))
1.4: processForSchedule:void
for(int i=0;i<infoValue.size();i++)
1.5: list:=bfcService:com.twowin.framework.collection.TDataProtocol
wapper:MiddleBFCWSWapper
1.5.1: <default constructor>
1.5.2: bfcService:com.twowin.framework.collection.TDataProtocol
userInfoBFC:UserInfoBFC
1.6: <constructor>
1.7: wallTime:=processNoTxn:com.twowin.framework.collection.TDataProtocol

2) HTC 작업 제출

(1) 기능 흐름

- 사용자는 프로젝트 목록을 가져와 화면에 보여주며, HTC 작업을 제출할 수 있는 화면을 보여준다.
- 자원직접선택 버튼을 누르면 자원을 선택할 수 있는 화면을 팝업으로 보여준다.
- 추가 버튼을 누르면 다른 자원을 추가 선택할 수 있도록 화면에 보여준다.
- 자원선택 버튼을 누르면 자원을 선택하고, 노드수와 큐를 선택할 수 있는 화면을 팝업으로 보여준다.
- 내용작성 버튼을 누르면 작업제출내용을 입력할 수 있는 화면을 팝업으로 보여준다.
- 실행파일에 파일설정 버튼을 누르면 파일을 선택 할 수 있는 화면을 팝업으로 보여준다.
- 추가정보 작성을 누르면 추가정보를 입력할 수 있는 화면을 보여준다.
- 입력 파일에서 추가를 누르면 입력 파일을 추가할 수 있는 박스가 추가 생성된다.
- 입력 파일에서 파일설정을 누르면 파일을 선택 할 수 있는 화면을 팝업으로 보여준다.
- 출력 파일에서 추가를 누르면 출력 파일을 추가할 수 있는 박스가 추가 생성된다.
- 출력 파일에서 파일설정을 누르면 파일을 선택 할 수 있는 화면을 팝업으로 보여준다.

(2) 관련DB 테이블

[PROJECT_MT] [JOB_MT] [JOB_CONTENTS] [JOB_STATUS_HISTORY] [ARGUMENT_INFO] [INPUT_FILE_INFO] [OUTPUT_FILE_INFO] [SCHEDULE_INFO] [JOB_ACCT_HT]

(3) 프레임워크 경로

```
com.kisti.web.jobMng.JobManagerCMD.requestForJob()
com.kisti.web.jobMng.JobManagerCMD.selectForCluster()
com.kisti.web.fileMng.FileMngCMD.startApplet()
```

다음의 웹서비스를 통해 프로젝트목록을 조회하여 가져온다.

bfcService = "com.kisti.middle.bfc.projectmanager. ProjectManagerBFC, selectForProject()"

bfcService = "com.kisti.middle.bfc.monitoring.MonitoringBFC, selectForClusterInfo()"

(4) 관련JSP

- /jobManager/jobRequest/htc/requestForJob.jsp
- /jobManager/exeFile/FlexExeFile.jsp
- /jobManager/input/FlexJobFile.jsp
- /jobManager/output/FlexOutFile.jsp
- /jobManager/workDir/FlexDir.jsp
- /jobManager/jobRequest/common/selectForCluster.jsp, requestForSchedule.jsp

(5) Control정보 (forward.cfg)

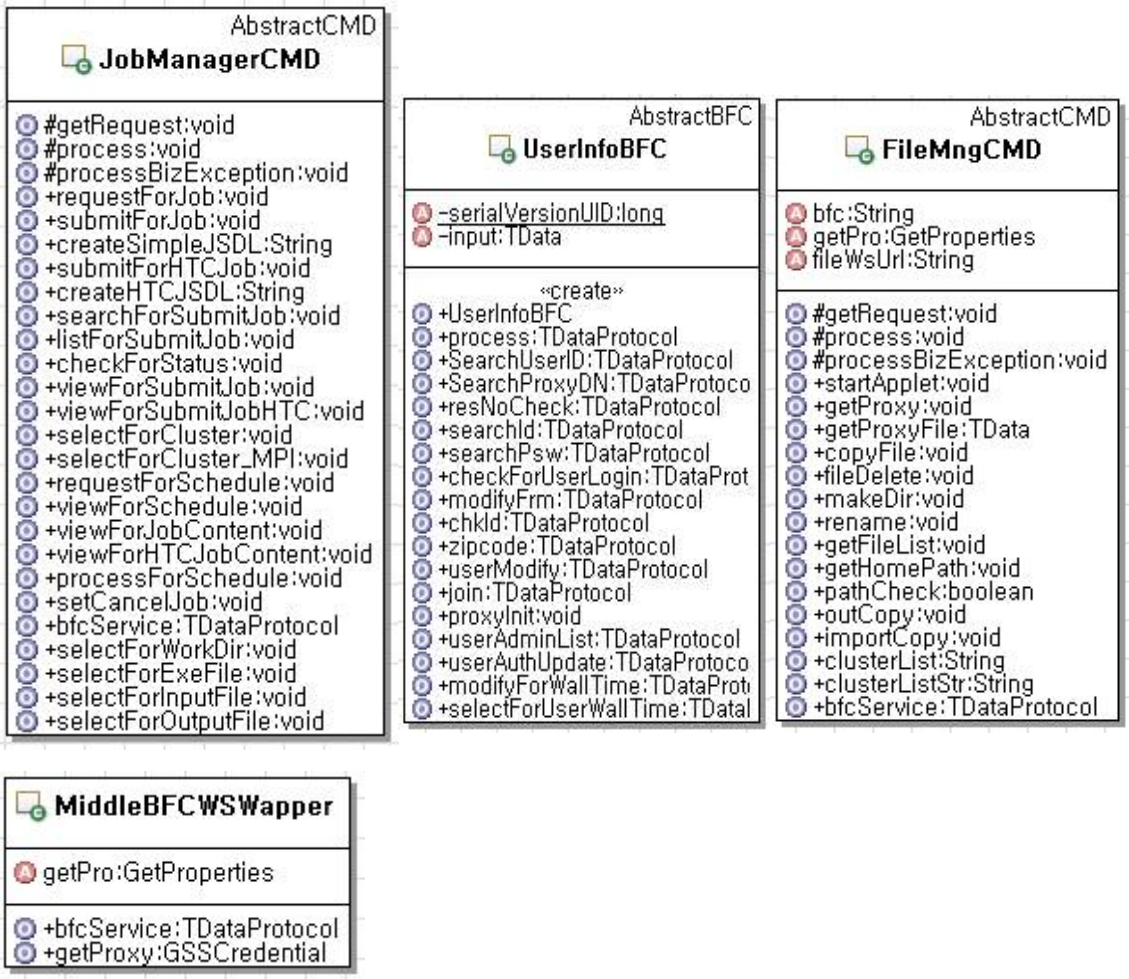
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" requestForHTCJob"

WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" selectForCluster"

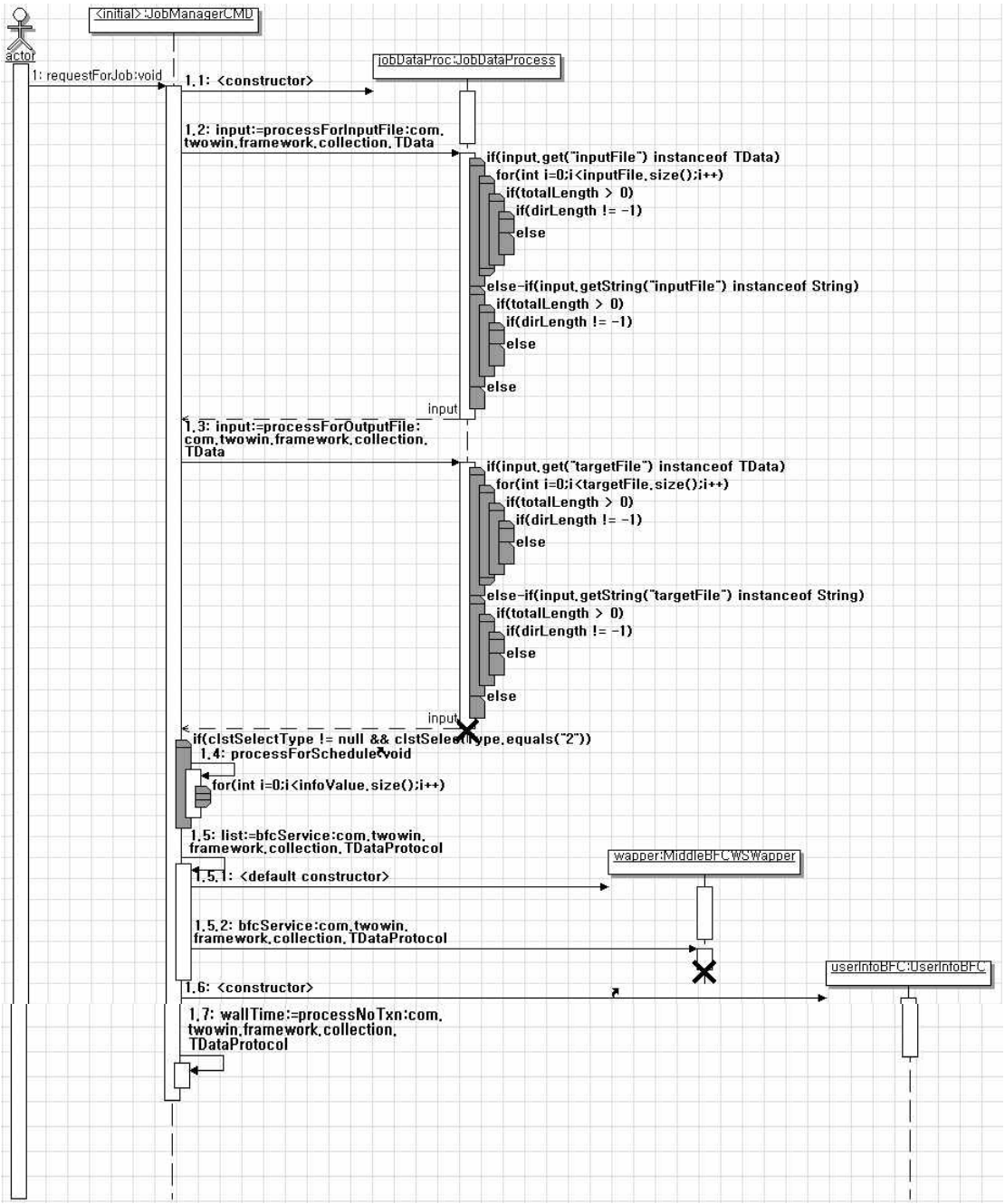
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" selectForQueueInfo"

WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" selectForQueueInfo"

(6) 클래스 다이어그램


AbstractCMD
JobManagerCMD
#getRequest:void
#process:void
#processBizException:void
+requestForJob:void
+submitForJob:void
+createSimpleJSDL:String
+submitForHTCJob:void
+createHTCJSDL:String
+searchForSubmitJob:void
+listForSubmitJob:void
+checkForStatus:void
+viewForSubmitJob:void
+viewForSubmitJobHTC:void
+selectForCluster:void
+selectForCluster_MPI:void
+requestForSchedule:void
+viewForSchedule:void
+viewForJobContent:void
+viewForHTCJobContent:void
+processForSchedule:void
+setCancelJob:void
+bfcService:TDataProtocol
+selectForWorkDir:void
+selectForExeFile:void
+selectForInputFile:void
+selectForOutputFile:void
AbstractBFC
UserInfoBFC
-serialVersionUID:long
-input:TData
«create»
+UserInfoBFC
+process:TDataProtocol
+SearchUserID:TDataProtocol
+SearchProxyDN:TDataProtoco
+resNoCheck:TDataProtocol
+searchId:TDataProtocol
+searchPsw:TDataProtocol
+checkForUserLogin:TDataProt
+modifyFrm:TDataProtocol
+chkId:TDataProtocol
+zipcode:TDataProtocol
+userModify:TDataProtocol
+join:TDataProtocol
+proxyInit:void
+userAdminList:TDataProtocol
+userAuthUpdate:TDataProtoco
+modifyForWallTime:TDataProt
+selectForUserWallTime:TDatal
AbstractCMD
FileMngCMD
bfc:String
getPro:GetProperties
fileWsUrl:String
#getRequest:void
#process:void
#processBizException:void
+startApplet:void
+getProxy:void
+getProxyFile:TData
+copyFile:void
+fileDelete:void
+makeDir:void
+rename:void
+getFileList:void
+getHomePath:void
+pathCheck:boolean
+outCopy:void
+importCopy:void
+clusterList:String
+clusterListStr:String
+bfcService:TDataProtocol
MiddleBFCWSWapper
getPro:GetProperties
+bfcService:TDataProtocol
+getProxy:GSSCredential

(7) 시퀀스 다이어그램


actor
<initial>:JobManagerCMD
jobDataProc:JobDataProcess
1: requestForJob:void
1.1: <constructor>
1.2: input:=processForInputFile:com.twowin.framework.collection.TData
if(input.get("inputFile") instanceof TData)
for(int i=0;i<inputFile.size();i++)
if(totalLength > 0)
if(dirLength != -1)
else
else-if(input.getString("inputFile") instanceof String)
if(totalLength > 0)
if(dirLength != -1)
else
else
input
1.3: input:=processForOutputFile:com.twowin.framework.collection.TData
if(input.get("targetFile") instanceof TData)
for(int i=0;i<targetFile.size();i++)
if(totalLength > 0)
if(dirLength != -1)
else
else-if(input.getString("targetFile") instanceof String)
if(totalLength > 0)
if(dirLength != -1)
else
else
input
if(clstSelectType != null && clstSelectType.equals("2"))
1.4: processForSchedule:void
for(int i=0;i<infoValue.size();i++)
1.5: list:=bfcService:com.twowin.framework.collection.TDataProtocol
wapper:MiddleBFCWSWapper
1.5.1: <default constructor>
1.5.2: bfcService:com.twowin.framework.collection.TDataProtocol
userInfoBFC:UserInfoBFC
1.6: <constructor>
1.7: wallTime:=processNoTxn:com.twowin.framework.collection.TDataProtocol

3) MPI 작업 제출

(1) 기능 흐름

- 사용자는 프로젝트 목록을 가져와 화면에 보여주고, MPI 작업을 제출할 수 있는 화면을 보여준다.
- 자원선택 버튼을 누르면 자원을 선택하고, 노드수와 큐, MPI 라이브러리를 선택할 수 있는 화면을 팝업으로 보여준다.
- 스케쥴러의뢰 버튼을 누르면 스케쥴러 의뢰시 요구사항을 입력할 수 있는 화면을 팝업으로 보여준다.
- 실행파일에 바이너리 버튼을 누르면 파일을 선택 할 수 있는 화면을 팝업으로 보여준다.
- 실행파일에서 소스버튼을 누르면 자동, 수동을 선택할 수 있도록 제공하고, 자동, 수동에 따른 파일 선택과 입력을 할 수 있는 화면을 팝업으로 보여준다.
- 추가정보 작성을 누르면 추가 정보를 입력할 수 있는 화면을 보여준다.
- 입력 파일에서 추가를 누르면 입력 파일을 추가할 수 있는 박스가 추가 생성된다.
- 입력 파일에서 파일설정을 누르면 파일을 선택 할 수 있는 화면을 팝업으로 보여준다.
- 출력 파일에서 추가를 누르면 출력 파일을 추가할 수 있는 박스가 추가 생성된다.
- 출력 파일에서 파일설정을 누르면 파일을 선택할 수 있는 화면을 팝업으로 보여준다.

(2) 관련DB 테이블

[PROJECT_MT] [JOB_MT] [JOB_CONTENTS] [JOB_STATUS_HISTORY] [ARGUMENT_INFO] [INPUT_FILE_INFO] [OUTPUT_FILE_INFO] [SCHEDULE_INFO] [JOB_ACCT_HT] [MPI_COMFILE_INFO]

(3) 프레임워크 경로

```
com.kisti.web.jobMng.JobManagerCMD.requestForJob()
com.kisti.web.jobMng.JobManagerCMD.selectForCluster()
com.kisti.web.fileMng.FileMngCMD.startApplet()
```

다음의 웹서비스를 통해 프로젝트목록을 조회하여 가져온다.

bfcService = "com.kisti.middle.bfc.projectmanager. ProjectManagerBFC, selectForProject()"

bfcService = "com.kisti.middle.bfc.monitoring.MonitoringBFC, selectForClusterInfo()"

(4) 관련 JSP

- /jobManager/jobRequest/simple/requestForJob.jsp
- /jobManager/exeFile/FlexExeFile.jsp
- /jobManager/input/FlexJobFile.jsp
- /jobManager/output/FlexOutFile.jsp
- /jobManager/workDir/FlexDir.jsp
- /jobManager/jobRequest/mpi/lisetForCluster.jsp, lisetForMPILibrary.jsp, selectForCluster.jsp,selectForCompilingType.jsp
- /jobManager/jobRequest/common/requestForSchedule.jsp

(5) Control정보 (forward.cfg)

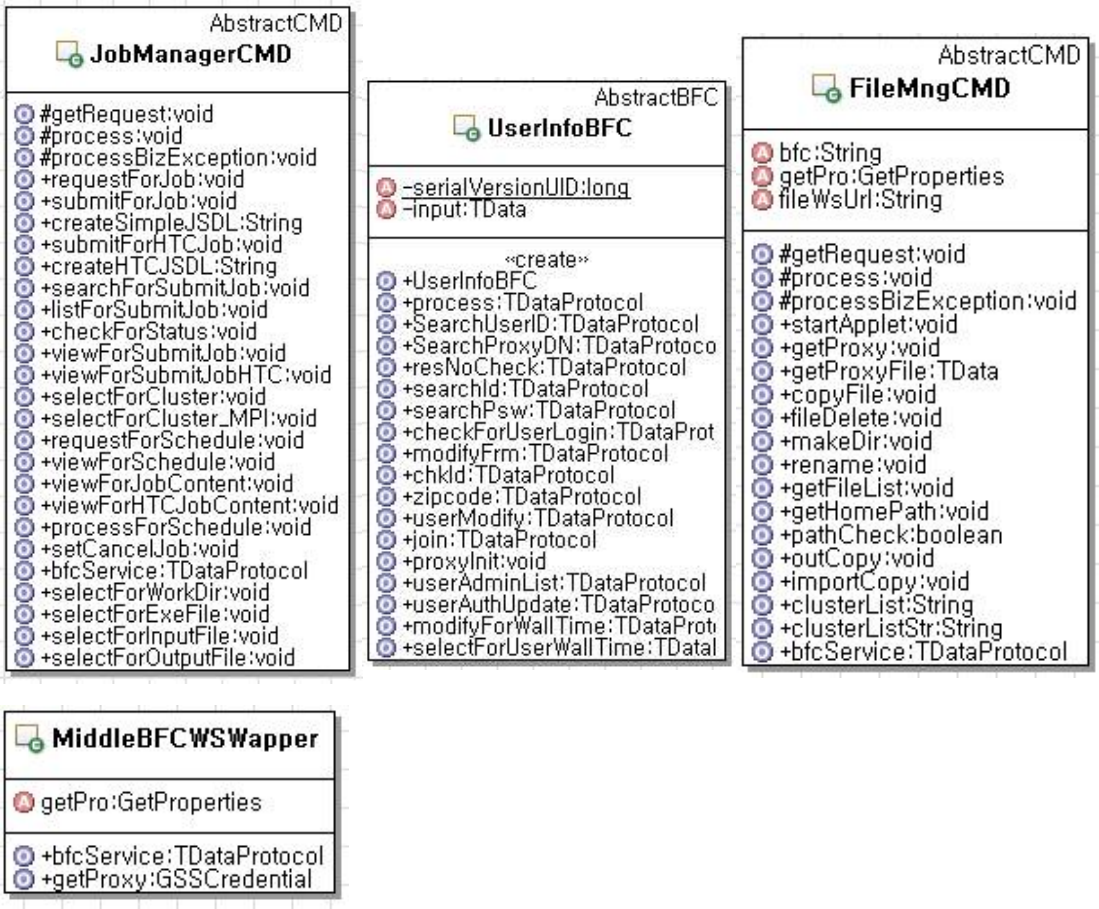
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" requestForMPIJob"

WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" selectForCluster"

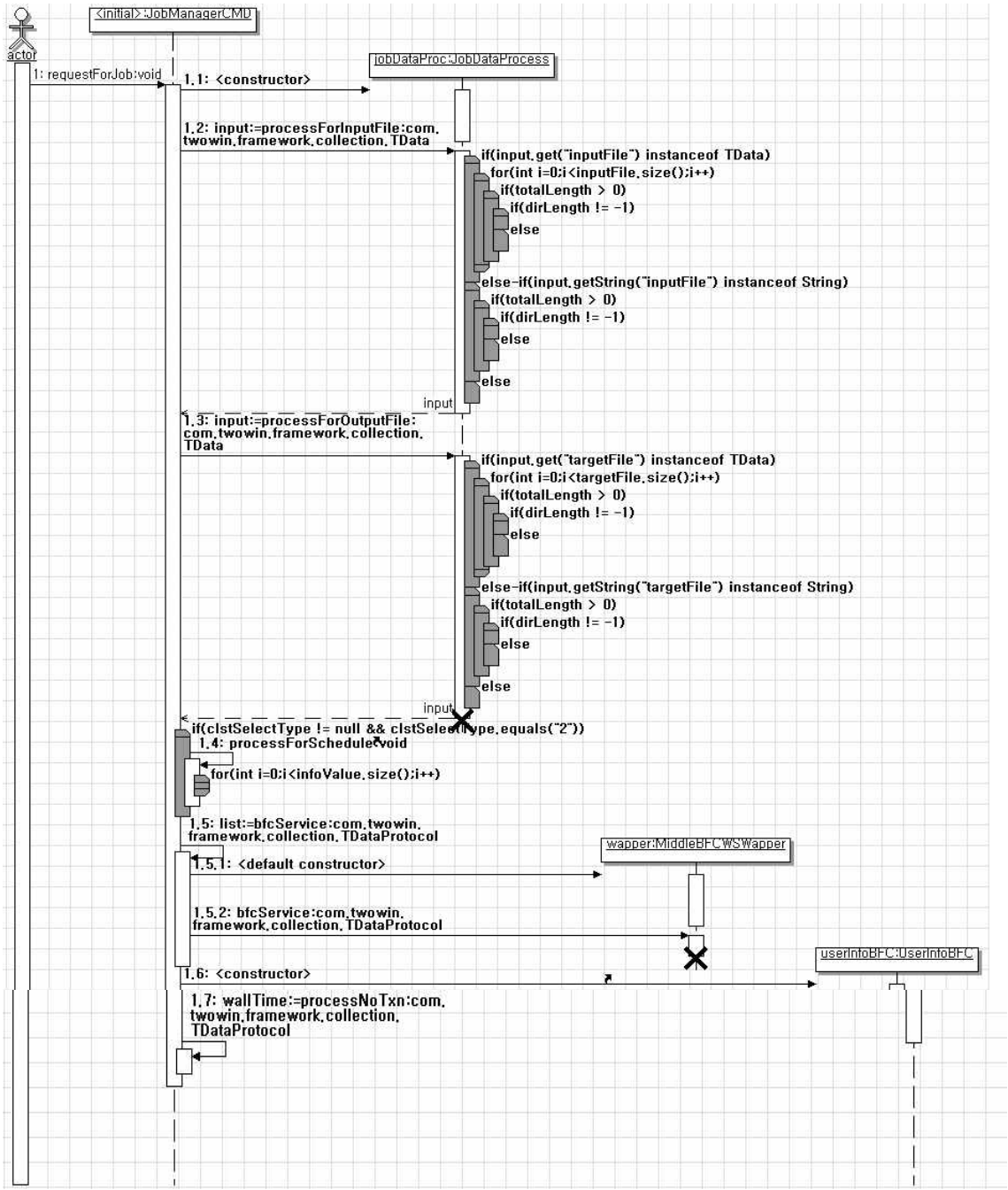
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" selectForQueueInfo"

WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name=" selectForQueueInfo"

(6) 클래스 다이어그램


AbstractCMD
JobManagerCMD
#getRequest:void
#process:void
#processBizException:void
+requestForJob:void
+submitForJob:void
+createSimpleJSDL:String
+submitForHTCJob:void
+createHTCJSDL:String
+searchForSubmitJob:void
+listForSubmitJob:void
+checkForStatus:void
+viewForSubmitJob:void
+viewForSubmitJobHTC:void
+selectForCluster:void
+selectForCluster_MPI:void
+requestForSchedule:void
+viewForSchedule:void
+viewForJobContent:void
+viewForHTCJobContent:void
+processForSchedule:void
+setCancelJob:void
+bfcService:TDataProtocol
+selectForWorkDir:void
+selectForExeFile:void
+selectForInputFile:void
+selectForOutputFile:void
AbstractBFC
UserInfoBFC
-serialVersionUID:long
-input:TData
«create»
+UserInfoBFC
+process:TDataProtocol
+SearchUserID:TDataProtocol
+SearchProxyDN:TDataProtoco
+resNoCheck:TDataProtocol
+searchId:TDataProtocol
+searchPsw:TDataProtocol
+checkForUserLogin:TDataProt
+modifyFrm:TDataProtocol
+chkId:TDataProtocol
+zipcode:TDataProtocol
+userModify:TDataProtocol
+join:TDataProtocol
+proxyInit:void
+userAdminList:TDataProtocol
+userAuthUpdate:TDataProtoco
+modifyForWallTime:TDataProt
+selectForUserWallTime:TDatal
AbstractCMD
FileMngCMD
bfc:String
getPro:GetProperties
fileWsUrl:String
#getRequest:void
#process:void
#processBizException:void
+startApplet:void
+getProxy:void
+getProxyFile:TData
+copyFile:void
+fileDelete:void
+makeDir:void
+rename:void
+getFileList:void
+getHomePath:void
+pathCheck:boolean
+outCopy:void
+importCopy:void
+clusterList:String
+clusterListStr:String
+bfcService:TDataProtocol
MiddleBFCWSWapper
getPro:GetProperties
+bfcService:TDataProtocol
+getProxy:GSSCredential

(7) 시퀀스 다이어그램


actor
<initial>:JobManagerCMD
jobDataProc:JobDataProcess
1: requestForJob:void
1.1: <constructor>
1.2: input:=processForInputFile:com.twowin.framework.collection.TData
if(input.get("inputFile") instanceof TData)
for(int i=0;i<inputFile.size();i++)
if(totalLength > 0)
if(dirLength != -1)
else
else-if(input.getString("inputFile") instanceof String)
if(totalLength > 0)
if(dirLength != -1)
else
else
input
1.3: input:=processForOutputFile:com.twowin.framework.collection.TData
if(input.get("targetFile") instanceof TData)
for(int i=0;i<targetFile.size();i++)
if(totalLength > 0)
if(dirLength != -1)
else
else-if(input.getString("targetFile") instanceof String)
if(totalLength > 0)
if(dirLength != -1)
else
else
input
if(clstSelectType != null && clstSelectType.equals("2"))
1.4: processForSchedule:void
for(int i=0;i<infoValue.size();i++)
1.5: list:=bfcService:com.twowin.framework.collection.TDataProtocol
wapper:MiddleBFCWSWapper
1.5.1: <default constructor>
1.5.2: bfcService:com.twowin.framework.collection.TDataProtocol
userInfoBFC:UserInfoBFC
1.6: <constructor>
1.7: wallTime:=processNoTxn:com.twowin.framework.collection.TDataProtocol

2.4.2 작업 모니터링

1) 작업 제출 내역 목록

(1) 기능 흐름

- 사용자는 작업 제출 내역에 해당하는 데이터를 조회하여 작업 제출 내역 목록을 보여준다.
- 작업 제출 내역 목록 중 작업명의 링크를 통해 해당 작업 내용에 대해 상세 조회를 할 수 있다.

(2) 관련DB 테이블

[JOB_MT]

(3) 프레임워크 경로

com.kisti.web.jobMng.JobManagerCMD.listForSubmitJob()

다음의 웹서비스를 통해 해당 사용자의 작업 제출 내역을 조회하여 레코드를 가져온다.

bfcService = "com.kisti.middle.bfc.jobmanager.jobManagerBFC, listForSubmitJob"

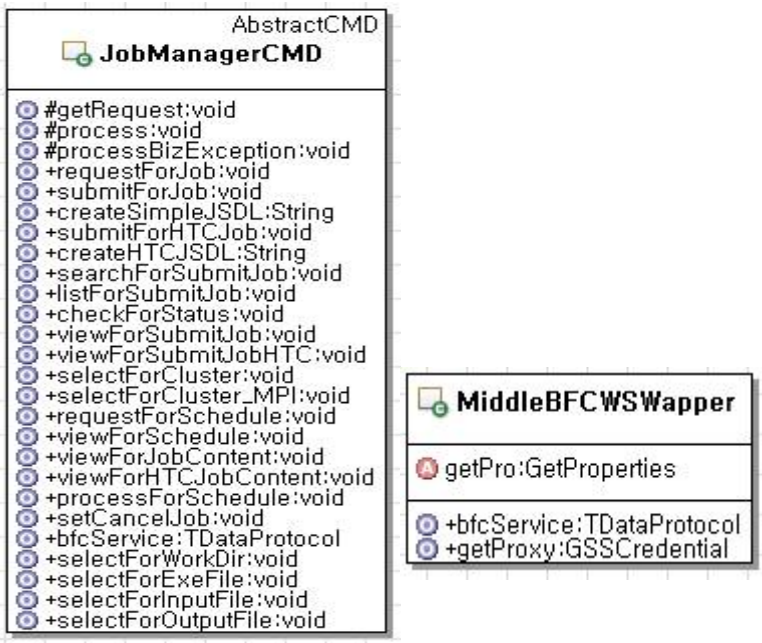
(4) 관련 JSP

- /jobManager/listForSubmitJob.jsp
- /jobManager/searchForSubmitJob.jsp

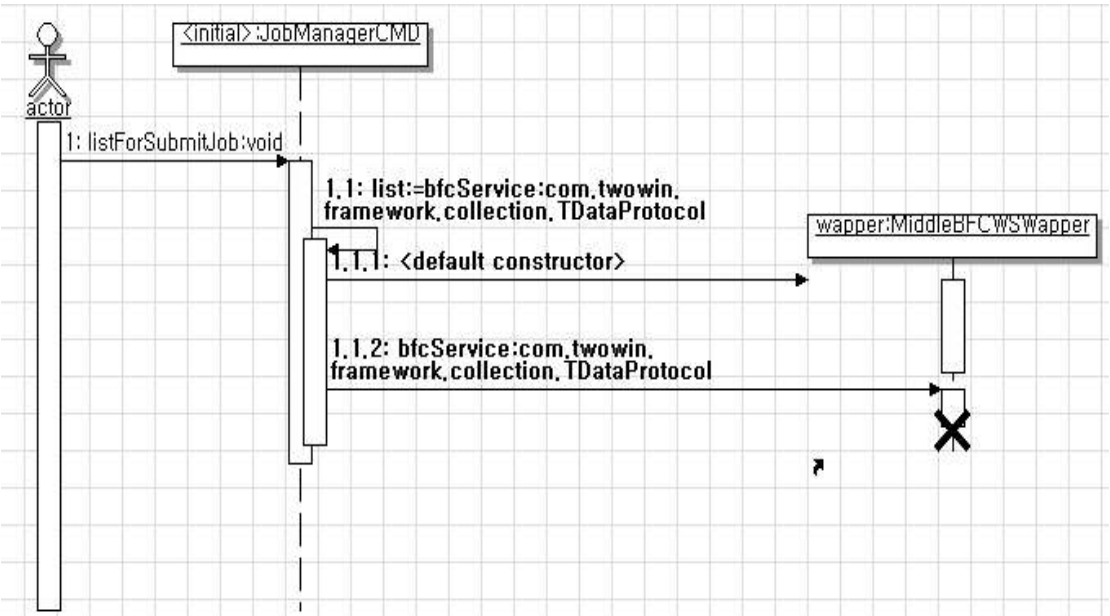
(5) Control정보 (forward.cfg)

WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name="listForSubmitJob"

(6) 클래스 다이어그램



(7) 시퀀스 다이어그램



2) 작업 제출 내역 상세보기

(1) 기능 흐름

- 사용자는 작업 제출 내역에서 해당 작업명의 링크를 이용하여 작업의 상세 내용을 볼 수 있다.
- 작업에 대한 기본 내용과 상태에 대한 정보를 실시간으로 볼 수 있는 화면을 제공한다.
- 작업 제출시 발생한 에러에 관한 내용을 보여준다.
- 작업 제출시 작성한 내용을 보여주고, 작업 결과물을 바로 다운로드 받을 수 있는 화면을 팝업으로 보여준다.
- 작업재전송 버튼을 이용해 해당 작업을 다시 전송 할 수 있는 페이지로 링크를 제공한다.

(2) 관련 DB 테이블

[JOB_MT]

(3) 프레임워크 경로

단순 작업, MPI 작업의 경우:

`com.kisti.web.jobMng.JobManagerCMD.viewForJobcontent()`

HTC 작업의 경우:

`com.kisti.web.jobMng.JobManagerCMD.viewForHTCJobContent()`

다음의 웹서비스를 통해 해당 사용자의 작업 제출 내역을 조회하여 레코드를 가져온다.

bfcService = "com.kisti.middle.bfc.jobmanager.jobManagerBFC, listForSubmitJob"

bfcService = "com.kisti.middle.bfc.jobmanager.jobManagerBFC, viewForJobContent"

bfcService = "com.kisti.middle.bfc.jobmanager.jobManagerBFC, selectForInputFile"

bfcService = "com.kisti.middle.bfc.jobmanager.jobManagerBFC, selectForOutputFile"

bfcService = "com.kisti.middle.bfc.jobmanager.jobManagerBFC, selectForProject"

(4) 관련 JSP

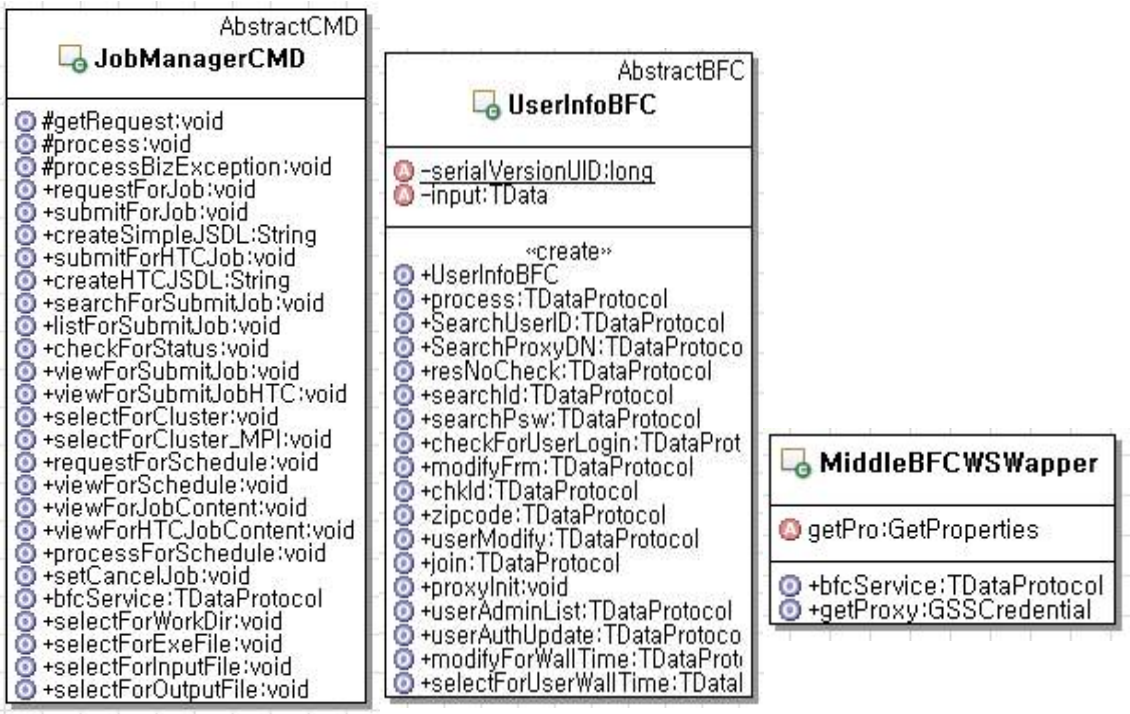
- /jobManager/listForSubmitJob.jsp
- /jobManager/searchForSubmitJob.jsp
- 각 작업 타입에 따라

단순 작업: /jobManager/jobRequest/simple/viewSimple.jsp
HTC 작업: /jobManager/jobRequest/simple/viewHtc.jsp, viewInfo.jsp
MPI 작업: /jobManager/jobRequest/simple/viewMpi.jsp

(5) Control정보 (forward.cfg)

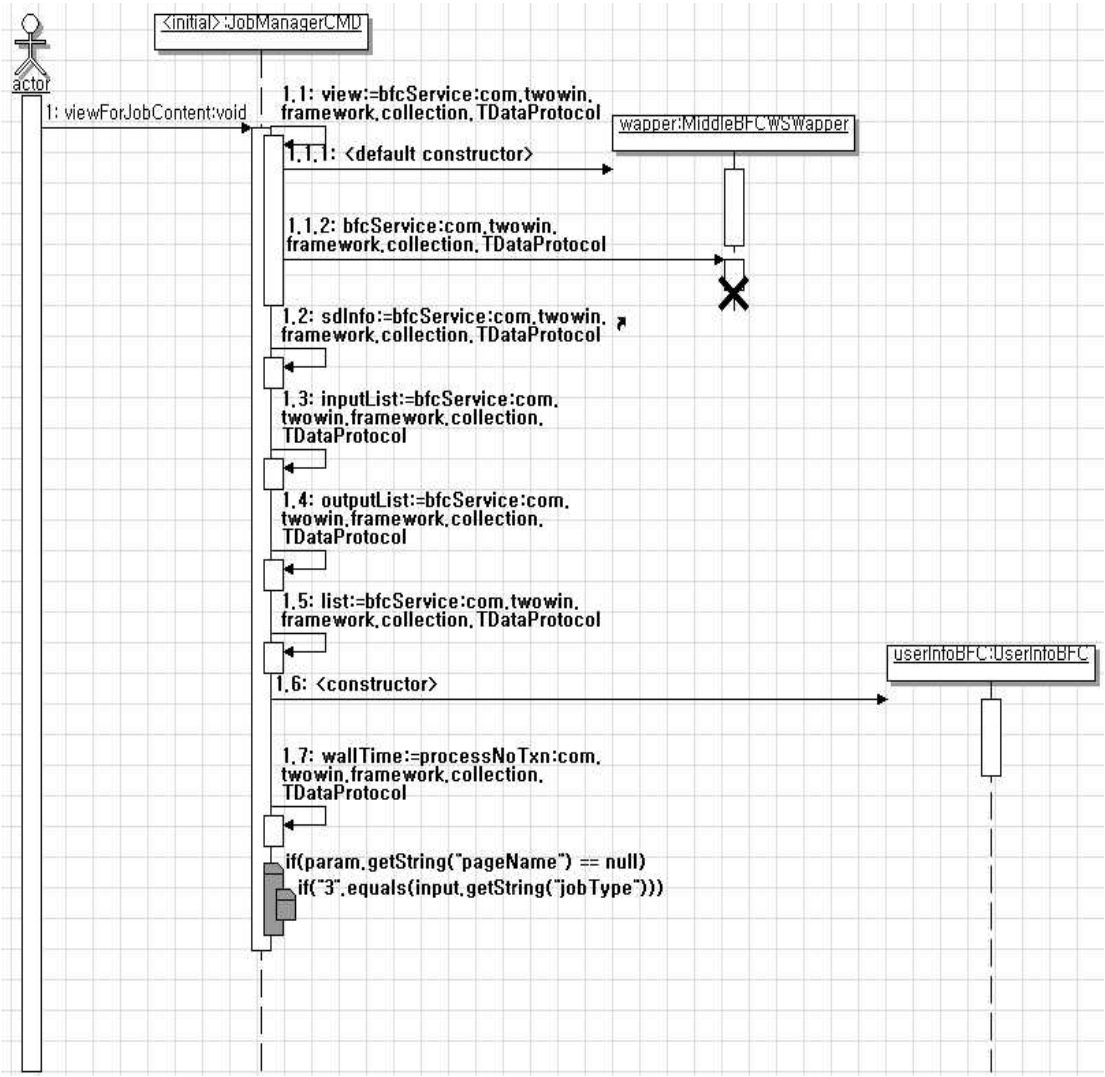
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name="listForSubmitJob"
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name="viewForJobContent"
WebRoot/WEB-INF/twfenv/servlet/jobManager/jobNavigation.xml, name="viewForHTCJobContent"

(6) 클래스 다이어그램

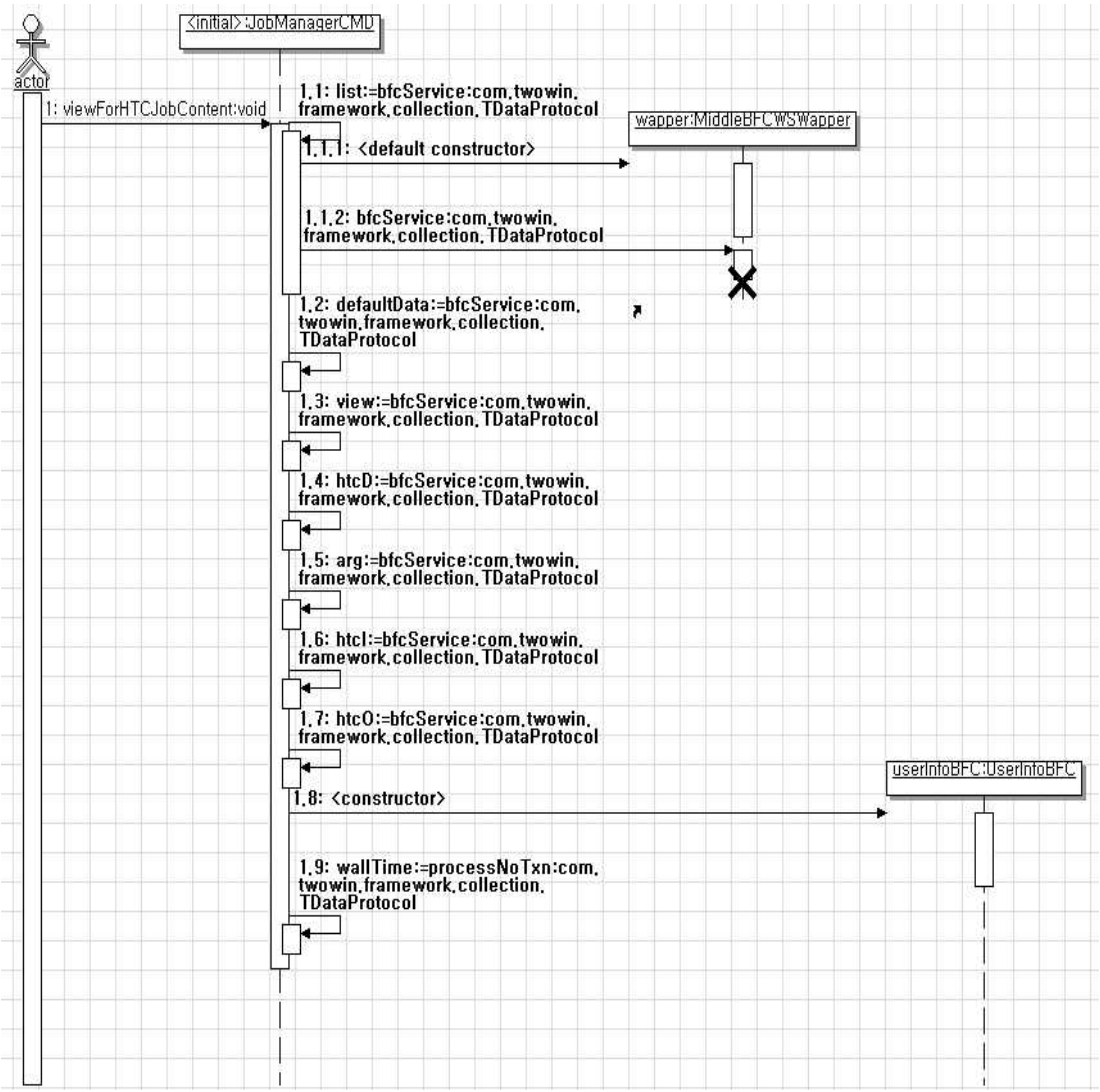


(7) 시퀀스 다이어그램

단순 작업, MPI 작업:



HTC 작업:


actor
<initial>:JobManagerCMD
1: viewForHTCJobContent:void
1.1: list:=bfcService:com.twowin.framework.collection.TDataProtocol
wapper:MiddleBFCWSWapper
1.1.1: <default constructor>
1.1.2: bfcService:com.twowin.framework.collection.TDataProtocol
1.2: defaultData:=bfcService:com.twowin.framework.collection.TDataProtocol
1.3: view:=bfcService:com.twowin.framework.collection.TDataProtocol
1.4: htcD:=bfcService:com.twowin.framework.collection.TDataProtocol
1.5: arg:=bfcService:com.twowin.framework.collection.TDataProtocol
1.6: htcI:=bfcService:com.twowin.framework.collection.TDataProtocol
1.7: htcO:=bfcService:com.twowin.framework.collection.TDataProtocol
1.8: <constructor>
userInfoBFC:UserInfoBFC
1.9: wallTime:=processNoTxn:com.twowin.framework.collection.TDataProtocol

3. 결론

o 그리드 작업관리서비스는 그리드 서비스 포털을 개발하는데 있어서 핵심 미들웨어 기술 요소임

o 본 기술 지원 보고서는 KISTI가 개발한 “WSRF 기반 그리드 작업관리서비스”의 서비스 구현 설계와 핵심 모듈별 세부 서비스 구현을 소개하고 있음

- 그리드 작업 제출, 그리드 작업 모니터링과 관련된 작업 형태별(단순 작업, HTC 작업, MPI 작업) 기능 흐름, 관련 DB 테이블, 서비스 프레임워크 경로, 클래스 다이어그램, 시퀀스 다이어그램 소개

첨부1: DB 테이블

[JOB_CONTENTS]: 작업제출내용

COLUMN NAME	한글명	TYPE	SIZE	NULL	KEY	비고
SUB_JOB_ID	서브 작업 아이디	INTEGER		N		
WORK_DIR	작업 디렉토리	VARCHAR2	1000	Y		
EXC_FILE	실행 파일	VARCHAR2	1000	Y		
ENVIRONMENT	환경변수	VARCHAR2	1000	Y		
LIBLARY	라이브러리	VARCHAR2	100	Y		
ST_ERR	표준 에러	VARCHAR2	100	Y		
ST_OUT	표준 출력	VARCHAR2	100	Y		
JOB_STATUS	작업 상태	NUMBER	3	Y		
JOB_ID	작업 아이디	INTERGER		Y		
EXC_SERVER	실행파일 서버	VARCHAR2	1000	Y		
SRC_SEL_TYPE	자원선택유형	CHAR	1	Y		
LOCAL_JOB_MANAGER	스케줄러 타입	VARCHAR2	50	Y		
QUEUE	큐 정보	VARCHAR2	50	Y		
COMPILING_TYPE	컴파일 방법	CHAR	1	Y		
EXE_SOURCE	소스	VARCHAR2	1000	Y		
EXE_COMPILING	컴파일방법	VARCHAR2	1000	Y		
MPI_LIBRARY	MPI LIBRARY	VARCHAR2	500	Y		
ST_FILE	표준 파일	VARCHAR2	500	Y		
ST_SERVER	표준 파일 서버	VARCHAR2	1000	Y		
JOB_STATUS_LOG	작업상태 로그	CLOB		Y		
ERR_COMMENT	작업에러 로그	CLOB		Y		
SGE_P_E_NAME	SGE P E NAME	VARCHAR2	100	Y		
WALL_TIME_USER	사용자가 입력한 WALL TIME	INTEGER		Y		
WALL_TIME_PROCESS	작업이 실행된 시간	INTEGER		Y		
WALL_TIME_ACCOUNT	ACCOUNT 로그에 올라온 시간	INTEGER		Y		

[JOB_MT]: 작업정보관리

COLUMN NAME	한글명	TYPE	SIZE	NULL	KEY	비고
JOB_ID	작업 아이디	INTEGER		N		
JOB_NM	작업 명	VARCHAR2	100	Y		
JOB_STATUS	작업 상태	INTEGER		Y		
PROJECT_ID	프로젝트 아이디	INTEGER		Y		
JOB_TYPE	작업 타입	VARCHAR2	18	Y		
START_DATE	작업 시작 시간	VARCHAR2	20	Y		
END_DATE	작업 종료 시간	VARCHAR2	20	Y		
EXITCODE	종료코드	INTEGER		Y		
ERR_COMMENT	에러 코멘트	VARCHAR2	40000	Y		
UU_ID	UU ID	VARCHAR2	100	Y		
JSDL_COMMENTS	JSDL	CLOB		Y		
RSL_COMMENTS	RSL	CLOB		Y		
PROXYDN	PROXY 정보	VARCHAR2	255	Y		
CANCELBIT	취소 플래그	CHAR	1	Y		Defaul : ‘F’

[INPUT_FILE_INFO]: 입력파일정보

COLUMN NAME	한글명	TYPE	SIZE	NULL	KEY	비고
SUB_NUM	INPUT FILE 카운트	INTEGER		N		
SAVE_DIR	저장 위치	VARCHAR2	1000	Y		
FILE_NM	파일 명	VARCHAR2	100	Y		
NODE_NUM	노드 수	VARCHAR2	20	N		
SUB_JOB_ID	서브 작업 아이디	INTEGER		N		
SOURCE_FILE	저장될 파일 명	VARCHAR2	100	Y		
INPUT_FLAG	저장 플래그	CHAR	1	Y		
INPUT_DEL_YN	파일 삭제 여부	CHAR	1	Y		

[OUTPUT_FILE_INFO]: 출력파일정보

COLUMN NAME	한글명	TYPE	SIZE	NULL	KEY	비고
SAVE_DIR	저장 위치	VARCHAR2	1000	Y		
SUB_NUM	OUTPUT 파일 카운트	INTEGER		N		
FIEL_NM	파일 명	VARCHAR2	100	Y		
SUB_JOB_ID	서브 작업 아이디	INTEGER		N		
OUTPUT_FILE	저장될 파일명	VARCHAR2	100	Y		
OUTPUT_FLAG	저장 플래그	CHAR	1	Y		
OUTPUT_DEL_YN	파일 삭제 여부	CHAR	1	Y		

[MPI_CONFIG_INFO]: MPI작업 부가 정보

COLUMN NAME	한글명	TYPE	SIZE	NULL	KEY	비고
SRC_FILE	파일명	VARCHAR2	50	Y		
COMFILE_TYPE	컴파일 방법	VARCHAR2	2000	Y		
SUB_JOB_ID	서브 작업 아이디	INTEGER		N		

[SCHDULE_INFO]: 스케줄러의뢰 정보

COLUMN NAME	한 글 명	TYPE	SIZE	NULL	KEY	비고
SUB_JOB_ID	서브 작업 아이디	INTEGER		N		
INFO_VALUE	의뢰 정보 값	VARCHAR2	100	N		

[JOB_ACCT_HT]: 어카운팅 클러스터 작업 현황

COLUMN NAME	한 글 명	TYPE	SIZE	NULL	KEY	비고
ACC_ID	계정 아이디	VARCHAR2	20			
USR_ID	사용자 아이디	VARCHAR2	20			
CLST_NM	클러스터명	VARCHAR2	50			
CLST_HOST	클러스터 호스트	VARCHAR2	50			
QUEUE_NM	큐 이름	VARCHAR2	50			
APPLICATION_NM	어플리케이션명	VARCHAR2	50			
CPU_NM	CPU 명	INTEGER				
SUBMIT_DATE	제출 일시	VARCHAR2	20			
START_DATE	시작시간	VARCHAR2	20			
END_DATE	종료시간	VARCHAR2	20			
WAIT_TIME	대기 시간	INTEGER				

CPU_TIME	프로세서 시간	INTEGER				
WALL_TIME	Wall Time	INTEGER				
MEM_USE	메모리 사용량	INTEGER				
SWAP_USE	Swap 사용량	INTEGER				
USE_DATE	사용일시	VARCHAR2	20			
JOB_STATUS	작업 상태	NUMBER	3			
CHARGE_YN	과금 여부	CHAR	1			
QUEUE_DATE	큐잉 일시	VARCHAR2	20			
CHARGE_USE	과금 사용량	INTEGER				
ACCOUNT_UPDATE_YN	자료수집 여부	CHAR	1			한번이라도 수집됐으면 Y
SUB_JOB_ID	서브 잡 아이디	INTEGER		N.N	P.K	
CLST_ID	클러스터 아이디	INTEGER				
LOCAL_JOB_ID	클러스터 작업 아이디	INTEGER				

[PROFILE_MNG]: 프로파일 관리

COLUMN NAME	한 글 명	TYPE	SIZE	NULL	KEY	비고
PROFILE_ID	프로파일 아이디	INTEGER	50	N		
PROFILE_NM	프로파일 명	VARCHAR2		Y		
JOB_ID	작업 아이디	INTEGER		Y		
PROJECT_ID	프로젝트 아이디	INTEGER		Y		
PROFILE_COMMENT	프로파일 설명	VARCHAR2	1000	Y		