Софийски университет „Св. Климент Охридски”

Факултет по математика и информатика

Радослава Данаилова Горанова

Услуги на приложно ниво в грид за научни изчисления (Среда за управление на бизнес-процеси в грид)

ДИСЕРТАЦИОНЕН ТРУД

за присъждане на образователна и научна степен „Доктор”

Научна специалност: 01.01.12 (Информатика)

Научен ръководител: доц. д-р Владимир Димитров

София, 2012

Съдържание

СЪДЪРЖАНИЕ **2**

СПИСЪК НА ФИГУРИТЕ **4**

СПИСЪК НА ТАБЛИЦИТЕ **5**

УВОД **6**

Актуалност на проблема 6
Цели на дисертацията 8
Задачи на дисертацията 8
Кратко съдържание на работата 9

ГЛАВА 1. ОБЗОР **11**

1.1 Грид 11
1.2 Ориентирана към услуги архитектура 14
1.2.1 Принципи на ориентирания към услуги подход 16
1.2.2 Архитектура на Уеб услуги 20
1.2.3 Стандарти за Уеб услуги 23
1.3 Управление на бизнес-процеси 32
1.4 Среди за управление на бизнес-процеси 36
1.4.1 Среда за разработка 37
1.4.2 Среда за изпълнение 39
1.5 Ориентиран към услуги грид 42
1.6 Грид средата g-Lite 45
1.7 Изводи 51

ГЛАВА 2. СРЕДИ ЗА ПРОЕКТИРАНЕ И ИЗПЪЛНЕНИЕ НА БИЗНЕС-ПРОЦЕСИ В ГРИД **53**

2.1 CRAB 53
2.2 g-Eclipse 55
2.3 P-GRADE 57
2.4 Triana 59
2.5 Taverna 62
2.6 Kepler 64
2.7 Сравнение на разгледаните среди 68
2.8 Изводи 70

ГЛАВА 3. МОДЕЛ НА СРЕДА ЗА УПРАВЛЕНИЕ НА БИЗНЕС-ПРОЦЕСИ В ГРИД **72**

3.1 Модули и компоненти на среда за управление на бизнес-процеси в EGI 72
3.1.1 Модул Регистър на услуги 74
3.1.2 Модул Среда за разработка 75
3.1.3 Модул Среда за изпълнение 77
3.2 Роли и случаи на употреба в среда за управление на бизнес-процеси в EGI 79
3.2.1 Роли в среда за управление на бизнес-процеси 79
3.2.2 Случаи на употреба в регистър на услуги 80
3.2.3 Случаи на употреба в среда за разработка 82
3.2.4 Случаи на употреба в среда за изпълнение 86
3.3 Допълнителни изисквания към среда за управление на бизнес-процеси в EGI 88
3.4 Проектиране на архитектура на среда за управление на бизнес-процеси в EGI 90
3.5 Изводи 93
3.6 Приноси 93

ГЛАВА 4. РЕАЛИЗАЦИЯ И ПРИЛОЖЕНИЕ НА СРЕДА ЗА УПРАВЛЕНИЕ НА БИЗНЕС-ПРОЦЕСИ В EGI **94**

4.1 Модул Адаптери за достъп до g-Lite 94
4.1.1 Използвани технологии и инструменти 94
4.1.2 Услуги за достъп към ROOT 95
4.1.3 Услуги за достъп към g-Lite 98
4.2 Модул Регистър на услуги 103

4.2.1 Подход за публикуване на услугите от BDII в модул Регистър ... 107
4.3 Модул Среда за разработка на бизнес-процеси ... 109
4.4 Модул Среда за изпълнение на бизнес-процес ... 121
4.5 Разработване на примерни бизнес-процеси в средата ... 125
4.6 Приноси ... 131

ЗАКЛЮЧЕНИЕ ... 132

БИБЛИОГРАФИЯ ... 134

АВТОРСКА СПРАВКА ... 142

Анотация на получените резултати ... 142
Научни и научно-приложни приноси ... 142
Приложни приноси ... 142
Публикации ... 143

ПРИЛОЖЕНИЯ ... 145

Приложение 1 - Релационен модел на данни ... 145
Приложение 2 - Конфигуриране на връзка до WSRR регистър ... 147
Приложение 3 - Генериране за проект за наблюдение в среда за разгръщане ... 156
Приложение 4 - Разработени модели на бизнес-процеси ... 158
Приложение 5 - Работно пространство на грид-потребител ... 165

Списък на фигурите

Фиг. 1. Слоеве на архитектура на грид ... 13
Фиг. 2. Взаимодействие на услуги в ориентирана към услуги архитектура ... 15
Фиг. 3. Оркестрация ... 17
Фиг. 4. Хореография ... 18
Фиг. 5. Механизъм на взаимодействие между доставчик и консуматор ... 21
Фиг. 6. Архитектурен стек на Уеб услуги ... 22
Фиг. 7. WSDL документ ... 23
Фиг. 8. SOAP съобщение ... 25
Фиг. 9. UDDI Регистър ... 26
Фиг. 10. WS-BPEL документ ... 27
Фиг. 11. WS-CDL Пакет ... 30
Фиг. 12. Жизнен цикъл на ориентираната към услуги архитектура ... 33
Фиг. 13. Компоненти на среда за управление на бизнес-процеси базирана на ориентирана към услуги архитектура ... 37
Фиг. 14. Услуги в g-Lite ... 45
Фиг. 15. Връзки между основните услуги в средата g-Lite ... 48
Фиг. 16. Архитектура на CRAB ... 54
Фиг. 17. Бизнес-процес моделиран в g-Eclipse ... 56
Фиг. 18. P-Grade редактор за работни потоци ... 58
Фиг. 19. Workflow Manager портлет на P-GRADE портала ... 58
Фиг. 20. Архитектура на Triana ... 59
Фиг. 21. Разпределни компоненти в Triana ... 60
Фиг. 22. Потребителски интерфейс на Triana ... 61
Фиг. 23. Архитектура на Taverna 2 ... 62
Фиг. 24. Бизнес-процес моделиран в Taverna ... 63
Фиг. 25. Адаптиране на Kepler за g-Lite ... 67
Фиг. 26. Бизнес-процес моделиран с Kepler ... 67
Фиг. 27. Компонентен модел на среда за управление на бизнес-процеси в EGI ... 73
Фиг. 28. Модули и компоненти на среда за управление на бизнес-процес в EGI ... 74
Фиг. 29. Взаимодействие между услугите в среда за изпълнение в EGI ... 78
Фиг. 30. Роли в среда за управление на бизнес-процеси и базови дейности в средата ... 80
Фиг. 31. Основни случаи на употреба за регистър ... 81
Фиг. 32. Основни случаи на употреба за среда за проектиране ... 83
Фиг. 33. Основни случаи на употреба за среда за разгръщане ... 85
Фиг. 34. Основни случаи на употреба за среда за изпълнение ... 87
Фиг. 35. Архитектура на среда за управление на бизнес-процеси в EGI ... 90
Фиг. 36. Модули и компоненти на среда за управление на бизнес-процес в EGI адаптирани за g-Lite ... 91
Фиг. 37. Примерна среда за управление на бизнес-процеси в EGI ... 92
Фиг. 38. Архитектура на модул Адаптери за достъп до g-Lite ... 94
Фиг. 39. Диаграма на Java класовете на ROOT услуги ... 98
Фиг. 40. Модел "Същност-връзки" на данните ... 99
Фиг. 41. Релационен модел на данните ... 100
Фиг. 42. Диаграма на Java пакети и класове на g-Lite услуги ... 103
Фиг. 43. Връзка между WSRR и жизнения цикъл на ориентираната към услуги архитектура ... 104
Фиг. 44. Уеб потребителски интерфейс за достъп до WSRR ... 105
Фиг. 45. Публикуване на услуга през Уеб потребителския интерфейс ... 106
Фиг. 46. Създаване не проект в среда за проектиране IBM WebSphere Business Modeler ... 110
Фиг. 47. Структура на проект в средата за проектиране IBM WebSphere Business Modeler ... 110
Фиг. 48. Диалог за търсене в регистър с резултатът от търсените услуги ... 111
Фиг. 49. Структура на проекта на грид-потребител, след добавяне на услуги от регистъра ... 112
Фиг. 50. Палитра в редактора на бизнес-процеси ... 113
Фиг. 51. Пример на моделиран бизнес-процес в среда за проектиране ... 114
Фиг. 52. Свойства на бизнес-процеса *Draw Histogram* ... 115
Фиг. 53. Проекти генерирани автоматично за проекта *Root* от среда за разгръщане ... 116
Фиг. 54. BPEL код за бизнес-процеса *Draw Histogram* ... 117
Фиг. 55. Потребителски интерфейс към бизнес-процеса *Draw Histogram* ... 118

Фиг. 56. Проекти на потребителския интерфейс генериран за процеса *Draw Histogram* 119
Фиг. 57. Генериран форма за извикване на бизнес-процеса *Draw Histogram* 119
Фиг. 58. Разгръщане на бизнес-процеса *Draw Histogram* в среда за изпълнение 121
Фиг. 59. Разгърнат бизнес-процес *Draw Histogram* в среда за изпълнение 121
Фиг. 60. IBM WebSphere Процесен сървър 122
Фиг. 61. Задача от тип човек-услуга в среда за изпълнение 123
Фиг. 62. Работно пространство на грид-потребителя 124
Фиг. 63. Изпълнение на бизнес-процеса *Draw Histogram* в среда за изпълнение 124
Фиг. 64. Бизнес-процесът Grid Job Submit Test 126
Фиг. 65. Извикване на бизнес процеса *Grid Job Submit Test* 126
Фиг. 66. Резултат от бизнес процеса *Grid Job Submit Test* 127
Фиг. 67. Прототип на бизнес-процес за NA62 130

Списък на таблиците

Таблица 1. Дейности в WS-BPEL 29
Таблица 2. Описание на g-Lite услуги в информационния индекс на средата 50
Таблица 3. Елементи в g-Eclipse за моделиране на бизнес-процес 55
Таблица 4. Оркестратори в Kepler 65
Таблица 5. Компоненти в Kepler 66
Таблица 6. Езици за описание на бизнес-процеси в среди за проектиране и изпълнение в EGI 69
Таблица 7. Среди за проектиране и изпълнение на бизнес-процеси в EGI 70
Таблица 8. ROOT услуги 97
Таблица 9. Адреси на достъп до WSDL описанията на ROOT услуги 97
Таблица 10. g-Lite услуги 102
Таблица 11. Адреси за достъп до WSDL описанията на g-Lite услуги 102
Таблица 12. Профили на регистъра WSRR 107
Таблица 13. Примерен фрагмент от код на Java за извикване на регистъра като Уеб услуга 109
Таблица 14. Общи свойства на дейности в бизнес-процес 114
Таблица 15. Типове на задачи за хора в среда за изпълнение 123
Таблица 16. Бизнес-процеси за изпращане на задача в грид 125
Таблица 17. Описание на таблицата Gridusers 145
Таблица 18. Описание на таблицата Proxy 146
Таблица 19. Описание на таблицата Jobs 146
Таблица 20. Описание на таблицата CAS 147

Увод

Актуалност на проблема

Съгласно [23] (стр.18), гридът е „*софтуерна и хардуерна инфраструктура, предлагаща надежден, последователен, всеобхватен и евтин достъп до високопроизводителни изчислителни ресурси*”. В [24] (стр. 2) авторите допълват концепцията за грид с “*координирано споделяне на ресурси и решаване на задачи в динамични виртуални организации*”. Под споделяне на ресурси авторите имат предвид не просто обмен на файлове, а директен достъп до компютри, софтуер, данни и сензори с цел използването им за решаване на общи научни или индустриални задачи. Споделянето на ресурсите е контролирано. Ясно е дефинирано какъв ресурс се споделя, кой го споделя и какви са условията, при които става споделянето.

В [24] (стр. 2) също е дефиниран и терминът "виртуална организация". Това са "*група от хора, институции и научни организации, които се дефинират според правилата за споделяне на ресурси*". Грид-потребителите принадлежащи към една виртуална организация споделят общи цели и интереси, изследват еднакви задачи и използват един и същ специализиран софтуер за разрешаването на тези задачи.

Реализация на грид е Европейската грид инфраструктура (European Grid Infrastructure - EGI). Тя обединява ресурсите на Европейски научни институти и организации от 24 страни. Основна част на грид инфраструктурата са клъстери от компютри, свързани посредством високоскоростен оптичен интернет, които предоставят изчислителни ресурси и дисково пространство за съхраняване на данни. Достъпът до тях се осъществява посредством специализиран софтуер наречен грид среда (grid middleware). Базовата грид среда инсталирана към момента на всички клъстери от EGI е средата g-Lite (Lightweight Middleware for Grid Computing).

Инфраструктурата EGI наследява грид инфраструктура за научни изследвания, изградена по проекта Enabling Grid for E-sciencE (EGEE). Първоначално грид инфраструктурата е била създадена за да посрещне нуждата от изчислителни ресурси на потребители от областта на Физиката на високите енергии. В последствие EGI е интензивно използвана и от грид-потребители - биолози, био-информатици, химици, ботаници и сеизмолози.

Ролята на Европейската грид инфраструктура е значима. Грид-потребителите използват инфраструктурата, за реализиране на изследователски проекти. Примери за

такива изследователски проекти са WISDOM – откриване на нови лекарства, Climate-G – анализиране на промени в климата, CMS – изследване съществуването на бозон на Хигс, ATLAS - изучаване на основните сили, които формират Вселената.

Проблемите, които потребителите решават в тези изследователски проекти често пъти са свързани с изпълнение на задачи, които протичат на няколко стъпки. Част от тези задачи могат да бъдат автоматизирани, докато друга част изискват човешка намеса.

Последователността от "*логически свързани задачи, разработени за постигането на определен бизнес-резултат*" [17] (стр. 4) наричаме бизнес-процес. Пример за бизнес-процес от изследователски проект в областта на Физиката на високите енергии е процес състоящ се от три стъпки. На първа стъпка се обработват данни, получени в резултат на физичен експеримент. При този тип експерименти, специализиран софтуер записва генерираните данни в EGI. На втора стъпка, съхранените данни подлежат на анализи, за които се използват изчислителните ресурси на грид инфраструктурата. На трета стъпка, резултатите от анализа на данни в грид могат да бъдат визуализирани и подложени на допълнителен анализ на работната станция на потребителя, посредством софтуер за визуализация и анализ на данни – ROOT [86]. Действията по-всяка една от тези стъпки са свързани с конкретен софтуер. Той е различен, за всяка една от стъпките и в повечето случай е вече наличен. В конкретния случай, реализацията на бизнес-процеса изисква интеграция на три различни софтуера.

Ориентираната към услуги архитектура [18] е модел, в който логиката на автоматизация се разделя на добре обособени, независими логически единици, наречени услуги, които могат да бъдат самостоятелно използвани и композирани в бизнес-процеси. Ориентираната към услуги архитектура предоставя подвижна среда, която улеснява интеграцията на наличен софтуер и е адаптивна към промени.

Управлението на бизнес-процеси включва "*методите, техниките и софтуера за проектиране, разработка, контрол и анализ на бизнес-процеси, при които се интегрират хора, организации, приложения, документи и други източници на информация*" [63](стр. 4). В контекста на ориентираната към услуги архитектура, средите за управление на бизнес-процеси предоставят софтуерни средства, които поддържат управление на бизнес-процеси, като покриват пълният жизнен цикъл на ориентираната към услуги архитектура и предоставят функционалност за моделиране,

разгръщане, изпълнение, наблюдение и оптимизиране на бизнес-процесите, чрез поддръжка на стандарти за Уеб услуги.

Гридът се координира от Open Grid Forum (OGF), организация една от целите, на която е разработването на отворени стандарти за грид. Изискванията, които налага OGF обществото е гридът да бъде ориентиран към услуги. Към настоящия момент EGI е частично ориентиран към услуги грид и не предоставя среда за управление на бизнес-процеси в дефинираните до момента термини.

От изключително значение е грид-потребителите да могат да управляват бизнес-процесите си в EGI, без да са необходими познания по програмиране или намеса на специалисти по информационни технологии. Наличието на среда за управление на бизнес-процеси в грид инфраструктурата ще предостави възможност на грид-потребителите да дефинират, проектират, разгръщат, наблюдават и изпълняват бизнес-процесите си в грид, без необходимост от познания по програмиране. Това от една страна, ще позволи грид-потребителите да се концентрират върху естеството и спецификата на самите процеси. А от друга страна, ще скрие сложността от използването на командите за достъп до грид.

Цели на дисертацията

Целта на дисертацията е да се предложи модел и архитектура на среда за управление на бизнес-процеси в грид, в контекста на ориентирана към услуги архитектура. Целта включва, въпросите свързани с дефинирането, проектирането и изпълнението на бизнес-процеси в грид и по специално в Европейската грид инфраструктура. Извън обхвата на изследването остават ред специфични въпроси, като сигурност, ефективност на изпълнение на бизнес-процеса и други.

Задачи на дисертацията

За постигане на тази цел, трябва да бъдат решени следните задачи:

- Изследване на среди за управление на бизнес-процеси и техните архитектури;
- Формулиране на основните изисквания към среди за управление на бизнес-процеси, базирани на ориентирана към услуги архитектура;
- Проучване на съществуващи услуги, които средата EGI предоставя;
- Изследване и анализ на съществуващи среди за проектиране и изпълнение на бизнес-процеси в EGI;
- Разработване на модел на среда за управление на бизнес-процеси в EGI;

- Проектиране на архитектурно решение на среда за управление на бизнес-процеси в EGI, базирана на разработения модел;
- Проектиране и реализиране на примерни композитни Уеб услуги за изпълнение на задачи в EGI;
- Проектиране и реализиране на примерни композитни Уеб услуги към компоненти от софтуер на Физиката на високите енергии и по специално софтуера ROOT;
- Реализиране на примерен бизнес-процес на базата на реализираните Уеб услуги.

Кратко съдържание на работата

В Глава 1 са разгледани основните концепции на грид и ориентирана към услуги архитектура. Представени са принципите на ориентирания към услуги подход, както и реализация на архитектурата чрез Уеб услуги. Представени са стандартите за Уеб услуги. Дефинирани са понятията услуга, бизнес-процес, управление на бизнес-процеси. Формулирани са основните изисквания към среди за управление на бизнес-процеси, базирани на ориентирана към услуги архитектура. Изследвани са среди за управление на бизнес-процеси и е изведен обобщен модел на среда за управление на бизнес-процеси, базирана на ориентирана към услуги архитектура. Представен е стандарта за ориентиран към услуги грид – OGSA и някои негови реализации. Проучени са съществуващите услуги, които грид средата g-Lite предоставя. Поставени са някои изследователски въпроси свързани с проектирането и изпълнението на бизнес-процеси в EGI. Изследванията от Глава 1 са публикувани в [1].

В Глава 2 са проучени и представени съществуващите среди за проектиране и изпълнение на бизнес-процеси в EGI. На базата на дефинираните в Глава 1 изисквания е направен сравнителен анализ на средите. Обобщени са предимствата на средите и са отличени причините, по които средите не отговарят на изискванията за управление на бизнес-процеси в контекста на ориентирана към услуги архитектура. Изследванията от Глава 2 са публикувани в [31].

В Глава 3 е представен модел на среда за управление на бизнес-процеси в Европейската грид инфраструктура. Разгледани са основните модули на средата: регистър на услуги, среда за разработка и среда за изпълнение и компонентите им по отношение на грид инфраструктурата EGI. Дефинирани са роли и случаи на употреба в средата, както и някои допълнителни изисквания. Проектирано е архитектурно

решение на среда за управление на бизнес-процеси в EGI. Изследванията от Глава 3 са публикувани в [30].

В Глава 4 са показани възможности за използване и приложения на средата за управление на бизнес-процеси в EGI. Разгледани са някои аспекти от реализацията на средата. Демонстрирано е приложение на средата за управление на бизнес-процеси в грид инфраструктурата. Проектирани са примерни композитни Уеб услуги за изпълнение на задачи в EGI и примерни композитни Уеб услуги към софтуера ROOT. Показано е приложение на разработените Уеб услуги композирани в бизнес-процеси. Демонстрирани са примерни бизнес-процеси за изпълнение на задачи в грид и бизнес-процес от областта на Физиката на високите енергии. Изследванията от Глава 4 са публикувани в ([32], [33]).

В Заключение, са обобщени резултатите от дисертационния труд и са представени някои от предимствата на използване на средата.

В Приложения са показани екрани от релационния модел на данни използван при реализацията на модул от средата, екрани от настройки на средите за проектиране и разгръщане, екрани на разработени бизнес-процеси и екрани от работно пространство на грид-потребителя.

Глава 1. Обзор

В изследователските проекти от областта на Физиката на високите енергии се провеждат физични експерименти. При този тип експерименти, се използват високо енергийни ускорители за ускоряване на елементарни частици. В резултат на физичния експеримент се генерират голям брой събития, от които едва малка част са значими. За да се определи, кои от тях са значими са необходими значителни дискови ресурси за съхраняване на данните и изчислителни ресурси за анализ на съхранените данни. Примери за такива изследователски проекти [3] са ATLAS (A Toroidal LHC ApparatuS) - изучава структурата на материята и основните сили, които формират Вселената, CMS (Compact Muon Solenoid) – изследва съществуването на бозон на Хигс и доказателство за супер-симетрията, NA62 - измерва много рядък каонен разпад.

Други примери, извън областта на Физиката на високите енергии, при които се налага използване на голяма изчислителна мощ и дисково пространство са задачи от областта на био-информатиката и геологията. В изследователските проекти от био-информатиката, например WISDOM, се извършват симулации и анализи на данни за откриване на нови лекарства. В изследователските проекти от областта на геологията, например Climate-G, се извършат анализ и визуализация на данни свързани с промените в климата. Отново за съхраняване и обработката на данни са необходими значителни дискови и изчислителни ресурси, които нито една институция или организация не може да осигури.

Тези изисквания, налагат използването на инфраструктура, която да предоставя надежден достъп до огромни изчислителни ресурси и дисково пространство. Такава инфраструктура е гридът.

1.1 Грид

Гридът е софтуерна и хардуерна инфраструктура, разпределена между различни институции и организации, която предоставя достъп до изчислителни ресурси, дискови пространства за съхраняване и трансфер на данни, услуги и сензори. Гридът не е обект на централизиран контрол, т.е. достъпът до ресурсите не се контролира от една институция или организация. Споделянето на ресурсите в грид е координирано. Ясно е дефинирано какъв ресурс се споделя и какви са условията, при които става споделянето.

Групата от хора, институции и научни организации, които се дефинират според правилата за споделяне на ресурси се нарича виртуална организация. Потребителите, принадлежащи на една виртуална организация, споделят общи цели и интереси, изследват еднакви проблеми и използват един и същ специализиран софтуер, за разрешаването на тези проблеми. Пример за виртуални организации са:

- Множеството от всички потребители на грид от Югоизточна Европа (South East Europe Grid - SEEGRID);
- Множеството от всички потребители на грид, които се занимават с анализ на данни и симулация на събития от физиката на високите енергии (ATLAS, CMS);
- Множеството от всички потребители на грид, които изследват и анализират биохимични процеси (BIOMED).

Един потребител може да бъде член на повече от една виртуална организация, според ролята и задачите, които изпълнява. Съгласно [25], една система е грид, ако отговаря на следните критерии:

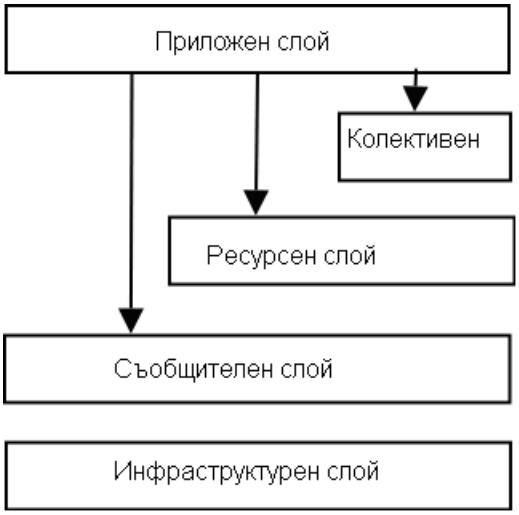
- Да предоставя координирано използване на ресурси, които не са обект на централизиран контрол;
- Да използва отворени стандарти, протоколи и интерфейси;
- Да предоставя нетривиални качествени услуги.

Гридът е система, която интегрира и координира ресурси и потребители, които са част от различни административни области. По тази причина и контролът върху тези ресурси е децентрализиран, но координиран. Всяка организация има собствени механизми за сигурност и достъп до ресурсите. Гридът не заменя тези механизми, а се интегрира с тях.

Гридът е и система, изградена от протоколи и интерфейси, които реализират удостоверяване, даване на право за достъп до ресурс, откриване на ресурси и достъп до самия ресурси. Тези протоколи трябва да са стандартни и отворени, за да бъдат поддържани от всяка една организация.

Гридът позволява част от ресурсите да бъдат използвани по координиран начин с цел доставяне на качествени услуги. Под качествени услуги визираме, винаги налични, сигурни и с добро време на отговор услуги. Целта на грид е да посрещне нуждите на потребителя, така че ползата от достъпа до инфраструктурата да е по-голяма от индивидуалния достъп до отделните ресурси.

Представената от [23] архитектура на грид е протоколна. Авторите дефинират архитектурата, като специфицират основни системни компоненти и връзката между тях. Протоколите дефинират базови механизми, за взаимодействие на потребителите с ресурсите в грид. Предложената архитектура на грид на Фиг. 1 се състои от пет слоя.



Фиг. 1. Слоеве на архитектура на грид

Инфраструктурният слой предоставя изчислителни ресурси, системи за съхраняване на данни, каталози, мрежови ресурси и сензори. Ресурсите са логически единици. Достъпът до тях се осъществява посредством грид-протоколи. Ресурс може да бъде и разпределена файлова система, изчислителен клъстер и други.

Съобщителният слой предоставя основните комуникационни протоколи и протоколи за удостоверяване, които са необходими за достъп до ресурсите. Комуникационните протоколи, позволяват обмена на данни между ресурсите от инфраструктурния слой. Протоколите за удостоверяване предоставят механизми за сигурност при достъпа до ресурси. Тези механизми включват верифициране на потребителите и техните права и верифициране на ресурсите. Грид-потребителите инициират задачи, които изискват достъп до множество от ресурси. Основно изискване към съобщителния слой е да предоставя възможност грид-потребителя да се удостовери само веднъж и да има достъп до ресурсите. Тази възможност се нарича единична точка на свързване (single sign-on). Други изисквания са слоят да предоставя функционалност за делегиране на права и възможност за интеграция с локални решения за сигурност.

Ресурсният слой предоставя протоколи за управление на ресурсите и информационни протоколи. Информационните протоколи се използват за предоставяне на информация за структурата и статуса на ресурсите – конфигурации, политики за

достъп, натоварване и други. Протоколите за управление се използват за да специфицират достъпа до споделените ресурси и операциите, които могат да бъдат извършвани над тях.

Колективният слой предоставя услуги и протоколи, които не са свързани с конкретен ресурс, а с взаимодействието в група от ресурси. Пример за такива услуги са: директориини услуги, брокер-услуги, услуги за наблюдение и диагностика, услуги за репликация на данни, услуги за откриване на софтуер и системи за управление на работни потоци. Колективният слой също предоставя и приложно-програмни интерфейси (Application Programming Interface - API) и средства за разработване на софтуер (Software Development Kit - SDK), които грид-приложенията могат да използват.

Приложният слой обхваща приложения, разработени посредством приложно-програмните интерфейси, за достъп до грид-ресурси. Приложенията могат да използват, която и да е услуга или протокол от по-долните слоеве.

Трябва да отбележим, че в своето естество гридът е динамична инфраструктура. Предоставените ресурси не винаги са налични. Възможно е нови ресурси да се включват и изключват, по всяко време от инфраструктурата. Въпреки това, в повечето изградени грид инфраструктури, например EGI, се разчита именно на постоянната наличност и надеждност на ресурсите (24 часа, 7 дни в седмицата).

Гридът се координира от OGF, организация една от целите, на която е разработването на отворени стандарти за грид. Особена роля има стандартът на архитектурата за отворени грид-услуги (Open Grid Service Architecture - OGSA) [27], която ще разгледаме по-натам.

Софтуерът, който предоставя възможност за управление и достъп до ресурсите в грид се нарича грид среда (grid middleware). В зависимост от начина, по който е реализирана, грид средата може да бъде ориентирана към услуги, частично ориентирана към услуги или не поддържаща ориентация към услуги.

1.2 Ориентирана към услуги архитектура

Ориентираната към услуги архитектура [18] е архитектурен стил, който се прилага при изграждане на системи и приложения. Основна характеристика на модела е разделяне на логиката на автоматизация на добре обособени логически единици, които могат да бъдат самостоятелно използвани и независими една от друга. Отделната логическа единица трябва да удовлетворява набор от принципи позволяващи

самостоятелното й съществуване. В термините на ориентираната към услуги архитектура, такива логически единици се обозначават като услуги.

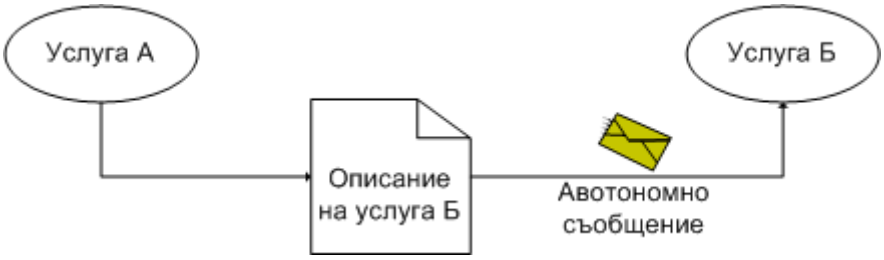
Дефиниция. *Услугите са функции или операции достъпни през мрежата. Те имат добре дефинирани интерфейси, добре дефинирана функционалност и изисквания към качеството на услугата (Quality of Service - QoS), както и добре определени крайни точки на достъп.* [2] (стр. 19)

Услугата се дефинира посредством описание. В описанието се съдържа информация за протоколите за достъп до услугата, името на услугата и типа на данните, които може да обменя при взаимодействие. Услугите притежават следните характеристики:

- *Автономност* – услугите са отделни единици, които могат да бъдат самостоятелно използвани;
- *Съставност* – услугите могат да бъдат интегрирани в композиция от услуги;
- *Изолираност* – услугите могат да бъдат напълно независими от други услуги;
- *Разширяемост* – услугите могат да допълват логиката, която капсулират;

Взаимодействието между услугите се осъществява, посредством обмен на съобщения. Съгласно [18](стр. 35) комуникацията между услугите трябва да бъде минимизирана за сметка на големината на съобщението. По този начин се предотвратява претоварване на комуникационната подсистема. В този смисъл, когато една услуга изпрати съобщение, тя няма контрол върху това какво се случва със съобщението, което от своя страна означава, че съобщенията също са автономни.

На Фиг. 2 [18](стр. 35) е представен механизма на взаимодействие между две услуги.



Фиг. 2. Взаимодействие на услуги в ориентирана към услуги архитектура

Услугата Б, предоставя определена функционалност. Информацията, каква функционалност предоставя услугата и как тя може да бъде извикана от *Услугата А*, се

предоставя от описанието на *Услугата Б*. Средата, в която си взаимодействат услугите, предоставя стандартни механизми за откриване на услуги. Това могат да бъдат регистри, директории или други услуги. Посредством тях, *Услугата А* може да открие описанието на *Услугата Б* и да разбере каква функционалност излага *Услугата Б.*

Посредством обмен на автономни съобщения, чиито формат също е дефиниран в описанието на *Услугата Б, Услугата А* изпраща данни, получава данни и извика конкретна функционалност на *Услугата Б*. Единственото, което свързва двете услуги е описанието на *Услугата Б*. Това позволява функционалността, която *Услугата Б* предоставя да бъде допълвана, без това да се отрази на *Услугата А*.

Важно е да отбележим, че при архитектурата ориентирана към услуги, преди да взаимодействат, услугите нямат информация една за друга. Тази информация се получава в момента на взаимодействие. Това свойство е основна характеристика на архитектурата и се обозначава като слаба свързаност.

1.2.1 Принципи на ориентирания към услуги подход

Това, което отличава ориентираната към услуги архитектура от други подобни архитектури е начинът на проектиране на трите основни елементи на модела: услуга, описание и съобщение. Ориентираният към услуги подход, предполага спазването на определени правила, чрез които се дефинира позицията и проектирането на услугите. Съгласно [19] подходът се базира на следните принципи:

Договор

Услугите изразяват целите си и функционалността, която капсулират, посредством *Договор*. Този принцип гарантира, че логиката на услугите и функционалността, която те излагат е описана последователно. *Договорът* на услугата се състои от група от документи, всеки от които описва част от услугата. Той се споделя между услугите. От голямо значение при проектирането на *Договора* е прецизно да се опише услугата, като се специфицират семантиката, типовете данни и политиките за достъп на услугата.

Слаба свързаност

Този принцип позволява логиката, която услугите капсулират, да бъде променяна без това да се отрази или минимално да повлияе върху другите услуги. Слабата свързаност гарантира независимо проектиране и допълване на логиката на

услугата. Това позволява, услугите да бъдат по-лесно композирани и повторно използвани.

Абстракция

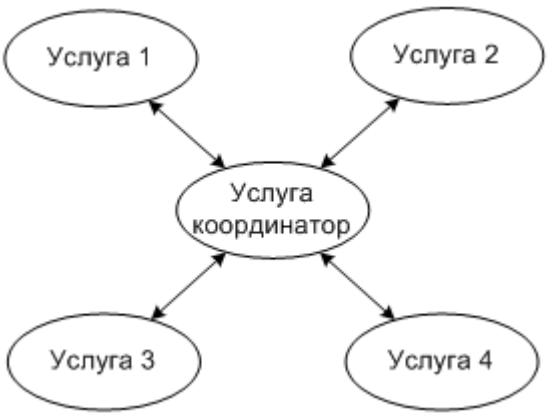
Прилагането на този принцип гарантира капсулиране на логиката на услугата. *Абстракцията*, позволява услугите да се разглеждат като черни кутии, като скрива детайлите от външния свят и показва определена функционалност. Единственият документ, който съдържа достъпната и публична информация за функционалността на услугата е *Договорът*. Така, чрез спазване на принципа за *Абстракция*, услугите изнасят публично, единствено логиката, която е описана в документите на *Договора*, а всяка друга логика е скрита и недостъпна отвън.

Композиция

Способността за изграждане на композиция от услуги е изключително важна, особено при разработването на по-сложни приложения. Този принцип гарантира, че услугите могат да участват в композиция. Тази композиция от услуги също е услуга. Процесът на композиране на услуги може да е рекурсивен. Той може да включва както съществуващи услуги, така и композиции от услуги.

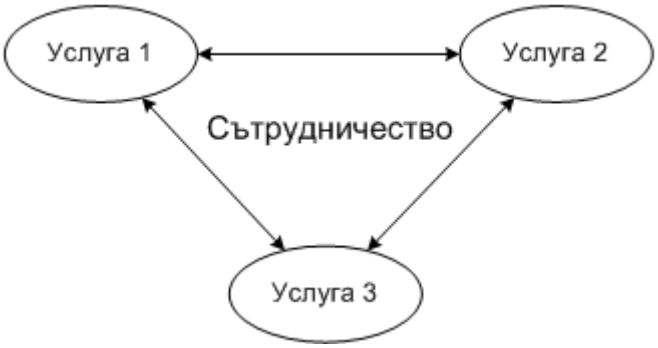
Съгласно [41] (стр. 20), при композицията се различават два модела на организиране на съставните услуги: оркестрация и хореография.

Оркестрацията на Фиг. 3 е композиция, при която имаме централизиран контрол, който се осъществява от услуга координатор. При оркестрацията услугата за композиране контролира съставните услуги и координира изпълнението на операциите им. Координаторът се грижи и за реда на извикване на услугите. Тъй като координатора също е услуга, той може да взаимодейства и с други услуги.



Фиг. 3. Оркестрация

Хореографията на Фиг. 4 е композиция, при която имаме съвместно сътрудничество между съставните услуги, което се базира на обмен на съобщения. При хореографията нямаме централизирана услуга координатор. Вместо това всяка съставна услуга е отговорна за извършването на определена дейност и за координирането й с останалите услуги от композицията. За разлика от оркестрацията, при хореографията, контролът е споделен между отделните участници в композицията. Така, всяка услуга от композицията носи информацията за останалите съставни услуги.



Фиг. 4. Хореография

Многократна употреба

Този принцип подчертава позицията на услугите, като ресурс с функционален контекст. Принципът за многократна употреба, задължава услугите да бъдат проектирани по такъв начин, че да могат да бъдат използвани повече от веднъж.

Свойството на услугите да бъдат повторно употребявани рефлектира и върху другите принципи. Спазването на принципа гарантира, че различните услуги няма да капсулират една и съща функционалност.

Автономност

За да могат услугите, да изнесат надеждно и последователно логиката, която капсулират, те трябва да могат да осъществяват контрол върху средата и ресурсите, които използват. Обикновено логиката, която услугата управлява, е в определени граници. Принципът за автономност гарантира, че услугата е самостойна и има независима, изолирана логика от логиката на останалите услуги.

Често пъти услугите използват обща среда и общи ресурси. По тази причина могат да бъдат разгледани два типа автономност: на ниво услуга и пълна автономност.

Автономност на ниво услуга се наблюдава, когато услугите споделят общ ресурс. Например услуги, които капсулират функционалност, която използва автономен ресурс. Автономен ресурс е ресурс, чието използване не е обвързано с

участието му в дадена услуга, който може да се използва независимо и който е съществувал преди създаването на услугата. Автономността на ниво услуга гарантира, че услугата има пълен контрол върху функционалността, която предоставя и частичен върху автономния ресурс.

Пълна автономност се наблюдава, когато лежащата отдолу логика е под пълен контрол и собственост на услугата. Това е типично за случаите, когато лежащата отдолу функционалност не се основава на наследен софтуер и се поддържа изцяло от услугата.

Липса на състояние

Този принцип гарантира, че услугите не съхраняват информация за състоянието си. Състоянието на услугите се представя чрез състояние на променливи и зависи от работата на услугата преди това. Управлението на прекомерна информация и съхраняване на състояния за дадена операция, може да доведе до понижаване качеството на услугата. Също така би било неадекватно, услугата да реагира по различен начин при всяко следващо извикване. По тази причина, за предпочитане е при проектиране на услугите, принципът да бъде спазен и услугите да не запазват състояние, дори и ако това означава управлението на състоянията за тази услуга да се изнесе извън услугата. Ако една услуга е отговорна за запазване на състоянието за по дълъг период от време, тя ще бъде по-трудно достъпна за другите услуги. Спазване на принципа гарантира спазване и на принципа за многократна употреба.

Откриваемост

Принципът за *откриваемост* предполага услугите да се проектират с такова описание, че да могат да бъдат лесно откривани посредством налични в средата механизми за откриване на услуги. Както вече беше споменато това могат да бъдат регистри, директории или други услуги. Независимо обаче от съществуващите механизми на средата, принципът трябва да гарантира, че услугата е описана с достатъчно информация, така че другите услугите да знаят как да я намерят и извикат. Това условие е силно зависимо от *Договора* на услугата, който съдържа пълната информация за услугата.

Принципите на ориентираната към услуги архитектура са тясно свързани един с друг. Неспазването на един от тези принципи може да доведе до невъзможност да бъдат изпълнени и останалите принципи.

Например, ако не се спази принципът за *Абстракция* и не се скрие част от функционалността, която услугата предоставя, това ще доведе до зависимост на другите услуги от промените върху функционалността на услугата. Зависимостта от друга страна, ще доведе до неизпълнение на принципа за *Слаба свързаност*, а нарушаване на принципа за *Многократна употреба*, до нарушаване на принципа за *Композиция*. Спазването на принципите за ориентираност към услуги, при изграждането на системи, приложения или архитектурни решения, гарантира гъвкавост, платформена независимост и възможност за разработване на ориентирани към услуги решения.

1.2.2 Архитектура на Уеб услуги

Една от най-широко разпространените реализации на ориентираната към услуги архитектура е чрез Уеб услугите. Архитектурата на Уеб услугите е специфицирана в документ Web Service Architecture [10], разработен от работна група на W3C консорциум.

Дефиниция. *Уеб услуга наричаме софтуерна единица, разработена да взаимодейства с други единици, необвързано от платформа и език. Уеб услугата е описана посредством документ, който специфицира интерфейсите ѝ. Взаимодействието с Уеб услугата, става посредством обмен на съобщения.* [10]

Уеб услугата е описана абстрактно и се реализира чрез конкретен код. В софтуерния код е специфицирано как услугата може да получава и изпраща съобщения, а самата услуга е представена, като едно абстрактно множество от операции. Така една и съща услуга (множество от операции) може да бъде реализирана чрез различен код (езици за програмиране). Този подход гарантира запазване на нивото на абстракция на услугата и, че при промяна на кода услугата няма да се измени. Друго предимство на подхода е, че така дефинирана услугата поддържа независимост от платформа и език.

В зависимост от контекста, в които се използва Уеб услугата тя може да приема различни роли. Например, една услуга може да е инициатор на съобщение, да е посредник или да получава съобщение. Не е необичайно, Уеб услугите да сменят ролите си. В едни случаи, услугата, може да предоставя функционалност, а в други да се нуждае от функционалността, която другите услуги предоставят.

В среда за обмен на съобщения при процеса на взаимодействие услугите могат да бъдат в две основни роли: на доставчик или на консуматор. Една Уеб услуга може да бъде и в едната и в другата роля повече от веднъж.

Доставчикът на услугата публикува описанието на услугата, като предоставя информация за нейните свойства и поведение. Консуматорът открива услугата и я извиква. Взаимодействието между доставчика и консуматора се осъществява по следния механизъм на Фиг. 5.

Фиг. 5. Механизъм на взаимодействие между доставчик и консуматор

Уеб услугата в ролята на консуматор открива услуга, предоставяща дадена функционалност. В зависимост от механизмите на средата за откриване на услуги, консуматорът ще се обърне към регистър, директория или друга услуга. Посредством описанието на услугата, консуматора разбира къде е локализирана услугата, какви входни параметри изисква и как може да я извика. Чрез обмен на съобщения, консуматорът извиква услугата.

В архитектурата на Уеб услугите понятията доставчик и консуматор са дефинирани по-абстрактно. Така доставчикът е лице или организация, които предоставя подходящ код за реализирането на конкретна услуга. Консуматорът е лице или организация, които имат нужда от Уеб услугата, която доставчика предоставя. Обменът на съобщенията между консуматора и доставчика се осъществява между техните кодове. В повечето случаи кода на консуматора инициира обмяната на съобщения.

На Фиг. 6 са представени четирите слоя от архитектурния стек на Уеб услугите.

Фиг. 6. Архитектурен стек на Уеб услуги

Първият слой – *Комуникация* е среда за обем на съобщения между доставчика и консуматора. Протоколът, който най-често се използва за тази част на архитектурата е Hypertext Transfer Protocol (HTTP) [22].

Вторият слой – *Съобщения* отговаря за извикване на услугата, посредством трансфер на съобщения между консуматора и доставчика. Протоколът, който се използва най-често за тази част на архитектурата е Simple Object Access Protocol (SOAP) [34].

Третият слой – *Описания* предоставя описания на услугите и механизми за откриване на описанията на тези услуги. Стандартите, който се използват в тази част на архитектурата са WSDL (Web Service Definition Language) [16] и UDDI (Universal Description Discovery and Integration)[9].

Четвъртият слой – *Процеси* обхваща действия върху повече от една услуги. Например оркестрация на услуги, хореография на услуги и др. Стандартите, който се използват в тази част на архитектурата са WS-BPEL [40] и WS-CDL [44].

Допълнителни аспекти в архитектурният стек на Уеб услугите са сигурност и управление на Уеб услуги. Сигурността на Уеб услуги, може да бъде постигната, чрез XML-базирани механизми за сигурност. Те трябва да решават проблемите с идентификацията, контрол на достъп до услугите, прилагане на политики за сигурност, сигурност на ниво съобщения и други. Управлението на Уеб услуги включва, наблюдение, контрол на качеството на услугите, надеждност на услугите и други.

В този контекст, трябва да отбележим, че има и допълнителни спецификации за Уеб услуги. Такива например са WS-Security – който се грижи за целостността и сигурността на съобщенията, WS-ReliableMessaging – който отговаря за доставянето на

надеждни съобщения и WS-Policy - който позволява дефинирането на политики, имащи отношение към качеството на услугите.

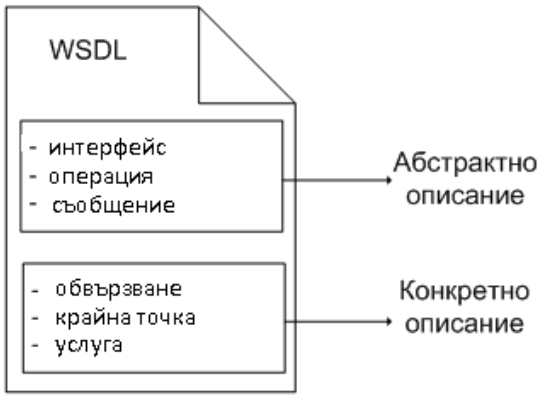
Ориентираната към услуги архитектура, реализирана чрез архитектурата за Уеб услуги предоставя един стандартизиран модел за изграждане на гъвкави, надеждни, лесно разширяеми и платформено независими приложения.

1.2.3 Стандарти за Уеб услуги

Общото между стандартите за Уеб услуги е езикът XML [12]. XML е отворен стандарт разработен от W3C, който дефинира правила за създаване на специализирани маркиращи езици и синтаксис, на който тези езици трябва да се подчиняват. XML указва как да бъде структуриран един документ и е широко приложим за съхраняване, форматиране и предаване на данни. В контекста на Уеб услугите, XML предоставя стандартен, гъвкав и разширяем формат на данни, който значително намалява различията между технологиите.

WSDL

WSDL е XML базиран и описва интерфейсите, които услугата предоставя, формата на съобщенията, както и физическото местоположение на услугата или нейните крайни точки. Описанието на услугата [18](стр. 133) може да бъде разделено на две категории: абстрактно описание и конкретно описание Фиг. 7. Абстрактното описание дефинира характеристиките на интерфейса на Уеб услугата, без да го обвързва с конкретна технология. Така се запазва целостта и независимостта на описанието на услугата от промените, които са възможни да настъпят върху лежащата отдолу реализация. Абстрактното описание се състои от три основни части: *интерфейс*, *операция* и *съобщение*.



Фиг. 7. WSDL документ

Интерфейсът (*portType*) е абстрактно множество от операции, които услугата поддържа. Операциите сами по себе си са абстрактно описание на функционалността, които услугата предоставя. Операциите имат входни и изходни параметри. Тъй като Уеб услугите взаимодействат помежду си само чрез съобщения, тези параметри се представят като съобщения. За това една операция използва множество входни или изходни съобщения. Съобщенията са абстрактно описание на данните, чрез които услугата комуникира.

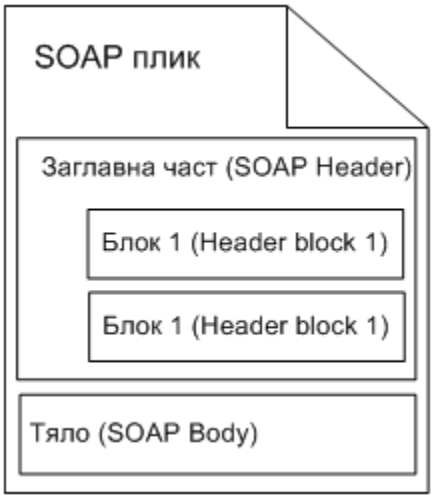
Конкретното описание на услугата се състои от три части: *обвързване*, *крайна точка* и *услуга*. Обвързването се осъществява чрез конкретен протокол за транспорт и формат на данните описани в интерфейса на услугата. Препоръчва се при реализацията на обвързване за дадена услуга да се използва SOAP протокола. Обвързване може да се приложи както към целия интерфейс, така и към конкретна операция. Крайната точка представлява физическия адрес, на който услугата може да бъде достъпна, а услугата е набор от свързани крайни точки.

Описанието на Уеб услугата не зависи от език и платформата. Самите Уеб услуги могат да бъдат реализирани чрез други технологии като COM, JMS или CORBA. Широкото разпространение на Уеб услугите, по начина по които са специфицирани в архитектурата за Уеб услуги се дължи на употребата на стандартни технологии и протоколи. Затова и най-често Уеб услугите се реализират именно чрез WSDL и SOAP върху HTTP.

SOAP

Както беше отбелязано, взаимодействието между услугите се базира на обмен на съобщения. По тази причина, изборът на среда за обмен на съобщения е от изключително значение. Инфраструктурата за обмен на съобщения от друга страна също трябва да е изключително гъвкава и разширяема.

Технология, която отговаря на изброените изисквания е SOAP. Стандартен протокол за пакетиране и обмен на XML съобщения - SOAP. В спецификацията на протокола [34] се дефинира стандартен формат за съобщения. Структурата на този формат е проста, но предоставя възможност за разширяване и промени. SOAP съобщенията са пакетирани в контейнери, наречени пликове и се състоят от заглавна част и тяло на Фиг. 8 [18](стр. 143).



Фиг. 8. SOAP съобщение

Заглавната част е област от съобщението, запазена за съхраняване на мета информация. Действителното съобщение се намира в тялото.

SOAP съобщенията съдържат информация, свързана с доставянето и обработката на съдържанието на съобщението. Тази информация може да включва действия засягащи сигурността на съобщението, данни за оркестрацията или хореографията.

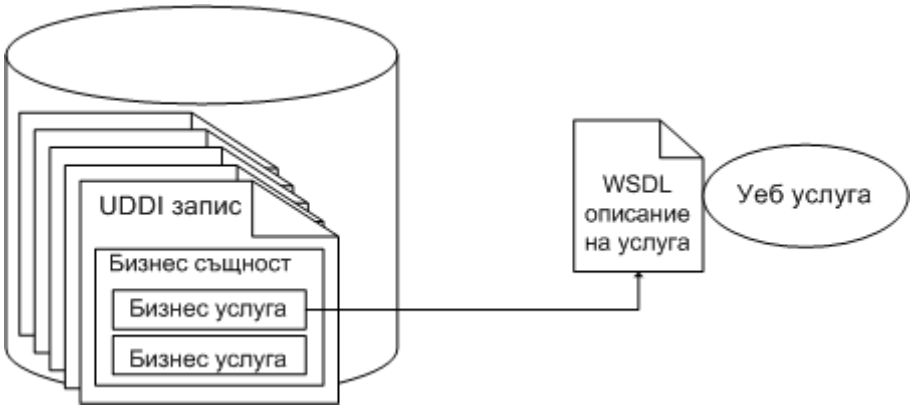
UDDI

Един от подходите за реализиране на механизъм за откриване на услуги, споменат в архитектурата за Уеб услуги е регистъра. Регистърът е хранилище под централизиран контрол за съхранение на информация. Публикуването на описание на услуга изисква действия от страна на доставчика на услугата – той трябва явно да публикува информация в регистъра, преди тази информация да стане достъпна за другите. Собственикът на регистъра, решава кой има права да публикува или обновява информацията в регистъра. Според това, регистърът може да бъде публичен или частен. Публичните регистри позволяват публикуване на услуги от всяка организация, докато частните публикуване на услуги само в рамките на организацията. Стандартът [9], който се използва за реализиране на регистър е UDDI.

UDDI е регистър от описания на услуги. Позволява публикуване и търсене на услуги. Текущата версия на UDDI (UDDI 3.0), позволява различни нива на видимост на услугите. Услугите в регистъра, могат да бъдат публични и частни.

В регистъра информацията се съхранява под формата на записи на Фиг. 9 [18](стр.138). Тези записи се съхраняват в XML формат. Всеки запис в регистъра се

състои от бизнес същност, която съдържа основна информация за организацията или доставчика на услугата. В нея също се съхранява и информация за бизнес услугите, които доставя съответната организация.



Фиг. 9. UDDI Регистър

Бизнес услугите могат да представят действителна Уеб услуга. Това става посредством структура tModel, която съдържа указател към описанието на услугата.
В регистъра се пазят всички версии на публикувани услуги. Регистърът може да бъде достъпен чрез приложно-програмни интерфейси.

WS-BPEL

Езикът WS-BPEL или накратко BPEL, е XML - базиран език за описание и изпълнение на бизнес-процеси в ориентирана към услуги среда.

Дефиниция. *Бизнес-процесът е последователност от координирани извиквания на услуги и свързаните с тях дейности, в рамките на една или няколко организации, която реализира конкретна бизнес цел.* [42] (стр. 8)

Езикът BPEL следва модела на оркестрация. Услуга-координатор координира изпълнението на всяка една от услугите в композицията. Често тази услуга се нарича и BPEL (процесен) сървър.

Езикът BPEL е стандартизиран от организацията OASIS. Първата версия на езика BPEL4WS 1.0 е от 2002 г. и се базира на езиците WSFL (разработен от IBM) и XLANG (Разработен от Microsoft). Във версия 2.0, езикът е преименуван на WS-BPEL и одобрен през 2007 г., като OASIS стандарт.

Посредством езикът BPEL могат да бъдат описани и изпълнявани композиции от услуги. Езикът е разширение на WSDL. Така, всеки BPEL бизнес-процес е и Уеб услуга. Според спецификацията на езика [40], BPEL предлага поддръжка както за изпълними, така и за абстрактни процеси. Изпълнимите бизнес-процеси могат да бъдат изпълнени посредством BPEL сървър. При абстрактните бизнес-процеси се описва публичния обмен на съобщения между участниците в процеса и някои външни аспекти от поведението на процеса. Абстрактните бизнес-процеси не са изпълними. Поради това те и рядко се използват. Абстрактни бизнес-процеси, могат да бъдат описани и чрез езика WS-CDL, разгледан в следващия раздел.

Фиг. 10. WS-BPEL документ

Основната структура на един BPEL документ на Фиг. 10 е както следва:

- Процес (**<process>**)
  - Връзки с партньори (**<partnerLinks>**). Партньори се наричат всички Уеб услуги, с които си комуникира BPEL процесът. Връзките с партньори описват начина на комуникация между процеса и неговите партньори;
  - Променливи (**<variables>**) - служат за съхраняване на текущото състояние на процеса и обменяните данни;
  - Обработки на грешки (**<faultHandlers>**) - набор от дейности, които се изпълняват при възникване на грешка;
  - Обработки на събития (**<eventHandlers>**) - набор от дейности, които се изпълняват при възникване на събитие;

- Компенсаторна обработка (**<compensationHandler>**) – набор от дейности, които се стартират от самия процеса за да бъдат отменени стъпки, които вече са завършили или за да се дефинират алтернативни стъпки, които да бъдат изпълнени;
- Обработка на необичайни прекъсвания (**<terminationHandler>**) – набор от дейности, които се изпълняват при настъпване на необичайно прекъсване на процеса;
- Корелации (**<correlationSets>**) – дефинира зависими дейности в процеса;
- Област (**<scope>)** – използва се за да дефинира вложени дейности в гореизброените елементи;
- Дейности – определят действията в процеса и хода му на изпълнение. Разделят се на основни и структурни дейности. Структурните дейности, определят реда на изпълнение на вложените в тях дейности.

В (Таблица 1) са представени основните и структурните дейности на езика WS-BPEL.

Основни дейности	
<receive>	дейност, която позволява бизнес процесът да чака пристигането на определено съобщение. Дейността завършва, когато пристигне съобщението.
<reply>	дейност, която позволява бизнес процесът да изпрати съобщение в отговор на <receive>
<invoke>	дейност, която позволява бизнес процесът да извика операция на Уеб услуга, която е дефинирана като партньор. Извикването може да бъде синхронно (заявка - отговор) или асинхронно (без да се очаква отговор)
<assign>	дейност, която присвоява стойност на променлива. Чрез нея могат да бъдат копирани данни от една променлива в друга
<throw>	дейност, която позволява бизнес-процесът да генерира грешка
<rethrow>	дейност за препращане на грешка. Използва се при обработки на грешки
<exit>	дейност, която прекратява бизнес-процесът, без да извършва обработки на грешка или събития
<wait>	дейност, която позволява бизнес-процесът да изчаква определен период от време преди да продължи

<pick>	дейност, която позволява бизнес-процесът да изчака пристигането на едно или няколко съобщения до определено време. След изтичането на времето, дейността приключва
<empty>	празна дейност в бизнес-процеса. Използва се при синхронизация на паралелни дейности
Структурни дейности	
<sequence>	дефинира колекция от дейности, който се изпълняват последователно
<flow>	дефинира колекция от дейности, който се изпълняват паралелно
<if>	дейност, която от множество от дейности, избира точна една, която да бъде изпълнена, според определено условие
<while>	дефинира дейност, която ще бъде повтаряна, докато определеното условие за край на дейността е истина
<repeatUntil>	дефинира дейност, която ще бъде повтаряна, докато определеното условие за край на дейността стане истина. Дейността се изпълнява поне веднъж
<forEach>	дефинира дейност, която се повтаря точно N пъти, където N е фиксирано число
Други дейности	
<validate>	използва се за проверка на стойност на променлива според зададена XML Schema

Таблица 1. Дейности в WS-BPEL

Езикът BPEL позволява интегрирането на Уеб услуги в бизнес-процеси. Процесът е колекция от дейности, които извикват конкретни Уеб услуги. Езикът BPEL позволява създаването на изпълними бизнес-процеси. Описанието на бизнес-процес описан, чрез езика BPEL е и Уеб услуга. Така, всеки бизнес-процес дефиниран чрез езика BPEL, може да бъде извикван и като Уеб услуга.

WS-CDL

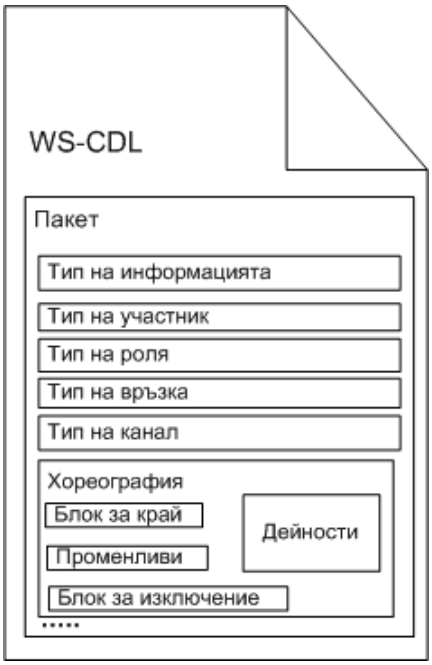
Езикът WS-CDL (Web Service Choreography Description Language) [44] следва модела на хореография. При хореографията услугите в композицията си сътрудничат за да се постигне конкретна бизнес цел.

Езикът WS-CDL специфицира декларативен XML базиран език, който да описва peer-to-peer колаборация между отделните участници. Езикът описва общото и

допълнително поведение на участниците в колаборацията, при която последователният обмен на съобщения води до постигане на обща бизнес цел. В спецификацията на езика WS-CDL са зададени и допълнителни изисквания, част от които са езикът да позволява:

- Многократна употреба - една и съща дефиниция на хореография да може да бъде използвана от различни участници в различни контексти и с различен софтуер.
- Съставност – Съществуващи хореографии да могат да бъдат комбинирани, за да формират нова хореография.
- Сътрудничество – Хореографията да дефинира реда на обмен на съобщения между два или повече участника, като описва взаимодействието по между им.
- Множествено сътрудничество - Хореографията да може да включва произволен брой участници или процеси.
- Семантика – Хореографията да може да включва документация и семантика за участниците.

В [44] е представен модел на езика WS-CDL. Документът в WS-CDL е множество от дефиниции, обединени в Пакет (Choreography Package), който се състои от следните елементи на Фиг. 11:



Фиг. 11. WS-CDL Пакет

- Променливи (variables), съдържат информация за наблюдавани обекти в колаборацията;

- Символи (tokens), представляват псевдоними, които могат да бъдат използвани като референция към части от Променливите;
- Тип на информация (informationType), дефинират структурата на съдържанието на Променливите.
- Тип на участник (participantType), дефинира тези части от поведението на участника, които трябва да бъдат реализирани от една и съща логическа единица или организация;
- Тип на роля (roleType), дефинира поведението на участника;
- Тип на връзка (relationshipType), дефинира типа на съгласуване между участниците в колаборацията;
- Канали (channels), специфицира къде и как се обменя информацията между участниците в колаборацията;
- Хореографии (choreographies), дефинира сътрудничеството между отделните участници:
    - Линия на живот (life-line), отразява прогреса на сътрудничеството. Първоначално, се осъществява сътрудничество между участниците, след което в рамките на колаборацията се извършва някаква дейност и на края, сътрудничеството завършва успешно или неуспешно;
    - Блок за изключение (exception block), дефинира допълнителни действия в случай че, хореографията завърши неуспешно;
    - Блок за край (finalizer blocks), дефинира допълнителни действия за успешно завършване на хореография.
    - Работна единица (work unit), описва ограниченията, които трябва да са изпълнени за да има прогрес по хореографията и реално да бъде извършена дадена дейност;
    - Дейности (activities), компоненти от най-ниско ниво в хореографията при които се осъществява действителната работа.
        - Структури за наредба (ordering structures), определят условията за наредба, по която трябва да се обмени информацията между участниците. Може да бъде: последователна, паралелна, на базата на избор – коя дейност да бъде изпълнена;
        - Взаимодействия (interaction activity), основен блок в хореографията, резултат на обмена на информацията между участниците;

- Семантика (semantics), позволява създаването на семантично описание към компонент на модела.

Чрез езика WS-CDL не могат да бъдат създавани изпълними бизнес-процеси.

В обобщение на този раздел може да кажем, че ориентираната към услуги архитектура, предоставя един стандартизиран модел за изграждане на гъвкави приложения. Едно от предимствата на този архитектурен стил е възможността за композиране на услуги в по-сложни такива композиции, които също са услуги. Има два модела на организиране на услугите в композиции: оркестрация и хореография. Първият модел (оркестрация) е централизиран. Услуга - BPEL сървър, координира изпълнението на всяка една от услугите в композицията. Вторият модел (хореография) е децентрализиран. Услугите в композицията си сътрудничат за да се постигне конкретна бизнес цел. Езикът WS-BPEL, реализира модела на оркестрация и чрез него могат да бъдат създавани изпълними бизнес-процеси. Езикът WS-CDL реализира модела на хореография и чрез него могат да бъдат създавани абстрактни бизнес-процеси.

1.3 Управление на бизнес-процеси

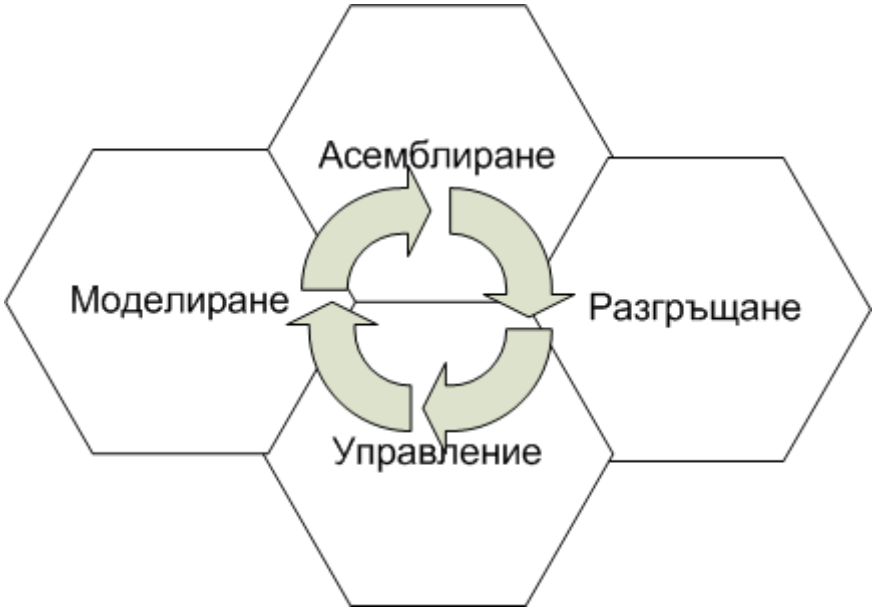
За бизнес организациите е важно информационните технологии да осигуряват приложения, които да поддържат бизнес-процесите от начало до край (end-to-end). Това включва, всички етапи от проектирането на бизнес-процеса до разгръщането му за изпълнение на процесен сървър и проследяването на бизнес-процеса след това. Един бизнес-процес може да включва дейности от различни приложения, обмяна на информация, интеграция на ресурси, дори и действия от човек (human tasks).

Дефиниция. *Управлението на бизнес-процеси включва "методите, техниките и софтуера за проектиране, разработка, контрол и анализ на бизнес-процеси, при които се интегрират хора, организации, приложения, документи и други източници на информация"* [63](стр. 4).

Поддръжката на бизнес-процеса трябва да бъде гъвкава, така че да отговаря на динамичните промени в организацията. Тези изисквания са в сила не само за процесите в една бизнес организация, но и за бизнес-процесите в изследователските проекти, които използват грид.

Ориентираната към услуги архитектура, покрива тези изисквания, като предоставя подвижна (agile) среда в която, логиката на автоматизация може да бъде обособена в услуги. Това прави средата гъвкава и адаптивна към промени.

В контекста на ориентираната към услуги архитектура, управлението на един бизнес-процес протича на четири етапа. Тези етапи се покриват от жизнен цикъл на ориентираната към услуги архитектура. В [2] (стр. 84) авторът представя четирите фази на жизнения цикъл: Моделиране, Асемблиране, Разгръщане и Управление. На Фиг. 12 [35](стр.20), са представени четирите фази на жизнения цикъл.



Фиг. 12. Жизнен цикъл на ориентираната към услуги архитектура

Във фазата моделиране се събират и анализират бизнес изискванията, които след това се използват за моделиране, симулация и оптимизация на бизнес-процесите. По време на фазата асемблиране се създават нови услуги различни от наличните за организацията. Тези услуги могат да бъдат използвани след това в оркестрацията на бизнес-процесите. Във фазата разгръщане се конфигурира средата за изпълнение на бизнес-процеса. В тази фаза се използват и специализирани услуги, които поддържат интеграцията на хора, процеси и информация.

При фазата на управление се управляват и наблюдават композитните приложения и ресурси от гледна точка на информационните технологии и бизнеса. На базата на бизнес метрики се изяснява поведението на процеса в реално време. Това от своя страна позволява процесът да бъде непрекъснато подобряван и оптимизиран.

В този контекст съвкупността от софтуерни средства, които поддържат пълният жизнен цикъл на ориентираната към услуги архитектура и предоставят

функционалност за моделиране, разгръщане, изпълнение, наблюдение и оптимизиране на бизнес-процеси, чрез поддръжка на стандарти за Уеб услуги наричаме среда за управление на бизнес-процеси.

Ориентираната към услуги архитектура [42], предоставя:

- Архитектура за интегриране на различните услуги, включително и от наследен софтуер;
- Възможност за описание на изнесената от приложенията функционалност (Уеб услуги), посредством стандарта WSDL;
- Възможност за откриване на услуги и многократна употреба на услуги (регистър на услуги), посредством стандарта UDDI;
- Възможност за композиране на Уеб услуги в бизнес-процеси, посредством стандарта BPEL;
- Инфраструктура за комуникация и управление на услугите (шина на услугите в предприятието).

В [2] (стр. 27), авторът представя шината на услуги в предприятието (Enterprise Service Bus - ESB), като „основният медиатор, който свързва услугите и който регулира потока от информация навсякъде и по всяко време прилагайки различни протоколи". Шината на услугите предоставя обща инфраструктура, която управлява достъпа до наличните услуги и играе роля на посредник между тях. Важна роля в ориентираната към услуги архитектура има регистърът, реализиран посредством стандарта UDDI. Той предоставя информация за версията на услугите, описанието на услугите и достъпа до тях. Интеграцията на услуги от наличен софтуер в архитектурата се реализира чрез адаптери. Адаптерите, могат да бъдат използвани и за достъп до външни за организациите системи.

Управлението на един бизнес-процес включва не само изпълнение на процеса, но и проследяване поведението на процеса в реално време. Проследяването на процеса, може да даде допълнителна информация, като време за изпълнение на процеса, средно време за изпълнение на процеса и друга статистическа информация. Това позволява бизнес-процесът да бъде разбран по-добре и съответно да бъде оптимизиран.

Бизнес-процесите могат да съдържат задачи за хора (human tasks). При по-сложни процеси, бизнес-процесът не може да продължи без намеса на човек. Поддръжка на задачи за хора (действия от човек) в бизнес-процесите, не се покрива от

езика BPEL, въпреки това е предимство, което всяка среда за управление на бизнес-процеси трябва да предоставя.

Важна част от бизнес-процесите са и бизнес-правилата. Бизнес-правилата са твърдения, който са валидни за даден процеса. Често пъти тези твърдения се променят във времето. В не ориентираните към услуги приложения, бизнес-правилата са вградени в самите приложения. Така, честата промяна на бизнес-правилата, води до честа промяна в приложенията, което е не ефективно. Поддръжката на управление на бизнес-правила, позволява бизнес-правилата да могат да бъдат лесно променяни дори и за разгърнати на процесен сървър бизнес-процеси.

Средите за управление на бизнес-процеси, базирани на ориентирана към услуги архитектура, поддържат всеки един етап от жизнения цикъл на архитектурата. Във фазата моделиране се създават бизнес-процесите. Софтуерът подържащ тази фаза, трябва да предоставя възможност на потребителя да моделира по графичен начин бизнес-процесите си. Също така, трябва да предоставя възможност за симулиране на бизнес-процесите и възможност за съхраняване на разработените от потребителя бизнес-процеси. Нещо повече, софтуерът трябва да предоставя възможност за транслиране на моделирания бизнес-процес към фаза асемблиране.

По време на фазата асемблиране се създават нови услуги различни от наличните за организацията. Софтуерът поддържащ тази фаза трябва да предоставя възможност на потребителя да асемблира наличните Уеб услуги в бизнес-процеси и да създава композитни приложения.

Във фазата разгръщане се конфигурира средата за изпълнение на бизнес-процеса. Софтуерът поддържащ тази фази трябва да предоставя услуга за изпълнение на бизнес-процесите, услуга за управление на бизнес-правила за процесите и обща инфраструктура за комуникация и управление на услугите и интеграция с външни приложения.

При фазата на управление се наблюдават композитните приложения. Софтуерът поддържащ тази фаза трябва да предоставя функционалност за наблюдение на бизнес-процеса. Бизнес метрики за процесът, могат да бъдат задавани във фаза моделиране на бизнес-процеса, могат да бъдат променяни във фаза асемблиране и наблюдавани във фаза разгръщане.

Важна част във всяка една от фазите заема и регистърът на услугите. Той предоставя възможност за публикуване на нови услуги и откриване на Уеб услуги, като по този начин спомага многократната употреба на регистрираните в него услуги.

Като обобщение на казаното до тук, формулираме основни изисквания към средите за управление на бизнес-процеси в контекста на ориентирана към услуги архитектура. Средите за управление на бизнес-процеси в контекста на ориентирана към услуги архитектура, покриват пълният жизнен цикъл на управление на бизнес-процеси, като предоставят поддръжка на:

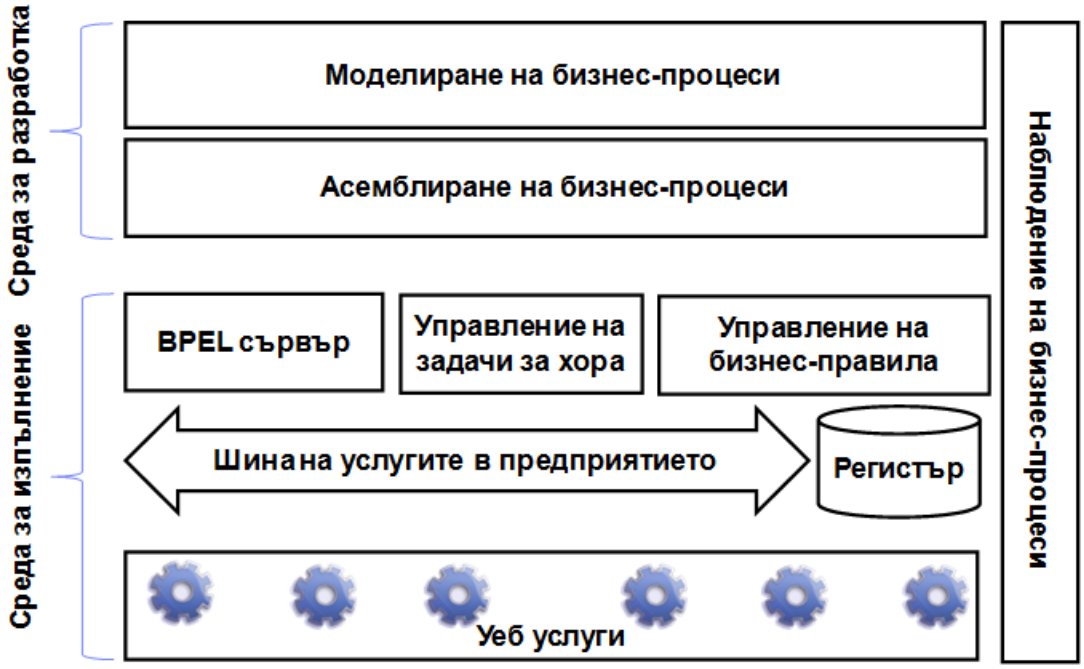
- Уеб услуги (WSDL) - за постигане на модулна и гъвкава архитектура;
- BPEL – за автоматизиране на бизнес процеса чрез композиране на Уеб услуги;
- Регистри – за откриване на услуги и многократна употреба на услуги;
- Адаптери – за достъп до външни системи;
- Шина на услугите (ESB) – инфраструктура за комуникация между услуги и бизнес-процеси;
- Задачи за хора (действия от човек) в бизнес-процес;
- Наблюдение на бизнес-процес;
- Управление на бизнес-правила за бизнес-процес.

Примери за реализации на среди за управление на бизнес-процеси, които покриват гореизброените изисквания са IBM SOA Foundations [81] и Oracle SOA Suite [85]. На базата на тях в следващата глава ще обобщим основните компоненти на средите за управление на бизнес-процеси, базирани на ориентирана към услуги архитектура. Други реализации на среди за управление на бизнес-процеси, които частично покриват гореизброените изисквания са Fiorano SOA Platform [69] и не-комерсиалната среда JBoss Enterprise SOA Platform [83]. Тези среди няма да разглеждаме в детайли.

1.4 Среди за управление на бизнес-процеси

В контекста на ориентираната към услуги архитектура средите за управление на бизнес-процеси поддържат Уеб услуги, BPEL като стандарт за описание на изпълними бизнес-процеси и UDDI регистър за откриване и публикуване на услуги.

Примерни среди за управление на бизнес-процеси, които реализират тези изисквания са IBM SOA Foundations [81] и Oracle SOA Suite [85]. Компонентите на средите са представени съответно в [37] и [56]. На Фиг. 13 представяме, общите и за двете среди компоненти, които приемаме за базисни в средите за управление на бизнес-процеси, базирани на ориентирана към услуги архитектура.



Фиг. 13. Компоненти на среда за управление на бизнес-процеси базирана на ориентирана към услуги архитектура

Средите за управление на бизнес-процеси предоставят два основни модула: среда за разработка и среда за изпълнение. По-долу представяме накратко всяка една от тях.

1.4.1 Среда за разработка

Средата за разработка ([13], [42]) предоставя инструментални средства за проектиране и за разгръщане на бизнес-процесите. Средата за разработка, може да бъде реализирана в едно инструментално средство или както е показано в случая в две различни инструментални среди.

Средата за проектиране (моделиране на бизнес-процеси) е графичен редактор, която поддържа моделиране на бизнес-процеси, управление на задачи за хора в бизнес-процеса, управление на бизнес-правила и въвеждане на метрики за наблюдение на бизнес-процеса. Средата за проектиране също така поддържа и функционалност за симулиране на създадените бизнес-процеси на базата на предварително зададени параметри.

Базовите дейностите необходими при графичното моделирането на бизнес-процесите в средите за проектиране са: основни дейности, структурни дейности и събития.

Основни дейности са:

- Задача (<дейност>) – стъпка от бизнес-процеса, обозначаваща дадено действие;
- Задача за хора – стъпка от бизнес-процеса, извършвана от човек;
- Задача бизнес-правило – стъпка от бизнес-процеса за изпълняване на бизнес-правило;
- Задача за получаване (<receive>) – стъпка от бизнес-процеса за асинхронно получаване на съобщение;
- Задача за изпращане (<reply) – стъпка от бизнес-процеса за изпращане на съобщение;
- Услуга (<invoke>) – стъпка от бизнес-процеса за синхронизирано извикване на Уеб услуги;
- Под-процес (<под-процес>)– стъпка от бизнес-процеса, обозначаваща йерархия в процес.

Структурни дейности са:

- Разклоняване на бизнес-процеси (<flow>) – за моделиране на паралелни потоци в бизнес-процесите;
- Сливане на бизнес-процеси (<flow>) – за обединяване на паралелни потоци в бизнес-процесите;
- Условни разклонения (<if>) – за условно разклоняване на потока на информация в бизнес процеса;
- Цикъл (<while>) – за многократно изпълнение на дадена дейност в бизнес-процес.

Дейности събития са:

- Начало – стъпка от бизнес-процеса, обозначаваща начало на процеса;
- Таймер (<wait>) – стъпка от бизнес-процеса, спираща процеса за известно време;
- Край (<exit>) – стъпка от бизнес-процеса, обозначаваща край на процеса.

В скобите е посочено съответствие със структурите на езика BPEL, представени във втори раздел на настоящата глава.

При моделиране на бизнес-процесите в средата за проектиране, могат да се използват и налични Уеб услуги и артефакти (данни, документи, обекти) публикувани в регистър. Средата за проектиране предоставя функционалност за свързване с регистъра, както и функционалност за транслиране на моделирания бизнес-процеса към фаза асемблиране. Транслирането на бизнес-процеса, обхваща не само транслиране на

процеса към програмен код, но и транслиране на всички свързани с него данни, метрики и обект.

Средата за разгръщане (асемблиране на бизнес-процеси), позволява на специалистите по информационни технологии да реализират моделирания бизнес-процес. Това включва, създаване на реален програмен код, които да реализира обозначените дейности (задачи) в бизнес-процеса. За всяка стъпка от бизнес-процеса се съпоставя BPEL елемент, ако има съответствие. Както видяхме по-горе, съответствие с BPEL елемент нямат задачи за хора и задача за бизнес-правило. Въпреки това, за тях в средата за разгръщане също се предоставят съответстващи компоненти.

Средата за разгръщане предоставя възможност на специалистите да разработват и нови бизнес-процеси и нови приложения независимо от средата за проектиране. Това включва готови примитиви за филтриране на съобщения, извикване на услуги, адаптери към бази от данни, трансформация на данни и други, които специалистите могат да използват.

В средата за разгръщане могат да бъдат разработвани и приложения и услуги, които да са директно насочени към отделните компоненти на средата за изпълнение, т.е. приложения за BPEL сървър, приложения за управление на бизнес-правила и приложения за шина на услугите в предприятието.

Средата за разгръщане предоставя също така и функционалност за свързване с регистъра, като по този начин позволява реализирането на статично или динамично свързване към услугите, при изпълнение на бизнес-процеса в средата за изпълнение.

1.4.2 Среда за изпълнение

Ядрото на средата за изпълнение е шина на услугите в предприятието ([42], [55]). Шината на услугите в предприятието е инфраструктура, която интегрира приложения и услуги. Шината на услугите е основният медиатор, който свързва услугите и който регулира потока на информация между тях. Достъпът до услугите през шината е виртуализиран. Така приложните клиенти могат да достъпват услугите без да знаят крайната точка на услугата. Чрез абстракция на крайните точки на услугите, шината позволява услугите да бъдат разгръщани на различни места без това да се отразява на приложните клиенти на тези услуги. Чрез регистъра, шината на услугите реализира механизмът на динамично свързване към услугите при изпълнението на бизнес-процесите.

Шината на услугите спомага интеграцията на налични приложения, като предоставя адаптери за достъп до бази от данни и външни системи. От гледна точка на ориентираната към услуги архитектура, шината на услугите реализира многократната употреба и слабата свързаност. Основните функции на шината са:

- Трансформация от един формат на данни към друг;
- Преобразуване от един транспортен протокол към друг;
- Маршрутизиране на съобщенията между услугите.

Компонентите BPEL сървър, управление на задачи за хора и управление на бизнес-правила ползват основните функции на шината на услугите. Така те използват единна инфраструктура за обмен на съобщения.

Бизнес-процесите се състоят от BPEL дейности, задачи за хора и бизнес-правила. В средите за управление на бизнес-процеси, задачите за хора и бизнес-правилата, могат да бъдат интегрирани директно в BPEL процеса. Изпълнението на бизнес-процеса в средата за изпълнение, представлява извикване на съответните компоненти от средата, които интерпретират дейностите в бизнес-процеса. Така BPEL дейност се интерпретира от BPEL сървъра, задачите за хора от компонента за управление на задачи и бизнес-правилата от компонента за управление на бизнес-правила. Често пъти трите компонента се реализират в един - наречен процесен сървър.

Компонентът BPEL сървър интерпретира BPEL дейностите от бизнес-процеса. Където е необходимо BPEL сървъра извършва синхронизирано извикване на Уеб услуги, изпраща и получава съобщения. Инфраструктурата за обмен на съобщения, която ползва е именно шината на услугите в предприятието.

Компонентът управление на задачи за хора, реализира интегрираните в BPEL процеса задачи за хора. Задачите за хора ([42], [55]) се инициират от човек или от услуга. Те могат да бъдат интегрирани в BPEL процеса или да бъдат самостоятелни услуги извън BPEL процеса. Задачите за хора могат да бъдат пренасочвани от един потребител към друг. Те могат да бъдат временно спирани и подновявани. Задачите за хора могат да имат срок на валидност за даден период от време и да бъдат препратени от един потребител към друг. Компонентът за управление на задачи за хора, се грижи за тези дейности.

Задачите за хора могат да бъдат:

- Човек-човек - задачи, които отразяват съвместна дейност между потребителите;
- Услуга-човек - задачите за хора интегрирани в BPEL процесите;
- Човек-услуга - задачи, при които потребител извиква услуга.

Пример за такава задача (човек-услуга) е потребител, който извиква BPEL процес. Разновидност на този тип задачи са административните задачи. Чрез тях потребителите могат да извършват административни дейности, например стартиране на BPEL процеси, спиране на BPEL процес и други.

Компонентът за управление на бизнес-правила реализира функционалност, която интерпретира бизнес-правилата и позволява промяна на бизнес-правилата по време на изпълнение на бизнес-процеса. Бизнес-правилата са твърдения, които ограничават бизнес-процеса. Бизнес-правилата са атомарни. Те се дефинират във фаза моделиране и асемблиране за бизнес-процеса. Най-често са във вид на условие за бизнес-процеса.

Компонентът за управление на бизнес-правила интерпретира, бизнес-правилата като независими от бизнес-процеса компоненти. Това позволява бизнес-правилата да могат да бъдат променяни динамично, по време на изпълнение на процеса. Обикновено чрез Уеб-базиран потребителски интерфейс потребителят може да променя бизнес-правилата за даден процес.

Регистърът на услуги е централизирано хранилище за услуги и свързаната с тях информация. Регистърът предоставя услуги за съхраняване, публикуване и откриване на услуги. Той присъства и в четирите фази на жизнения цикъл на ориентираната към услуги архитектура. Във фаза моделиране и асемблиране, от регистъра могат да бъдат използвани нови Уеб услуги и свързаната с тях информация, както и да бъдат публикувани моделираните и разработените бизнес-процеси. Във фаза на разгръщане, регистърът служи за динамична селекция на действителните услуги, които ще бъдат изпълнение в бизнес-процеса. Обикновено действителните услуги се избират от множество от кандидат услуги, на базата на предварително зададени критерии. Във фаза на управление, регистърът спомага за по-ефективно управление на наличните услуги, разработени във фаза на моделиране и асемблиране.

Наблюдението на бизнес-процеса, също присъства в четирите фази на жизнения цикъл на ориентираната към услуги архитектура. Той предоставя информация в реално време за състоянието на бизнес-процеса и за резултатите от изпълнението на бизнес-процеса. Наблюдението на бизнес-процеса се извършва на база на ключови за

производителността на процеса индикатори. Те се указват по време на моделиране и асемблиране на бизнес-процеса. Във фаза на разгръщане и управление, компонентът за наблюдение на бизнес-процеси, посредством шината на услугите в предприятието обменя съобщения с процесния сървър, като по този начин събира информация по ключовите индикатори за изпълняващите се на сървъра бизнес-процеси.

1.5 Ориентиран към услуги грид

Бързото развитие на грида, като инфраструктура в е-бизнеса и е-науката, наложи установяването на стандарти, които да регламентират необходимите услуги за поддържане на грид системи и приложения. Разработването на такъв стандарт започва през 2002. В [26] авторите представят, част от тези разработки, като архитектура за отворени грид услуги (OGSA).

Архитектурата за отворени грид услуги дефинира, понятието грид услуга като Уеб услуга с разширен интерфейс, който да поддържа жизнен цикъл, състояние и асинхронни събития, които са задължителни при моделиране и контролиране на динамични грид ресурси.

Основното предимство на OGSA е, че всички ресурси в грид се представят като грид услуги. Изчислителните ресурси, ресурсите за съхранение на данни, мрежи, програми, бази от данни – всичко е услуга. Този подход опростява представянето на ресурсите. Това позволява последователен достъп до ресурси, намиращи се на хетерогенни платформи, както и композиране на услуги в състав от услуги.

В [27] стандартът OGSA е допълнен, като са дефинирани в детайли основните услуги за достъп до ресурси в грид, както и изискванията, които те трябва да покриват. В архитектурата отворена към грид услуги са дефинирани услуги за сигурност, услуги за управление на данни, и други. Услугите, които OGSA предоставя са групирани логически в три слоя.

Първият слой, обхваща базовите ресурси. Базови са тези ресурси, които се поддържат от физически единици извън контекста на OGSA: процесор, памет, дискове, лицензи и други. Тези ресурси обикновено са локално управлявани и могат да бъдат споделяни. В [54] този слой се нарича още и „машинен” и се представя от широко множество хетерогенни високопроизводителни изчислителни ресурси, които обединени заедно съставляват хардуерната част на грид инфраструктурата.

Във втория слой се представят услуги от по-високо ниво на опростяване на представянето и логическа абстракция. Това са услугите, които имат отношение към

грид инфраструктурата. В [54] този слой се нарича още и слой от грид услуги. Представлява софтуер, който свързва хетерогенните ресурси от „машинния” слой и грид приложенията и предоставя множество от услуги, като регистри, услуга за управление на ресурси, услуги за наблюдение, информационни услуги и други. Комуникацията между услугите се базира на стандарта за обмен на съобщения между Уеб услуги - SOAP. Ориентираността към услуги на OGSA гарантира, че виртуалните ресурси представени, като услуги могат да бъдат достъпвани, както такива независимо от слоя в които се намират.

Третият слоя в логическото представяне на OGSA са приложения и софтуер, които използват дефинираните от междинния слой услуги. В [54] е обозначен като „приложен” слой. Състои се от грид портали и приложения, които комуникират с услугите от междинния слой посредством обмен на SOAP съобщения.

В [27] авторите споменават и услуга за композиране на услуги, но не специфицират конкретна реализация. Авторите подчертават също, че няма да дефинират нови стандарти за композиране, а ще разчита на съществуващите вече такива. Както видяхме във втори раздел на първа глава, в ориентираната към услуги архитектура има два модела на организиране на услугите в композитни услуги: оркестрация и хореография. Съответните стандарти, които реализират двата модела са WS-BPEL и WS-CDL.

Архитектурата ориентирана към грид услуги предвижда възможност за композиране на услуги. Въпреки това, спецификацията на архитектурата представена в [27] не дава яснота по въпросите свързани с композирането на услуги в бизнес-процеси и тяхното изпълнение. Отворени остават въпросите свързани с мястото на услугата в грид архитектурата и взаимодействието й с другите услуги в грид.

Паралелно с разработването на архитектура отворена към грид услуги е разработена и спецификация за описание на грид услуги (Open Grid Service Infrastructure - OGSI). Спецификацията е представена в [62] и разработена от Global Grid Forum (GGF). Целта е OGSI да се наложи, като стандартна технология за разработване на грид услуги.

Спецификацията OGSI е реализирана от Globus Alliance в трета версия на продукта Globus Toolkit. Софтуерът Globus Toolkit е с отворен код и представлява набор от компоненти и услуги, чрез които може да бъде създадена и изградена грид среда. Предишната версия на продукта Globus Toolkit 2 е използвана в реализацията на

грид среди, като European DataGrid (EDG) и LHC Computing Grid Project (LCG) част от Европейска грид инфраструктура (EU DataGrid).

За съжаление, същото не може да се каже за Globus Toolkit 3, които въпреки, че е реализация на архитектура отворена към грид услуги не успява да се наложи като стандарт. Концепциите заложени в OGSI са преразгледани и в [57] авторите предлагат едно по-добро решение за изграждане на грид услуги, базирано на Уеб услуги и колекцията от стандарти Web Service Resource Framework (WSRF). Колекцията от стандарти WSRF дефинира отворена среда за достъп до грид ресурси посредством Уеб услуги. Стандартите WSRF разширяват стандартите за Уеб услуги, така че да поддържат грид концепцията. Софтуерът, които реализира новият вид Уеб услуги е Globus Toolkit 4.

Globus Toolkit 4 предоставя услуги за наблюдение, услуги за сигурност, услуги за обработка на задачи, услуги за трансфер на данни. Globus Toolkit 4, включва пълна реализация на WSRF спецификациите. Въпреки че, софтуерът се фокусира изцяло върху реализацията с Уеб услуги, все още има компоненти които са реализирани различно. Така например GridFTP компонентът, използва протоколи, различни от тези дефинирани в спецификациите.

Разработки базирани на Globus Toolkit 4, в които се разглеждат изследователските въпроси за композиране на WSRF грид услуги са ([20] и [59]). В [20] авторите разглеждат как езикът BPEL може да бъде приложен за композиране на WSRF-базирани грид услуги в био-информатиката. Друг аспект, който разглеждат авторите е как BPEL процес, може да бъде изложен като WSRF грид услуга. В [59] авторите изследват оркестрирането на грид услуги, чрез BPEL сървър в Globus Toolkit 4 среда.

За съжаление Globus Toolkit 4 също като неговите предшественици не успява да се наложи като стандарт на среда за достъп до грид. Това доведе до реализирането на нова подобрена версия на софтуера в Globus Toolkit 5, в която концепцията от Globus Toolkit 4 е запазена, но някои от компонентите на софтуера са заменени с цел подобрения.

От казаното до тук може да заключим, че OGSA реализациите на Globus Toolkit не успяват да се наложат като стандарти при разработването на грид среди и грид услуги. Въпреки това, архитектурата за отворени грид услуги въвежда една концепция на прилагане на ориентираната към услуги архитектура в грид и към момента си остава единственият стандарт на ориентиран към услуги грид.

1.6 Грид средата g-Lite

Друга реализация на архитектурата на грид представена в първа глава е средата g-Lite. Към момента, g-Lite е и базовата грид среда инсталирана на всички клъстери в Европа, които са част от Европейската грид инфраструктура (EGI). Трябва да отбележим, че g-Lite не реализира OGSA. Въпреки представянето на средата като ориентирана към услуги в [45] услугите, които средата предоставя, съвсем не могат да бъдат определени като ориентирани към услуги. Можем да кажем, че средата g-Lite е частично ориентирана към услуги.

На Фиг. 14 [45] са показани услугите на средата групирани тематично в пет групи. Средата предоставя услуги за управление на задачи, информационни услуги, услуги за сигурност, услуги за достъп и услуги за управление на данни. По-долу ще представим накратко всяка една от услугите в тези пет групи.

Фиг. 14. Услуги в g-Lite

Услугите за сигурност, предоставят функционалност за удостоверяване на потребител в средата, даване на права на потребител за достъп до ресурс или услуга и функционалност за събиране и анализиране на информация при нарушение на сигурността. Сигурността в g-Lite [14] се базира на инфраструктура за грид сигурност (Grid Security Infrastructure - GSI) [80]. Тя позволява идентифициране на потребител в средата посредством грид сертификат, сигурна комуникация, делегиране на права, както и възможност потребителят да се удостовери само веднъж в средата (single sign-on). За да има достъп до EGI, всеки потребител трябва да притежава грид сертификат и да е член на поне една виртуална организация.

Услугите за достъп, включват:

- Потребителски интерфейс (User Interface - UI) [14] – входна точка към грид инфраструктурата. Обикновено представлява компютър, до който потребителят има достъп. Компонентът предоставя функционалност за показване на всички налични ресурси в грид, за изпълнение на задача в грид, за проследяване на състоянието на задачата и други.
- Приложно-програмни интерфейси (ППИ) – позволяват разработката на грид-базирани приложения.

Услугите за управление на задачи включват:

- Изчислителен елемент (Computing Element - CE);
- Система за управление на разпределението на задачите (Workflow Management System - WMS);
- Отчетност (Accounting) предоставя информация за броя задачи, които са изпълнени от даден клъстер, информация за потребителите, които са пускали задачи както и друга статистическа информация;
- Управление на пакети, позволява динамично внедряване на приложен софтуер в g-Lite средата. Услугата е прототип и все още не е реализирани в средата.

Изчислителният елемент [14] е изчислителен ресурс, обикновено клъстер от компютри, в който един от възлите е главен и разпределя задачите. Най-често механизма на разпределяне на задачите е в пакетна обработка. Изчислителният ресурс, може да бъде един компютър, клъстер от компютри или суперкомпютър. Изчислителният елемент се състои от следните компоненти:

- общ интерфейс към клъстер от компютри (Grid Gate - GG);
- механизъм за пакетна обработка (Local Resource Management System - LRMS);
- клъстер от компютри – изчислителни възли (Worker Nodes - WN) [76];

Софтуерни пакети реализиращ изчислителен елемент в g-Lite са LCG CE [72] и CREAM CE [71]. Вторият е ориентиран към услуги.

Системата за управление на разпределението на задачите (Workflow Management System, WMS) [77] разпределя задачите на наличните изчислителни елементи, според предпочитанията на потребителя. В предпочитанията си, потребителя може да указва брой на процесорите, дали задачата е паралелна, минимална памет и други. Това става посредством език за описание на задачи (Job Description Language, JDL) [53]. Системата също пази следи от изпълнението на задачата, посредством водене на журнал и

проследяване. Основен компонент на системата е Workload Manager (WM) [14], който приема заявки за изпълнение на задачи.

Услуги за управление на данни, включват:

- Елемент за съхранение (Storage Element - SE);
- Файлови каталози и репликация. В каталозите се пази информация за това къде са локализирани данните, както и съответстващите им мета данни;
- Управление на данни, включва услугата за трансфер (File Transfer Service - FTS) [14], чрез която данните могат да бъдат копирани от един елемент за съхранение на друг.

Елементът за съхранение предоставя достъп до ресурс за съхранение, който може да бъде диск или третична памет. Поддържа различни протоколи за достъп до данните, като GSIFTP [78], RFIO [14] и други. Ресурсите за съхранение се управляват от услуга наречена Storage Resource Manager (SRM) [14]. Тя се грижи за резервиране на ресурс за съхранение, за копиране на файл от дисков сървър до третична памет и други. В грид средата g-Lite към момента има три реализации на елемент за съхранение:

- Disk Pool Manager (DPM) [74] – за елементи с достъп до дискови ресурси за съхранение;
- CASTOR [14] - за елементи с достъп до третична памет;
- dCache [75] - за елементи с достъп до дискови ресурси и третична памет;

Информационните услуги предоставят механизми за наблюдение на ресурси и задачи, откриване на ресурси и извличане на информация. Реализирани са посредством два основни компонента:

- Информационен индекс (Berkeley Database Information Index, BDII) [70];
- Релационна система за наблюдение (Relational Grid Monitoring Architecture, R-GMA) [73].

Релационната система за наблюдение е информационна услуга, която следва модела Регистър-Доставчик-Клиент. Доставчикът е източник на информация. Той публикува данни в системата. Централизиран регистър посредничи между клиентът и доставчикът. Данните се предават директно между доставчикът и клиентът. Релационната схема за наблюдение е реализирана във версия 3.1 на средата g-Lite. От версия 3.2 тази услуга не се поддържа.

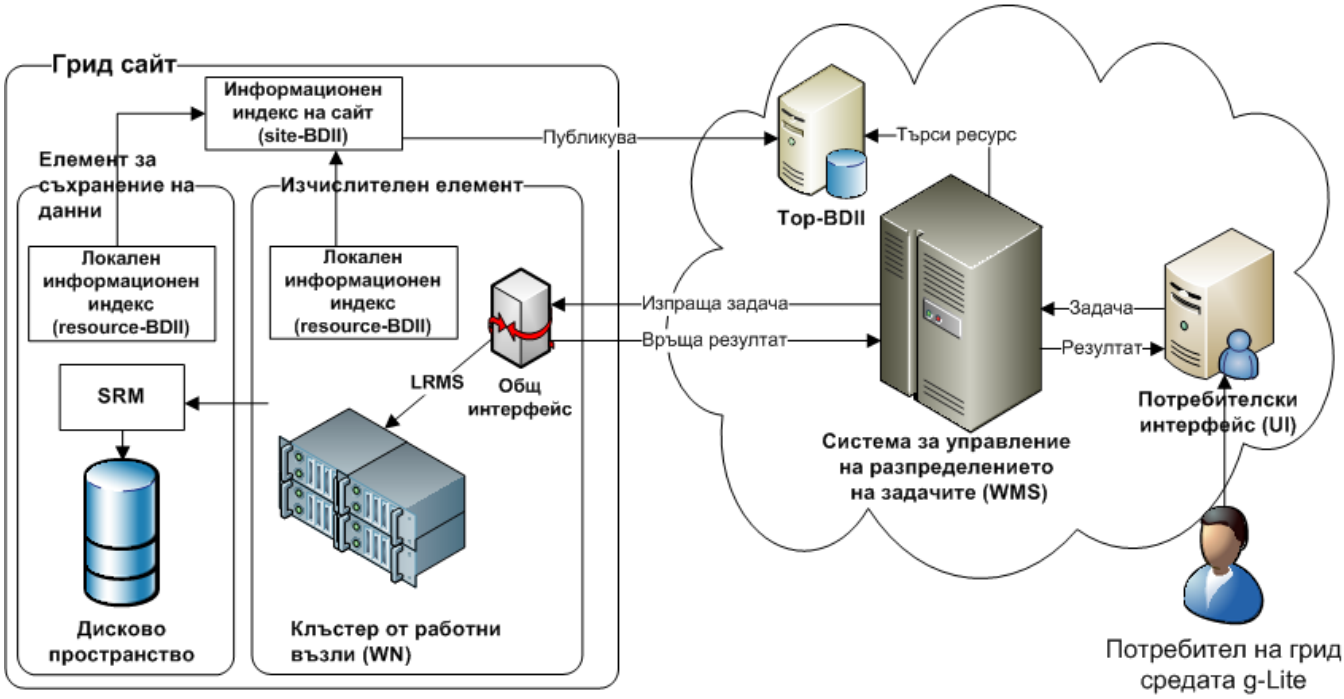
Информационният индекс е стандартна LDAP (Lightweight Directory Access Protocol) [64] база от данни. Тя се обновява от външни процеси, чрез извличане на

информация от различни източници и сливане на информацията в централизирана база от данни. Средата g-Lite предоставя стандартни клиенти за достъп до LDAP базата.

Групата от g-Lite услуги (изчислителен елемент, елемент за съхранение на данни, информационен индекс, система за управление на разпределението на задачите), които всяка една организация поддържа, в контекста на грид инфраструктурата EGI, се нарича сайт.

Информационната система е йерархична и се състои от три нива. Грид услугите, включително и ресурсите в грид - изчислителен елемент и елемент за съхраняване на данни, събират информация за себе си в локален информационен индекс (resource-level BDII). Тази информация се събира от информационен индекс от второ ниво или още информационен индекс на ниво сайт (site-level BDII). Информационният индекс от най-високо ниво (top-level BDII) събира информация от всички информационни индекси на наличните сайтове и съхранява тази информация в LDAP база от данни.

Минималните услуги, които един сайт трябва да поддържа са: изчислителен елемент, елемент за съхранение на данни и информационен индекс на сайта (site-BDII). Към момента, в EGI инфраструктурата участват над 200 сайта, 9 от които в България. На Фиг. 15 са представени основните услуги на средата g-Lite в рамките на един сайт и връзките им с другите услуги от средата.



Фиг. 15. Връзки между основните услуги в средата g-Lite

Средата g-Lite е зависима от операционната система. Командите за достъп до грид, които средата предоставя са обвързани с познания за конкретна операционна система – Scientific Linux 5, а изпълнението на самата задача с език за описание на задачи (JDL) [53]. Това често пъти създава неудобства при използването на средата и води до де-мотивиране на крайния потребител да я използва.

Услугите на средата g-Lite се базират на компоненти разработени в грид проектите EGEE, DataGrid (EDG), Globus и Condor. В [11] авторите представят някои от тези компоненти. Дори и към момента в g-Lite се използват компоненти от Globus Toolkit 2. Средата g-Lite, подлежи на непрекъснати подобрения. Това включва подобрения на надеждността и качеството на предлаганите услуги, както и въвеждане на нови ориентирани към услуги компоненти, базирани на Globus Toolkit 4. Примерни такива подобрени са разгледани в [5], [6] и [7].

В [5] авторите представят тип качествени услуги, които са под контрол на потребителя. Използвайки тези услуги, потребителят има по-добър контрол върху услугите от EGI, които използва. В [6] и [7] авторите представят архитектурата на услуга за проследяване на задачи в грид средата, която улеснява доставянето на качествени услуги в грид инфраструктурата. Услугата използва среда за обмен на съобщения.

Визията за g-Lite, представена в [45] е тя да бъде една изцяло ориентирана към услуги грид среда, която да поддържа съвместимост със стандарти като OGSA. За съжаление наличните, към момента грид услуги на средата, не отговарят на тези изисквания и не позволяват поддържането на ориентирани към услуги приложения. Ориентираният към услуги подход, предполага спазването на определени правила, наречени принципи. Споменаваме отново някои от тях.

Договор. Описанието на услугата е изключително важно. То е непосредствено свързано с откриване на услугата, композицията и многократна употреба. В описанието на услугата се съдържа информация за функционалността, която услугата излага, входни и изходни параметри за всяка една от операциите на услугата и как услугата може да бъде извикана.

В (Таблица 2) са показани основните g-Lite услуги, крайни точки на услугите, индикация дали имат WSDL описание и индикация дали WSDL описанието им може да бъде извлечено от информационния индекс на средата.

	Описание на услугата в BDII		
g-Lite Услуга	**Крайна точка**	**WSDL**	**WSDL в BDII**
lcg-CE	ldap://ce001.grid.uni-sofia.bg:2170/mds-vo-name=BG05-SUGrid,o=grid	не	не
CREAM CE	https://cr1.ipp.acad.bg:8443/ce-cream/services	да	да
SRM	httpg://se001.grid.uni-sofia.bg:8446/srm/managerv2	да	не
FTS	https://fts-t2-service.cern.ch:8443/glite-data-transfer-fts/services/FileTransfer	да	да
WMS	https://wms.ipp.acad.bg:7443/glite_wms_wmproxy_server	да	да
MyProxy	myproxy.ipp.acad.bg:7512	не	не
BDII	ldap://bdii.ipp.acad.bg:2170/mds-vo-name=local,o=grid	не	не
VOMS	https://lcg-voms.cern.ch:8443/vomses/	да	не

Таблица 2. Описание на g-Lite услуги в информационния индекс на средата

От представеното по-горе се забелязва, че само част от услугите в g-Lite имат WSDL описание. Две от тези услуги, които си заслужава да отбележим са WMS и CREAM CE. Тези услуги са реализирани като Уеб услуги и следват стандартите на ориентираната към услуги архитектура.

Описанията (WSDL файловете) на част от услугите в g-Lite, могат да бъдат извлечени от информационния индекс на средата. Така например, в информационният индекс за някои от услугите (WMS, CREAM и FTS) има описание на пътя до WSDL файла, за другите (lcg-CE, VOMS, SRM, BDII, MyProxy) такова описание не е налично.

Откриваемост. Информационният индекс на g-Lite предоставя възможност за откриване на налични услуги. Информацията, която индексът предоставя, обаче е непълна и не достатъчна. От извлечената информация, могат да се направят заключения, кои са наличните ресурси (към кой сайт, коя виртуална организация поддържат), но не и как те могат да бъдат достъпени. Някой от услугите в g-Lite имат крайна точка за достъп. Такива са вече споменатите VOMS, WMS, CREAM CE и FTS. Но повечето от услугите нямат.

Липсата на централизирано хранилище (регистър) за наличните описания на всички услуги в средата възпрепятства разработването на грид приложения за g-Lite и извличането на информация за предоставената от услугите функционалност. Липсата

на регистър прави невъзможно и публикуването на услуги дефинирани от потребителя и поддържането на ориентирани към услуги приложения.

Композиция. g-Lite средата не предоставя възможност за композиране на други услуги. Също така не е предвидена такава услуга в архитектурата на средата.

Липсата на услуга за композиране, на услуга за откриване, както и липсата на описания на част от услугите в средата са ясни индикатори, че g-Lite не е ориентирана към услуги. Дори наличието на двете услуги (WMS и CREAM CE) не е достатъчна предпоставка средата да поддържа ориентирани към услуги приложения. Пълната ориентираност към услуги на g-Lite предполага реализацията на услуга за композиране, на услуга за публикуване и откриване - регистър, както и промени в реализацията на вече съществуващи услуги. За съжаление, разработването на всички услуги на средата, като ориентирани към услуги е задача, зависеща от много фактори. От една страна, средата се разработва от различни екипи и всяка промяна в архитектурата й ще повлияе на съществуващите и внедрени вече решения. От друга страна, средата е във фаза на подобрения на наличните услуги в посока добавяне на нови ориентирани към услуги компоненти.

Пълната ориентираност към услуги на g-Lite ще позволи изграждането на гъвкави грид приложения. Въвеждането на централизиран регистър съдържащ описанията на всички услуги, които следват стандартите на ориентираната към услуги архитектура ще подобри изграждането на ориентирани към услуги приложения. Наличието на услуга за композиране ще позволи създаването и изпълняването на бизнес-процеси дефинирани от потребителя в грид инфраструктурата EGI.

1.7 Изводи

От стратегическо значение е в Европейската грид инфраструктура да бъде реализиран механизъм за управление на бизнес-процеси. Това от една страна, ще позволи грид-потребителите да се концентрират върху естеството и спецификата на самите процеси. А от друга страна, ще скрие сложността от използването на командите за достъп до грид.

Реализацията на средата g-Lite е частично ориентирана към услуги. Грид услуги на средата, не отговарят на изисквания поставени в ориентираната към услуги архитектура и не позволяват поддържането на ориентирани към услуги приложения.

Ясни индикатори за това са:

- липсата на услуга за композиране;
- липсата на услуга за откриване;
- липсата на регистър за услуги;
- липсата на описания на част от услугите в средата.

Средите за управление на бизнес-процеси, базирани на ориентираната към услуги архитектура, предоставят функционалност за дефиниране, изпълнение, наблюдение и оптимизиране на бизнес-процеси, чрез поддръжка на стандарти за Уеб услуги. В настоящата глава бяха формулирани основните изисквания към тези среди и изведени базовите им компоненти.

Наличието на среда за управление на бизнес-процеси в EGI, ще предостави възможност на грид-потребителите да дефинират и проектират сами процесите си в грид, без да е необходимо познания по програмиране или намеса на специалисти по информационни технологии. Изследователски интерес представляват въпросите свързани с дефинирането, проектирането и изпълнението на бизнес-процеси в Европейската грид инфраструктура, за която g-Lite е базова среда. В средата няма дефинирана услуга за композиране и не всички услуги на средата, следват принципите на ориентираната към услуги архитектура. Тези проблеми възпрепятстват дефинирането и изпълнението на бизнес-процеси в EGI.

В следващата глава, ще разгледаме някои разработки от други изследователи, в които са засегнати въпросите за проектиране и изпълнение на бизнес-процеси в EGI. Някои от тях предоставят частични решения на поставените проблеми, като се фокусират върху дефинирането и изпълнението на бизнес-процеси в EGI, без да се ангажират с дефинирането на услуга за композиране. Тези решения не се базират на ориентирания към услуги подход и съответно не поддържат Уеб услуги и шина на услугите. Други решения, също не ориентирани към услуги, предоставят възможност за автоматизиране на част от потребителските процеси при изпълнението им в грид. При тях се разчита на услугите на средата g-Lite за изпълнение на процеса. Има и решения, при които е засегната идеята за ориентираност към услуги, но тя е частично развита за EGI.

Глава 2. Среди за проектиране и изпълнение на бизнес-процеси в грид

В настоящата глава, са разгледани и сравнени среди предоставящи възможности за композиране на услуги в бизнес-процеси и изпълнението им в Европейската грид инфраструктура (EGI). При описанието на средите е следвана обща схема. Описани са:

- Модул за проектиране на бизнес-процеси – графичен редактор, който средата предоставя за създаване на бизнес-процесите;
- Език, който се използва в средата при моделиране на бизнес-процесите;
- Модул за изпълнение на бизнес-процесите – сървър за изпълнение на процеси.

При сравнителния анализ на представените среди са взети предвид и следните изисквания, в сила при средите за управление на бизнес-процеси, базирани на ориентираната към услуги архитектура:

- Поддръжка на Уеб услуги - за постигане на модулна и гъвкава архитектура;
- Поддръжка на BPEL – за автоматизиране на бизнес-процеса чрез композиране на услуги;
- Поддръжка на регистри – за откриване на услуги и многократна употреба на услуги;
- Поддръжка на адаптери – за достъп до външни системи;
- Поддръжка на шина на услугите – за комуникация между услугите и процеси;
- Поддръжка на задачи за хора (действия от човек) в бизнес-процеса;
- Поддръжка на наблюдение на бизнес-процеса;
- Поддръжка на управление на бизнес-правилата за процеса.

2.1 CRAB

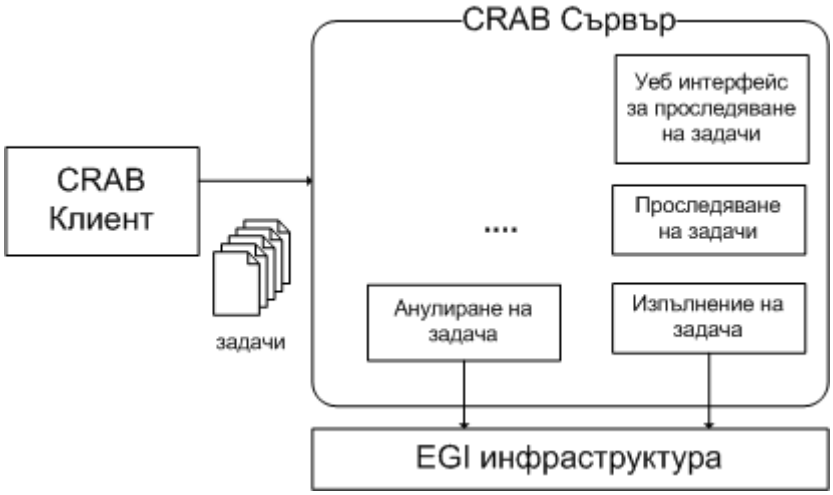
Проектът CMS е един от двата целеви научни проекта, които анализират данни, резултат от работата на ускорителят LHC. Данните от ускорителя се съхраняват на елементи за съхранение на данни (SE) в грид инфраструктурата EGI. Така за да извършват анализите си, грид-потребителите трябва да използват командите за достъп до грид средата g-Lite. Анализът на самите данни също се извършва в грид.

Един типичен процес за анализ на данни на CMS в грид, може да бъде представен в три стъпки. Първо, потребителят създава програмен код за анализ на

данни. На втора стъпка, разработеният от потребителя програмен код се изпълнява на грид сайта, където се намират реалните данни. И в последната стъпка, потребителят визуализира резултата от анализа посредством софтуер за визуализация и анализ на данни (ROOT).

Софтуерът CRAB (CMS Remote Analysis Builder)[58] автоматизира, втората стъпка от процеса. Посредством него, потребителят може да създава задачи (tasks), които да изпълнява в средата g-Lite. В контекста на CRAB задачите са множество от подзадачи (jobs), който могат да бъдат изпълнени в грид среда. Софтуерът CRAB е разработен специално за проекта CMS с цел да опрости процеса на създаване и изпълнение на CMS задачи за анализ в EGI.

На Фиг. 16 е показана архитектурата на CRAB, която е клиент-сървър архитектура.



Фиг. 16. Архитектура на CRAB

Клиентът е приложение инсталирано на потребителския интерфейс (glite-UI) в средата g-Lite. Посредством него потребителят създава задачи за анализ на данни и ги изпраща към сървъра CRAB. Софтуерът не предоставя графичен интерфейс за създаване на задачите. Вместо това потребителят създава конфигурационен файл – текстов файл, в който се зададени път до данните, които ще бъдат анализирани и как да бъдат обработвани данните последователно или паралелно. При изпращането на задача към сървъра CRAB, потребителят изпраща и конфигурационен файл.

Сървърът се състои от отделни компоненти, които отговарят за изпълнение на задачите в g-Lite средата, проследяване на състоянието на задачите, както и други компоненти. Софтуерът предлага Уеб интерфейс посредством, който потребителят

може да проследява състоянието на задачите, както и да получава допълнителна информация за използваните грид ресурси.

Софтуерът CRAB не поддържа нито един от гореописаните критерии на среда за управление на бизнес-процеси. Въпреки това се спряхме на него, поради автоматизацията, което предлага на стъпка от процеса за анализ на данни в CMS.

2.2 g-Eclipse

Софтуерът g-Eclipse [39] е Eclipse базирана среда, която позволява разработването на грид приложения. Софтуерът поддържа достъп до EGI грид инфраструктурата. Като част от средата се предоставя възможност за моделиране и изпълнение на бизнес-процеси (workflows) в грид. В контекста на g-Eclipse, бизнес-процесът представлява последователност от изпълними задачи в грид. Изпълнимите задачи в грид се описват посредством абстрактен модел за описание на задачи (Job Description). Моделът предоставя стандартна структура, в която се описват параметрите на задачата: изпълним код, входни данни и изходни данни. Моделът е стандартизиран от OGF като език за дефиниране на задача за изпълнение (Job Submission Definition Language - JSDL) [4]. Всички задачи в g-Eclipse се съхраняват като JSDL файлове. Така, бизнес-процесът се моделира като последователност от JSDL задачи. Това позволява повторна употреба на вече дефинирани задачи в средата. Самият процес също се съхранява като JSDL файл. Средата g-Eclipse предоставя графичен редактор за моделиране на бизнес-процеси. Моделирането на процеса става посредством четири основни елемента, представени в (Таблица 3). Те са: задача-процес (Workflow Job), входна точка, изходна точка, връзка.

Workflow Job	Задача-процес. Контейнер с описание на задача. По подразбиране, описанието на задачата е абстрактно. Потребителят трябва да дефинира по време на моделиране на бизнес-процеса, конкретна задача за изпълнение.
Input Port	Входна точка. Входни данни за задачата-процес.
Output Port	Изходна точка. Изходни данни за задачата-процес.
Link	Връзка. Показва потока на данни от една задача към друга.

Таблица 3. Елементи в g-Eclipse за моделиране на бизнес-процес

На Фиг. 17 е показан примерен бизнес-процес моделиран със средата g-Eclipse.

Фиг. 17. Бизнес-процес моделиран в g-Eclipse

В средата g-Eclipse е реализиран и компонент, който изпраща моделираният бизнес-процес за изпълнение в EGI. Компонентът трансформира JSDL описанието на процеса към JDL. Трансформираният бизнес-процес, в JDL формат, съдържа описание на задачите от процеса, както и допълнителен атрибут, в който са описани зависимостите между тези задачи. Самият бизнес-процес се изпълнява от WMS услугата на g-Lite. В [39] авторите, описват и някои допълнителни изисквания, при моделиране на бизнес-процес за g-Lite в средата g-Eclipse. Такива са:

- Зависимостите между задачите в JDL файла трябва да са описани, като ориентиран ацикличен граф (Directed Acyclic Graph - DAG). Този начин на описание е приложим за процеси, при които входните данни, изходните данни и изпълнението на задача от процеса, зависят от изпълнението на друга задача от процеса;
- Всяка задача от бизнес-процеса трябва да има валиден изпълним файл и валидни данни, до които потребителя да има достъп;
- Изпълнението на задача от бизнес-процеса, зависи от наличност на свободен ресурс в g-Lite средата. Това може да доведе до изчакване на твърде голям период от време за завършване на задачата и дори до не завършване на процеса.
- След изпращане на процеса към грид средата отговорността за изпълнението на бизнес-процеса е на g-Lite.

От горните наблюдения на авторите, могат да се направят следните заключения. Средата g-Eclipse, предоставя възможност за моделиране на бизнес-процеси на базата на езика JSDL. Няма възможност за добавяне на цикли или условни разклонения в процеса. Сървъра отговорен за изпълнение на бизнес-процеса е WMS услугата на g-Lite средата. Разчита се на услугите за наблюдение на задачи, които средата g-Lite предоставя. Няма дефинирани механизми за прихващане на грешки и изключения по време на изпълнение на самия бизнес-процес.

Софтуерът g-Eclipse предоставя възможност за съхраняване на разработени от потребителите процеси локално на файловата система на потребителя. Не се

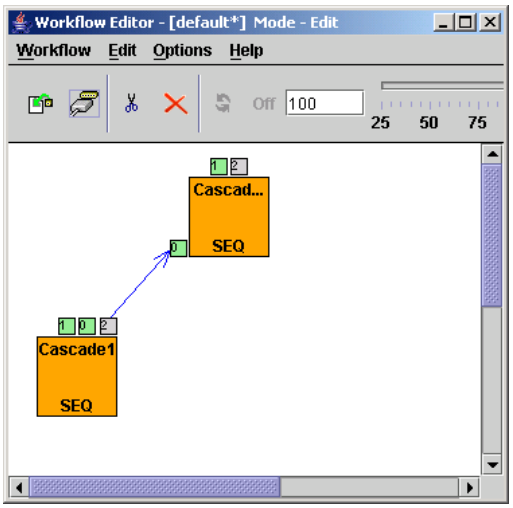
предоставя поддръжка на адаптери за достъп до външни системи. В описанието на процеса, задачите са асоциирани с конкретен изпълним код, който се указва по време на моделиране на процеса. Няма поддръжка на Уеб услуги и поддръжка на шина на услугите. Не се предоставя функционалност за наблюдение на състоянието на процеса и не се поддържат задачи за хора (действия от човек) в бизнес-процеса, както и управление на бизнес-правила за процеса.

2.3 P-GRADE

Софтуерът P-GRADE е Уеб базиран грид портал [52] за достъп до грид ресурси. Порталът е адаптиран и за EGI инфраструктурата и предоставя достъп до грид ресурсите на средата g-Lite (CE, SE, BDII, WMS). Архитектурата на портала е представена в [43]. Уеб частта на портала, представлява набор от портлети интегрирани в среда за разработка на портали (GridSphere). Посредством тях, потребителят може да създаде прокси за достъп до g-Lite средата, както и да създава, изпълнява и наблюдава собствени процеси.

В контекста на софтуера P-GRADE, процесът се обозначава като работен поток (workflow). Той се представя, като ориентиран ацикличен граф (DAG), в който всеки възел представлява конкретна задача в грид, зад която стои конкретен изчислителен ресурс и изпълним код, който трябва да бъде изпълнен на този ресурс. Възлите в потока се свързват чрез дъги. Те показват зависимостите между възлите и посоката на трансфер на файлове между тях. Всеки възел има портове, които указват броя на входните и изходните файлове за този възел. Когато възел в потока, зависи от резултата на друг възел, дъгата свързва изходният порт на единия възел с входният порт на другия възел.

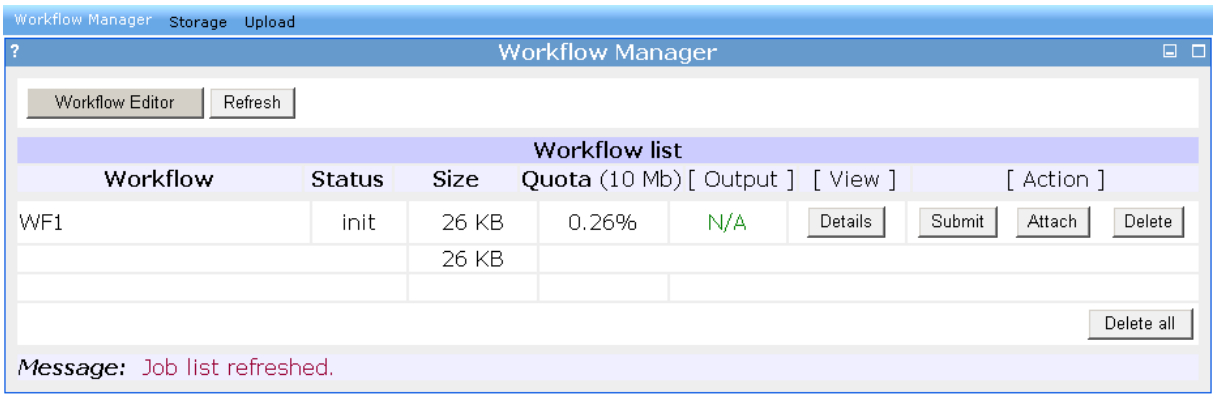
Работен поток се създава, чрез редактор. Редакторът е настолно приложение, което се стартира от настолния компютър на потребителя. На Фиг. 18 [52] е показан редакторът за работни потоци в средата. Работният поток показан на фигурата се състои от два възела. Първият възел има два входни и един изходен порт. Изходните портове са маркирани в сиво, входните портове в зелено.



Фиг. 18. P-Grade редактор за работни потоци

Структурата на графа определя реда на изпълнение. Възможни са последователни и паралелни потоци. При разклонения в графа на потока се задава паралелно изпълнение. Възможни са две нива на паралелизъм на ниво поток и на ниво изпълним код асоциран с конкретен възел. В описанието на работния поток не може да има цикли, нито разклонения на базата на някакво условие. Езикът за описание на работния процес, който се използва от P-GRADE е DAG базиран. В последните версии на софтуера, се предлага и поддръжка на рекурсия и вложени работни потоци.

Уеб частта на портала, предоставя възможност за съхранение на създаден през редактора работен поток, на сървър в хранилище за работни потоци. На Фиг. 19 [52] е показан портлет от Уеб частта на портала, през който работният поток може да се стартира, наблюдава и изтрива.



Фиг. 19. Workflow Manager портлет на P-GRADE портала

За изпълнение на работния поток, се използва сървър за управление на работни потоци. Доскоро във версии на софтуера, като сървър за управление на работни потоци е използвана Condor DAGMan. Сървърът интерпретира описанието на работния поток и

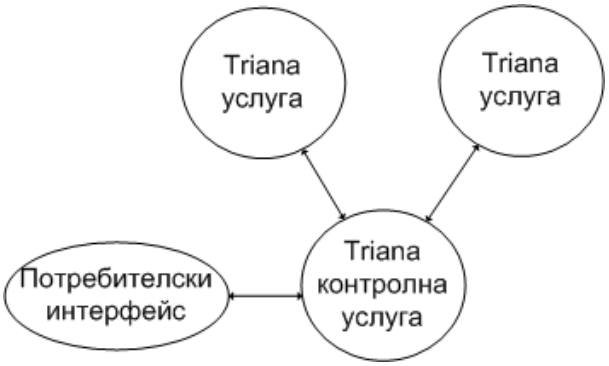
го изпълнява, като изпраща изпълнимия код на възела към система за пакетна обработка в грид. В последната версия на софтуера се използва собствена реализация на сървър за управление на работни потоци.

Софтуерът P-GRADE, предоставя възможност на потребителят да изпълнява собствени бизнес-процеси в EGI. Езикът за описание на процеси, който софтуера поддържа е DAG базиран, т.е. не поддържа BPEL. Софтуерът предоставя регистър за съхраняване на разработени от потребителите процеси. Не се предоставя поддръжка на адаптери за достъп до външни системи. В описанието на процеса, възлите са асоциирани с конкретен грид ресурс и изпълним код, които се указват по време на моделиране на процеса. По тази причина не може да се говори за ориентираност към услуги, поддръжка на Уеб услуги и поддръжка на шина на услугите. Софтуерът предоставя функционалност за наблюдение на състоянието на процеса и не поддържа задачи за хора (действия от човек) в бизнес-процеса, както и управление на бизнес-правила за процеса.

2.4 Triana

Софтуерът Triana е разпределена среда, която предоставя възможност на потребителите да създават и изпълняват процеси (workflows). Софтуерът е използван като базисен в GridLab [79], и е адаптиран за достъп до грид, включително и за EGI.

От архитектурна гледна точка [47] средата е изградена от три основни компонента: потребителски интерфейс, Triana услуга и Triana контролна услуга. Потребителският интерфейс е графичен редактор посредством, който потребителят създава собствени процеси. Потребителският интерфейс комуникира с Triana контролна услуга, която може да изпълни процеса или да го изпрати за изпълнение на някоя от разпределените Triana услуги. Архитектурата на Triana е представена на Фиг. 20.

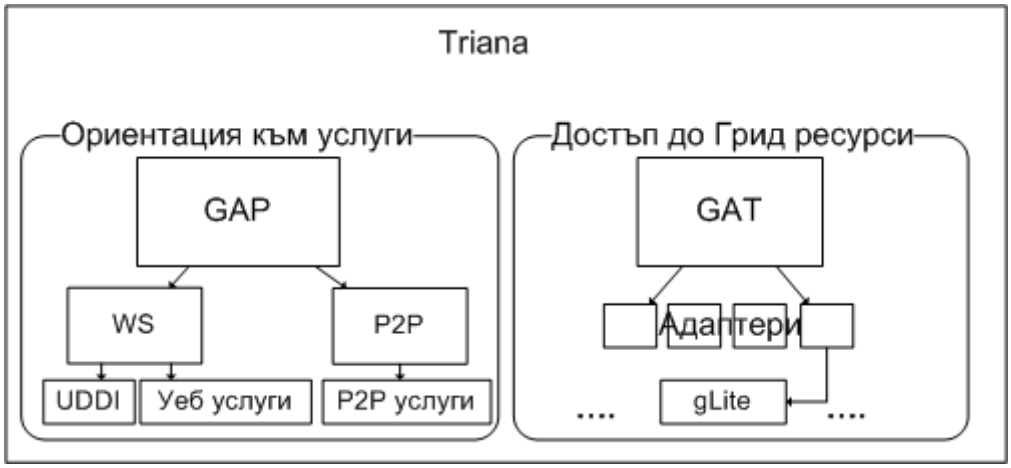


Фиг. 20. Архитектура на Triana

Потребителският интерфейс е графичен редактор, локално инсталиран на потребителския компютър, който предоставя на потребителя готови компоненти, който потребителя може да композира за да създаде процес. В Triana има два типа разпределени компоненти [61]: компоненти ориентирани към услуги (service-oriented) и компоненти за достъп до грид ресурси (grid-oriented).

Компонентите ориентирани към услуги, предоставят достъп до Уеб услуги и pear-to-pear (P2P) услуги. Софтуерът позволява, посредством графичният редактор да бъдат откривани Уеб услуги, да бъдат създавани Уеб услуги и да бъдат публикувани Уеб услуги. Също Уеб услуги могат да участват в процес, както и новосъздаден процес може да бъде публикуван като Уеб услуга. Стандартът, който се следва е UDDI. Интерфейсът, който реализира достъпа до Уеб услугите в Triana е GAP (Grid Application Prototype).

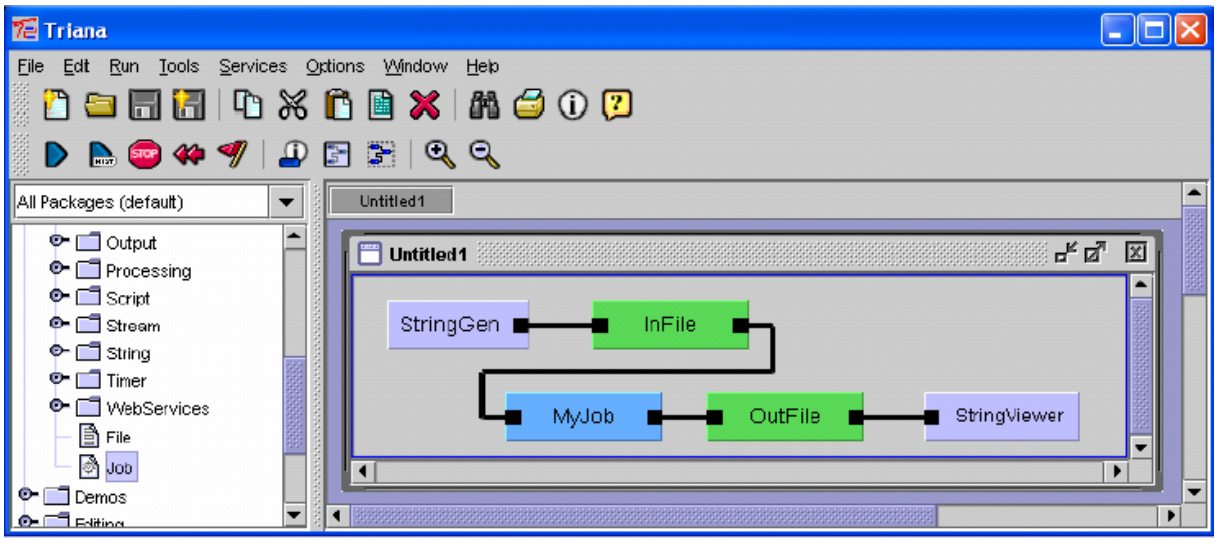
Компонентите за достъп до грид ресурси са отделни задачи, които Triana изпълнява в грид среда. Интерфейсът, който реализира достъпа до грид ресурси в Triana е GAT (Grid Application Toolkit) [8]. Интерфейсът е съвкупност от приложно-програмни интерфейси за достъп до грид услуги. Достъпът до тях се осъществява посредством адаптери. На Фиг. 21 са представени двата интерфейса в Triana.



Фиг. 21. Разпределни компоненти в Triana

Архитектурата на Triana позволява средата да бъде допълвана с нови компоненти и лесно да бъде интегрирана с външни системи. Реализацията на двата интерфейса от друга страна, позволява добавяне на нови адаптери – поддръжка на различни грид среди, както и добавяне на нови услуги.

Графичният редактор на Triana Фиг. 22 [61], предоставя и функционалност за стартиране и спиране на процес.



Фиг. 22. Потребителски интерфейс на Triana

Разработените от потребителите процеси се съхраняват във вид на XML файлове, локално на файловата система на потребителя. Езикът за описание на процеси, който се използва е специфичен за Triana и е XML базиран. Интерпретира се като ориентиран цикличен граф (Directed Cyclic Graph – DCG). Въпреки това, софтуерът поддържа BPEL и DAG описания на процеси. За изпълнение на процеса се използва сървър, който е част от Triana услугата.

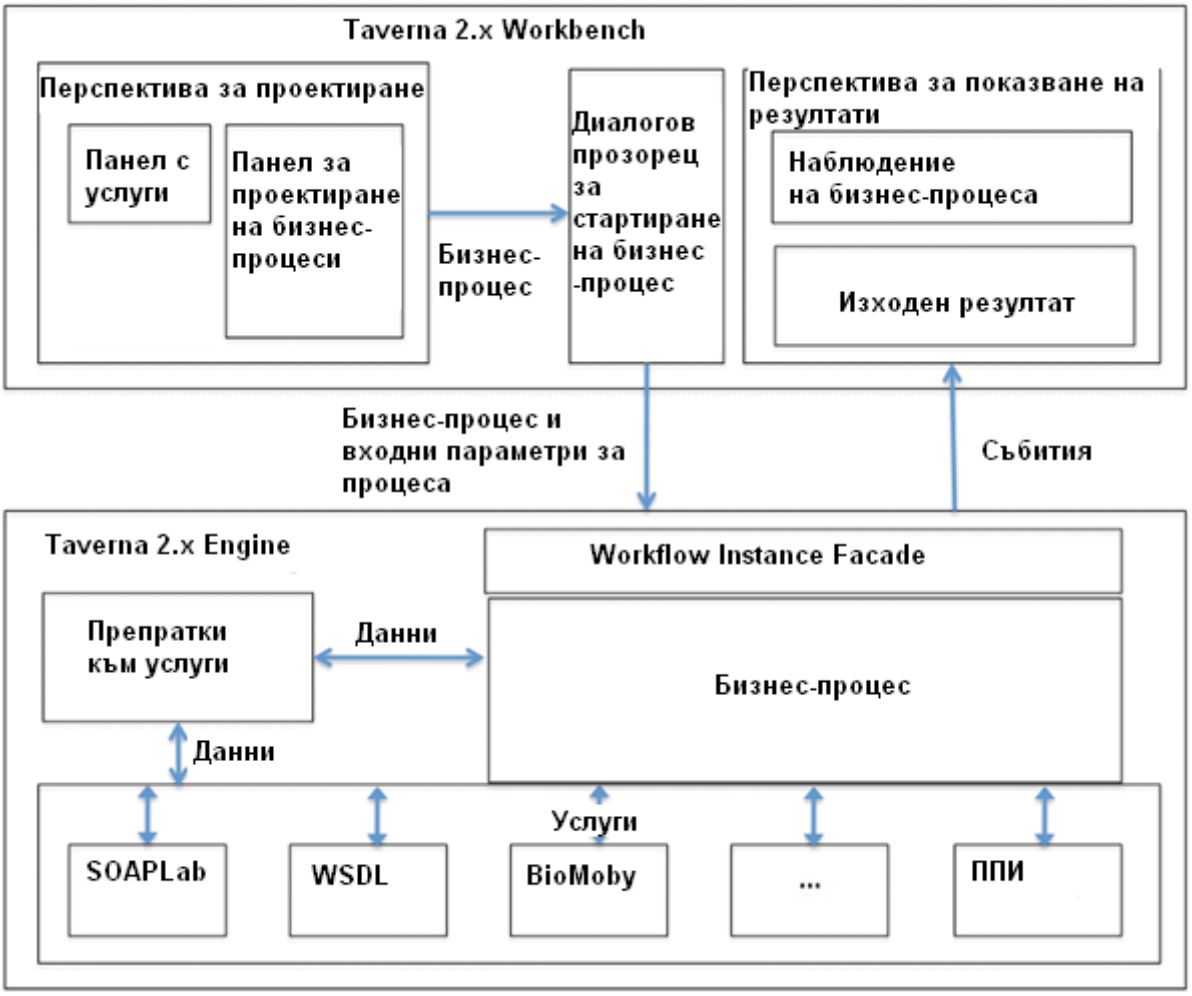
Приложение което използва Triana, като среда за изпълнение на процеси в g-Lite е грид порталът GENIUS [21]. Порталът GENIUS е разработен изцяло за EGI инфраструктурата. Той предоставя лесен достъп до услугите на g-Lite средата. Посредством Уеб-базиран интерфейс потребителят може да изпълнява задачи в g-Lite, да проверява състояние на задачите, да извършва трансфер на файлове в грид. Порталът предоставя и графичен редактор за създаване на процеси, като поддържа създаване и изпълняване на DAG процеси, чрез Triana в g-Lite.

Прави впечатление че, изброените по-горе предимства на Triana – поддръжка на BPEL, UDDI и композиране на Уеб услуги в процеси, не са пълноценно използвани за средата g-Lite. Според [21] процесите, който могат да бъдат изпълнени, чрез Triana в g-Lite се последователност от JDL файлове подредени като ориентиран ацикличен граф. В описанието на процеса по този начин (DAG), не могат да бъдат добавяни цикли и условни разклонения, което от своя страна възпрепятства създаването на по-сложни процеси. Не се поддържат и задачи за хора (действия от човек) в процеса.

Софтуерът Triana представя регистър за услуги, който поддържа UDDI. От разгледаната архитектура на софтуера, в която са реализирани два интерфейса за достъп до компоненти (GAP и GAT), може да се направи извод, че Уеб услугите и UDDI регистъра, не се използват за грид среди и в частност за средата g-Lite.

2.5 Taverna

Софтуерът Taverna [49] предоставя среда за проектиране и изпълнение на бизнес-процеси в грид. Софтуерът се инсталира локално на настолния компютър на потребителя. От архитектурна гледна точка, софтуерът се състои от два модула: модул за проектиране на процеси (Taverna Workbench) и сървър за изпълнение на процеси (Taverna Engine). На Фиг. 23 [49] е представена архитектурата на Taverna, версия 2.



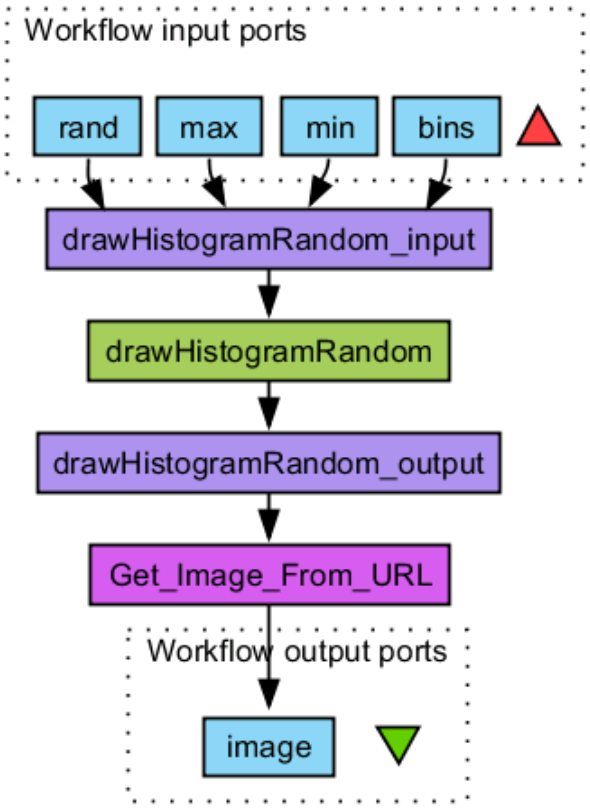
Фиг. 23. Архитектура на Taverna 2

Модулът Taverna Workbench предоставя възможност за моделиране на процеси. Потребителят, моделира процеса, като избира от наличните услуги в средата и ги

подрежда за последователно изпълнение. Наличните услуги в средата са показани в панела с услуги. В перспектива за проектиране, потребителят може да добави нова услуга или да изгради композиция от услуги. Софтуерът позволява да бъдат добавяни и WSDL услуги. Резултатът от композицията на нови услуги е процес (workflow), който се съхранява като t2flow файл – XML базиран. Модулът Taverna Workbench предоставя възможност за стартиране на процеса. В перспектива за показване на резултати, потребителят може да наблюдава процеса и да разглежда резултата от вече изпълнения процес.

Вторият модул Taverna Engine реализира компонент (Workflow Instance Facade), който интерпретира t2flow файла. Интересно е, че в Taverna 2 реално няма отделен сървър, който да изпълнява процес. Бизнес-процесът се изпълнява самостоятелно, като всяка една от услугите в процеса се извиква последователно. Ролята на компонента WorkflowInstanceFacade е да идентифицира, конкретна инстанция на процеса, който се изпълнява. За трансфер на данни в процеса, Taverna 2.x използва препратки към услуги. Модулът Taverna Engine предоставя и допълнителна информация за времето, за което процеса е бил изпълнен, междини данни от изпълнението на процеса, както и крайните резултати от изпълнението на процеса.

На Фиг. 24 е показан примерен процес моделиран с Taverna.



Фиг. 24. Бизнес-процес моделиран в Taverna

За софтуера Taverna 2.0 е разработен допълнителен пакет (TavernaGrid Plugins) за създаване на процеси в грид среда. Пакетът е разработен и представен в [36] и предоставя достъп до услуги, които се изпълняват в g-Lite. За съжаление предоставените препратки в документацията на TavernaGrid не са налични, което възпрепятства проучването на реализираните пакети.

Друга подобна разработка, която позволява изпълнение на процеси в g-Lite се базира на по-стара версия на софтуера Taverna 1.x и сървър за изпълнение на процесите Moteur. Сървърът Moteur е специално разработен [29] за изпълнение на процеси в грид среда, включително и в грид инфраструктурата EGI. Процеси изпълнявани от сървъра Moteur, могат да бъдат проектирани и променяни единствено, чрез графичния редактор на Taverna. Езикът за описание на процеси, който се поддържа от Taverna 1.x и Moteur е SCUFL (Simple Conceptual Unified Flow Language). Езикът е XML базиран.

От казаното до тук, могат да бъдат направени следните изводи. Средата Taverna е инструментално средство за моделиране на научни бизнес-процеси. Предоставя се възможност на потребителя да създава процеси на базата на съществуващи услуги, както и да добавя нови услуги от налични описания на услуги. Средата предоставя локален регистър, инсталиран на настолния компютър на потребителя, който предоставя достъп до всички налични услуги. Потокът на предаване на информация при изпълнение на процеса е последователен, но е възможно и паралелно изпълнение на услуги. Потребителят няма възможност да задава условни разклонения в процеса, но има възможност да включва в процеса WSDL услуги, включително услуги от средата g-Lite. Софтуерът Taverna, предоставя функционалност за наблюдения на процеса и не предоставя възможност за намеса в процеса от човек или възможност за управление на правила за процеса. Езикът, който се използва за описание на процеса е XML базиран и не поддържа BPEL. В [60] авторите изтъкват и причини поради, които Taverna не поддържа BPEL. В заключение, въпреки възможността за композиране на услуги за g-Lite, Taverna, не поддържа инфраструктура за комуникация и управление на услугите.

2.6 Kepler

Софтуерът Kepler [28] е инструментална среда, която предоставя възможност на потребителите да създават, изпълняват и споделят модели на процеси (workflows). Средата предоставя графичен интерфейс за проектиране на процесите, който се инсталира локално на настолния компютър на потребителя. Процесите се моделират

графично и могат да бъдат описани посредством графична нотация, състояща се от пет основни компонента. Това са: оркестратори (directors), актьори (actors), параметри, връзки и портове. Най-важни са първите два компонента. Оркестраторите контролират изпълнението на процеса. Актьорите предоставят определена функционалност и изпълняват инструкциите на оркестратора. Всеки процес има един оркестратор.

В софтуера Kepler има дефинирани шест типа оркестратори. В (Таблица 4) [38], са изброени само три от тях, които са и най-често използвани при моделиране на процеси.

Оркестартор	**Описание**
	SDF (Synchronous Dataflow) – последователно изпълнение на компонентите в процеса
	PN (Process Networks) - паралелно изпълнение на компонентите в процеса
	DDF (Dynamic Dataflow) – последователно изпълнение на процес, в който има цикли или разклонения.

Таблица 4. Оркестратори в Kepler

Софтуерът предоставя и готова реализирана функционалност във вид на актьори. Това позволява един и същи компонент – актьор да бъде многократно използван в различни процеси. Всеки актьор има портове посредством, които комуникира с другите актьори. Чрез портовете актьорите, приемат и изпращат данни помежду си. Актьорите се свързват в процес, посредством портовете си. Линията, която ги свързва се нарича канал. Каналът, показва потока на данни от един актьор към друг. Портовете могат да бъдат:

- Входни – за приемане на данни;
- Изходни – за предаване на данни;
- Входно/изходни – за приемане и предаване на данни.

Връзките позволяват на потребителите да разклоняват потока на данни. Така едни и същи данни могат да бъдат изпратени паралелно към различни актьори на процеса. Параметрите са променливи, които се асоциират с конкретен актьор, оркестратор, дори и процес. Предимство в Kepler е, че могат а бъдат разработвани и добавяни нови актьори към средата.

В (Таблица 5) [28] е показана графичната нотация на някои компонентите от Kepler.

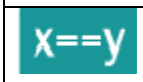
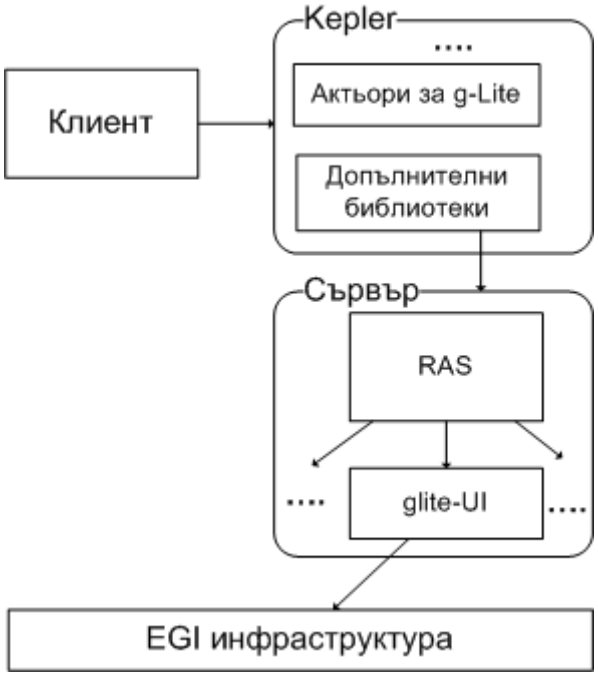
Компонент	**Описание**
	Оркестратор. Компонент който контролира изпълнението на другите компоненти (актьори).
	Актьор - условие. Позволява разклонение в процеса, на базата на зададено логическо условие. В случая равенство.
	Съставен актьор. Компонент, който се състои от множество актьори, които съвместно изпълняват по-сложени операции. Аналог на под-процес.
	Актьор – адаптер. Позволява достъп до външни системи (R, SAS и MATLAB). В случая MATLAB.
	Актьор – Уеб услуга. Позволява изпълняването на Уеб услуга.
	Актьор – визуализация. Компонент за визуализиране на резултат. Използват се за визуализиране на изходния резултат от процеса в текстов или графичен формат.

Таблица 5. Компоненти в Kepler

Разработените от потребителите процеси се съхраняват във вид на XML файлове, локално на файловата система на потребителя. Езикът за описание на процеси, който се използва в Kepler е MoML [46]. Езикът е разширение на XML и се използва за моделиране на процеси (workflows). Всеки актьор от процеса освен чрез графична нотация е описан и чрез MOML файл, в който са дефинирани: параметрите, портовете и други допълнителни настройки.

Софтуерът Kepler е разширен в проекта Euforia (Serpens) [15] за изграждане и изпълнение на процеси в грид. За средата Kepler са разработени нови актьори ([48], [51]), които предоставят поддръжка на различни грид среди, една от които е и средата g-Lite. Разширенията включват стандартни дейности, като изпълнение на задача в грид средата, създаване на прокси, наблюдение на задача, трансфер на данни в g-Lite, проверка на състояние на задача и извличане на резултат.

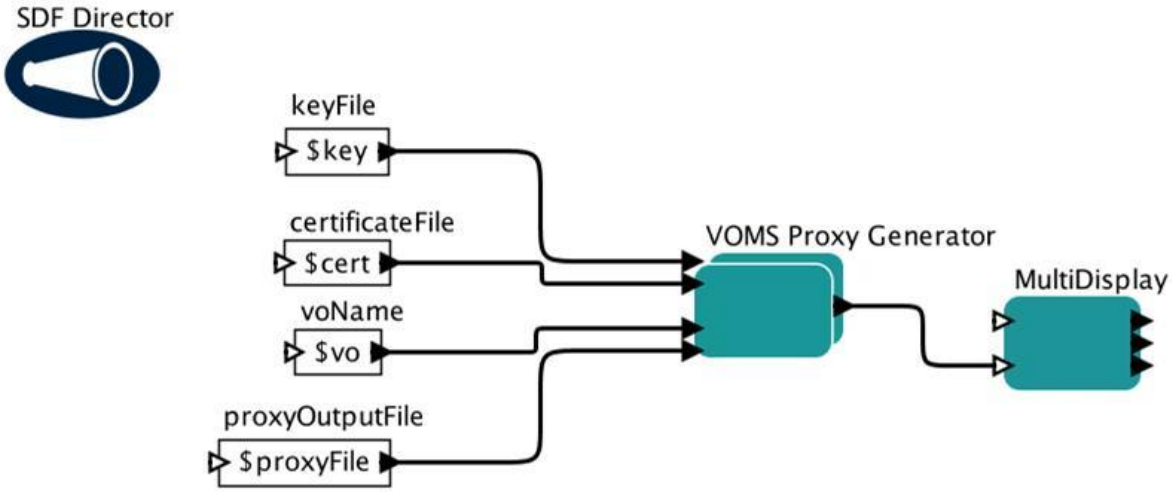
На Фиг. 25 [15] е показана част от архитектурата, разработена по проекта Euforia, която позволява изпълняване на процеси (workfows) в грид средата g-Lite, на базата на Kepler.



Фиг. 25. Адаптиране на Kepler за g-Lite

Моделираният от потребителя процес се изпълнява от вграден в средата Kepler сървър. За достъп до грид услуги на средата g-Lite, е разработен отделен сървър (Roaming Access Server, RAS), който играе роля на посредник между Kepler и услугите в g-Lite.

На Фиг. 26 [50] е показан примерен процес, за създаване на прокси в g-Lite грид среда, моделиран чрез софтуера Kepler.



Фиг. 26. Бизнес-процес моделиран с Kepler

Софтуерът Kepler с разширенията от модула Serpens, предоставя възможност на потребителя да изпълнява собствени бизнес-процеси в g-Lite среда. Езикът за описание на процеси, който софтуера поддържа е MoML – XML базиран. Няма поддръжка на регистър на услуги. Предоставя се частична поддръжка на адаптери за достъп до външни системи (само до R, SAS и MATLAB). В описанието на процеса, могат да участват и актьори-услуги, което позволява изграждането на процеси от Уеб услуги. Няма единна среда за обмен на съобщения между Уеб услугите в процеса, т.е., не се поддържа шина на услугите. Софтуерът Kepler предоставя функционалност за наблюдение на състоянието на процеса и не поддържа възможност за задаване в процеса на задачи за хора (действия от човек) или възможност за управление на отделни правила за процеса.

2.7 Сравнение на разгледаните среди

Така разгледаните среди могат да бъдат условно разделени на две групи. Към първата група спадат среди, които не предоставят готови услуги за композиране в бизнес-процеси. Такива са CRAB, g-Eclipse и P-GRADE. Чрез тези среди, потребителят може да създаде процес, като последователност от изпълними задачи, които да бъдат изпълнени в EGI. При описанието на процеса, потребителя задава JDL файлове или конкретни изпълними файлове за изпълнение. При този тип среди, потребителят не може да усложни процеса с добавяне на логика, която изисква цикъл или условни разклонение.

Към втората група спадат среди, които предоставят готови услуги за композиране в бизнес-процес. Такива са Triana, Taverna и Kepler. Този тип среди предоставят и графичен редактор за създаване и стартиране на процеса. Потребителят създава процеса на базата на вече готови услуги (компоненти, актьори). Процесът се изпълнява от сървър специфичен за конкретната среда, който се намира извън грид средата. Сървърите са адаптирани за изпълнение на процеси в грид.

Прави впечатление, че при описанието на процеса във всички разгледани среди се акцентува на управлението на данните в процеса: потокът на данни между отделните услуги, но не и върху управлението на самия процес. По тази причина и езиците за описание на процеси, които се използват в тези среди, предоставят нотация, чрез която се специфицират входните данни, изходните данни и трансфера на данни между отделните услуги в процеса, но не и нотация за описание на по-сложна логика в

процеса. Такава нотация се предоставя от езика BPEL. Езикът BPEL е и единственият одобрен стандарт за описание на изпълними бизнес-процеси.

В (Таблица 6) са обобщени езиците за описание на бизнес-процеси, които се използват в разгледаните среди.

Среда	**CRAB**	**g-Eclipse**	**P-Grade**	**Triana**	**Taverna**	**Kepler**
Език за описание на бизнес-процеси	−	JSDL	DAG	DAG	T2FLOW	MoML

Таблица 6. Езици за описание на бизнес-процеси в среди за проектиране и изпълнение в EGI

Друго общо между разгледаните среди (Triana, Taverna и Kepler) е технологията, използвана за проектиране и изпълнение на процесите. Сървърите за изпълнение на процеси, които средите реализират следват концепцията на система за управление на потоци (Workflow Management System - WfMS), дефинирана и специфицирана от [65].

Според дефинициите в трети раздел на първа глава, бизнес-процесът може да включва автоматизирани дейности или дейности извършвани от човек. Основното при системите за управление на потоци е, че те управляват само автоматизираните дейности, докато дейностите извършвани от човек са извън техният обсег.

При средите за управление на бизнес процеси, това не е така. Нещо повече, не всички системи за управление на потоци предоставят инструментални средства за проверка на процеса или средства за диагностика на процеса. Някои от разгледаните среди предоставят функционалност за наблюдение на процеса, по време на изпълнението му (P-GRADE, Triana, Taverna и Kepler). Но нито една от тях не предоставя функционалност за оптимизиране на процеса. Разработването на един процес е продължителна задача. С времето, могат да се променят правила за процеса, да се добавят нови правила или нови дейности. По време на изпълнението на бизнес-процеса, може да бъде събрана допълнителна информация, която да спомогне оптимизиране и подобряване на самия процес. Тази функционалност се поддържа от средите за управление на бизнес-процеси, но не и от средите за управление на потоци.

Друг важен факт е, че нито една от разгледаните среди, не поддържа възможност за дефиниране на задачи за хора (действия от човек) в процеса или възможност за управление на отделни правила за процеса. Дори и средите ориентирани към услуги (Triana, Taverna и Kepler) не поддържат обща инфраструктура за комуникация и управление на услугите (шина на услугите в предприятието). В

софтуерът Triana се наблюдава подобна функционалност реализиране чрез интерфейс GAP, която обаче не е пълноценно приложена за грид.

В (Таблица 7), са обобщени резултатите от анализа на разгледаните среди, според изброените в началото на Глава 2 изисквания.

Критерии	**CRAB**	**g-Eclipse**	**P-Grade**	**Triana**	**Taverna**	**Kepler**
Поддръжка на Уеб услуги	–	–	–	✔	✔	✔
Поддръжка на BPEL	–	–	–	✔	–	–
Поддръжка на регистър	–	–	✔	✔	–	–
Поддръжка на адаптери	–	–	–	✔	–	✔
Поддръжка на шина на услугите	–	–	–	–	–	–
Поддръжка на задачи за хора (действия от човек) в бизнес-процеса	–	–	–	–	–	–
Поддръжка на наблюдение на бизнес-процеса	–	–	✔	✔	✔	✔
Поддръжка на управление на бизнес-правилата за процеса	–	–	–	–	–	–

Таблица 7. Среди за проектиране и изпълнение на бизнес-процеси в EGI

2.8 Изводи

В настоящата глава бяха изследвани и анализирани съществуващи среди за проектиране и изпълнение на бизнес-процеси в EGI, на базата на формулирани изисквания. Разгледаните среди предоставят възможност за създаване и изпълнение на бизнес-процеси в грид, включително и в грид инфраструктурата EGI. Управлението на един бизнес-процеси, обаче включва не само дефиниране и изпълняване на процеса, но и възможността за наблюдение и оптимизиране на бизнес-процеса. Бизнес-процесите могат да включват и задачи за хора (действия от човек). Тези функционалности не са предвидени и съответно не са реализирани в разгледаните среди. Причините за това са две. На първо място, сървърите за изпълнение на бизнес-процесите, които средите използват следват концепцията на система за управление на потоци. На второ място,

общата линия, която средите следват е насочена към управление на потока на данни в процеса не и към управление на самия процес.

Нито една от разгледаните среди не предоставя поддръжка на шина на услугите, поддръжка на задачи за хора (действия от човек) в бизнес-процеса или поддръжка на управление на бизнес-правила за процеса. Предимства, поддържани единствено от средите за управление на бизнес-процеси. Средите за управление на бизнес-процеси предоставят възможност на потребителя да променя динамично правилата към бизнес-процеса, да наблюдава процеса и да го оптимизира. Към настоящия момент EGI не предоставя среда, базирана на ориентираната към услуги архитектура, която да предоставя поддръжка на бизнес-процеси в разглежданите до момента термини.

В следващата глава представяме модел и архитектура на среда за управление на бизнес-процеси в грид, в контекста на ориентирана към услуги архитектура. Разгледани са въпросите свързани с дефинирането, проектирането и изпълнението на бизнес-процеси в грид и по специално в Европейската грид инфраструктура.

Глава 3. Модел на среда за управление на бизнес-процеси в грид

Представеният в първа глава модел на среда за управление на бизнес-процеси базирана на ориентираната към услуги архитектура, покрива пълния жизнен цикъл на управление на бизнес-процеси, като предоставя функционалност за моделиране, изпълнение, наблюдение и оптимизиране на бизнес-процеси, чрез поддръжка на стандарти за Уеб услуги. Следвайки този модел, в настоящата глава на дисертацията ще предложим модел на среда, базирана на ориентираната към услуги архитектура, която да поддържа бизнес-процеси в грид инфраструктурата EGI, чрез поддръжка на стандарти за Уеб услуги. Разработката на модела покрива следните въпроси:

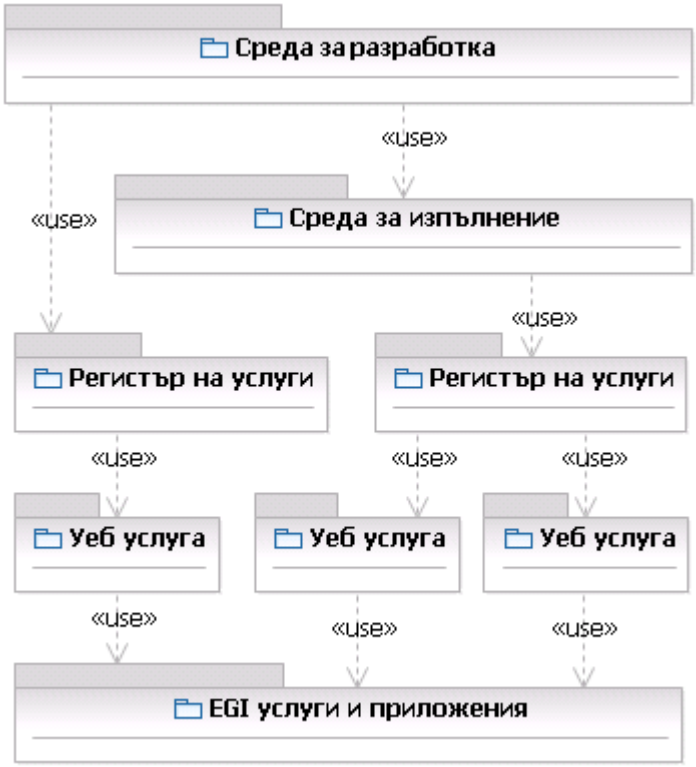
- Основни модули и компоненти на среда за управление на бизнес-процеси в грид инфраструктурата EGI;
- Роли и случаи на употреба в среда за управление на бизнес-процеси в грид инфраструктурата EGI;
- Допълнителни изисквания към среда за управление на бизнес-процеси в грид инфраструктурата EGI;

За представяне на базовата функционалност на компонентите в средата са използвани описателните възможностите на езика за моделиране UML.

3.1 Модули и компоненти на среда за управление на бизнес-процеси в EGI

Следвайки общият архитектурен модел на среда за управление на бизнес-процеси представен в първа глава, средата за управление на бизнес-процеси в EGI, предоставя три основни модула: среда за разработка, среда за изпълнение и регистър на услуги. Грид инфраструктурата EGI и прилежащите към нея услуги и приложения са налични. Те не могат да бъдат променени. В този контекст модулите на средата за управление на бизнес-процеси, могат да се разглеждат като надстройка над грид инфраструктурата на EGI.

На Фиг. 27 представяме трите модула на средата, като надстройка над EGI.



Фиг. 27. Компонентен модел на среда за управление на бизнес-процеси в EGI

Предложеният модел се състои от пет слоя:

- Първи слой – EGI услуги и приложения, включва всички услуги предоставяни от грид инфраструктурата EGI и грид приложения;
- Втори слой – Уеб услуги, включва всички разработени Уеб услуги за достъп до EGI услуги и приложения;
- Трети слой - Регистър на услуги, включва всички услуги-регистри, които предоставят възможност за търсене и публикуване на Уеб услуги;
- Четвърти слой - Среда за изпълнение, включва компонентите: шина на услугите в предприятието, услуги за изпълнение на бизнес-процеси, услуги за наблюдение на бизнес-процеси, услуги за управление на задачи за хора и услуги за управление на бизнес-правила;
- Пети слой - Среда за разработка, обхваща инструменталните средства за проектиране и разгръщане на бизнес-процеса в среда за изпълнение.

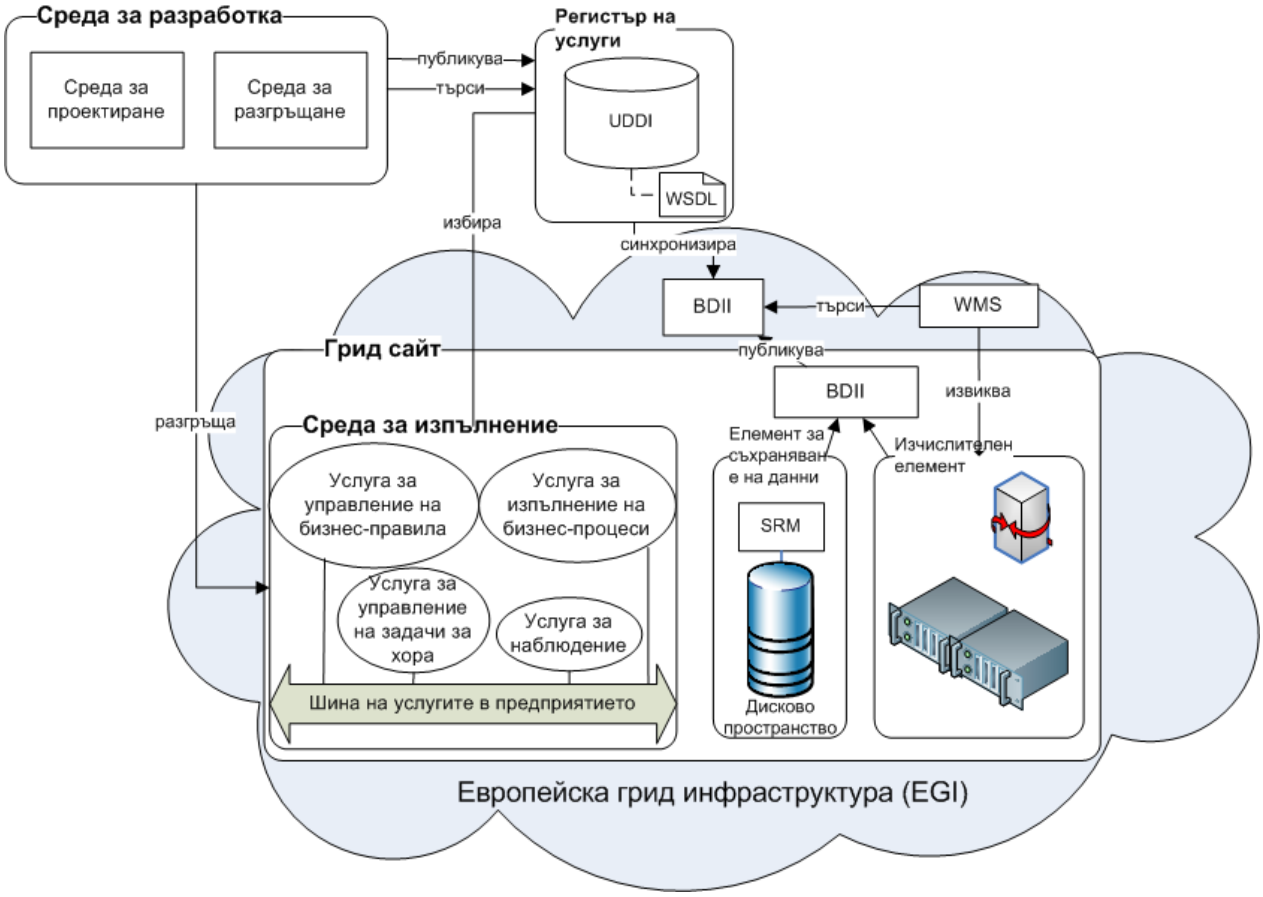
Първия слой в модела на среда за управление на бизнес-процеси в EGI е наличен. Той покрива вече изградена грид инфраструктура с грид среда за достъп до изчислителни и дискови ресурси в нея.

Вторият слой обхваща всички Уеб услуги за достъп до EGI и приложения. Уеб услугите са важна част от модела, на базата на тях се композират бизнес-процеси в по-горните слоеве. Уеб услугите могат да бъдат разработени от специалисти по информационни технологии или да бъдат налични (т.е. вече съществуващи за EGI).

Третият слой, предоставя Уеб услуги-регистри, в които са публикувани Уеб услугите от втория слой. Достъпът до услугите-регистри може да бъде осъществяван, както през среда за изпълнение така и през среда за разработка.

В четвъртия и петия слой от модела на средата се създават, изпълняват и управляват дефинираните от грид-потребителя бизнес-процеси. Те са подробно разгледани в следващите раздели на настоящата глава.

На Фиг. 28 са показани трите основни модули на средата и тяхното разположение в грид инфраструктурата EGI.



Фиг. 28. Модули и компоненти на среда за управление на бизнес-процес в EGI

3.1.1 Модул Регистър на услуги

Регистърът на услуги е хранилище за Уеб услуги и асоциираните с тях документи и политики. Всички налични услуги в EGI трябва да са част от регистъра, като Уеб услуги. Това предполага съществуването на Уеб услуги за всички основни

EGI услуги и интеграция на регистъра с информационната система на грид инфраструктурата.

За всяка Уеб услуга в регистъра трябва да се съхраняват: WSDL описание на услугата, версия на услугата и допълнително текстово описание за типа на услугата. Към всяка Уеб услуга може да има прилежащи документи, както и зависимости от други Уеб услуги. Регистърът на услуги, трябва да предоставя възможност за публикуването на Уеб услуги с прилежащи към тях документи и зависимости. Регистърът трябва да поддържа стандарта UDDI.

В грид инфраструктурата, достъпът на потребителите до ресурсите се регламентира според принадлежността на потребителя към виртуална организация и притежанието на грид сертификат. Като услуга, регистърът на услуги не е част от грид инфраструктурата. Той трябва да е свободно достъпен от всеки потребител. Грид-потребителите, трябва да могат свободно да разглеждат всички публикувани услуги в регистъра, без да е необходимо да се идентифицират, като грид-потребители.

Чрез регистъра на услуги, грид-потребителите могат да споделят общи документи и файлове. Достъпът до тях се регламентира от политиките за сигурност на съответните виртуални организации в EGI.

От гледна точка на грид инфраструктурата, регистърът е централизирано хранилище за WSDL файлове на Уеб услугите за достъп до EGI. Предимствата от използването на модула в грид инфраструктурата са многократна употреба на услугите и достъп до услугите при разработката на грид-базирани приложения.

3.1.2 Модул Среда за разработка

Средата за разработка не е част от грид инфраструктурата. Потребителите на EGI, трябва да могат свободно да я инсталират на локалните си компютри, без да е необходимо да се идентифицират с грид сертификат. Средата за разработка се състои от среда за проектиране и среда за разгръщане.

В среда за проектиране, посредством графичният интерфейс, потребителят създава бизнес-процеси. Бизнес-процесите се моделират на базата на вече разработени и съществуващи Уеб услуги, регистрирани в регистъра.

Основно изискване към средата за управление на бизнес-процеси в EGI е потребителят да може сам да проектира и управлява бизнес-процесите си без да са необходими познания по програмиране или намеса на специалисти по информационни

технологии. Предвид това дейностите, които могат да се използват от потребителя при моделиране на бизнес-процесите в среда за проектиране за EGI са:

- Всички структурни дейности, представени в Глава 1, раздел 4;
- Всички дейности събития, представени в Глава 1, раздел 4;
- Основните дейности:
    - Задача бизнес-правило – стъпка от бизнес-процеса за изпълняване на бизнес-правило;
    - Услуга (<invoke>) – стъпка от бизнес-процеса за синхронизирано извикване на Уеб услуги;
    - Под-процес (<под-процес>)– стъпка от бизнес-процеса, обозначаваща йерархия в процес, за който са в сила горните точки.

Липсата на основните дейности: задача, задача за хора, задача за получаване и задача за изпращане не е случайна. Ограниченият брой дейностите, които могат да участват в бизнес-процесите за EGI, позволяват на грид-потребителите да разгърнат бизнес-процесите в средата за изпълнение, без да са необходими познания по програмиране или намеса на специалист.

Трябва да отбележим, че не поддържането на основната дейност задача за хора в средата за проектиране, съвсем не означава не поддържане на задачи за хора в средата за управление на бизнес-процеси в EGI.

Задачите за хора могат да бъдат:

- Човек-човек - задачи, които отразяват съвместна дейност между потребителите;
- Услуга-човек - стъпка от бизнес-процеса, извършвана от човек;
- Човек-услуга - задачи, при които потребител извиква услуга.

Средата за управление на бизнес-процеси в EGI, поддържа единствено задачи за хора от типа човек-услуга. Примери за такива задачи са стартиране на бизнес-процес, спиране на бизнес-процес и други. Задачи за хора от типа: човек-услуга, се интерпретират от модула среда за изпълнение, разгледан в следващия раздел на настоящата глава.

В средата за проектиране, грид-потребителите могат да задават ключови за бизнес-процеса индикатори, по които той да бъде наблюдаван по време на изпълнение. Пример за такива индикатори са средно време за изпълнение на дейност от процеса, време за изпълнение на целия процес, средно време за изпълнение на целия процес и други.

В средата за проектиране, грид-потребителите могат да задават и бизнес-правила, като стъпка от процеса. Бизнес-правилата са твърдения, които ограничават бизнес-процеса. Бизнес-правилата са атомарни и задават условие, което трябва да бъде валидно за бизнес-процеса.

Средата за проектиране трябва да предоставя възможност на потребителите да съхраняват разработените от тях бизнес-процеси, както и да изнася разработените от тях бизнес-процеси към средата за разгръщане.

Средата за разгръщане е инструментално средство за разработка на бизнес-процеси, грид приложения и Уеб услуги. Тя може да се използва, както от грид-потребители (неспециалисти), така и от специалисти по информационни технологии.

За грид-потребителите средата за разгръщане, трябва да предоставя възможност да внесат, разработеният от тях бизнес-процес, при което автоматично да се генерира BPEL описанието на процеса, както и необходимия програмен код за дейностите в него. Ако при моделиране на бизнес-процеса, грид-потребителят е указал ключови индикатори за наблюдение или бизнес-правила за тях също трябва да се генерира съответния програмен код.

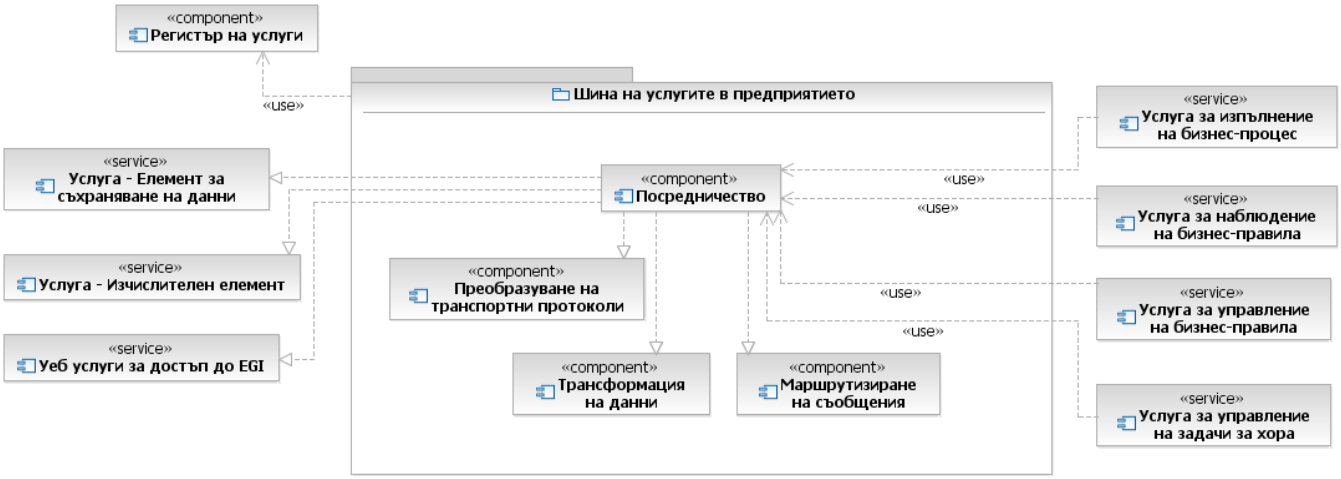
За специалистите по информационни технологии, средата за разгръщане предоставя инструментариум за създаване на Уеб услуги и грид приложения. Това включва и съответните клиенти за достъп до регистър и среда за изпълнение.

Средата за разгръщане трябва да предоставя на потребителите, клиенти за разгръщане на създадения от ИТ специалистите или генерирания за грид-потребителите програмен код в среда за изпълнение.

3.1.3 Модул Среда за изпълнение

Средата за изпълнение е част от грид инфраструктурата. Тя включва следните компоненти: услуга за изпълнение на бизнес-процеси, услуга за управление на бизнес-правила, услуга за наблюдение и услуга за управление на задачи за хора. Те използват обща инфраструктура за обмен на съобщения – шина на услугите в предприятието. Четирите услуги са непосредствено свързани една с друг. По тази причина те могат да бъдат разглеждани и като един компонент, който по-рано обозначихме като процесен сървър.

На Фиг. 29 е показано взаимодействието между четирите услуги в средата за изпълнение и услугите от грид инфраструктурата.



Фиг. 29. Взаимодействие между услугите в среда за изпълнение в EGI

Услугата за изпълнение на бизнес-процеси интерпретира BPEL описанието на бизнес-процеса, като извиква съответните грид услуги в него и проверява съответните бизнес-правила.

Услугата за наблюдение, позволява наблюдаване на бизнес-процеса по време на изпълнението му по ключови индикатори за наблюдение, указани при проектиран на бизнес-процеса. Комуникацията между услугата за наблюдение и услугата за изпълнение се осъществява през шина на услугите.

Услугата за управление на бизнес-правила, предоставя на грид-потребителя, възможност да променя бизнес-правилата по време на изпълнение на бизнес-процеса. Комуникацията между двете услуги, се осъществява през шина на услугите.

Услугата за управление на задачи за хора, позволява грид-потребителят да извиква бизнес-процес, например да стартира бизнес-процеси, да спира бизнес-процес и други. Комуникацията между услугата за управление на задачи и услугата за изпълнение, се осъществява през шина на услугите. Чрез услугата за управление, потребителите могат също да управляват задачите си, като временно ги спират, подновяват или им задават срок на валидност.

Средата за изпълнение се състои от набор от услуги. Тези услуги са локализирани физически на различни места, но са достъпни логически под едно и също име. Услугите от средата за изпълнение трябва да са публикувани в регистъра, като достъпа до тях се осъществява, както достъпа до всяка една от останалите услуги в регистъра.

Концептуално, компонентите на средата за изпълнение трябва да са част от грид инфраструктурата, инсталирани локално на всеки грид сайт от EGI. Те трябва да са основни елементи в грид инфраструктурата, каквито са елементите за съхраняване на данни и изчислителния елемент. Обменът на съобщения в грид сайта трябва да се осъществява през компонента шина на услугите в предприятието, така достъпа до всяка услуга в грид сайта ще се осъществява през шината. Комуникацията между услугите от средата за изпълнение и услугите на грид сайта, също трябва да се осъществява през шината. Шината на услугите в предприятието трябва да е надстройка над услугите в EGI грид инфраструктурата.

Тези изисквания предполагат промени в грид инфраструктурата и прилежащите към нея услуги и приложения. Както беше отбелязано в началото на настоящата глава, обаче първият и вторият слой в модела на среда за управление на бизнес-процеси в EGI са налични. Тези съображения са взети предвид при проектиране на софтуерната архитектура на средата, представена по-долу в настоящата глава.

3.2 Роли и случаи на употреба в среда за управление на бизнес-процеси в EGI

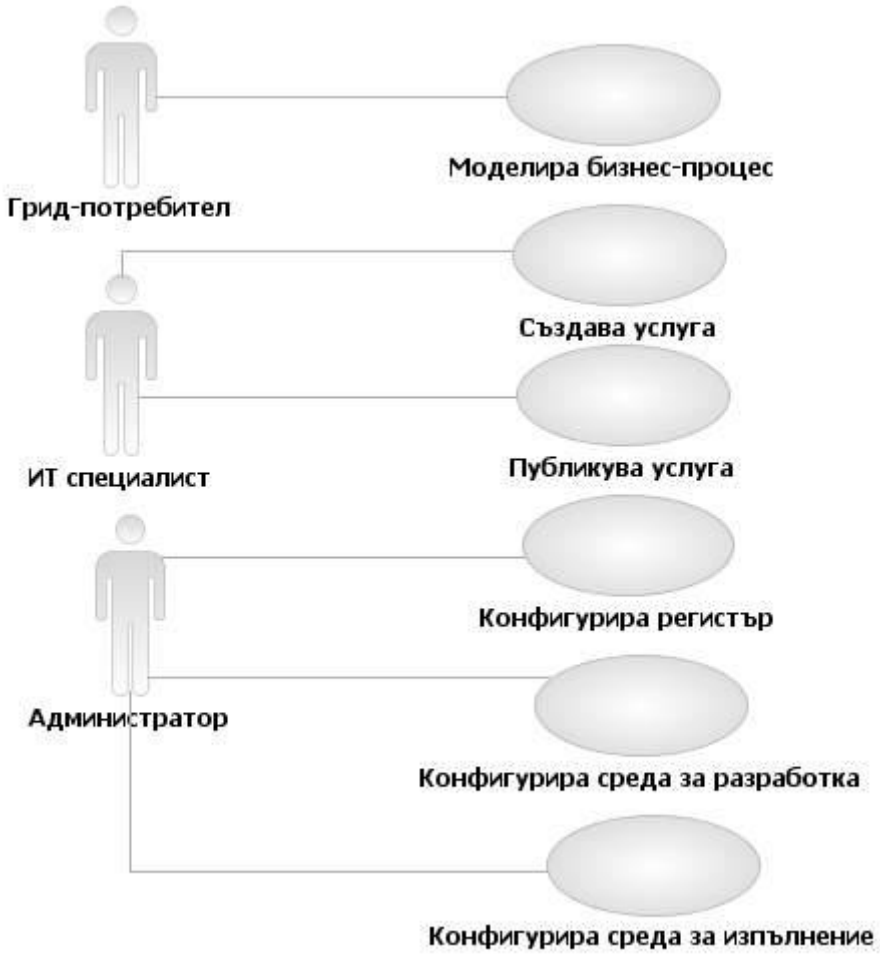
По-долу описваме роли и случаи на употреба в среда за управление на бизнес-процеси в EGI.

3.2.1 Роли в среда за управление на бизнес-процеси

Основните участници (роли) в средата за управление на бизнес-процеси в EGI са:

- Грид-потребител (grid user) – потребител на средата за управление. Грид-потребителят проектира бизнес-процеси в средата за разработка, на база на съществуващи Уеб услуги публикувани в регистъра. Грид-потребителят притежава сертификат за достъп до EGI и е регистриран към виртуална организация.
- Специалист по информационни технологии – ИТ специалист (developer) – разработва Уеб услуги в среда за разработка и ги публикува в регистъра.
- Администратор (administrator) – администрира бизнес-процеси на грид-потребители, задава права на достъп на грид-потребители и ИТ специалисти до модулите Регистър и Среда за изпълнение. Администраторът притежава сертификат за достъп до EGI и е регистриран към виртуална организация.

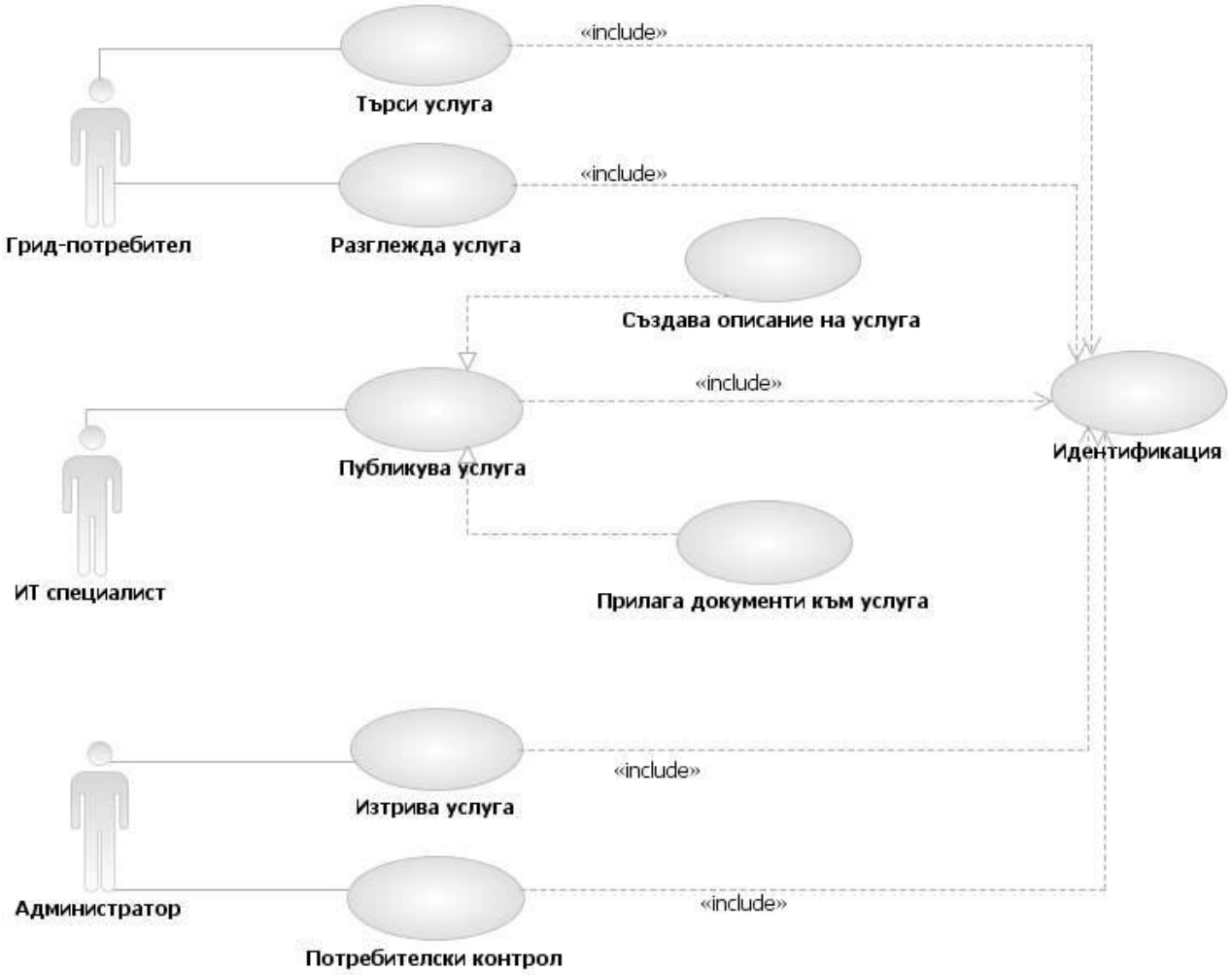
На Фиг. 30 е показана UML диаграма на ролите в среда за управление на бизнес-процеси в EGI и някои базови дейности в средата, според ролята на потребителя.



Фиг. 30. Роли в среда за управление на бизнес-процеси и базови дейности в средата

3.2.2 Случаи на употреба в регистър на услуги

Основната функционалност, която регистърът предоставя е публикуване на нови Уеб услуги и откриване на налични Уеб услуги. На Фиг. 31 са показани някои основни случаи на употреба за регистър, според ролята на потребителя.



Фиг. 31. Основни случаи на употреба за регистър

Достъпът до модула се контролира от услуга, която проверява ролята и правата на потребителите. Грид-потребителите имат достъп единствено до функционалност за търсене и разглеждане на съществуващи Уеб услуги в регистъра. Всеки регистриран грид-потребител има достъп до описанията на Уеб услугите в регистъра. За да достъпват тази функционалност не се изисква грид сертификат. Сигурността при извикване на Уеб услуга за достъп до грид инфраструктурата от грид-потребител се реализира на ниво услуга, а не на ниво регистър. Така за достъп до описанието на Уеб услуга от EGI не се изисква грид сертификат, но за извикване на Уеб услугата се изисква сертификат и принадлежност на потребителя към виртуална организация.

ИТ специалистите, които разработват грид приложения, могат да публикуват или редактират Уеб услуга в регистъра, като приложат WSDL файла на услугата и прилежащите й документи. Публикуването и редактирането на Уеб услуга в регистъра изисква идентификация на потребителя - ИТ специалист.

Изтриването на услуга в регистъра е ограничено само за потребители с роля администратор. Администраторът осъществява и потребителски контрол, като регистрира потребители и задава права за достъп до регистъра.

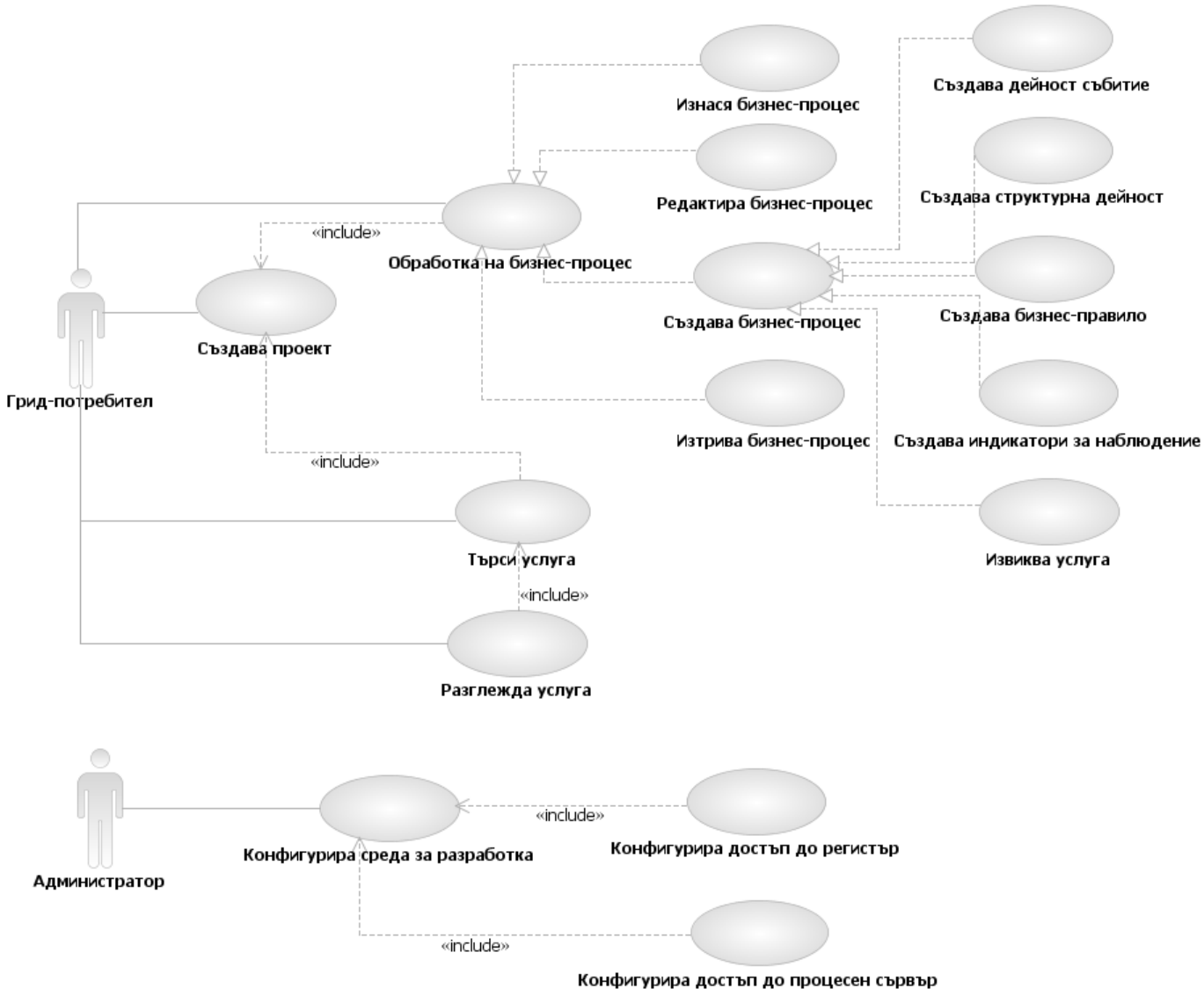
Регистърът е ключов компонент в средата за управление на бизнес-процеси в EGI. От гледна точка на грид-потребителите, регистърът е място където те могат да открият услугите, които са им необходими за да моделират бизнес-процесите си в среда за проектиране. От гледна точка на ИТ специалистите, регистърът е място където те могат да публикуват разработените от тях Уеб услуги в среда за разгръщане.

3.2.3 Случаи на употреба в среда за разработка

Средата за разработка се състои от среда за проектиране и среда за разгръщане. Посредством графичен интерфейс на средата за проектиране, грид-потребителите създават бизнес-процеси. Бизнес-процесът се моделира на базата на вече разработени и съществуващи услуги, регистрирани в регистъра. Основната функционалност, която средата за проектиране предоставя е:

- Създаване на проект – проектът е контейнер за всички бизнес-процеси и прилежащите му дейности, създадени от потребителя. Името на проекта е уникално в рамките средата.
- Създаване на бизнес-процес – изисква създаден проект. Функционалността включва:
    - Създаване на дейност събитие;
    - Създаване на структурна дейност;
    - Създаване на основна дейност: бизнес-правило, услуга, под-процес;
    - Създаване на индикатори за наблюдение;
- Запазване на бизнес-процес – изисква създаден проект;
- Редактиране на бизнес-процес – изисква създаден проект;
- Изнасяне на бизнес-процес към среда за разгръщане;
- Изтриване на бизнес-процес – изисква създаден бизнес-процес;
- Търсене на услуга в регистъра – изисква предварително конфигуриран достъп до регистъра;
- Разглеждане на услуга от регистъра.

На Фиг. 32 са показани някои основни случаи на употреба в среда за проектиране, според ролята на потребителя.



Фиг. 32. Основни случаи на употреба за среда за проектиране

Потребителите на средата за проектиране, могат да бъдат в роля на грид-потребител и на администратор. Администраторът конфигурира средата за проектиране, като конфигурира достъпа до регистъра и достъпа до процесен сървър. Достъпът до процесен сървър, позволява извършване на дейности, като симулация и проследяване на модела на бизнес-процеса.

В средата за проектиране, грид-потребителят може да търси в регистъра услуги по зададен критерии и да разглежда услуги от регистъра. Грид-потребителя създава бизнес-процес чрез композиция на EGI услуги, структурни дейности и дейности събития. Грид-потребителя може да редактира бизнес-процеса, като добавя ключови

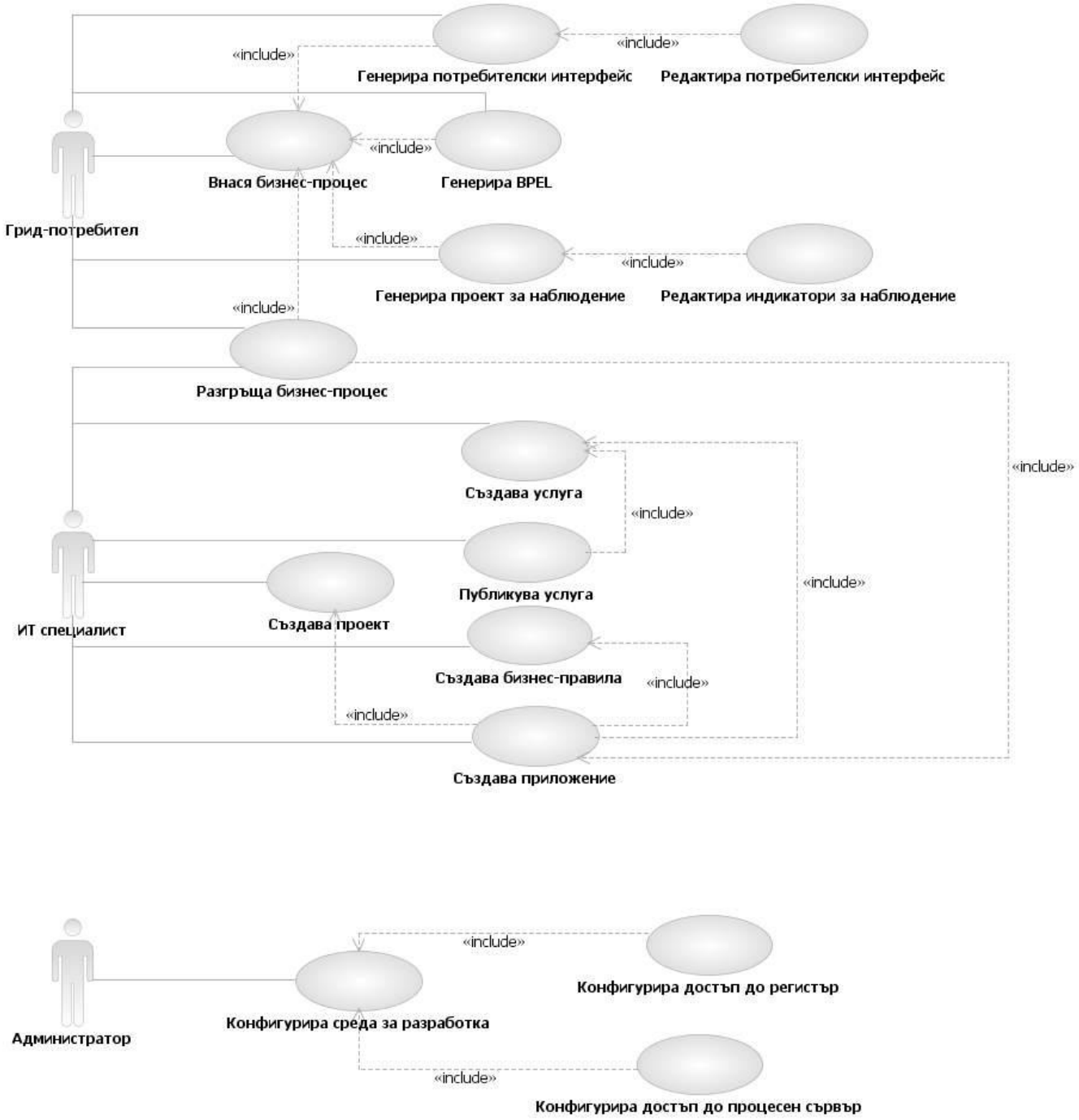
индикатори за наблюдение и бизнес-правила за процеса. Грид-потребителят може да изнесе проектираният бизнес-процес към среда за разгръщане. Тези действия изискват създаването на проект в средата за проектиране. В него се намират всички бизнес-процеси и асоциираните към него дейности, създадени от потребителя.

Средата за проектиране е настолна среда, инструментално средство, което всеки грид-потребител може да инсталира локално на компютъра си. По тези причини идентификацията на потребителите се извършва от операционната система и не се изисква идентификация на потребителя в средата.

Средата за разгръщане е инструментално средство за разработка на бизнес-процеси, грид приложения и Уеб услуги. Основната функционалност, която средата за разгръщане предоставя е:

- Внасяне на бизнес-процес, функционалността е предназначена за грид-потребителите. В средата за разгръщане, за внесен от потребителя бизнес-процес са възможни следните дейности:
    - Генериране на BPEL за бизнес-процеса;
    - Генериране на клиент за бизнес-процеса;
    - Генериране на проект за наблюдение за бизнес-процеса
- Разгръщане на бизнес-процес в среда за изпълнение, функционалността е предназначена както за грид-потребители, така и за ИТ специалисти.
- Създаване на проект – проектът е контейнер за всички Уеб услуги и прилежащите й документи, създадени от ИТ специалист. Възможни са следните дейности:
    - Създаване на Уеб услуга;
    - Публикуване на Уеб услуга в регистъра;
    - Създаване на приложение;

На Фиг. 33 са показани основни случаи на употреба в среда за разгръщане, според ролята на потребителя.



Фиг. 33. Основни случаи на употреба за среда за разгръщане

В средата за разгръщане грид-потребителя внася проектираният от него бизнес-процес и генерира потребителски интерфейс, BPEL и проект за наблюдение за процеса. Средата за разгръщане предоставя клиенти за достъп до средата за изпълнение, посредством които потребителят разгръща бизнес-процеса за изпълнение.

Администраторът конфигурира средата за разгръщане, като конфигурира достъпа до регистъра и достъпа до процесен сървър.

Средата за разгръщане предоставя на ИТ специалистите инструментариум за създаване на Уеб услуги и грид приложения. Създадените от ИТ специалистите Уеб услуги се публикуват в регистъра на услуги и могат да бъдат използвани от грид-потребителите при проектиране на бизнес-процесите им.

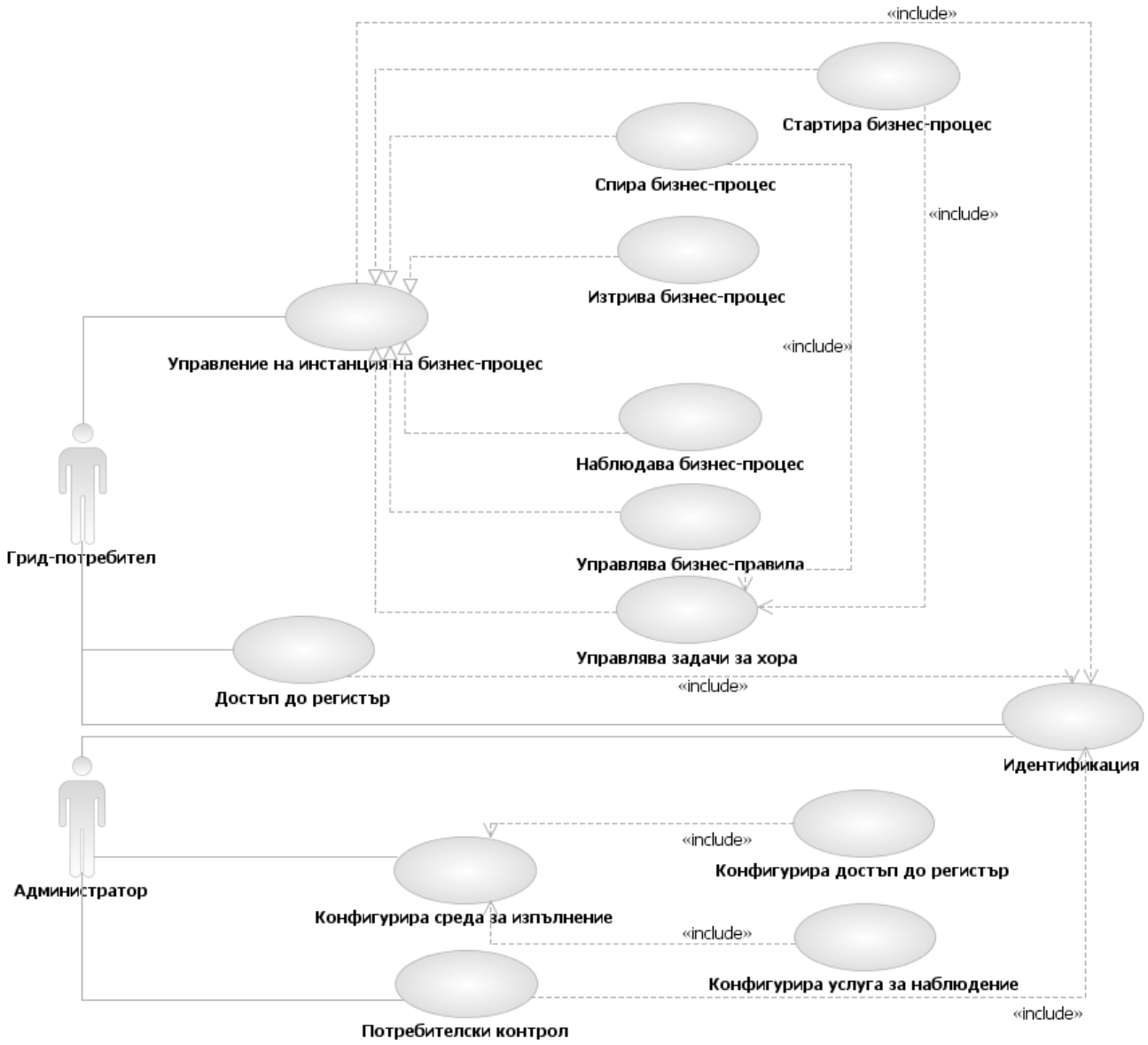
Средата за разгръщане е настолна среда, инструментално средство, което всеки грид-потребител или ИТ специалист може да инсталира локално на компютъра си. По тези причини идентификацията на потребителите се извършва от операционната система и не се изисква идентификация на потребител в средата.

3.2.4 Случаи на употреба в среда за изпълнение

Основната функционалност, която средата за изпълнение предоставя е:

- Управление на инстанция на бизнес-процес:
    - Стартиране на бизнес-процес;
    - Спиране на бизнес-процес;
    - Изтриване на инстанция на бизнес-процес;
    - Управление на бизнес-правила за бизнес-процес;
    - Наблюдение на бизнес-процес;
    - Управление на задачи за хора в бизнес-процес;
- Достъп до регистър на услуги;

На Фиг. 34 са показани основни случаи на употреба в среда за изпълнение, според ролята на потребителя.



Фиг. 34. Основни случаи на употреба за среда за изпълнение

Достъпът до средата за изпълнение се контролира от услуга, която проверява ролята и правата на потребителите. Администраторът конфигурира средата за изпълнение, като конфигурира достъпа до регистъра, услугата за наблюдение и осъществява потребителски контрол.

През потребителски интерфейс, предоставящ клиенти за достъп до среда за изпълнение, грид-потребителите могат да стартират и спират бизнес-процес. Те могат да управляват задачи за хора, да променят бизнес-правилата на процеса, по време на неговото изпълнение, както и да визуализират резултатите от метриките за наблюдение на бизнес-процеса (време за изпълнение, средно време за изпълнение и други).

3.3 Допълнителни изисквания към среда за управление на бизнес-процеси в EGI

Представеният до момента модел на среда за управление на бизнес процеси в Европейската грид инфраструктура, предполага съществуване на слой от Уеб услуги между средата за управление на бизнес-процеси и грид инфраструктурата. Както беше показано в шести раздел на първа глава, в текущата реализация на g-Lite такива Уеб услуги за достъп до EGI липсват. Това налага някои допълнителни ограничения, които не могат да бъдат пренебрегнати при използването и реализацията на среда за управление на бизнес-процеси в EGI. По-долу представяме някои от тях.

Неориентираност към услуги на EGI

Информационната система на EGI, съдържа информация за всички налични услуги в грид инфраструктурата. За съжаление, едва малка част от тях следват принципите на ориентираната към услуги архитектура. Така част от услугите в грид инфраструктурата имат WSDL описания и информация за крайната точка на услугите, но за по-голямата част от тях такава информация не е налична. Слоят от Уеб услуги за достъп до EGI и приложения е съществена част от модела на средата за управление на бизнес-процеси. На базата на тях се композират бизнес-процеси в по-горните слоеве на средата. Това ограничение налага допълнителни изисквания към средата за управление на бизнес-процеси в EGI, а именно изграждане на междинен слой, който да предоставя адаптери за достъп до грид инфраструктурата EGI чрез Уеб услуги.

Композиране на EGI услуги в бизнес-процеси

В модула Регистър на средата за управление на бизнес-процеси в EGI, могат да бъдат регистрирани всички налични услуги от информационната система на грид инфраструктурата, които имат WSDL описание. При публикуване на услуги от информационната система на EGI в регистъра, трябва да се вземат предвид следните съображения:

- Публикуваните услуги в информационната система и в регистъра, трябва да съответстват. Грид инфраструктурата е динамична. В информационната система, могат да се появят нови услуги или да бъдат премахнати такива в рамките на един ден.

- Описанията (WSDL файловете) на услугите не се съхраняват в информационната система. Информационната система, може да даде информация за крайните точки на услугите, но не и информация за описанието на тези услуги. Информацията за Уеб услуги с WSDL описание в EGI съдържа единствено данни за път до крайните им точки.

- Достъпът до информационната система на EGI изисква регистрация на потребителски интерфейс. Достъпът до информационната система се осъществява чрез клиент, който е локално инсталиран на потребителския интерфейс в g-Lite. За да бъде използван клиента не се изисква грид сертификат, въпреки това клиентите за достъп са инсталирани на елементи от грид инфраструктурата.

Тези съображения предполагат реализирането на допълнителна функционалност към модула Регистър, която да извършва публикуване на услугите от информационната система в регистъра, така че всяка промяна в информационната система да бъде своевременно отразена. Това включва:

- функционалност за регистриране на всички налични WSDL файлове на услуги от EGI в регистъра;
- функционалност за ежедневно обновяване на информацията в регистъра с актуалните (налични) услуги от EGI. Модулът Регистър поддържа стандарта UDDI, така в регистъра се пазят всички версии на публикуваните услуги.

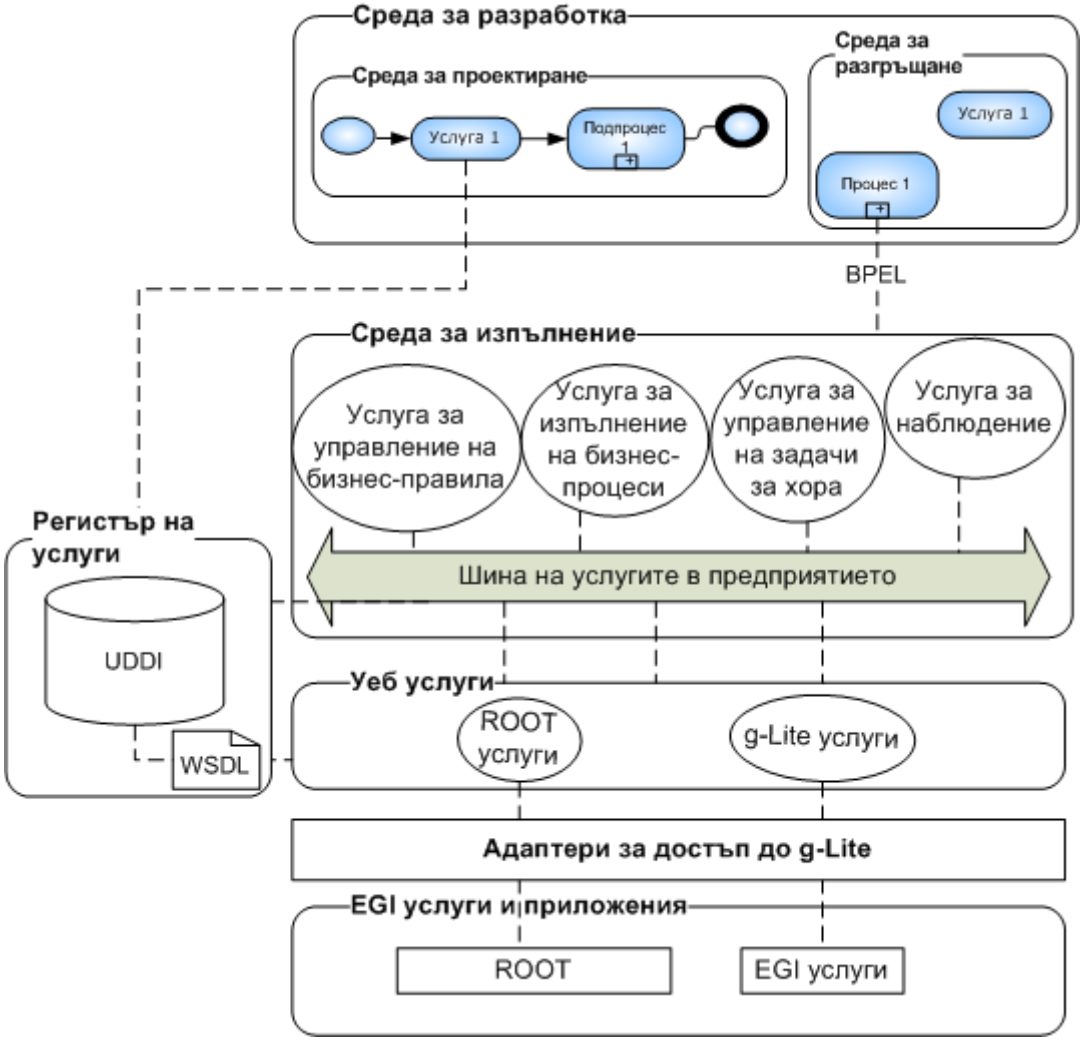
Наличието на WSDL описание и крайна точка за достъп до наличните услуги в EGI не е достатъчно условие те да бъдат лесно композирани в бизнес-процеси от грид-потребителя. Допълнително изисквания към Уеб услугите в средата за управление на бизнес-процеси е те да могат да участват в композиция.

EGI не позволява надстройка с шина на услугите

Шината на услугите в предприятието е ядрото на средата за изпълнение. Всички услуги от средата за изпълнение, комуникират през шината. Концептуално, изчислителният елемент, елементът за съхранение на данни трябва да са услуги закачени към шината. Текущата реализация на грид средата g-Lite, обаче не позволява надстройка с шина на услугите в предприятието. Споменатите до тук допълнителни изисквания са взети предвид при проектирането на архитектура на среда за управление на бизнес-процеси в EGI, която представяме по-долу.

3.4 Проектиране на архитектура на среда за управление на бизнес-процеси в EGI

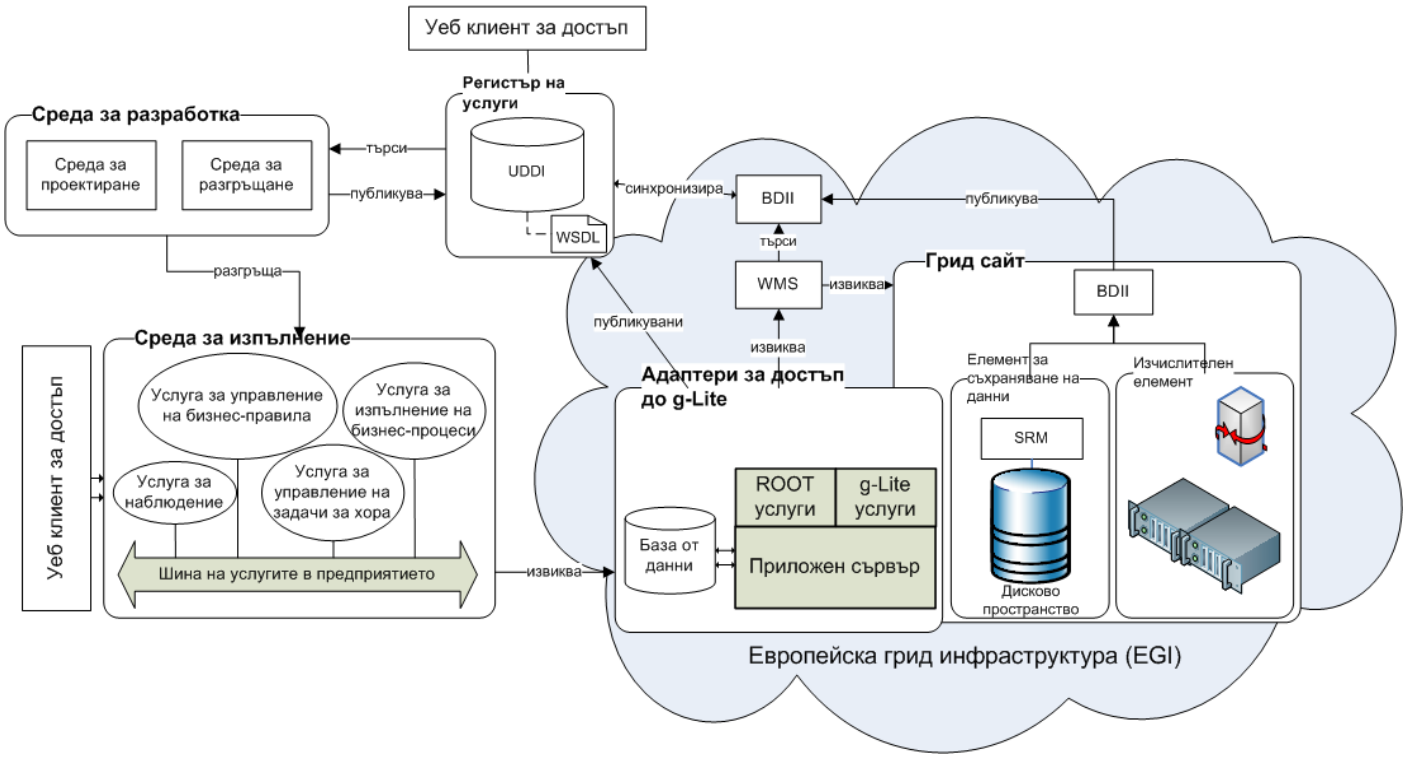
На Фиг. 35 представяме архитектурно решение на среда за управление на бизнес-процеси в EGI с оглед на основните модули и компоненти, която отразява описаните до момента изисквания.



Фиг. 35. Архитектура на среда за управление на бизнес-процеси в EGI

За да бъде моделът на среда за управление на бизнес-процеси приложим за g-Lite, средата за изпълнение трябва да бъде изнесен извън грид инфраструктурата. Също така трябва да бъде добавен нов допълнителен модул - Адаптери за достъп до g-Lite, който да отрази допълнителните изисквания.

На Фиг. 36 представяме модули и компоненти на среда за управление на бизнес-процеси в EGI адаптирани за g-Lite.



Фиг. 36. Модули и компоненти на среда за управление на бизнес-процес в EGI адаптирани за g-Lite

Новият модул Адаптери за достъп до g-Lite, предоставя адаптери за достъп до грид ресурсите в EGI като Уеб услуги, които да бъдат композирани в бизнес-процеси от грид-потребителя. С цел да бъде демонстрирана концепцията на среда за управление на бизнес-процеси в EGI, модулът предоставя както Уеб услугите за достъп до g-Lite, така и Уеб услуги за достъп до софтуер за визуализация и анализ на данни - ROOT.

Уеб услугите за достъп до ROOT са независими от EGI инфраструктурата и като такива, могат да бъдат изнесени извън грид. При Уеб услугите за достъп до g-Lite, обаче това не е така. При тях е от значение кой е потребителят, който ще използва услугите и към коя виртуална организация принадлежи.

Това налага модулът Адаптери за достъп до g-Lite да бъде част от грид инфраструктурата. Примерен ресурс от EGI, където може да бъде реализиран модулът е компонента на грид инфраструктурата потребителски интерфейс (glite-UI). Достъпът на потребителите до g-Lite Уеб услугите от модула се регламентира в база от данни, която съхранява потребителско име, сертификат за потребителя и виртуална организация, към която потребителят принадлежи. Така правилата за сигурност при достъпа до грид ресурси, остават не нарушени.

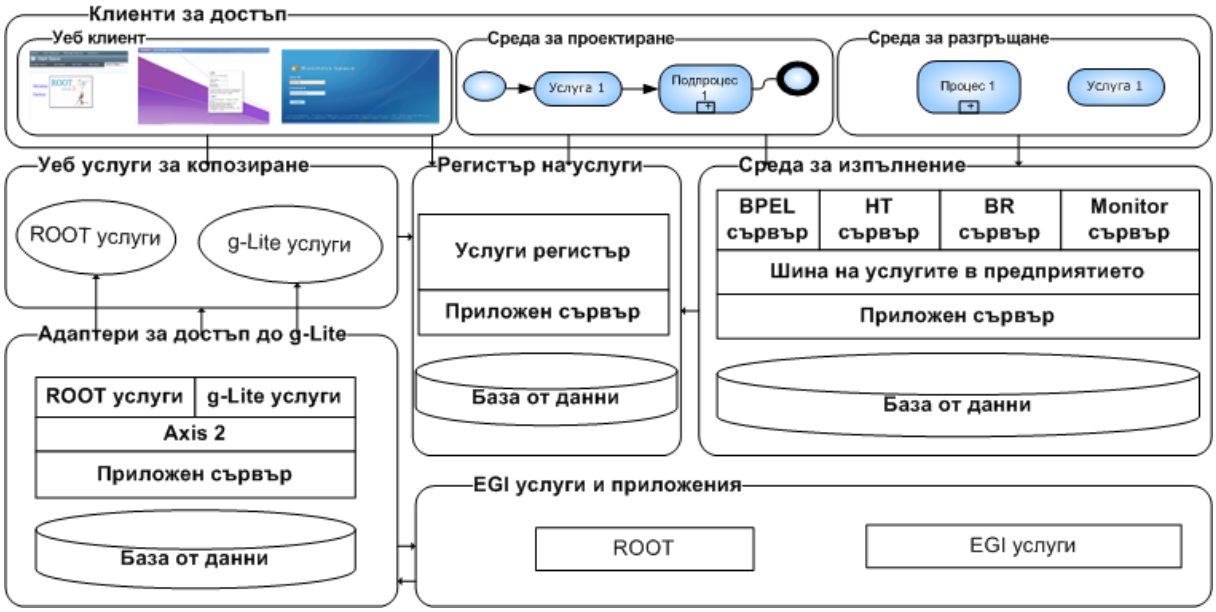
Уеб услугите за достъп до g-Lite, предоставят функционалност за изпълнение на задачи в g-Lite, за проследяване състояние на задачи, за създаване на прокси сертификат и други. Уеб услугите за достъп до ROOT, изнасят част от функционалността за изчертаване на хистограми и функции в софтуера ROOT, и предоставят нова функционалност за генериране на програмен код.

Модулът Адаптери за достъп до g-Lite предоставя необходимите Уеб услуги за композиране в бизнес-процеси. Те са публикувани в модул Регистър на средата и така могат да бъдат използвани от грид-потребителя при моделирането на бизнес-процеси в средата за проектиране.

Грид-потребителите могат да търсят услуги в регистъра посредством Уеб базирани клиенти за достъп до регистъра. Специалистите по информационни технологии през Уеб базираният клиент могат да публикуват нови услуги и свързаните с тях документи.

Аналогично в средата за изпълнение, посредством Уеб базирани клиенти за достъп, грид-потребителите могат да стартират и спират бизнес-процесите си, могат да променят бизнес-правилата на процеса, по време на неговото изпълнение, както и да визуализират резултатите от метрики за наблюдение на бизнес-процеса (време за изпълнение, средно време за изпълнение и други).

Примерна детайлизация на среда за управление на бизнес-процеси в EGI е представена на Фиг. 37.



Фиг. 37. Примерна среда за управление на бизнес-процеси в EGI

На базата на представения модел на среда за управление на бизнес-процеси в EGI, в Глава 4 са предложени възможности за използване и приложение на среда за управление на бизнес-процеси в EGI. За целите на изследването е използван софтуер на средата IBM SOA Foundations.

3.5 Изводи

Предложеният модел на среда за управление на бизнес-процеси в EGI, покрива пълният жизнен цикъл на ориентираната към услуги архитектура. Предимствата на модела пред разгледаните в Глава 2 среди са:

- Поддръжка на Уеб услуги (Модули "Регистър" и "Среда за разработка")
- Поддръжка на стандарти за Уеб услуги:
  - Поддръжка на BPEL (Модули "Среда за разработка" и "Среда за изпълнение")
  - Поддръжка на UDDI регистри (Модули "Регистър", "Среда за разработка" и "Среда за изпълнение")
- Поддръжка на адаптери (Модул "Среда за разработка")
- Поддръжка на шина на услугите (Модул "Среда за изпълнение")
- Поддръжка на задачи за хора в бизнес-процеса (Модули "Среда за разработка" и "Среда за изпълнение")
- Поддръжка на наблюдение на бизнес-процеса (Модули "Среда за разработка" и "Среда за изпълнение")
- Поддръжка на управление на бизнес-правилата за процеса (Модули "Среда за разработка" и "Среда за изпълнение")

3.6 Приноси

Приносите на тази глава могат да бъдат обобщени, както следва:

- Предложен е модел на среда за управление на бизнес-процеси в EGI;
- Разработено е архитектурно решение на среда за управление на бизнес-процеси в EGI;
- Проучени са съществуващите услуги в грид инфраструктурата EGI. Установена е необходимост от създаване на адаптери за достъп до услугите на грид инфраструктурата.

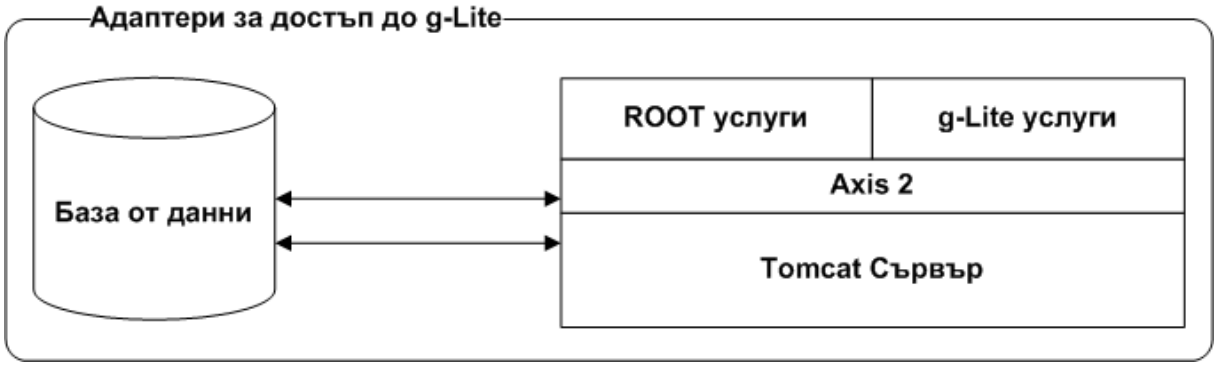
Глава 4. Реализация и приложение на среда за управление на бизнес-процеси в EGI

В тази глава на дисертацията ще покажем възможности за използване и приложения на средата за управление на бизнес-процеси в EGI, както и някои аспекти от реализацията й.

4.1 Модул Адаптери за достъп до g-Lite

Модулът Адаптери за достъп до g-Lite предоставя Уеб услуги за достъп до грид ресурсите в EGI. Уеб услугите покриват изискванията отбелязани в трети раздел на трета глава, а именно да могат да бъдат композирани в бизнес-процеси от грид-потребителя. С цел да бъде демонстрирана концепцията на среда за управление на бизнес-процеси в EGI, модулът предоставя както Уеб услугите за достъп до g-Lite, така и Уеб услуги за достъп до софтуера ROOT.

На Фиг. 38 е представена архитектурата на модула. Модулът е реализиран върху Tomcat 7 приложен сървър с Axis 2 контейнер за Уеб услуги. Модулът е инсталиран на операционна система Scientific Linux CERN 4 (SLC4) на компютър, който е част от грид инфраструктурата EGI.



Фиг. 38. Архитектура на модул Адаптери за достъп до g-Lite

4.1.1 Използвани технологии и инструменти

За реализацията на Уеб услугите от модула, бяха използвани следните инструменти и технологии:

- **MySQL 4.1.22 [84] -** Безплатна релационна система за управление на база от данни с отворен код, която предоставя възможност за съхранение и управление на данни;

- **Java 1.5 (JDK-1.5.0_22) [82] –** Обектно ориентиран език за програмиране и безплатен набор от технологии и средства за разработване на приложения, независими от операционната система;
- **Tomcat 7.0.8 [67] -** безплатен Уеб контейнер с отворен код, на който могат да бъдат внедрявани и стартирани Уеб приложения разработени на Java. За разлика от стандартните Уеб сървъри, които при HTTP заявка от клиента връщат статични файлове, сървърът Tomcat може да връща и динамични документи, създадени в резултат от изпълнението на Java код. Сървърът е разработен на Java, което го прави платформено независим;
- **Axis 2 / Java [66]–** безплатно приложение разработено на Java, което предоставя функционалност за създаване на Уеб услуги, включително и изпращане на SOAP съобщение от Уеб услуга, получаване на SOAP съобщение от Уеб услуга, извличане на WSDL описание на Уеб услуга. Като приложение Axis 2, може да съществува единствено на приложен сървър;
- **Eclipse 3.6 (Helios) [68] –** среда за разработка на Java-базирани приложения и добавки (plug-in) за Eclipse. Предоставя функционалност за интегриране с множество от приложни сървъри, включително и Tomcat, както и интеграция с Axis 2/ Java.

4.1.2 Услуги за достъп към ROOT

Средата ROOT е софтуер за визуализация и анализ на данни. В основата си, ROOT е обектно-ориентирана среда, базирана на C++, която предоставя специализирани пакети и готови класове. Софтуерът е платформено независим и поддържа операционните системи Linux, OpenSolaris, Mac OS X и Windows (Vista, XP, NT, 2000).

За да анализират данни, потребителите на средата ROOT, трябва да са запознати с концепциите на обектното-ориентираното програмиране и да имат познания по езика C++. Обикновено, потребителя създават C++ файлове, наречени макроси. В тези файлове, потребителя използва готовите C++ класове, дефинирани в средата ROOT и на базата на тях създава обекти и използва методи за работа с тези обекти.

Създадените от потребителя C++ файлове се интерпретират от средата ROOT. Като C++ интерпретатор се използва - CINT. Интерпретаторът е вграден в ROOT. Средата не предоставя текстов редактор, в който да бъдат създавани макроси.

Потребителят използва стандартните текстови редактори (например notepad) за изготвяне на C++ файла.

Средата ROOT се използва предимно от грид-потребители физици. Софтуерът ROOT се използва за представяне на стандартен входен формат на данните на задачи, които в последствие се изпълняват в грид инфраструктурата EGI, както и за представяне на стандартен изходен формат на данните, които са резултат от обработката на данни в EGI.

За целите на дисертацията са реализирани Уеб услуги, които изнасят част от функционалността на ROOT средата. Услугите предоставят следната функционалност:

- RootHistogram1D – услуга предоставяща функционалност за изчертаване на едномерни хистограми;
- RootFunction1D – услуга предоставяща функционалност за изчертаване на функции;
- RootDownloads – услуга предоставяща функционалност за записване на изчертаните от приложението функции и хистограми в ROOT и C++ файлове. Тази функционалност позволява, извършване на последващи допълнителни анализи на данните от съхранените файлове в средата ROOT.

В (Таблица 8) са представени услугите и някои от по-важните операции, които предоставят.

RootHistogram1D	
Операция	**Описание**
drawHistogramRandom	Изчертава хистограма по зададено произволно число. В резултат се връща URL на картинката.
drawHistogramFromFunction	Изчертава хистограма по зададена функция. В резултат се връща URL на картинката.
drawHistogramFromDataFile	Изчертава хистограма по зададени от потребителя файл с данни. В резултат се връща URL на картинката.
drawFittedHistogram	Приближава хистограма до функция. В резултат се връща URL на картинката.

RootFunction1D	
Операция	**Описание**
drawFunction	Изчертава функция. В резултат се връща URL на картинката.
drawSin	Изчертава функцията sin(x). В резултат се връща URL на картинката.
drawCos	Изчертава функцията cos(x). В резултат се връща URL на картинката.
RootDownloads	
Операция	**Описание**
getRootFileURL	По зададен URL на картинка, връща URL на ROOT файла
getCFileURL	По зададен URL на картинка, връща URL на C++ файла
downloadCFileByURL	По зададен URL на C++ файла, връща файлът в текстов вид

Таблица 8. ROOT услуги

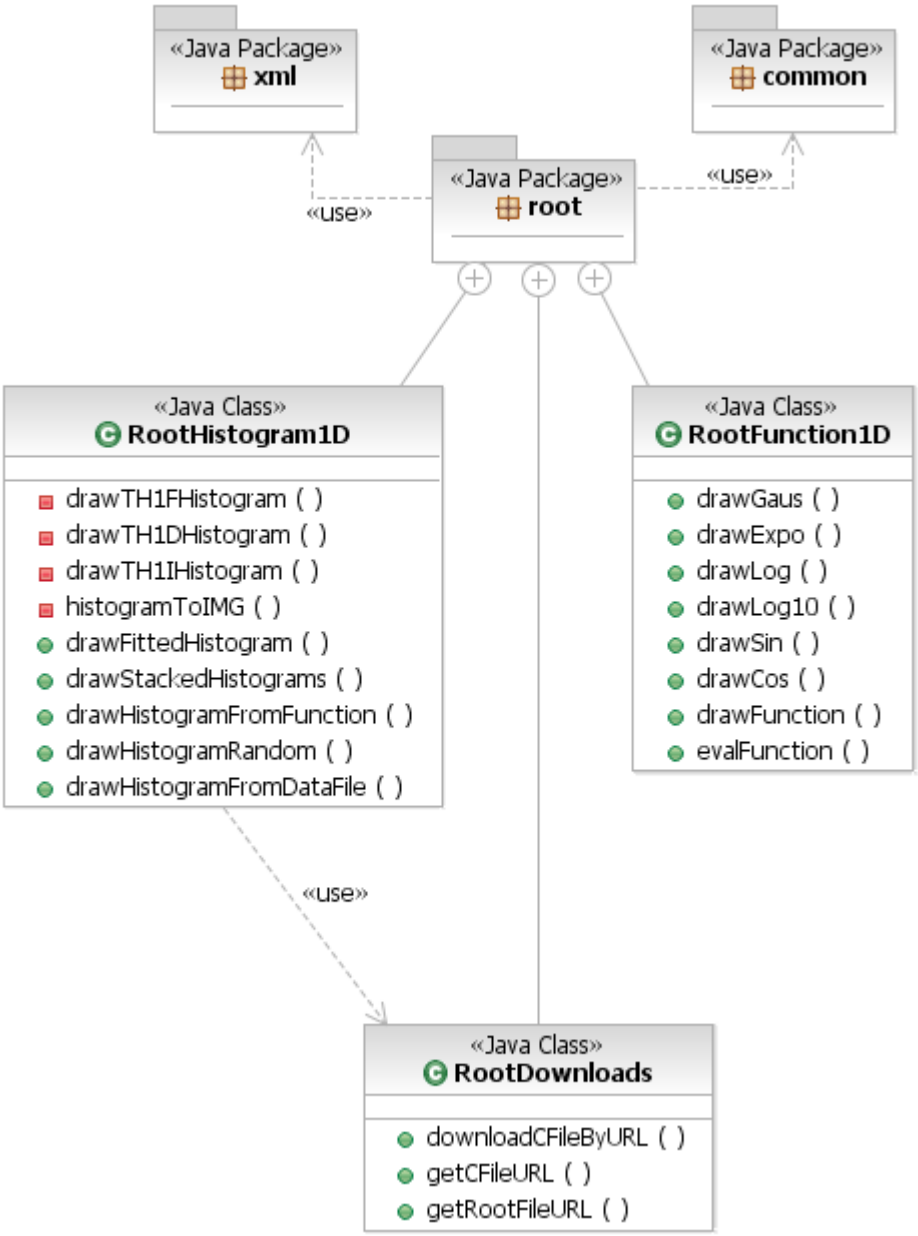
В (Таблица 9) са показани адресите за достъп до WSDL описанията на услугите.

ROOT услуги	**WSDL описание**
RootDownloads	http://myproxy.grid.uni-sofia.bg:8080/heph/services/RootDownloads?wsdl
RootHistogram1D	http://myproxy.grid.uni-sofia.bg:8080/heph/services/RootHistogram1D?wsdl
RootFunction1D	http://myproxy.grid.uni-sofia.bg:8080/heph/services/RootFunction1D?wsdl

Таблица 9. Адреси на достъп до WSDL описанията на ROOT услуги

Услугите са внедрени на Tomcat 7 приложен сървър с Axis2 контейнер. За реализиране на услугите са използвани среда за разработка Eclipse и език за програмиране Java. За достъп до услугите е реализиран Уеб потребителски интерфейс, достъпен на адрес: http://myproxy.grid.uni-sofia.bg:8080/na62/

На Фиг. 39 е показана диаграма на Java пакети и класове, които реализират ROOT услугите.



Фиг. 39. Диаграма на Java класовете на ROOT услуги

4.1.3 Услуги за достъп към g-Lite

Уеб услугите за достъп до ROOT са независими от EGI инфраструктурата и като такива, могат да бъдат изнесени извън грид. При Уеб услугите за достъп до g-Lite, обаче това не е така. При тях е от значение кой е потребителят, който ще използва услугите и към коя виртуална организация принадлежи. Това налага модулът Адаптери за достъп до g-Lite да бъде част от грид инфраструктурата.

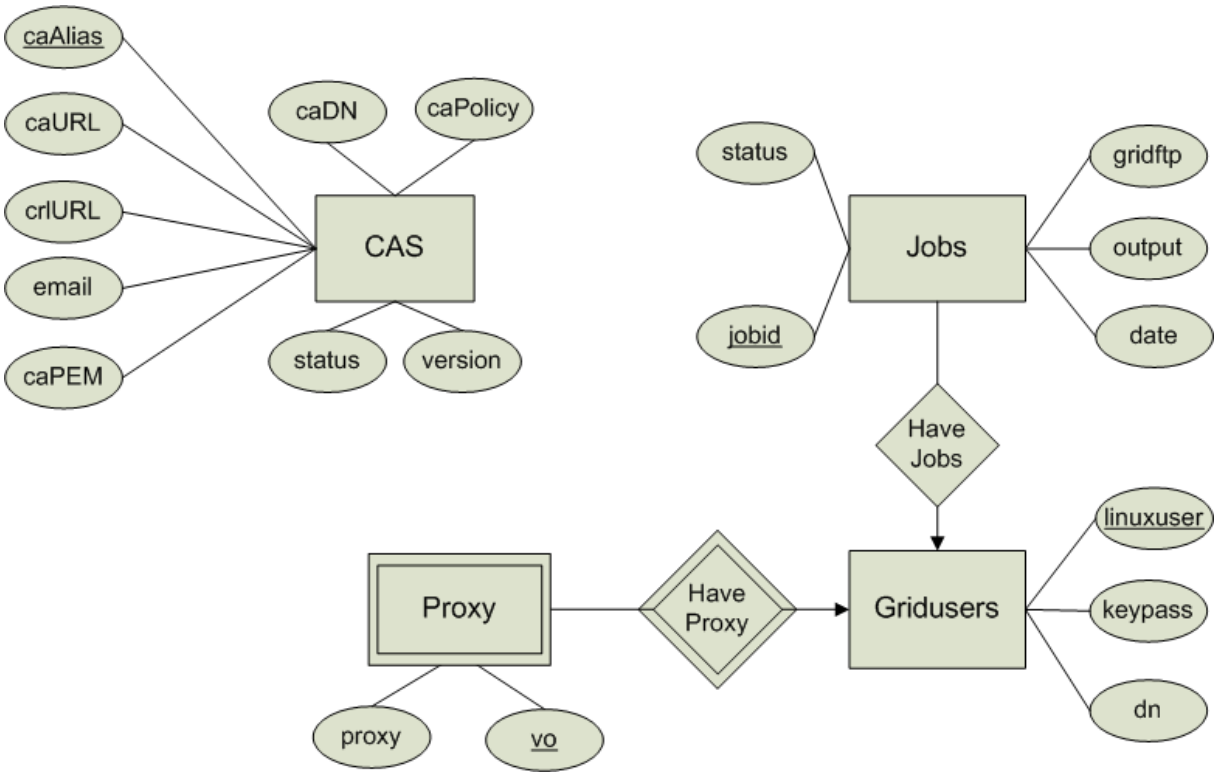
Достъпът на потребителите до g-Lite Уеб услугите от модула се регламентира в база от данни. Тя съхранява потребителско име, сертификат за потребителя и виртуална организация, към която потребителят принадлежи. Базата от данни се отнася за

текущият модул и обслужва текущо създадените Уеб услуги от модула, по тези причини представеният по-долу модел на данни е значително опростен.

Състои се от четири основни множества от същности:

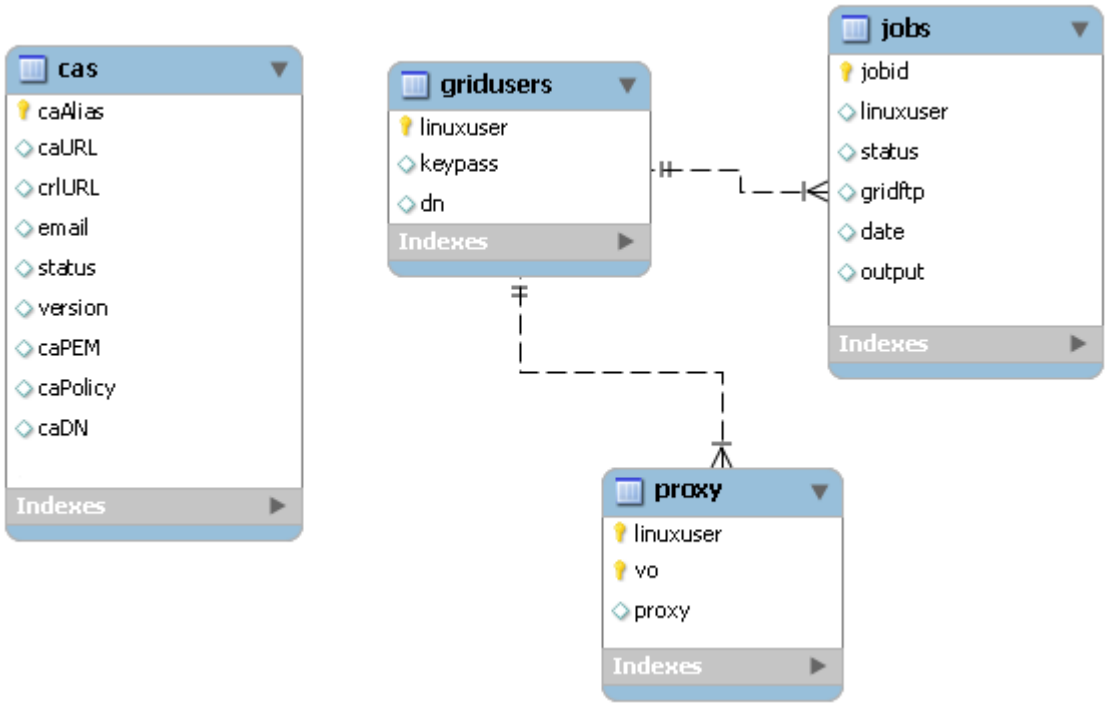
- Сертифицираща организация (CAS) – множество от същност, което съдържа информация за сертифициращите организации признати в EGI. Ключ за множеството е името на сертифициращата организация (caAlias), което е уникално в рамките на грид инфраструктурата;
- Грид-потребители (Gridusers) - множество от същности, което съдържа информация за грид-потребителя;
- Прокси сертификати на потребители (Proxy) – множество от същности, което съдържа информация за прокси файловете създадени от потребителя;
- Грид задачи на потребители (Jobs) – множество от същности, което съдържа информация за задачите на потребителя.

На Фиг. 40 е показан моделът "Същност-връзки" на проектираната база от данни.



Фиг. 40. Модел "Същност-връзки" на данните

На Фиг. 41 е показан релационният модел на проектираната база от данни. Подробно описание на таблиците от релационния модел е поместено в Приложение 1.



Фиг. 41. Релационен модел на данните

За целите на дисертацията са реализирани следните Уеб услугите за достъп до g-Lite:

- ProxyService – услуга, която предоставя функционалност за създаване и съхраняване в базата от данни на прокси сертификат;
- WebFTP- услуга, която предоставя функционалност за записване на файл на приложния сървър;
- UserService – услуга, която предоставя функционалност за промяна на потребителския DN (Distinguished Name) в базата от данни;
- JobStatus – услуга, която предоставя функционалност за проверка на състоянието на задача и за изпращане на съобщение до e-mail на потребителя при успешно приключена задача;
- JobManager – услуга, която предоставя функционалност за изпращане на задача в грид.

Достъп до g-Lite Уеб услугите, могат да имат единствено грид-потребители на грид инфраструктурата EGI. Необходимо е грид-потребителя да има регистрация (потребителска сметка) на операционната система SLC4, на която е инсталиран приложния сървър, регистрация в базата от данни, както и инсталиран на операционната система грид сертификат.

В (Таблица 10) са изброени три от g-Lite услугите и някои от по-важните операции, които те предоставят.

ProxyService	
Операция	**Описание**
proxyCreate	Създава прокси файл по зададени име на потребител, име на виртуална организация и парола.
proxyCreateUpload	Създава прокси файл по зададени име на потребител, име на виртуална организация и парола и го записва в базата от данни
proxyView	Връща прокси файла от базата от данни по зададени име на потребител, име на виртуална организация и парола
proxyValidate	Проверява валидността на прокси файла
JobManager	
Операция	**Описание**
gliteJDLGenerate	Генерира тестов JDL файл по зададено име на потребител
gliteJDLExecGenerate	Генерира тестов JDL файл, по зададено име на потребител и път до изпълним файл (без възможност за указване на входни аргументи към изпълнимия файл)
gliteJDLInputGenerate	Генерира тестов JDL файл, по зададено име на потребител и път до изпълним файл (с възможност за указване на входни аргументи към изпълнимия файл)
gliteJDLInput2Generate	Генерира тестов JDL файл, по зададено име на потребител, път до изпълним файл и имена на файлове обработвани в изпълнимия файл (с възможност за указване на входни аргументи към изпълнимия файл). Файловете трябва да съществуват на приложния сървър.
gliteJDLVerify	Проверява валидността на JDL файла
gliteJobList	По зададени име на потребител, име на виртуална организация, парола и състояние връща списък със задачите на потребителя в съответното състояние
gliteJobListMatch	По зададени JDL файла, име на потребител, име на виртуална организация и парола връща списък с изчислителни ресурси
gliteJobSubmit	По зададени JDL файла, име на потребител, име на виртуална организация и парола изпраща задача за изпълнение в грид. Връща уникален идентификатор на задача (JOBID)

gliteJobGetOutput	По зададени JOBID, име на потребител, име на виртуална организация и парола връща адрес на директория от приложния сървър, в която се намират изходните файлове на задачата
JobStatus	
Операция	**Описание**
gliteJobOutputEmailer	По зададени JOBID, име на потребител, име на виртуална организация, парола и e-mail връща състоянието на задачата и изпраща e-mail при състояние приключена

Таблица 10. g-Lite услуги

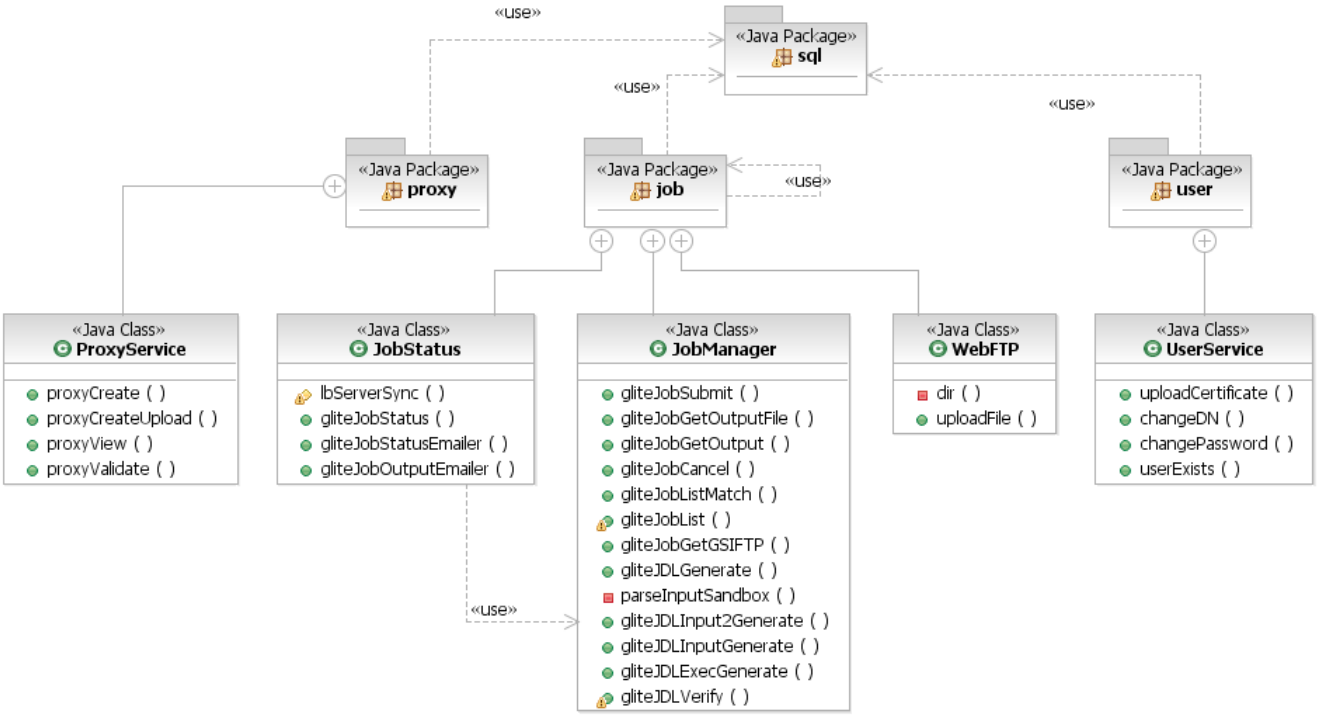
В (Таблица 11) са показани адресите за достъп до WSDL описанията на услугите.

g-Lite услуги	**WSDL описание**
ProxyService	http://myproxy.grid.uni-sofia.bg:8080/heph/services/ProxyService?wsdl
WebFTP	http://myproxy.grid.uni-sofia.bg:8080/heph/services/WebFTP?wsdl
UserService	http://myproxy.grid.uni-sofia.bg:8080/heph/services/UserService?wsdl
JobStatus	http://myproxy.grid.uni-sofia.bg:8080/heph/services/JobStatus?wsdl
JobManager	http://myproxy.grid.uni-sofia.bg:8080/heph/services/JobManager?wsdl

Таблица 11. Адреси за достъп до WSDL описанията на g-Lite услуги

При проектирането на Уеб услугите е следван общ принцип: входните и изходни параметри на операциите да са прости типове (символен низ, число). Това позволява Уеб услугите да бъдат лесно композирани в бизнес-процес от грид-потребителите, без да са необходими познания по програмиране.

Услугите са внедрени на Tomcat 7 приложен сървър с Axis2 контейнер. За реализиране на услугите са използвани среда за разработка Eclipse и език за програмиране Java. При реализацията на g-Lite Уеб услугите са използвани съществуващи приложно-програмни интерфейси на Java и клиенти за достъп до g-Lite средата. На Фиг. 42 са показани диаграми на Java пакети и класове, които реализират g-Lite услугите.



Фиг. 42. Диаграма на Java пакети и класове на g-Lite услуги

Уеб услугите са публикувани в модул Регистър на средата за управление на бизнес-процеси в EGI.

4.2 Модул Регистър на услуги

Като регистър на услуги в среда за управление на бизнес-процеси в EGI е използван софтуерът IBM WebSphere Service Registry and Repository (WSRR).

Софтуерът IBM WebSphere Service Registry and Repository е част от IBM SOA Foundations. IBM SOA Foundations е среда за управление на бизнес-процеси, която реализира пълният жизнен цикъл на ориентираната към услуга архитектура. Средата IBM SOA Foundations предоставя всички необходими инструментални средства, за проектиране, изпълнение и наблюдение на бизнес-процеси, което я прави подходяща за целите на изследването. Пълните версии на инструменталните средства от средата са достъпни безплатно за изследователски цели, по академична програма на IBM.

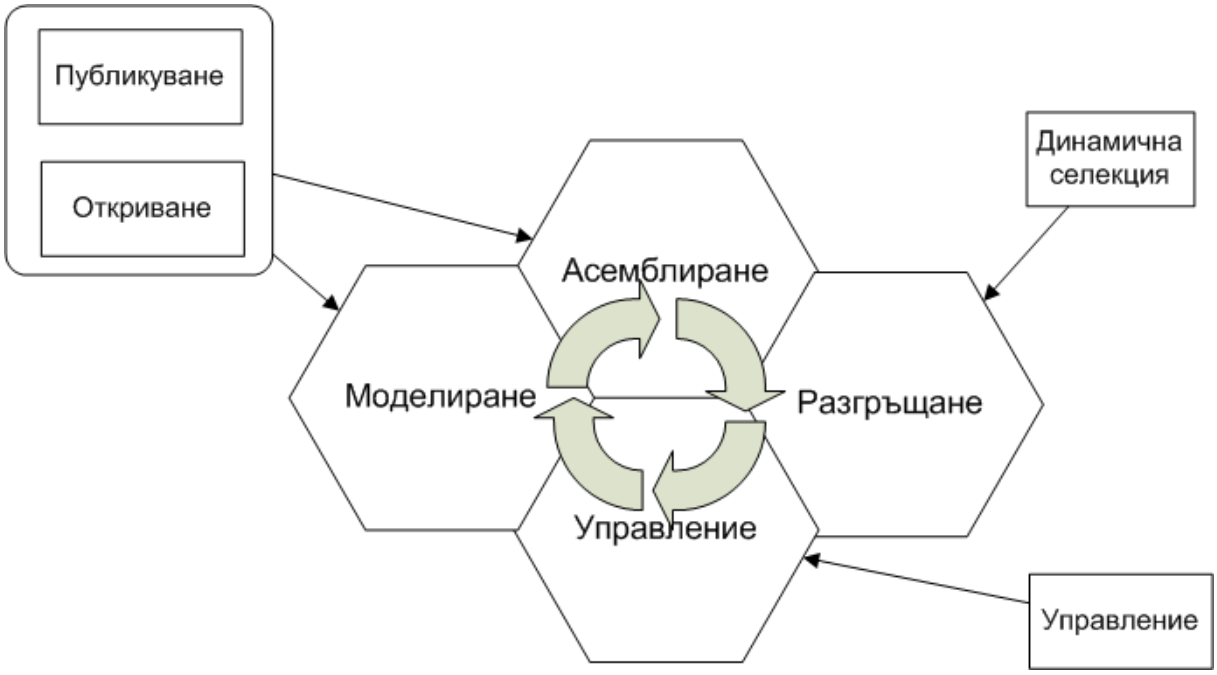
Софтуерът WSRR е едновременно и регистър и хранилище (repository) на данни. Като хранилище, WSRR съхранява артефакти свързани с Уеб услугите. Примерни артефакти са WSDL описания, XSD схеми, XML документи, BPEL описания на бизнес-процеси. Като регистър, WSRR съхранява мета данни за артефактите, т.е. информация, която описва самите артефакти. Регистърът поддържа стандарта UDDI. Реализирането

на регистър и хранилище в един компонент, позволява поддържането на консистентни данни за артефактите и техните мета данни.

Софтуерът WSRR, предоставя следната функционалност:

- Публикуване (Publish) на услуги;
- Откриване (Find) на съществуващи услуги;
- Динамична селекция (Enrich) на услуги в средата за изпълнение;
- Управление (Manage) на услуги.

На Фиг. 43 [37] е показана връзката между предоставената от софтуера WSRR функционалност и жизненият цикъл на ориентираната към услуги архитектура.



Фиг. 43. Връзка между WSRR и жизнения цикъл на ориентираната към услуги архитектура

Механизмът за публикуване и откриване на услуги, позволява създаването и модифицирането на бизнес-процеси от вече съществуващи услуги. Динамичната селекция на услуги във фаза разгръщане, позволява средата за изпълнение да се обръща към регистъра по време на изпълнени на бизнес-процеса и така да реагира на настъпили промени в услугите от процеса. Функционалността за управление на услуги във фаза управление, контролира връзките между отделните услуги и събира информация за наблюдение на услугите.

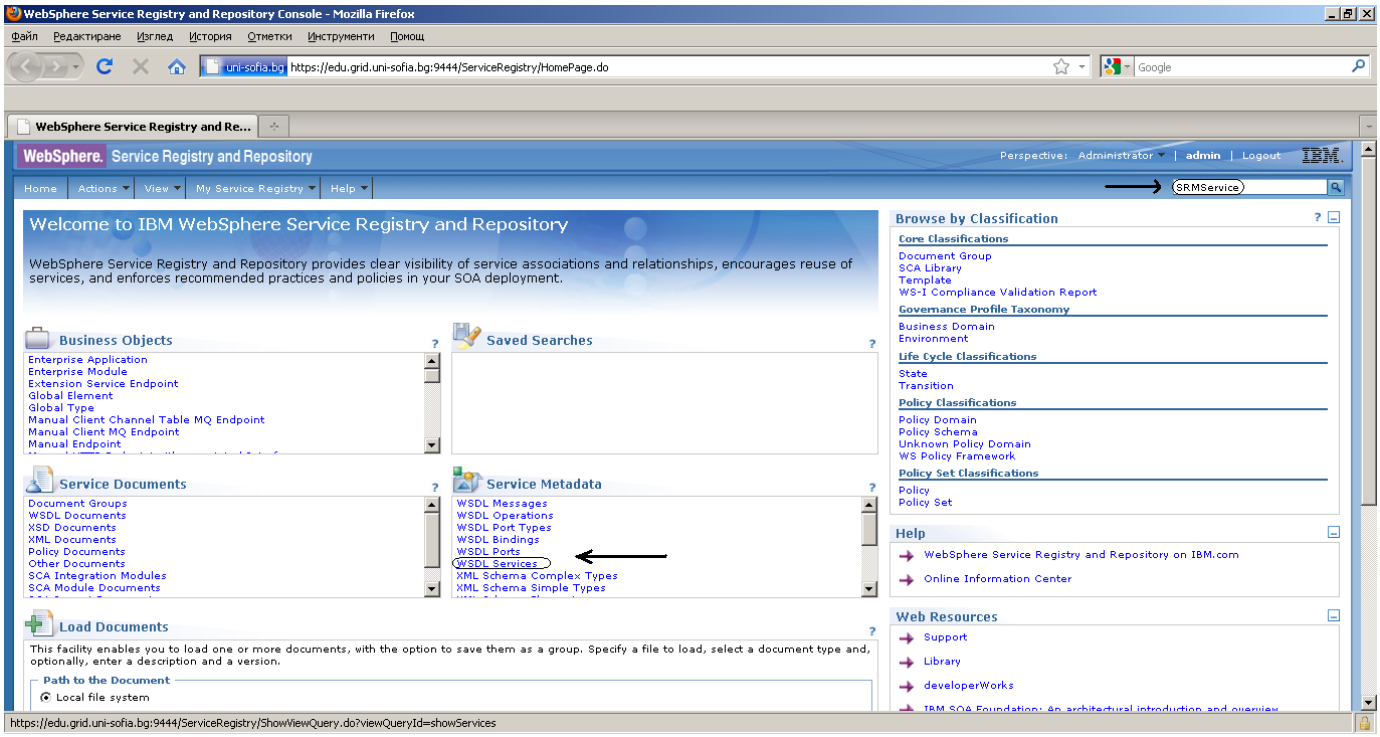
Достъпът до регистъра WSRR може да бъде осъществен:

- През Уеб потребителски интерфейс;
- Посредством инструменталните средства на средата IBM SOA Foundations. Като част от IBM SOA Foundations, WSRR е интегриран с всички инструментални средства на средата.
- Посредством добавка за Eclipse;
- Посредством приложно-програмни интерфейси за извикване на регистъра като Уеб услуга.

Първите два метода на достъп до регистъра, са подходящи за грид-потребители. Последните два метода на достъп до регистъра, са насочени към специалистите по информационни технологии.

Грид-потребителите на EGI, могат да достъпят регистъра посредством Уеб потребителският интерфейс, както и посредством другите инструментални средства на средата IBM SOA Foundations. Осъществяване на достъп до регистъра през инструменталните средства на средата IBM SOA Foundations е разгледан в последващите глави.

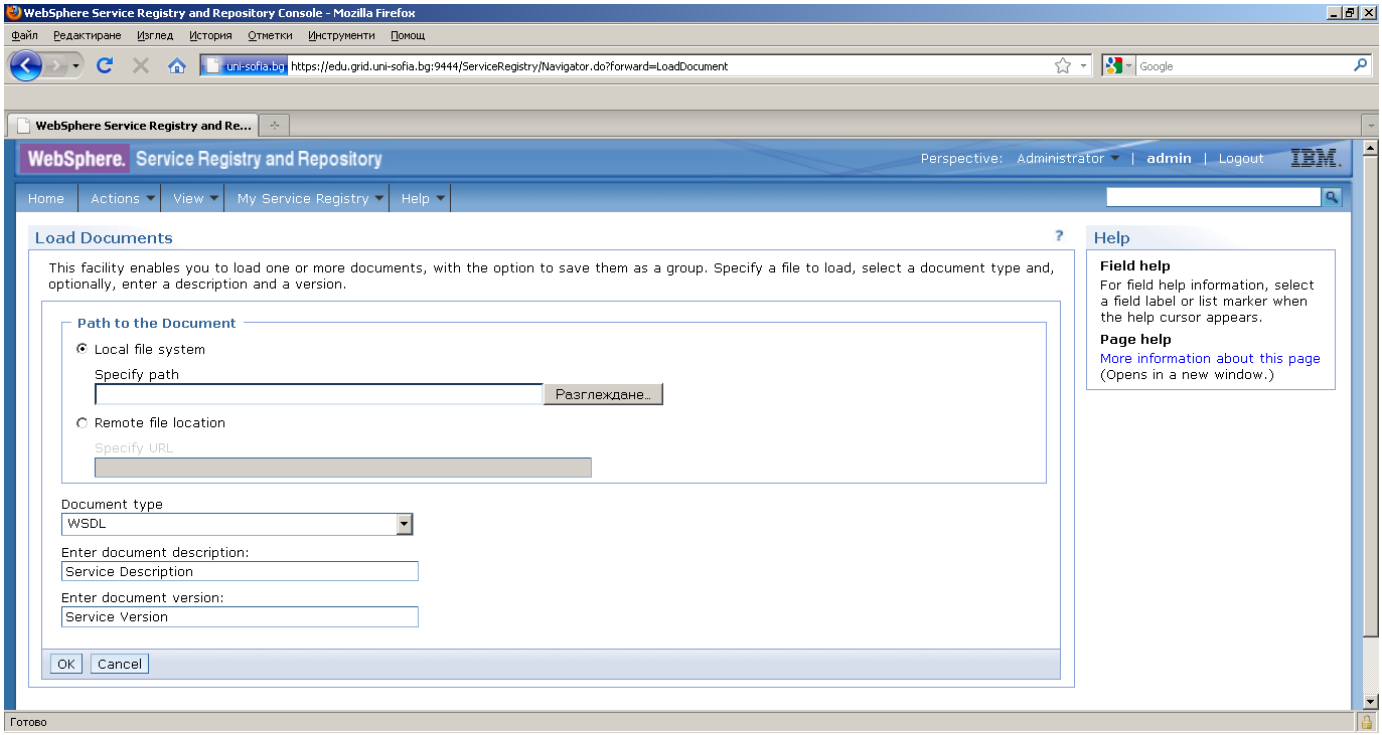
Посредством Уеб интерфейса, грид-потребителите могат да търсят услуга по зададен критерии и да разглеждат публикуваните в регистъра услуги. На Фиг. 44 е показан екран от Уеб потребителския интерфейс на регистъра.



Фиг. 44. Уеб потребителски интерфейс за достъп до WSRR

В Уеб потребителския интерфейс грид-потребителят може да търси за налични услуги през полето за търсене, като въведе име на услугата или като изведе всички налични услуги в регистъра. За публикуваните услугите в регистъра, грид-потребителят може да разгледа в графичен вид предоставяните от услугата операции.

Специалистите по информационни технологии, също могат да използват Уеб потребителския интерфейс за откриване и публикуване на услуга в регистъра. На Фиг. 45 е показан екран за публикуване на услуга през Уеб потребителския интерфейс на WSRR. Чрез указване на пътя до WSDL файла (от локалната файлова система или на отдалечен сървър), ИТ специалистът може да публикува Уеб услуга.



Фиг. 45. Публикуване на услуга през Уеб потребителския интерфейс

Специалистите по информационни технологии, които разработват грид услуги за EGI, могат да достъпят регистъра и през средата за разработка – Eclipse. Софтуерът WSRR предоставя добавка за Eclipse, чрез която разработчиците могат да публикуват и откриват ресурси в регистъра. Примерни екрани на използване на добавката за Eclipse са показани в Приложение 2.

В приложенията, които разработват специалистите по информационни технологии могат да извикат регистъра и като Уеб услуга. Именно този подход е използван при разработването на функционалност за публикуване на услугите от информационната система BDII на g-Lite в регистъра. Подходът е описан в следващия раздел.

С цел да бъде демонстрирана концепцията на среда за управление на бизнес-процеси в EGI, софтуерът IBM WebSphere Service Registry and Repository 7.0 е инсталиран на операционна система Scientific Linux CERN 4 (SLC4) на компютър, който не е част от грид инфраструктурата EGI.

Създадени са два профила на регистър:

- WSRRSrv01 – е профил на регистър, за който е реализирана сигурност. За достъпа до регистъра през Уеб потребителския интерфейс се изисква потребителско има и парола. От потребителите не се изисква да притежават грид сертификат. В този профил са публикувани Уеб услугите от модул Адаптери за достъп до g-Lite;
- WSRRSrv02 – е профил на регистър, за който не е реализирана сигурност. В този профил са публикувани всички налични Уеб услуги от информационната система на EGI. За достъпа до регистъра през Уеб потребителския интерфейс не се изисква потребителско има и парола, нито грид сертификат. За да бъдат извикани EGI услугите обаче, грид сертификатът е задължителен.

И двата профила предоставят сигурен HTTPS достъп до регистъра.

В (Таблица 12) са показани адресите за достъп до двата профила на регистъра WSRR.

WSRRSrv01 профили	
Уеб достъп (HTTP)	http://edu.grid.uni-sofia.bg:9080/ServiceRegistry
Уеб достъп (HTTPS)	https://edu.grid.uni-sofia.bg:9443/ServiceRegistry
Реализирана сигурност	Да
WSRRSrv02 профил	
Уеб достъп (HTTP)	http://edu.grid.uni-sofia.bg:9081/ServiceRegistry
Уеб достъп (HTTPS)	https://edu.grid.uni-sofia.bg:9444/ServiceRegistry
Реализирана сигурност	Не

Таблица 12. Профили на регистъра WSRR

4.2.1 Подход за публикуване на услугите от BDII в модул Регистър

В регистъра на средата за управление на бизнес-процеси освен услугите от модула Адаптери за достъп до g-Lite, могат да бъдат публикувани и всички налични Уеб услуги от EGI. Един възможен подход за това е да се извлекат от информационната

система на BDII само тези услуги, които притежават WSDL описания и те да бъдат публикувани в регистъра. Предвид представените в трети раздел на трета глава изисквания, тази функционалност трябва да предоставя:

- възможност, за регистриране на всички налични WSDL файлове на услуги от средата g-Lite в регистъра;
- възможност за ежедневно обновяване на информацията в регистъра с актуалните (налични) услуги от g-Lite средата.

За целите на дисертацията тази функционалност беше реализирана. Създаденият програмен код:

- Извлича всички пътища до WSDL описания на услуги от информационната система BDII;
- Извлича WSDL описанията от указаните пътища. В програмния код е предвидено, че за извличането на някои от WSDL описанията може да се изисква грид сертификат;
- Публикува WSDL описанията на услугите в регистъра WSRRSrv02, като извиква регистъра като Уеб услуга.

За реализиране на функционалността са използвани среда за разработка Eclipse и език за програмиране Java, както и предоставената от Linux обвивка - BASH. Примерен фрагмент от код на Java за извикване на регистъра като Уеб услуга е показан в (Таблица 13).

```
package wsrr;


import sdo.commonj.*;
import com.ibm.www.xmlns.prod.serviceregistry._6._0.sdo.*;
import com.ibm.www.xmlns.prod.serviceregistry._6._0.ws.sdo.*;
.............................................................
public class ServiceRegistry {
public void publishWSDLDocument(String wsdlLocation, String wsdlFilePath)
{
.......................................................
   bsrURI = findWSDLDocument(wsdlLocation);
   if(bsrURI != null) return; // --> without versions
   // version = wsdlVersion; // ---> with versions

   try {
     aURL = new URL(wsdlLocation);
     filePath = aURL.getFile();
     file = new File (filePath);
     fileName = file.getName().replace('?', '.');
```

```
      if (wsdlLocation != null)doc.setLocation(wsdlLocation);
      if (wsdlcontent != null) doc.setContent(wsdlcontent.getBytes());

      doc.setName(fileName);
      doc.setDescription(doc.getDescription());
      doc.setBsrURI(new Id(idref));
      doc.setVersion(version);
      artefactArray[0] = doc;
      wsrr.setArtefacts(artefactArray);
      wsrr.setRoot(new IDRef(idref));


      dg.setWSRR(wsrr);
       bsrURI = proxy.create(dg);
       System.out.println("Returned bsrURI is " + bsrURI);
    }
    catch (Exception e) {
           System.out.println("Message: " + e.getMessage());
    }
}
public static void main(String[] args){
String location =
"http://myproxy.grid.uni-sofia.bg:8080/heph/services/ProxyService?wsdl";
String path = "ProxyService.wsdl";
ServiceRegistry r = new ServiceRegistry();
r.publishWSDLDocument(location, path);
}
}
```

Таблица 13. Примерен фрагмент от код на Java за извикване на регистъра като Уеб услуга

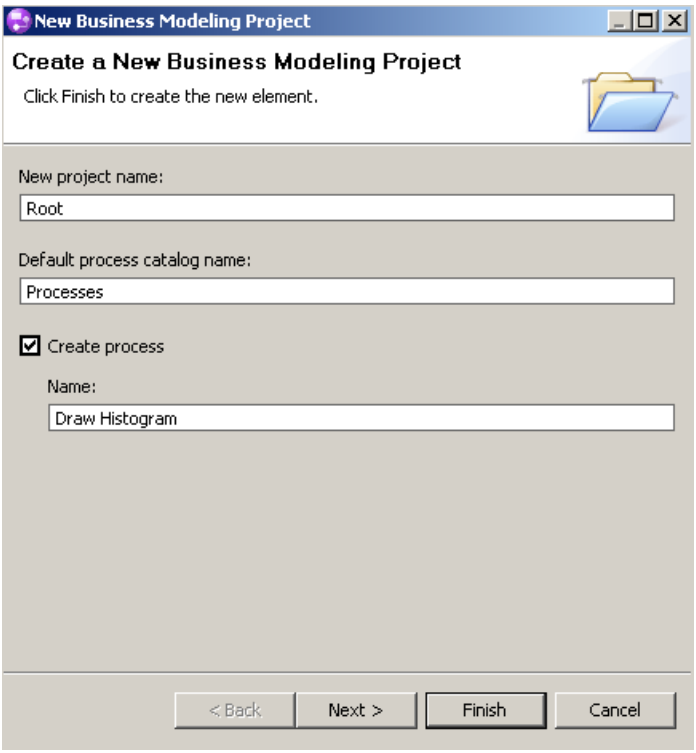
4.3 Модул Среда за разработка на бизнес-процеси

Като среда за разработка на бизнес-процеси в EGI са използвани инструменталните среди IBM WebSphere Business Modeler и IBM WebSphere Integration Developer (WID), част от IBM SOA Foundations.

Софтуерът WebSphere Business Modeler 7.0 е среда за проектиране на бизнес-процеси, която предоставя възможност за моделиране на бизнес-процеси и артефактите свързани с тях. За целите на дисертацията софтуерът е инсталиран на настолен компютър с операционна система Windows XP, който не е част от EGI грид инфраструктурата.

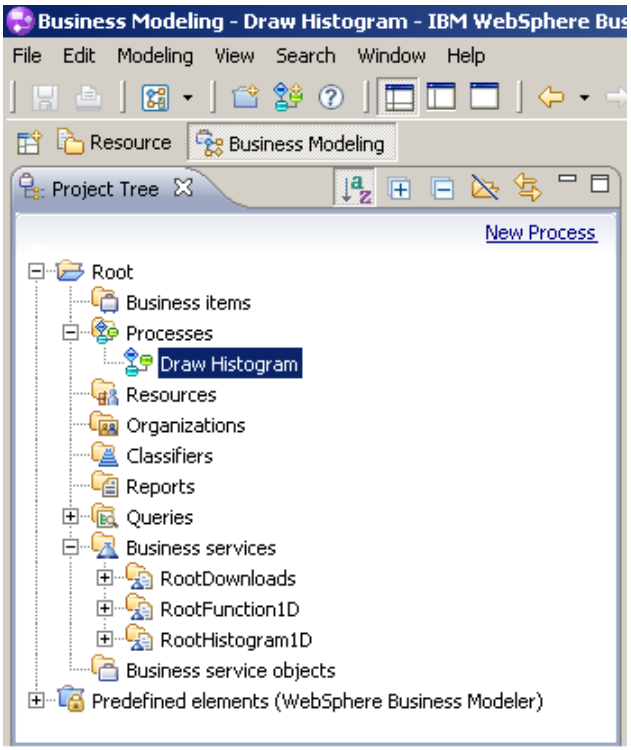
Средата за проектиране е базирана на Eclipse. Платформата Eclipse е и общата инструментална рамка, която се следва от инструменталните средства в IBM SOA Foundation. Проектирането на бизнес-процес в средата от грид-потребител е обвързано

със създаването на проект. На Фиг. 46 е показан примерен екран за създаването на проекта *Root* с бизнес-процес *Draw Histogram*



Фиг. 46. Създаване не проект в среда за проектиране IBM WebSphere Business Modeler

Всички проекти в средата имат обща структура. На Фиг. 47 е показана структурата на проекта Root.



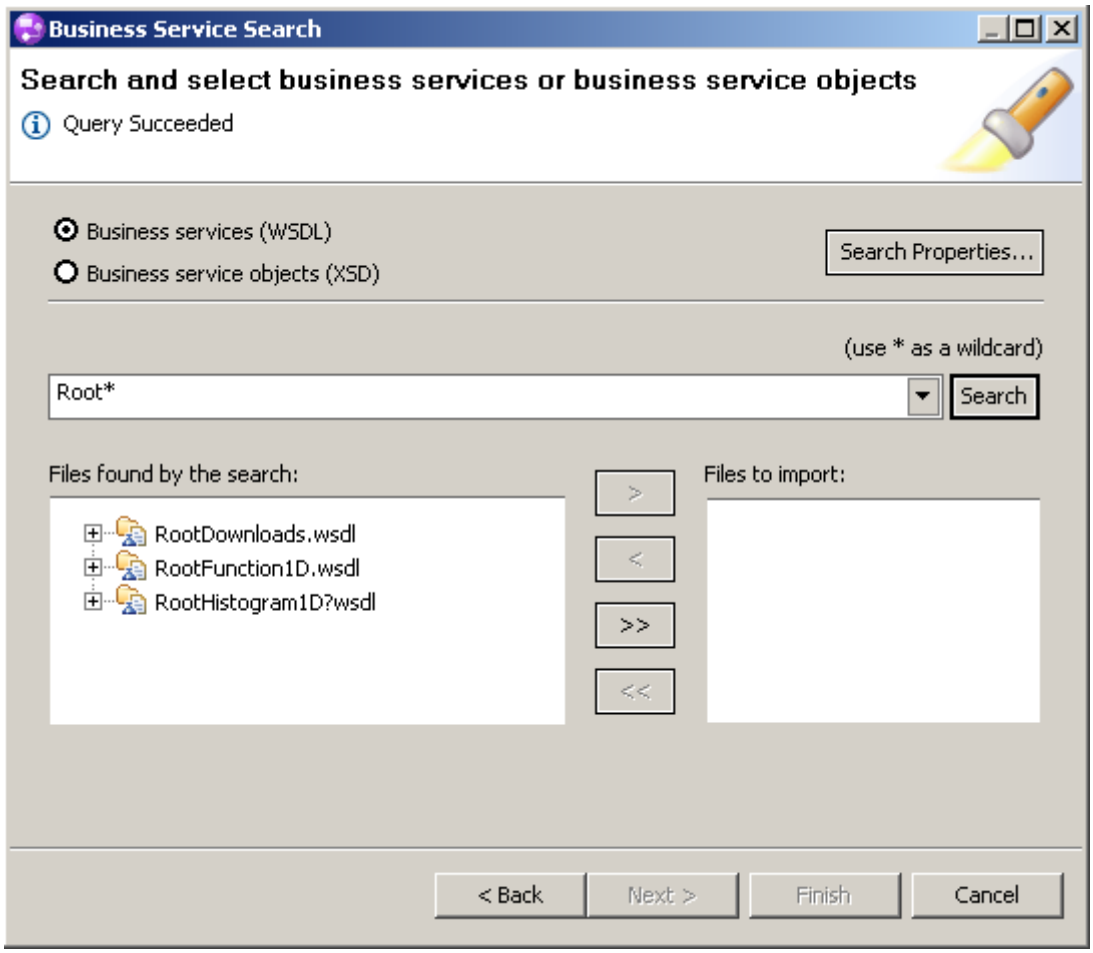
Фиг. 47. Структура на проект в средата за проектиране IBM WebSphere Business Modeler

За грид-потребителите две от папките в проекта са от съществено значение:

- Processes – която съдържа проектирания от потребителя бизнес-процес;
- Business services – която съдържа наличните Уеб услуги, които потребителя може да използва при моделирането на бизнес-процеса.

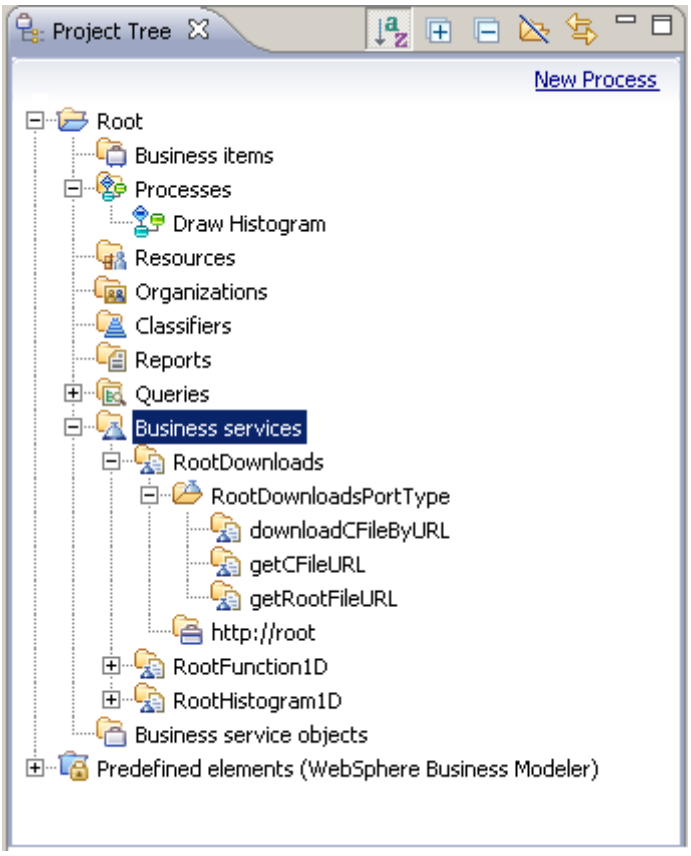
Средата за проектиране предоставя на грид-потребителя интуитивен графичен интерфейс за добавяне на Уеб услуги от регистъра към проекта. Средата поддържа интеграция с WSRR регистър. Необходима е еднократна настройка на средата за проектиране, при която да се укаже кои са наличните регистри, в които потребителя може да търси Уеб услуги. Тази настройка се извършва от администратора на средата. Примерни екрани на конфигуриране на настройката в средата са поместени в Приложение 2.

Грид-потребителите на EGI, могат да търсят за налични услуги в регистъра и да добавят желаните от тях Уеб услуги в папката *Business services*. На Фиг. 48 е показан диалогов прозорец за търсене в регистър с резултатът от търсените услуги.



Фиг. 48. Диалог за търсене в регистър с резултатът от търсените услуги

На Фиг. 49 е показана структурата на проекта на грид-потребител, след добавяне на Root услугите от регистъра.

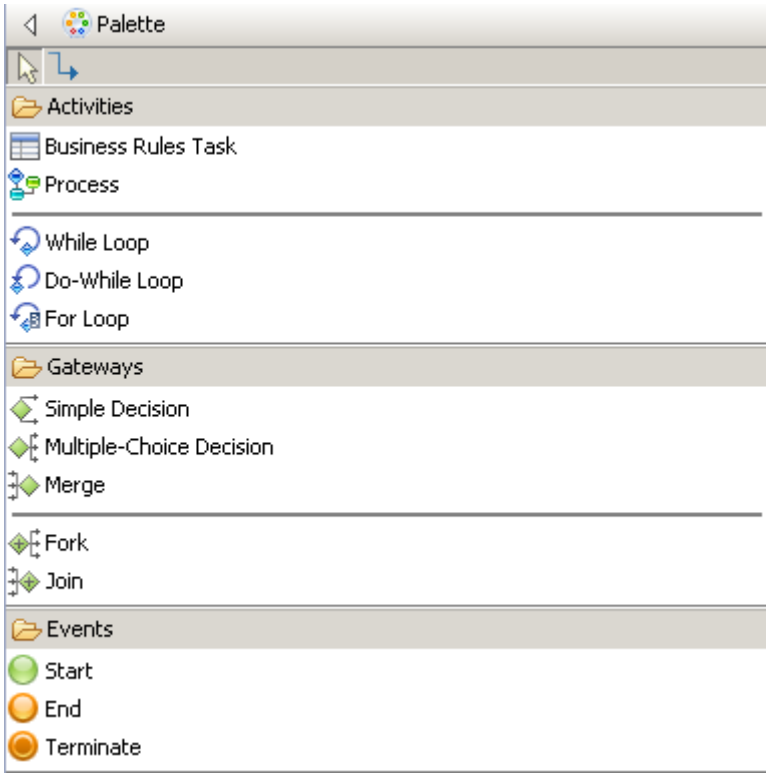


Фиг. 49. Структура на проекта на грид-потребител, след добавяне на услуги от регистъра

Средата за проектиране предоставя и интуитивен графичен интерфейс за създаване на нов бизнес-процес от грид-потребителя. Моделирането на бизнес-процеса от грид-потребителя в средата, става в редактора на бизнес-процеси. При проектирането на бизнес-процеса на грид-потребителя се предоставя възможност да избира от разнообразни дейности. Дейностите, които могат да се използват при моделирането на бизнес-процеси са всички структурни дейности, всички дейности събития, както и следните основни дейности:

- Задача бизнес-правило – стъпка от бизнес-процеса за изпълняване на бизнес-правило;
- Услуга (<invoke>) – стъпка от бизнес-процеса за синхронизирано извикване на Уеб услуги;
- Под-процес (<под-процес>)– стъпка от бизнес-процеса, обозначаваща йерархия в процес, за който са в сила горните ограничения.

На Фиг. 50 представяме част от дейностите в палитрата на редактора на бизнес-процеси, които имат отношение към проектирането на бизнес-процеси от грид-потребител.

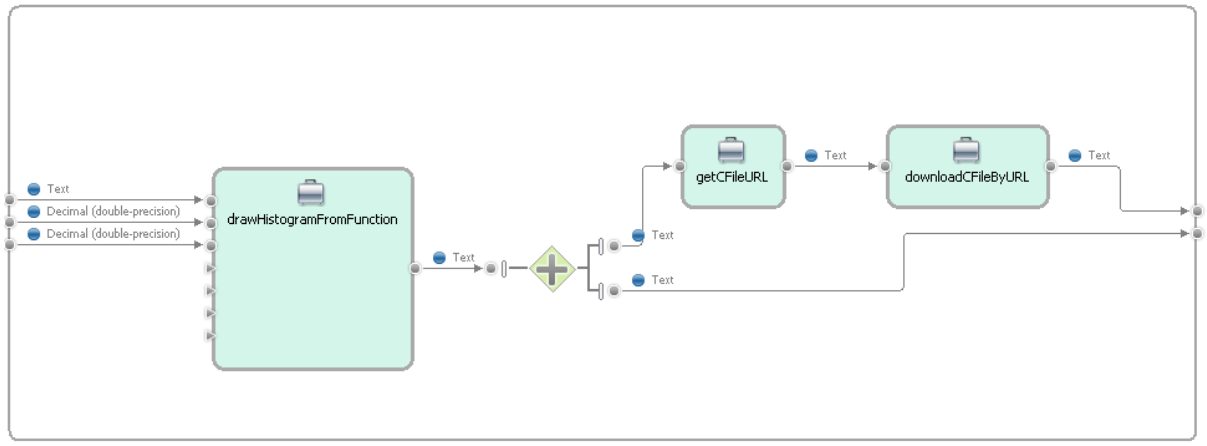


Фиг. 50. Палитра в редактора на бизнес-процеси

Моделирането на самият бизнес-процес от грид-потребителя в редактора на бизнес-процеси е графично. Посредством взимане и поставяне на операция на услуга или дейност, потребителят може да проектира своя бизнес-процес. За всяка една дейност от бизнес-процеса (с изключение на структурните дейности) и за самия бизнес-процес, грид-потребителят може да укаже бизнес метрики, за които да се следи в средата за изпълнение. Такива бизнес метрики са например: състояние на бизнес-процеса (дейността), време на стартиране, време на приключване, общо време на работа за бизнес-процеса (дейността) и други.

На Фиг. 51 е показан пример на моделиран бизнес-процес, който използва Уеб услугите за достъп до ROOT. Бизнес-процесът приема три входни параметъра: функция и интервал (минимална стойност на интервала, максимална стойност на интервала). Операцията *drawHistogramFromFunction* на услугата *RootHistogram1D*, приема седем входни аргумента, три от тях са подадените като вход на бизнес-процеса, а останалите са зададени като константи. Това се указва в свойствата на дейността *drawHistogramFromFunction* от бизнес-процеса. Използването на структурната дейност

за разклоняване (Fork), позволява резултатът от извикването на услугата да се върне като резултат от бизнес-процеса и паралелно с това да бъде входен параметър за друга операция на услуга. В резултат от изпълнението на бизнес-процесът се извеждат два изходни параметъра: URL до картинката на изчертаната хистограма и C файла на изчертаната хистограма.



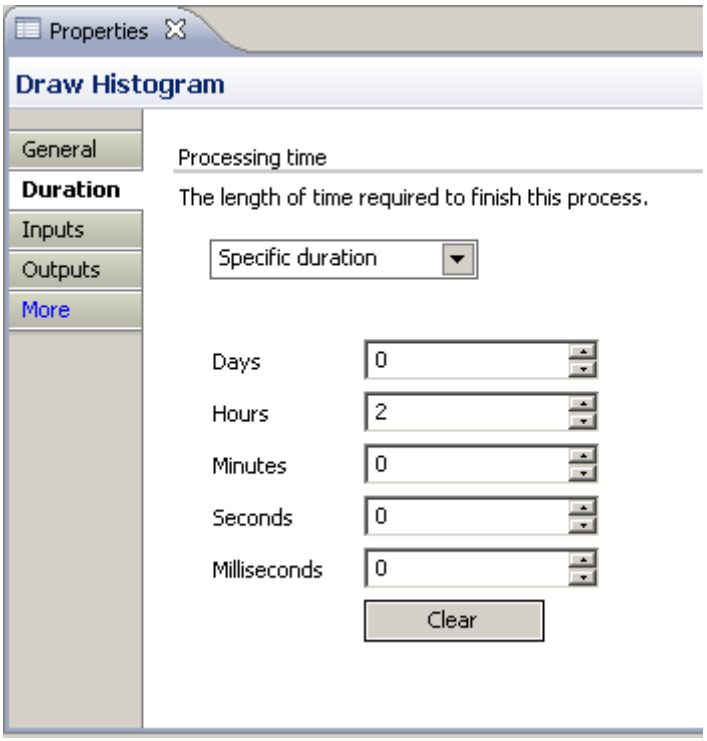
Фиг. 51. Пример на моделиран бизнес-процес в среда за проектиране

За всяка от дейностите в бизнес-процеса и за самия процес са в сила следните общи свойства (Таблица 14). Те могат да бъдат променяни в изглед на средата за проектиране.

Свойства (Properties)	**Описание**
General	Обща информация за бизнес-процеса (дейността). Например име на бизнес-процес и описание
Duration	Времето необходимо за завършване на бизнес-процеса (дейността)
Input	Входни параметри за бизнес-процеса.
Output	Изходни параметри за бизнес-процеса

Таблица 14. Общи свойства на дейности в бизнес-процес

На Фиг. 52 е показан изглед на средата за проектиране със свойствата на бизнес-процеса *Draw Histogram*. В случая е зададена продължителността на работа на бизнес-процеса, която е 2 часа.



Фиг. 52. Свойства на бизнес-процеса *Draw Histogram*

Средата за проектиране IBM WebSphere Business Modeler, предоставя и възможности за създаване на класификатори, симулиране на бизнес-процеси, създаване на бизнес-обекти, създаване на роли, моделиране на сложни бизнес-процеси с задачи за хора и други. Поради наложените в Глава 3 изисквания, тези дейности няма да разглеждаме в детайли.

Средата за проектиране предоставя интуитивен графичен интерфейс на грид-потребителите да изнесат разработения от тях бизнес-процес във формат за разгръщане на процесен сървър. Необходимо условие за това е режимът на моделиране на бизнес-процесът в средата за проектиране да е режим за разгръщане на процесен сървър.

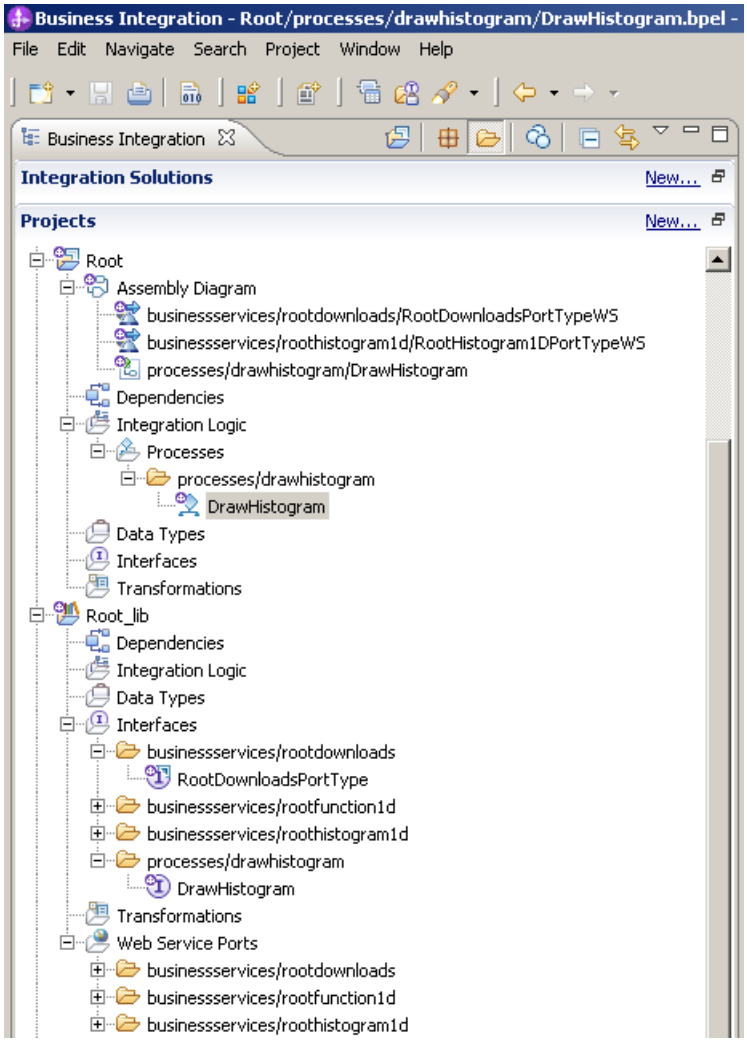
Софтуерът WebSphere Integration Developer 7.0 е среда за разгръщане на бизнес-процеси, която предоставя възможност за разгръщане на разработения от потребителя бизнес-процес в среда за изпълнение, както и генериране на потребителски интерфейс за моделирания бизнес-процес. За целите на дисертацията софтуерът е инсталиран на настолен компютър с Windows XP операционна система, който не е част от EGI грид инфраструктурата.

За специалистите по информационни технологии, средата за разгръщане е мощно средство за разработване на бизнес-процеси и приложения, които в последствие могат да бъдат разгърнати на процесен сървър. Средата за разгръщане също предоставя настройка за интеграция с WSRR регистър, което позволява специалистите по

информационни технологии да публикуват в регистъра разработени от тях услуги и бизнес-процеси. Тази настройка се извършва от администратора на средата. Примерни екрани на конфигуриране на настройката в средата са поместени в Приложение 2.

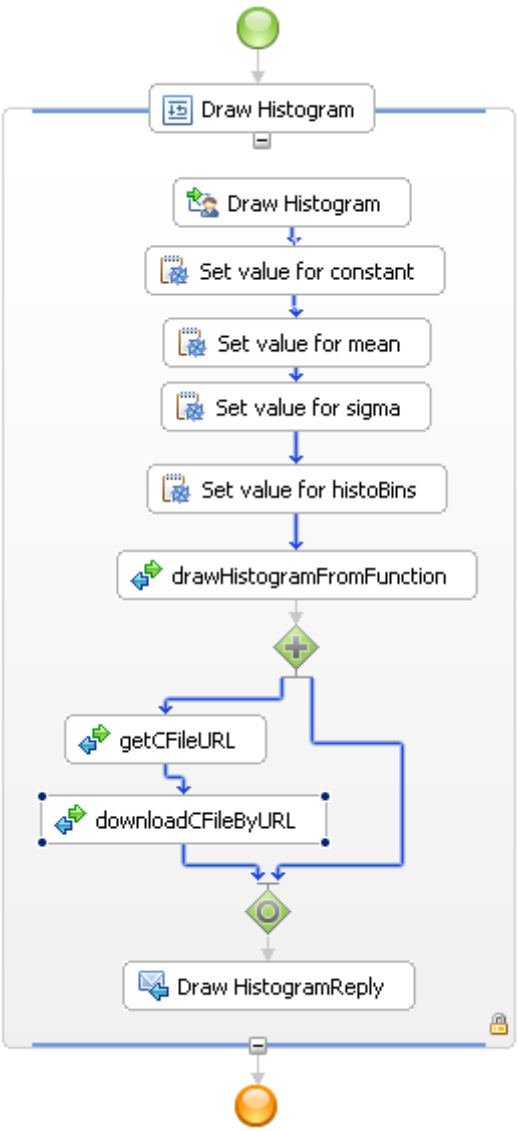
Използването на WebSphere Integration Developer по гореописания начин изисква познания по програмиране. В оставащата част от настоящата глава, ще разгледаме възможностите на средата, които имат отношение към разгръщането на бизнес-процеси от неспециалисти, които нямат познания по програмиране, каквито в повечето случаи са грид-потребителите.

Софтуерът WebSphere Integration Developer, предоставя интуитивен графичен интерфейс на грид-потребителя за внасяне на моделирания бизнес-процес от средата за проектиране. Средата за разгръщане автоматично генерира, необходимият програмен код и необходимите проекти за грид-потребителя. На Фиг. 53 е показан екран от автоматично генерираните от средата за разгръщане програмен код за проекта *Root*.



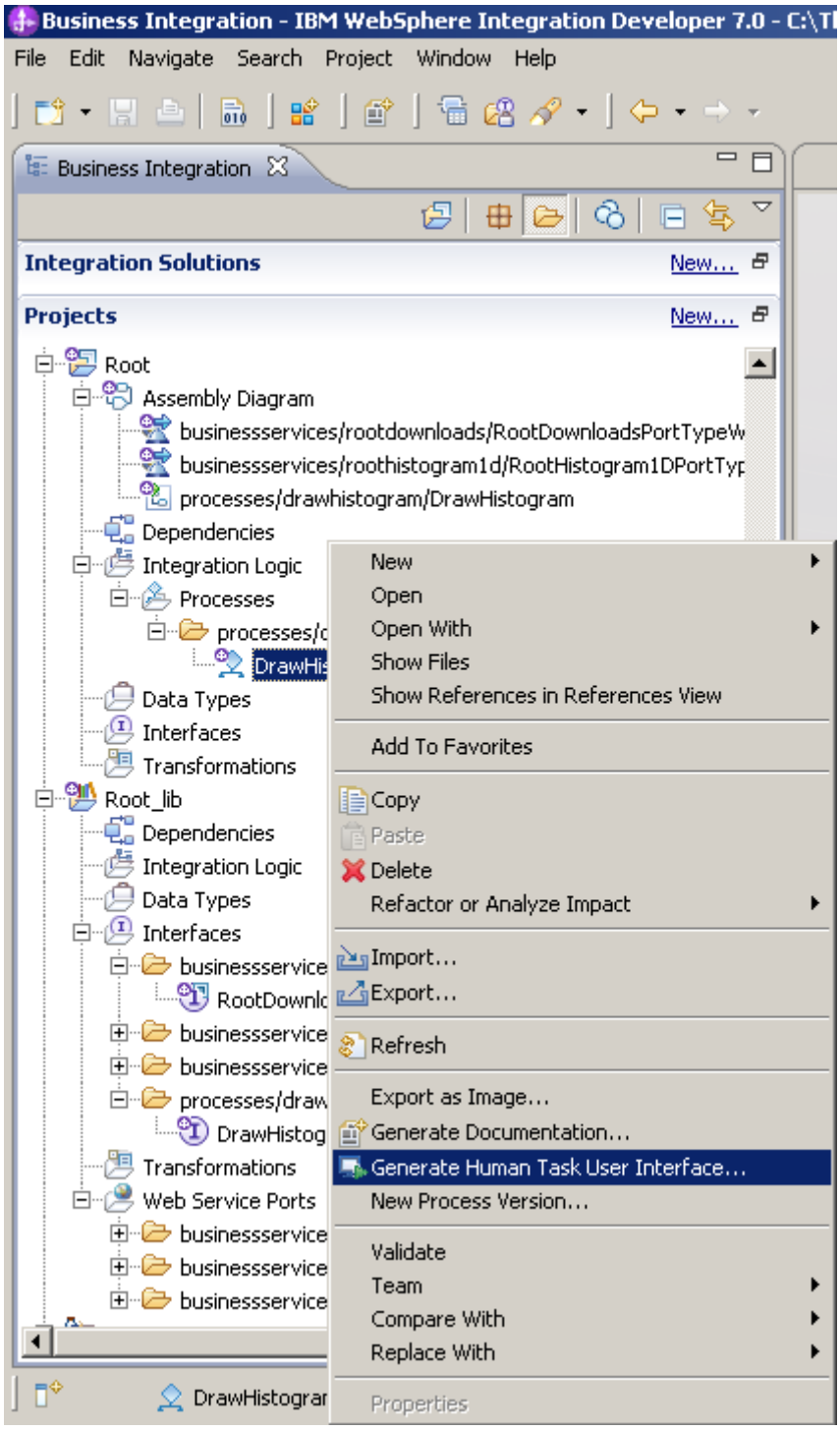
Фиг. 53. Проекти генерирани автоматично за проекта *Root* от среда за разгръщане

Автоматично от средата за разгръщане се генерират три проекта, в случая *Root*, *Root_lib* и *RootApp*. Проектите с разширение *_lib*, съдържат артефакти, типове на данни интерфейси на услуги и друга информация, която се използва в бизнес-процеса. В проекта *Root*, се съдържа бизнес-процеса в BPEL формат, както и генерирани компоненти за извикване на съответните Уеб услуги. В проекта *RootApp*, се съдържа пакетиран бизнес-процес готов за разгръщане на процесен сървър. На Фиг. 54 е показан генерирания BPEL код за бизнес-процеса *Draw Histogram*.



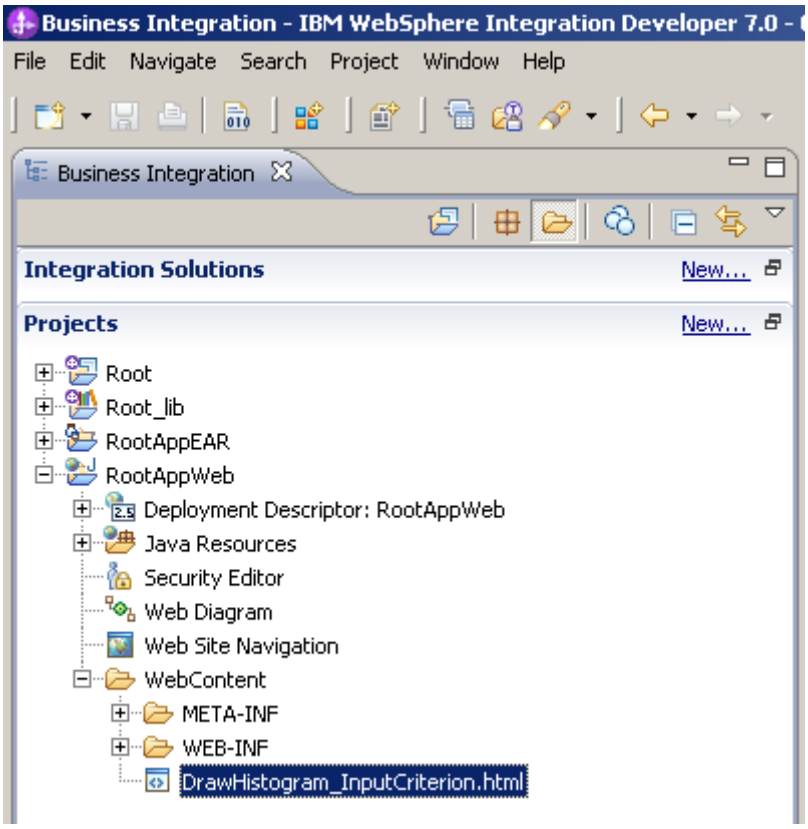
Фиг. 54. BPEL код за бизнес-процеса *Draw Histogram*

Средата за разгръщане предоставя на грид-потребителя и интуитивен графичен интерфейс за генериране на потребителски интерфейс към моделирания бизнес-процес. На Фиг. 55 е показан екран за генериране на потребителски интерфейс към бизнес-процеса *Draw Histogram*.



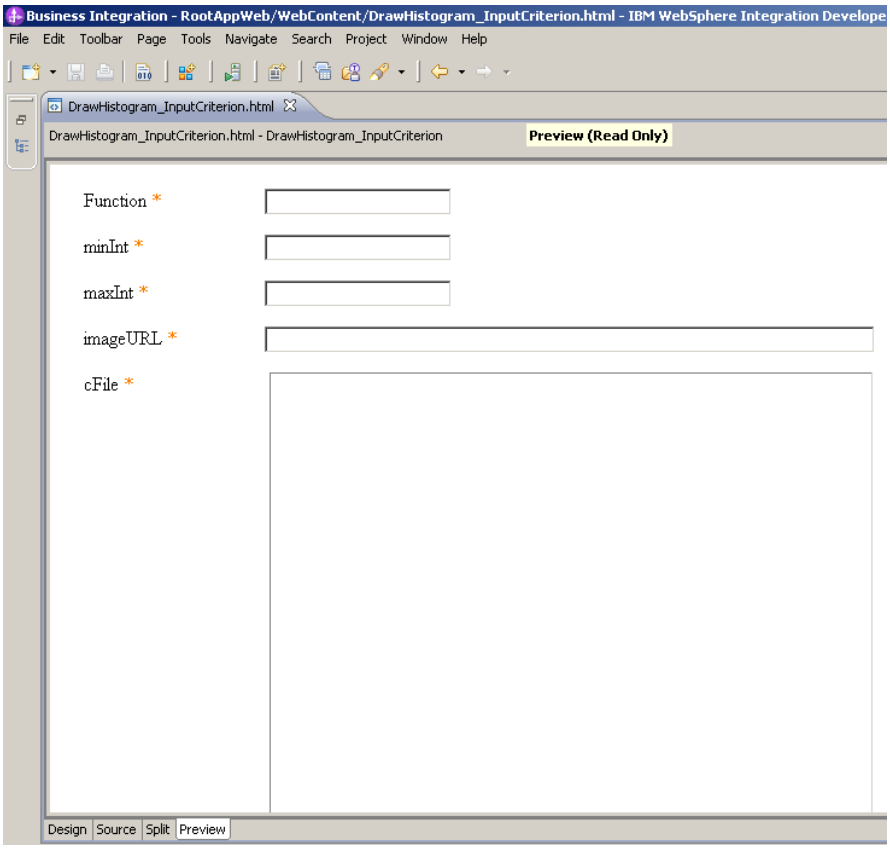
Фиг. 55. Потребителски интерфейс към бизнес-процеса *Draw Histogram*

Средата за разгръщане генерира два проекта *RootAppWeb* и *RootAppWebEAR*. В първия се съхранява HTML кода на потребителския интерфейс, а във втория се съдържа пакетиран потребителския интерфейс готов за разгръщане на процесен сървър. Генерираните проекти са показани на Фиг. 56.



Фиг. 56. Проекти на потребителския интерфейс генериран за процеса *Draw Histogram*

На Фиг. 57 е показан екран на генерираната форма в HTML формат



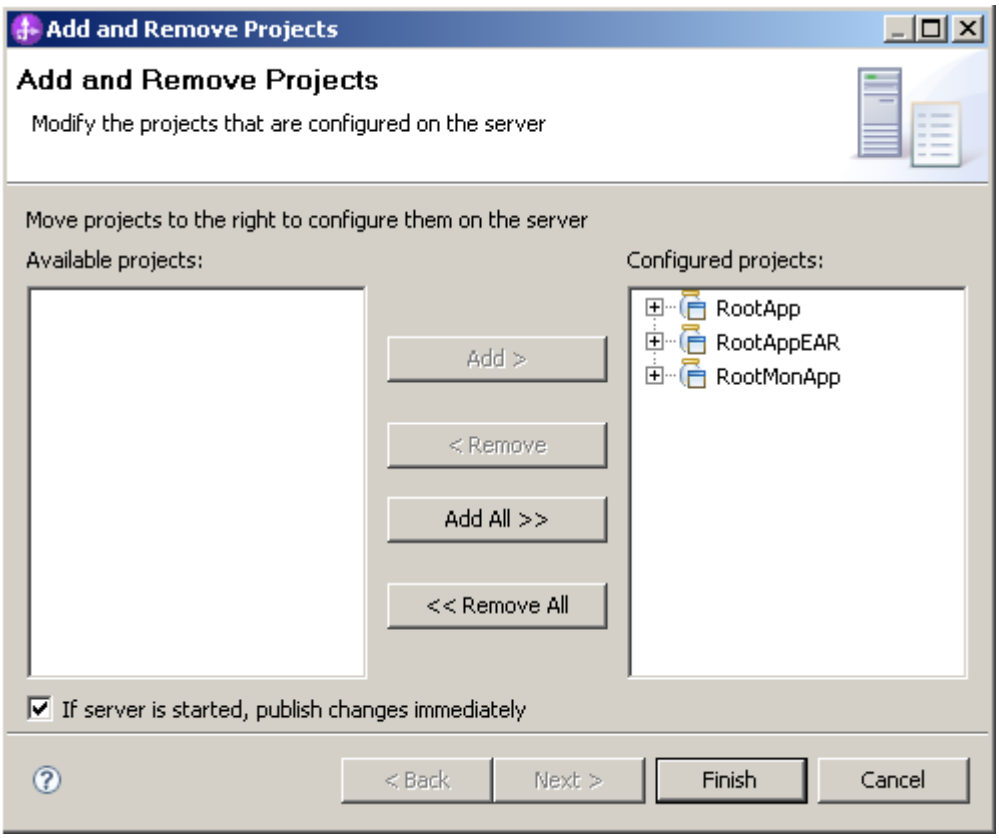
Фиг. 57. Генериран форма за извикване на бизнес-процеса *Draw Histogram*

Във формата присъстват полета за указване на входните параметри за бизнес-процеса, както и полета за показване на резултата от изпълнението на бизнес-процеса. По подразбиране полетата за показване на изходния резултат от бизнес-процеса, са видими след изпълнение на бизнес-процеса, но не и при въвеждане на входните параметри. Формата може да бъде променяна според предпочитанията на потребителя.

Средата за разгръщане WebSphere Integration Developer, предоставя различни технологични решения за формата на генерирания потребителски интерфейс. Това са JSF страници, портлети, Lotus форми и HTML-Dojo страници. Използваното решение в случая е HTML-Dojo страници.

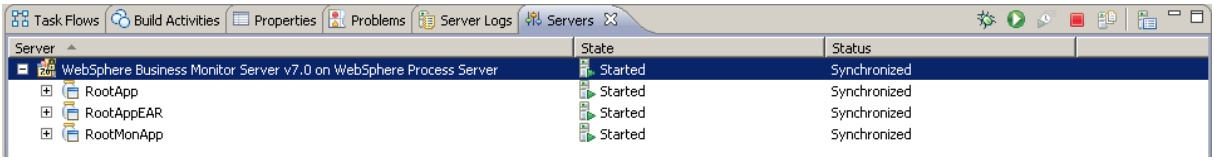
При моделирането на бизнес-процеса, грид-потребителя може да укаже метрики за всяка една дейност от бизнес-процеса и за самия бизнес-процес. Това грид-потребителят може да направи, както в средата за проектиране така и в средата за разгръщане. В резултат на указаните бизнес метрики, в средата за изпълнение ще бъдат наблюдавани състояние на бизнес-процеса, време на стартиране, време на приключване, общо време на работа за бизнес-процеса и други. Средата за разгръщане предоставя функционалност на грид-потребителя да генерира готови проекти съдържащи описанията на посочените метрики и пакети за тяхното разгръщане на процесен сървър. Примерни екрани за генериране на проект за наблюдение в средата за разгръщане са показани в Приложение 3

Средата за разгръщане, предоставя на грид-потребителя и интуитивен графичен интерфейс за разгръщане на бизнес процеса в среда за изпълнение. На Фиг. 58 е показан екран за разгръщане на цялата генерирана функционалност от средата за разгръщане за бизнес-процеса *Draw Histogram* в средата за изпълнение.



Фиг. 58. Разгръщане на бизнес-процеса *Draw Histogram* в среда за изпълнение

На Фиг. 59 е показан екран от разгърнатия вече бизнес-процес



Фиг. 59. Разгърнат бизнес-процес *Draw Histogram* в среда за изпълнение

4.4 Модул Среда за изпълнение на бизнес-процес

Като среда за изпълнение на бизнес-процеси в EGI е използван IBM WebSphere Process Server (WPS), част от IBM SOA Foundations. Софтуерът WebSphere Process Server 7.0, предоставя услуга за изпълнение на бизнес-процеси, услуга за наблюдение, услуга за управление на бизнес-правила и услуга за управление на задачи за хора. За целите на дисертацията софтуерът е инсталиран на настолен компютър с операционна система Linux (SLC4), който не е част от EGI грид инфраструктурата.

Реализацията на IBM за процесен сървър се базира на надстройка на приложен сървър и шина на услугите в предприятието. Надстройката е показана на Фиг. 60 [37]. Шината на услугите (IBM WebSphere Enterprise Service Bus) е неделима част от средата

за изпълнение. Тя предоставя единна инфраструктура за интеграция и свързаност на приложения и услуги.

Фиг. 60. IBM WebSphere Процесен сървър

Разгърнатият от грид-потребителя бизнес-процес на процесен сървър, се изпълнява от услуга за изпълнение на бизнес-процеси. Услугата интерпретира BPEL описанието на бизнес-процеса, като извиква съответните услуги в него. Ако за бизнес-процеса е реализиран модел за наблюдение, то услуга за наблюдение (IBM WebSphere Business Monitor) наблюдава бизнес-процеса по време на изпълнението му. Бизнес-процесът се наблюдава по ключови индикатори указани при проектиране на бизнес-процеса. Комуникацията между услугата за наблюдение и услугата за изпълнение, се осъществява през шината на услугите.

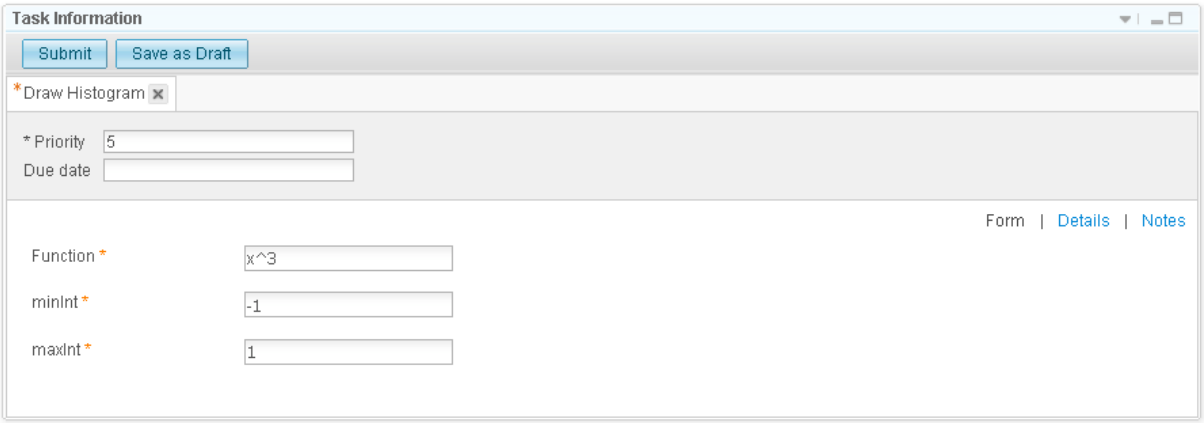
Услугата за управление на бизнес-правила, предоставя на грид-потребителя, възможност да променя бизнес-правилата по време на изпълнение на бизнес-процеса. Комуникацията между двете услуги, също се осъществява през шина на услугите.

Услугата за управление на задачи за хора, позволява грид-потребителя да извиква бизнес-процес. В (Таблица 15) са показани основните типове на задачи за хора. Средата за управление на бизнес-процеси в EGI, поддържа единствено задачи за хора от типа човек-услуга.

Тип на задача		Описание
	Услуга-човек	Задачите за хора интегрирани в BPEL процесите
	Човек-услуга	Задачи, при които потребител извиква услуга
	Човек-човек	Задачи, които отразяват съвместна дейност между потребителите

Таблица 15. Типове на задачи за хора в среда за изпълнение

Пример за задача човек-услуга е потребител, който извиква BPEL процес. Такава задача се реализира, при генерирането на потребителския интерфейс за моделирания от грид-потребителя бизнес-процес в средата за разработка. На Фиг. 61 е показан задача от типа човек-услуга за извикването на бизнес-процеса *Draw Histogram.*



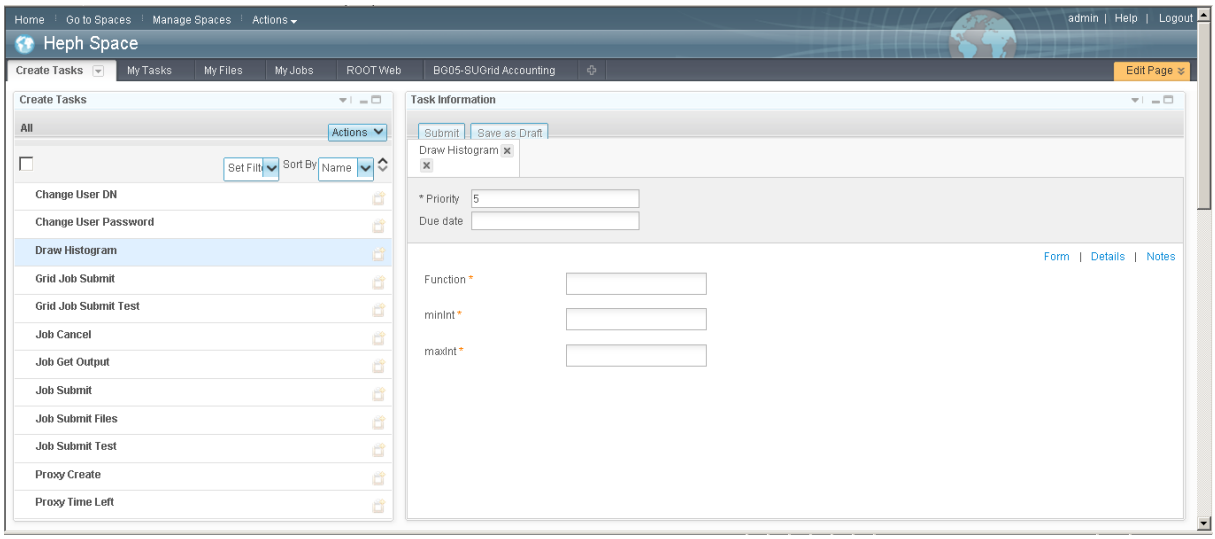
Фиг. 61. Задача от тип човек-услуга в среда за изпълнение

Задачите от типа човек-услуга са интегрирани в работно пространство на потребителя – IBM WebSphere Business Space, софтуер част от IBM SOA Foundations. През работното пространство, грид-потребителите могат да управляват задачи за хора от вида (човек - услуга) и така да стартират и да спират бизнес-процесите си. Работното пространство на потребителя е набор от Уеб базирани потребителски интерфейси, които се предоставят като част от средата за изпълнение (IBM WebSphere Process Server).

През работното пространство, грид-потребителя може да стартира бизнес-процес и да вижда резултата от изпълнението му. Може да проследява бизнес-метрики за процеса, може да управлява бизнес-правила свързани с процеса. За целите на

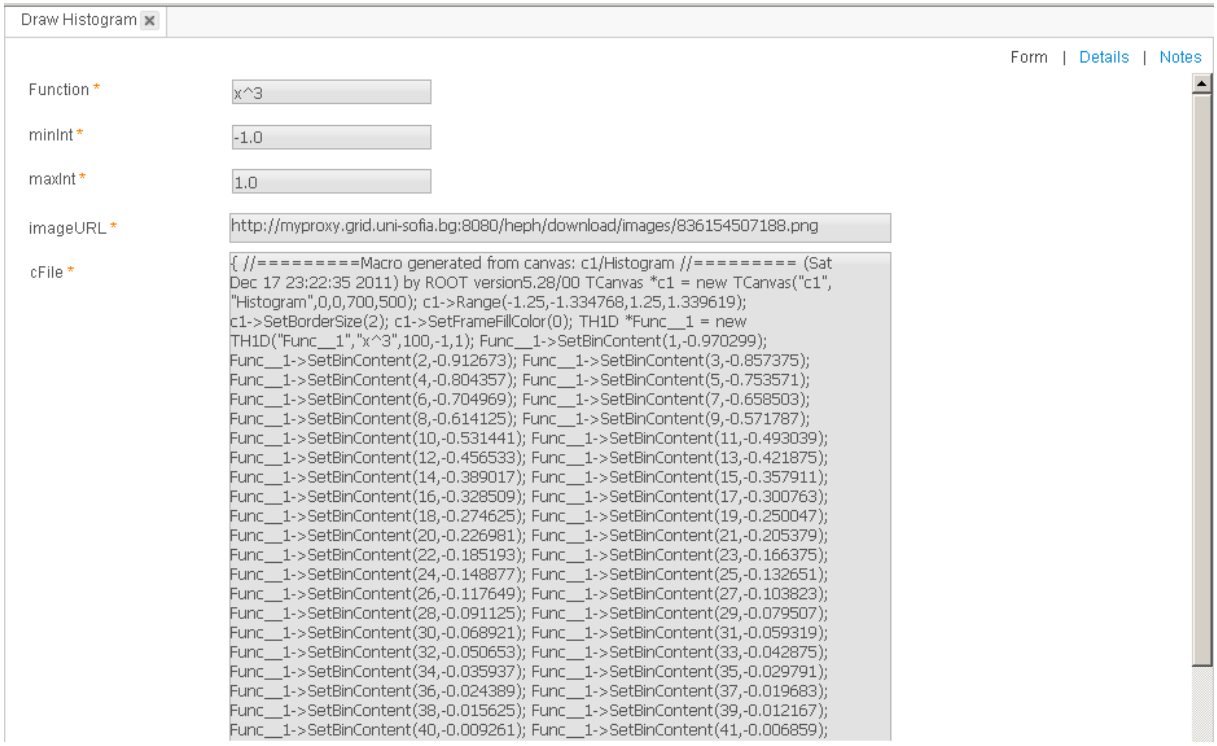
дисертацията, беше разработен профил на работно пространство за грид-потребителя. Екрани от работното пространство са представени в Приложение 5.

Работното пространство е достъпно на адрес: https://edu.grid.uni-sofia.bg:9446/BusinessSpace . На Фиг. 62 е показан примерен екран от работно пространство на грид-потребителя.



Фиг. 62. Работно пространство на грид-потребителя

На Фиг. 63 е показан резултат от изпълнението на бизнес-процеса *Draw Histogram* в среда за изпълнение.



Фиг. 63. Изпълнение на бизнес-процеса *Draw Histogram* в среда за изпълнение

4.5 Разработване на примерни бизнес-процеси в средата

В настоящата глава, ще представим два бизнес-процеса: примерен бизнес-процес за изпращане на тестова задача в грид и примерен бизнес-процес от Физиката на високите енергии, обслужващ експериментът NA62.

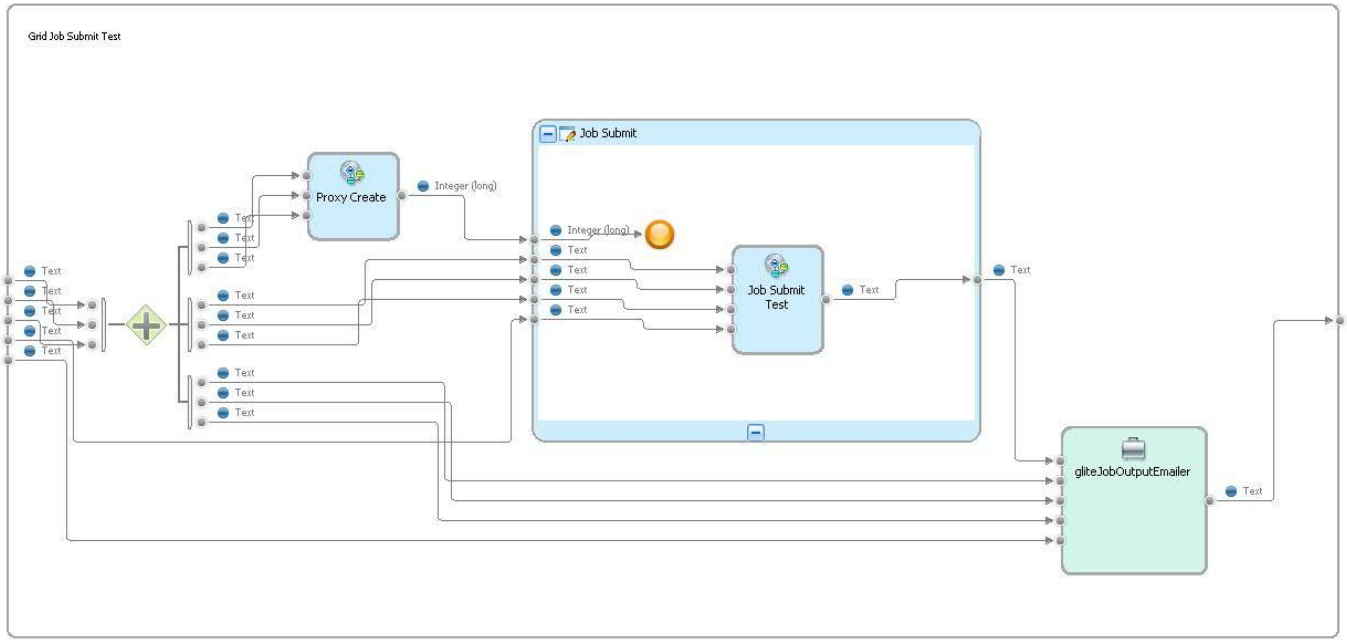
Примерен бизнес-процес за изпращане на тестова задача в грид

На базата на представените в Глава 4 Уеб услуги и средата за разработка са моделирани следните бизнес-процеси (Таблица 16).

Процес	**Описание**
Proxy Create	По зададени потребителско име, виртуална организация и парола, създава грид прокси за потребителя и го записва в базата от данни. Като резултат връща за колко време е валидно създаденото прокси
Job Submit	По зададени потребителско име, виртуална организация, парола, изпълним файл и входни аргументи към изпълнимия файл изпраща задача за изпълнение в грид. Като резултат връща уникален идентификатор на задачата (JobID)
Job Submit Test	По зададени потребителско име, виртуална организация, парола и изпълним файл, изпраща тестова задача в грид. Като резултат връща уникален идентификатор на задачата (JobID).
Job Submit Files	По зададени потребителско име, виртуална организация, парола, изпълним файл, имена на файлове обработвани в изпълнимия файл и входни параметри изпраща задача за изпълнение в грид. Като резултат връща уникален идентификатор на задачата (JobID).
Grid Job Submit	По зададени потребителско име, виртуална организация, парола, изпълним файл, имена на файлове обработвани в изпълнимия файл, входни параметри и e-mail на потребителя, създава прокси, изпраща задачата с изпълним файл в грид и e-mail на потребителя при успешно завършена задача.
Grid Job Submit Test	По зададени потребителско име, виртуална организация, изпълним файл и e-mail на потребителя, създава прокси, изпраща тестова задача с изпълним файл в грид и e-mail на потребителя при успешно завършена задача.

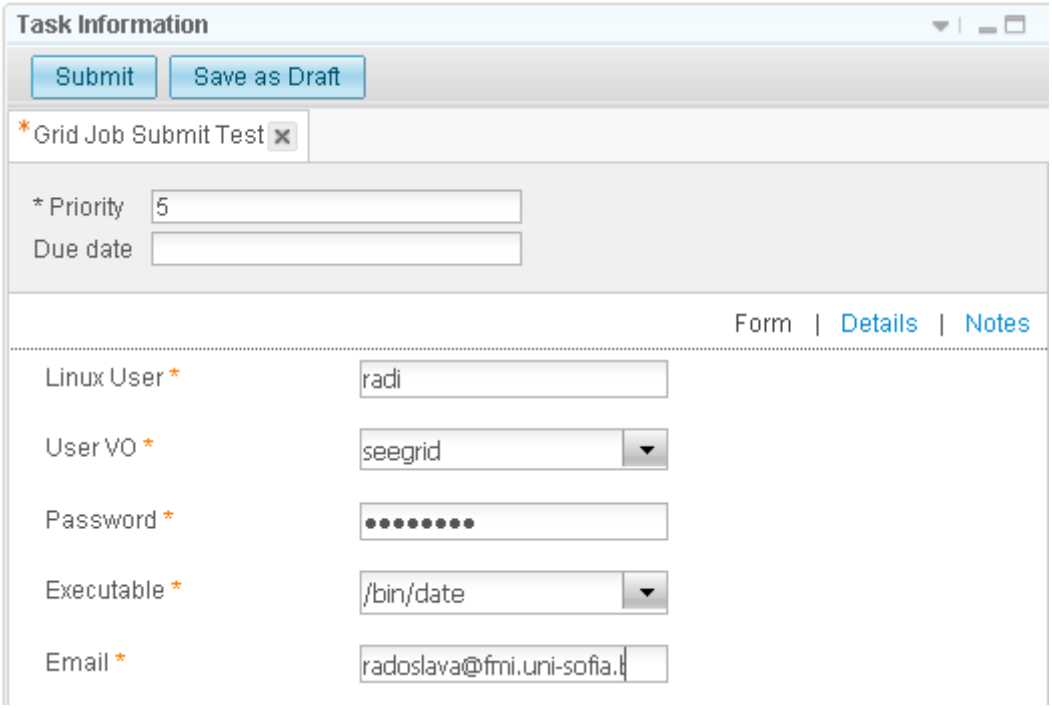
Таблица 16. Бизнес-процеси за изпращане на задача в грид

Моделите на бизнес-процесите са представени в Приложение 4. На Фиг. 64 е показан бизнес-процес Grid Job Submit Test, който се състои от два под-процеса – Proxy Create и Job Submit Test и услугата gliteJobOutputEmailer.

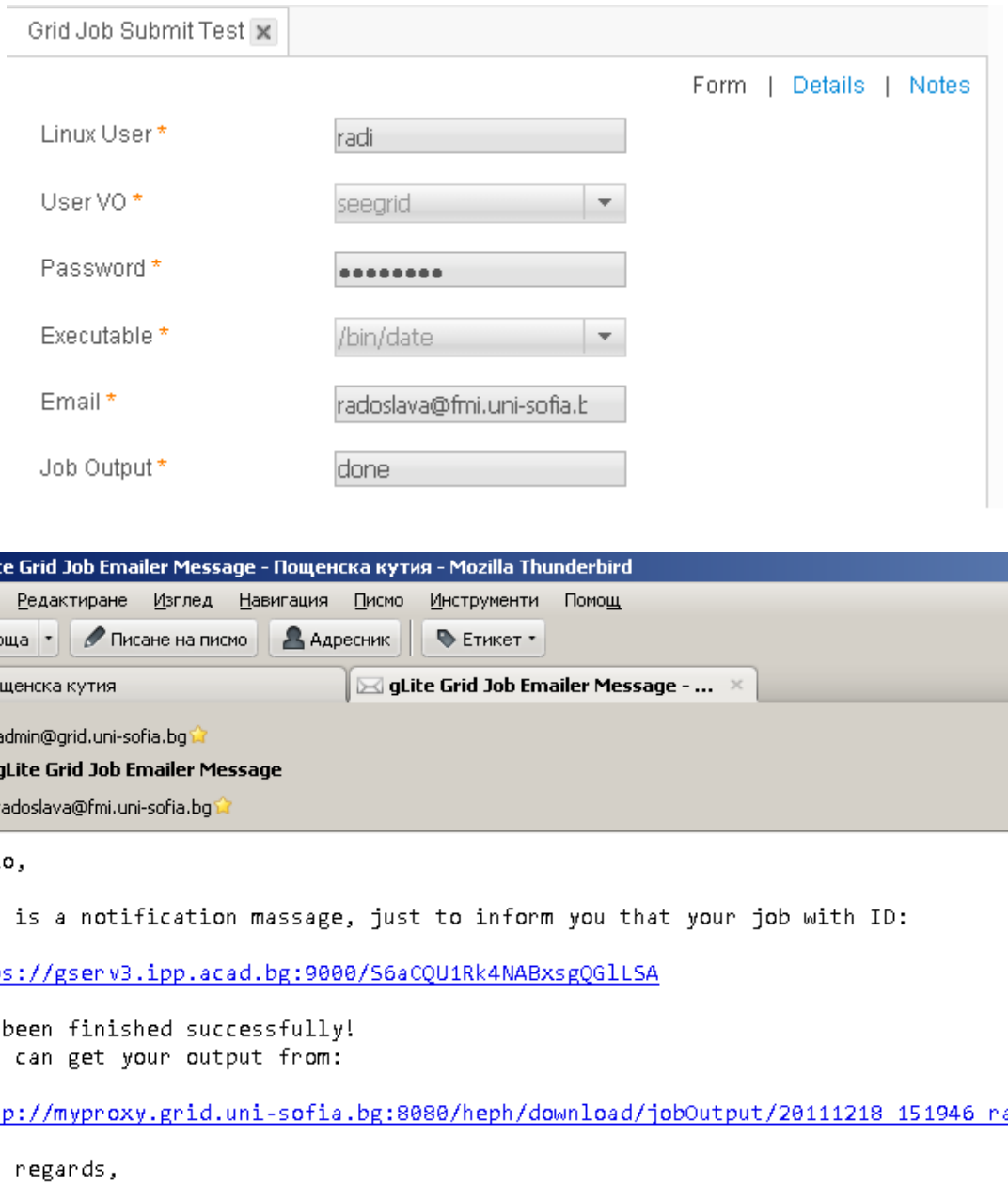


Фиг. 64. Бизнес-процесът Grid Job Submit Test

Бизнес-процесът е разгърнат в среда за изпълнение. На Фиг. 65 е показан екран от извикване на бизнес процеса през работно пространство на потребителя - Business Space, а на Фиг. 66 резултатът от изпълненият бизнес-процес.



Фиг. 65. Извикване на бизнес процеса *Grid Job Submit Test*



Фиг. 66. Резултат от бизнес процеса *Grid Job Submit Test*

Разработените бизнес-процеси се изпълняват в грид инфраструктурата EGI.

Примерен бизнес-процес от Физика на високите енергии, обслужващ експериментът NA62

Научният проект NA62 е проект от областта на Физиката на високите енергии, чиято цел е да изследва много рядък каонен разпад. За целта грид-потребителите от експеримента, анализират данни от ускорителя SPS. Ускорителят Super Proton Synchrotron (SPS) е вторият по големина ускорител в CERN. Той ускорява протони, както и извършва предварително ускоряване на частиците за сблъсък на снопове в

LHC. За експеримента NA62, ускорителят ще започне действителна работа през 2013 година.

В рамките на експеримента, ускорителят ще бъде използван за набор на реални данни. Детекторите на ускорителя, ще прочитат данните – генерирани в резултат от сблъсъка на протони и ще ги записват в локална ферма от компютри. Софтуерът, който ще извършва тази дейност е специфичен и е обвързан със самите детектори. Във фермата от компютри, ще се структурират данните от детектора под формата на събития. Към момента софтуерът е в процес на разработка.

Процесът на събиране на данни е обвързан с работата на ускорителя. На всеки 20 секунди ще бъдат засичани данни от детекторите. Тези данни ще се записват в локална ферма. Данните от фермата ще подлежат на допълнителна обработка, която ще включва реконструиране и филтриране на данните от допълнителен софтуер (Level 3). В резултат на това ще се получават обработени - реконструирани и филтрирани събития. Тези събития ще се записват на CASTOR – елемент за съхранение на данни от EGI. Данните ще се записват в ROOT формат. Всички ROOT файлове ще имат обща структура. Един набор на данни ще се прави за период от 3 месеца, в резултат на това ще бъдат генерирани около 300 000 ROOT файлове. След набора на данни грид-потребителите от експеримента NA62 ще извършват обработка на данните.

Грид-потребителите от експеримента NA62 могат да извършват следните типове обработка на данни:

- Повторна реконструкция – реконструкция от високо ниво. Генерираните от детекторите файлове, могат да подлежат на повторна реконструкция. Потребителя задава степен на окрупняване на данните. В резултат на повторната реконструкция се получават нови ROOT файлове, които също се записват на CASTOR.
- Повторно филтриране - Реконструираните данни могат да подлежат на повторно филтриране. Данните се филтрират, според зададен от потребителя начин на филтриране. В резултат на това се получават нови ROOT файлове, при които данните са повторно филтрирани.
- Анализ на експериментални данни – Потребителите извършват определени операции върху ROOT файловете, както и извличат определени данни от ROOT файловете.

Независимо от набора на експериментални данни от ускорителя, грид-потребителите могат да извършват и анализ на данни върху симулирани данни от

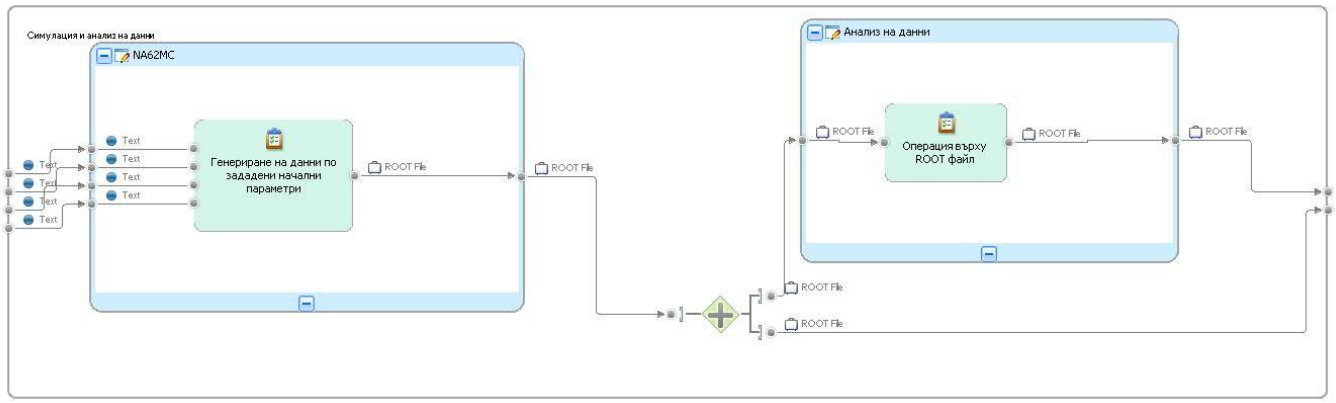
детектора. Софтуер NA62MC (Монте Карло) извършва симулация на данните от ускорителя, като генерира първични частици и симулира отклика на детектора. Резултатът от симулацията се записва в изходен файл. В конфигурационен файл, потребителят може да зададе начални параметри на софтуера: тип на сноб, брой частици, име на изходен файл. Резултатният файл подлежи на обработка по един от начините споменати в предходния параграф.

По-долу представяме как средата за управление на бизнес-процеси в EGI, може да бъде използвана от грид-потребители физици за целите на научния проект NA62. Описаните до момента действия на грид-потребителите физици, могат да бъдат реализирани като бизнес-процеси.

- Процес 1: Повторна реконструкция
    - Стъпка 1: Услуга, която предоставя функционалност за четене на файл от CASTOR. Входни параметри на услугата потребителско име и виртуална организация. Изходен параметър ROOT файл, при успешно прочетен файл или 0 при неуспешно прочетен файл;
    - Стъпка 2: Услуга, която предоставя функционалност за окрупняване на данни. Входен параметър на услугата ROOT файл, изходен параметър на услугата ROOT файл;
    - Стъпка 3: Услуга, която предоставя функционалност за записване на файл на CASTOR. Входни параметри на услугата ROOT файл, потребителско име и виртуална организация. Изходен параметър резултат от операцията на записване (успешно записан или неуспешно записан);
- Процес 2: Повторно филтриране
    - Стъпка 1: Услуга, която предоставя функционалност за четене на файл от CASTOR. Входни параметри на услугата потребителско име и виртуална организация. Изходен параметър ROOT файл, при успешно прочетен файл или 0 при неуспешно прочетен файл;
    - Стъпка 2: Услуга, която предоставя функционалност за филтриране на данни. Входен параметър на услугата ROOT файл, изходен параметър на услугата ROOT файл;
    - Стъпка 3: Услуга, която предоставя функционалност за записване на файл на CASTOR. Входни параметри на услугата ROOT файл, потребителско име и виртуална организация. Изходен параметър резултат от операцията на записване (успешно записан или неуспешно записан);

- Процес 3: Анализ на експериментални данни
    - Стъпка 1: Услуга, която предоставя функционалност за извършване на определена функционалност върху ROOT файловете (четене на данни от ROOT файловете)
- Процес 4: Симулация на експериментни данни
    - Стъпка 1: Услуга, която предоставя функционалност за симулиране на данни, чрез софтуера NA62MC. Входни параметри – параметри от конфигурационния файл. Изходен параметър ROOT файл

Поради предстоящия набор на данни, към момента няма реализиран софтуер за повторна реконструкция на данни и повторно филтриране. Поради това на Фиг. 67 представяме прототип на възможен бизнес-процеси, включващи симулация и анализ на експериментални данни.



Фиг. 67. Прототип на бизнес-процес за NA62

Софтуерът NA62MC е разработен, но единствено за Linux операционна система. Софтуер за анализ на данни още не е реализиран. Структурата на ROOT файла, в който ще се записват данните от симулациите и детекторите е в процес на уточнение. Поради тези съображения, към момента не е направена реализация на гореописания бизнес-процес. При разработване на подходящи Уеб услуги за софтуера NA62MC и Уеб услуги за анализ на данни върху ROOT файлове, средата за управление на бизнес-процеси в EGI е напълно приложима за бизнес-процесите от NA62.

4.6 Приноси

Приносите на тази глава са изцяло приложни. Те са както следва:

- Проектиран и разработен е метод за регистриране на Уеб услуги от информационната система на EGI в модул Регистър на услуги;
- Проектирани и реализирани са адаптери за достъп до услуги на грид инфраструктурата EGI;
- Проектирани и реализирани са услуги за достъп до компоненти от софтуер на Физиката на високите енергии, по специално ROOT;
- Проектирани и реализирани са примерни бизнес-процеси, които се изпълняват в грид инфраструктурата EGI;
- Демонстрирани са възможности за приложение и използване на средата за управление на бизнес-процеси в EGI;
- Демонстрирано е приложение на средата за създаване и изпълнение на примерен бизнес-процес за научен експеримент от Физиката на високите енергии;

Заключение

Изследванията представени в настоящия дисертационен труд, са насочени към разработването на модел и архитектура на среда за управление на бизнес-процеси в грид, в контекста на ориентирана към услуги архитектура. Разгледани са въпросите свързани с дефинирането, проектирането и изпълнението на бизнес-процеси в грид и по специално в Европейската грид инфраструктура.

В Глава 1 са изследвани и формулирани основните изисквания към средите за управление на бизнес-процеси, базирани на ориентираната към услуги архитектура. Изведен е обобщен компонентен модел на среда за управление на бизнес-процеси, базирана на ориентираната към услуги архитектура. Проучени са съществуващи услуги в грид инфраструктурата EGI.

В Глава 2 са изследвани и анализирани съществуващи среди за проектиране и изпълнение на бизнес-процеси в EGI, на базата на формулираните изисквания. Основен недостатък на представените среди е неориентираността към услуги. Нито една от разгледаните среди не предоставя поддръжка на шина на услугите, поддръжка на задачи за хора в бизнес-процеса или управление на бизнес-правила. Предимства, поддържани единствено от средите за управление на бизнес-процеси. В резултат на сравнителния анализ беше установено, че към момента EGI не предоставя среда, която да предоставя поддръжка на управление на бизнес-процеси, в контекста на ориентираната към услуги архитектура.

В Глава 3 е предложен модел и архитектура на среда за управление на бизнес-процеси в EGI. Разработени са основни случаи на употреба за средата. Установена е необходимост от създаване на адаптери за достъп до услугите на EGI. Предложено е архитектурно решение на среда за управление на бизнес-процеси в EGI. Демонстрирана е концепция за прилагане на ориентирана към услуги архитектура в грид. Предложеният модел предоставя среда, която поддържа бизнес-процеси в грид инфраструктурата EGI, чрез стандарти за Уеб услуги и предоставя възможност на грид-потребителите да дефинират и проектират сами процесите си в грид, без да е необходимо познания по програмиране или намеса на специалисти по информационни технологии.

В Глава 4 са предложени възможности за използване и приложение на средата в EGI. Проектирани са примерни композитни Уеб услуги за изпълнение на задачи в EGI. Проектирани са примерни композитни Уеб услуги към компоненти от софтуер на

физиката на високите енергии и по специално софтуера ROOT. Проектирани и реализирани са адаптери за достъп до услугите на грид инфраструктурата EGI. Реализирани са примерни бизнес-процес, на базата на реализираните Уеб услуги. Демонстрирани са възможности за приложение и използване на средата за управление на бизнес-процеси в EGI. Демонстрирано е приложение на средата за създаване и изпълнение на примерен бизнес-процес за научен експеримент от Физиката на високите енергии.

Библиография

[1] Горанова, Р. Д., "Обзор върху композиране на процеси от физиката на високите енергии в грид", Списание на Софийския Университет за електронно обучение, ISSN 1314-0086, Бр.4, 2010

[2] Димитров, В. "Ориентирана към услуги архитектура", ТехноЛогика, 2009.

[3] Приложения във Физика на високите енергии, [налично на: http://press.eu-egee.org/fileadmin/documents/infosheets/bg/hep_applications_bg.pdf, към дата: 19.12.2011, последно обновен на: 14.09.2007]

[4] Anjomshoaa, A., Drescher, M., Fellows, D., Ly, A., McGough, S., Pulsipher, D., Savva, A. "Job Submission Description Language (JSDL) Specification, Version 1.0", 28.07.2008, [налично на: http://www.gridforum.org/documents/GFD.136.pdf, към дата: 19.12.2011]

[5] Atanassov, E., Misev, A. "User level grid quality of service", Large Scale Scientific Computing, Lecture Notes in Computer Science, Volume 5910, pp. 507-514, 2010.

[6] Atanassov, E., Gurov, T., Dimitrov, D., "Job Track Service: Architecture and Features", SEE-GRID-SCI User Forum, 6-11 December 2009, Istanbul, Turkey, pp. 199-202, 2009.

[7] Atanassov, E., Gurov, T., Dimitrov, D., "Salute Grid Application using Message Oriented Middleware", AIP Conference Proceedings, vol. 1186, pp. 183-191, 2009.

[8] Beck-Ratzka A., Zangerl T. "GAT Beginners Guide", 28.02.2009, [налично на: http://www.aei.mpg.de/~alibeck/documents/gat-usage.pdf, към дата: 19.12.2011]

[9] Bellwood, T., Capell, S., *et al.* "UDDI Version 3.0.2", 2004, [налично на: http://www.uddi.org/pubs/uddi_v3.htm, към дата: 19.12.2011]

[10] Booth, D., Haas, H., *et al.* "Web Service Architecture", 2004, [налично на: http://www.w3.org/TR/ws-arch/, към дата: 19.12.2011]

[11] Boyanov, L., Nenkova, P. "On the employment of LCG GRID middleware", 2005, International Conference on Computer Systems and Technologies, CompSysTech' 2005, [налично на: http://ecet.ecs.ru.acad.bg/cst05/Docs/cp/SII/II.11.pdf, към дата: 19.12.2011]

[12] Bray, T., Paoli, J., *et al.* "Extensible Markup Language (XML) 1.1", 2006, [налично на: http://www.w3.org/TR/2006/REC-xml11-20060816/, към дата: 21.6.2012]

[13] Buelow, H., Das, M., Deb, M., *et al.* "Getting Started with Oracle BPM Suite 11g R1", Packt Publishing, 2010.

[14] Burke, S., Campana, S., Lanciotti, E., Litmaath, M., Lorenzo, P., Miccio, V., Nater, C., Santinelli, R., Sciaba, A. "gLite 3.2 User Guide Manuals Series, Version: 1.4.1", 03.08.2011, [налично на: https://edms.cern.ch/file/722398/1.4/gLite-3-UserGuide.pdf, към дата: 19.12.2011]

[15] Cabellos, L., Campos, I., Fernandez-del-Castillo E., Owsiak, M., Palak, B., Plociennik, M. "Scientific workflow orchestration interoperating HTC and HPC resources", Computer Physics Communications, Volume 182, Issue 4, pp. 890-897, 2011.

[16] Christensen, E., Curbera, F., *et al.* "Web Service Definition Language (WSDL) 1.1", 2001, [налично на: http://www.w3.org/TR/wsdl, към дата: 19.12.2011]

[17] Davenport, T., Short, J. "The New Industrial Engineering: Information Technology and Business Process Redesign" Sloan Management Review, Vol. 31, No. 4., 1990

[18] Erl, T. "Service-Oriented Architecture: Concepts, Technology, and Design", Prentice Hall Publishers, 2005.

[19] Erl, T. "Service-Oriented Architecture: Principles of Service Design", Prentice Hall Publishers, 2007.

[20] Ezenwoye, O., Sadjadi, S., Cary, A., Robinson, M. "Orchestrating WSRF-based Grid Services", Technical Report FIU-SCIS-2007-04-01, 2007.

[21] Falzone A. "Portals and Workflows: GENIUS and TRIANA", 2006, [налично на: http://www.bioinfogrid.eu/project-events/initial-training-course/presentations/14_falzone.pdf, към дата: 19.12.2011]

[22] Fielding, R., Gettys, J., *et al.*, "Hypertext Transfer Protocol (HTTP) v1.1", 1999 [налично на: http://www.w3.org/Protocols/rfc2616/rfc2616.html, към дата: 19.12.2011]

[23] Foster, I., Kesselman, C. "The Grid: Blueprint for a New Computing Infrastructure. First Edition", Morgan Kaufmann Publishers, 1998.

[24] Foster, I., Kesselman, C., Tuecke, S. "The Anatomy of the Grid Enabling Scalable Virtual Organizations", 2001, [налично на: http://www.globus.org/alliance/publications/papers/anatomy.pdf, към дата: 19.12.2011]

[25] Foster, I. "What is the Grid? A Three Point Checklist", 2002, [налично на: http://dlib.cs.odu.edu/WhatIsTheGrid.pdf, към дата: 19.12.2011]

[26] Foster, I., Kesselman, C., Nick, J., M., Tuecke, S. "The Physiology of the Grid. An Open Grid Service Architecture for Distributed Systems Integration", 2002, [налично на: http://www.globus.org/alliance/publications/papers/ogsa.pdf, към дата: 19.12.2011]

[27] Foster, I., Kishimoto, H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Maciel, F., Siebenlist, F., Subramaniam, R., Treadwell, J., Von Reich, J. "The Open Grid Service Architecture, Version 1.5", 2006, [налично на: http://www.gridforum.org/documents/GFD.80.pdf, към дата: 19.12.2011]

[28] Getting Started with Kepler, 12.05.2008 [налично на: https://code.kepler-project.org/code/kepler-docs/trunk/outreach/documentation/shipping/getting-started-guide.pdf, към дата: 19.12.2011]

[29] Glatard, T., Montagnat, J., Lingrand, D., Pennec, X. "Flexible and efficient workflow deployement of data-intensive applications on Grids with MOTEUR" (special issue: Special Issue on Workflows Systems in Grid Environments) in International Journal of High Performance Computing Applications (IJHPCA), 22 (3), pp. 347-360, 2008.

[30] Goranova, R. D., "Framework for service composition in g-Lite", American Institute of Physics, Conference Proceedings Volume 1404, ISBN 978-0-7354-0976-7, ISSN 0094-243X, 2011, pp. 218-224.

[31] Goranova, R. D., "Service composition tools in g-Lite", Conference Proceedings of the Fifth International Conference ISGT, ISSN 1314-4855, 2011, pp. 228-235.

[32] Goranova, R. D., Dimitrov, V.T., "An Approach to Developing Business Processes With Web Services in Grid", Conference Proceedings of the 4th International Conference "Distributed Computing and Grid Technologies in Science and Education", ISBN 978-5-9530-0269-1, 2011, pp. 114-120.

[33] Goranova, R. D., "Development of ROOT Services for Grid", American Institute of Physics, Conference Proceedings Volume 1301, ISBN 978-0-7354-0856-2, ISSN 0094-243X, 2010, pp. 661-668.

[34] Gudgin, M., Hadley, M., *et al.* "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", 2007, [налично на: http://www.w3.org/TR/soap12-part1/, към дата: 19.12.2011]

[35] High, R., Kinder, S., Graham, S., "IBM’s SOA Foundation An Architectural Introduction and Overview Version 1.0", IBM, 2005, [налично на:

http://download.boulder.ibm.com/ibmdl/pub/software/dw/webservices/ws-soa-whitepaper.pdf, към дата: 19.12.2011]

[36] Holland, R. "TavernaGrid Documentation", 2009, [налично на: https://grid.sara.nl/wiki/images/a/a2/TavernaGrid_docs.pdf, към дата: 19.12.2011]

[37] IBM BPM Overview, [налично на: ftp://ftp.software.ibm.com/software/za/soa/c1040-bpm-overview-v2.0.pdf, към дата: 19.12.2011]

[38] Jackson, A., "Kepler Directors", 28.10.2010 [налично на: http://scilla.man.poznan.pl:8080/confluence/display/euforia/02.+Kepler+Directors , към дата: 19.12.2011]

[39] Johnson D., Meacham K., Kornmayer H. "A Middleware Independent Grid Workflow Builder for Scientific Applications",Proceedings of 5th IEEE International Conference on e-Science Workshops, pp. 86-91, 2009.

[40] Jordan, D., Alves, A., *et al.* "Web Services Business Process Execution Language Version 2.0", 2007, [налично на: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf, към дата: 19.12.2011]

[41] Juric, M. "Business Process Execution Language for Web Services", Packt Publishing, 2006.

[42] Juric, M., *et al.* "WS-BPEL 2.0 for SOA Composite Applications with IBM WebSphere 7", Packt Publishing, 2010.

[43] Kacsuk, P., Farkas, Z. "P-GRADE Portal: A generic workflow system to support user communities", Journal Future Generation Computer Systems Volume 27 Issue 5, pp. 454-465, 2011, [налично на: http://portal.p-grade.hu/download/pgportal.pdf, към дата: 19.12.2011]

[44] Kavantzas, N., Burdett, D., *et al*. "Web Services Choreography Description Language Version 1.0", 2004, [налично на: http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/, към дата: 21.6.2012]

[45] Laure, E., Fisher, S., M., Frohner, A., Grandi, C., Kunszt, P., Krenek, A., Mulmo, O., Pacini, F., Prelz, F., White, J., Barroso, M., Buncic, M., Hemmer F., Di Meglio, A., Edlund, A. "Programming the Grid with gLite", Computational Methods in Science and Technology 12(1), pp. 33-45, 2006, [налично на: http://www.man.poznan.pl/cmst/2005/v_12_1/03Laure.pdf, към дата: 19.12.2011]

[46] Lee, E., Neuendorffer, S., Hylands, C. "Ptolemy II Heterogeneous Concurrent Modeling and Design in Java, Volume 1: Introduction to Ptolemy II", (Chapter 7.

Modeling Markup Language), pp.193-234, 2005, [налично на: http://ptolemy.eecs.berkeley.edu/papers/05/ptIIdesign1-intro/ptIIdesign1-intro.pdf, към дата: 19.12.2011]

[47] Majithia S., Shields M., Taylor I., Wang I. "Triana: A Graphical Web Service Composition and Execution Toolkit", Proceedings of IEEE International Conference on Web Services, pp. 514-521, 2004.

[48] Metery, R., Guillerminet, B. "Interfacing gLite services with the Kepler scientific workflow", Book of Abstracts of 3rd EGEE User Forum (11-15 February 2008), pp. 63, 2008.

[49] Missier, P., Soiland-Reyes, S., *et al.* "Taverna, reloaded", 2010, [налично на: http://www.taverna.org.uk/pages/wp-content/uploads/2010/04/T2Architecture.pdf, към дата: 19.12.2011]

[50] Owsiak, M. "Executing Grid workflows using Serpens", 12.10.2010 [налично на: http://scilla.man.poznan.pl:8080/confluence/display/euforia/03.+Executing+Grid+workflows+using+Serpens, към дата: 19.12.2011]

[51] Owsiak, M., Zok, T., Plociennik M. "Getting Started with Serpens extension for Kepler 2.2.x Working with gLite module", 20.07.2011 [налично на: http://code.kepler-project.org/code/kepler/releases/release-branches/glite-2.2/docs/glite.pdf, към дата: 19.12.2011]

[52] P-GRADE Portal User Manual, Version 2.10, 25.08.2010, [налично на: http://portal.p-grade.hu/manual/user/v210/UsersManualRelease2_10.html, към дата: 19.12.2011]

[53] Pacini, F. "JDL Attributes", 28.10.2003, [налично на: http://www.grid.org.tr/servisler/dokumanlar/JDL_Atributes_DataGrid.pdf, към дата: 19.12.2011]

[54] Prodan, R., Fahringer, T. "Grid Computing Experiment Management, Tool Integration, and Scientific Worklows", LNCS 4340, Springer, 2007.

[55] Reynolds, A., Wright, M. "Oracle SOA Suite 11g R1 Developer's Guide", Packt Publishing, 2010.

[56] Silver, B. "A first look at Oracle BPM Suite 11G", 2010, [налично на: http://www.oracle.com/us/corporate/analystreports/infrastructure/bruce-silver-bpm-suite-11g-083869.pdf, към дата: 19.12.2011]

[57] Sotomayor, B., Childers, L. "Globus® Toolkit 4 Programming Java Services", Elsevier, 2006.

[58] Spiga, D., Cinquilli, M., *et al.* "Automation of user analysis workflow in CMS", Journal of Physics: Conference Series, Volume 219 (072019), 2010.

[59] Tan, K., Turner, K. "Orchestrating Grid Services using BPEL and Globus Toolkit 4", PGNET 2006: Proceedings of the 7th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, pp. 31-36, 2006.

[60] Tan, W., Missier, P., Madduri, R., Foster, I. "Building Scientific Workflow with Taverna and BPEL: a Comparative Study in caGrid". Lecture Notes in Computer Science, vol. 5472, pp. 118-129, 2009.

[61] Triana User Manual, 2005, [налично на: http://www.gridlab.org/WorkPackages/wp-3/D3.8.pdf, към дата: 19.12.2011]

[62] Tuecke, S., Foster, I., Frey, J., Graham, S., Kesselman, C., Maquire, T., Sandholm, T., Snelling, D.,Vanderbilt, P. "Open Grid Services Infrastructure (OGSI), Version 1.0", 2003, [налично на: http://globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf, към дата: 19.12.2011]

[63] van der Aalst, M., P., W., ter Hofstede, H., M., A., Weske, M. "Business Process Management: A Survey", 2003, [налично на: http://bpt.hpi.uni-potsdam.de/pub/Public/PaperArchive/bpm2003.pdf, към дата: 19.12.2011]

[64] Wahl, M., Howes, T., *et al.* "Lightweight Directory Access Protocol (v3)", 1997 [налично на: http://www.faqs.org/rfcs/rfc2251.html, към дата: 19.12.2011]

[65] Workflow Management Coalition Terminology & Glossary, 1999, [налично на: http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf, към дата: 19.12.2011]

Уеб източници на програмни средства

[66] Apache Axis 2 / Java 1.6.1 [налично на: http://axis.apache.org/axis2/java/core/index.html, към дата: 19.12.2011]

[67] Apache Tomcat 7.0, [налично на: http://tomcat.apache.org/download-70.cgi, към дата: 19.12.2011]

[68] Eclipse Helios 3.6.2, [налично на: http://www.eclipse.org/downloads/packages/release/helios/sr2, към дата: 19.12.2011]

[69] Fiorano SOA Platform, [налично на: http://www.fiorano.com/products/fsoa/products_fioranosoa.php, към дата: 19.12.2011]

[70] g-Lite BDII V5, [налично на: https://twiki.cern.ch/twiki/bin/view/EGEE/BDII, към дата: 19.12.2011]

[71] g-Lite CREAM CE 3.2.14-1, [налично на: http://glite.cern.ch/glite-CREAM/, към дата: 19.12.2011]

[72] g-Lite LCG CE 3.1.46-0, [налично на: http://glite.cern.ch/lcg-CE/, към дата: 19.12.2011]

[73] R-GMA System Specification Version 6.2.0, [налично на: http://www.r-gma.org/documentation/specification.pdf, към дата: 19.12.2011]

[74] g-Lite SE DPM 1.8.2-3, [налично на: http://glite.cern.ch/glite-SE_dpm_disk/, към дата: 19.12.2011]

[75] g-Lite SE d-Cache 3.2.3-0 [налично на: http://glite.cern.ch/glite-SE_dcache_srm/, към дата: 19.12.2011]

[76] g-Lite WN 3.2.12-1, [налично на: http://glite.cern.ch/glite-WN/, към дата: 19.12.2011]

[77] g-Lite WMS 3.1.32-0, [налично на: http://glite.cern.ch/glite-WMS/, към дата: 19.12.2011]

[78] GridFTP, [налично на: http://www.globus.org/toolkit/docs/5.0/5.0.3/data/gridftp/developer/, към дата: 19.12.2011]

[79] GridLab Triana, [налично на: http://www.gridlab.org/WorkPackages/wp-3/index.html, към дата: 19.12.2011]

[80] Grid Security Infrastructure (GSI), [налично на: http://www.globus.org/security/overview.html, към дата: 19.12.2011]

[81] IBM SOA Foundations – WebSphere Software V7.0, [налично на: http://www-01.ibm.com/software/solutions/soa/offerings.html, към дата: 19.12.2011]

[82] Java SE 5.0, [налично на: http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase5-419410.html, към дата: 19.12.2011]

[83] JBoss Enterprise SOA Platform, [налично на: http://www.jboss.com/products/platforms/soa/, към дата: 19.12.2011]

[84] MySQL 4.1.22, [налично на: http://dev.mysql.com/doc/refman/4.1/en/, към дата: 19.12.2011]

[85] Oracle SOA Suite 11g, [налично на: http://www.oracle.com/us/technologies/soa/soa-suite/index.html, към дата: 19.12.2011]

[86] ROOT 5.28, [налично на: http://root.cern.ch/drupal/content/production-version-528, към дата: 19.12.2011]

Авторска справка

В настоящия раздел са описани приносите и публикациите, свързани с дисертацията.

Анотация на получените резултати

Научни и научно-приложни приноси

- Формулирани са основните изисквания към средите за управление на бизнес-процеси, базирани на ориентираната към услуги архитектура;
- Изведен е обобщен модел на среда за управление на бизнес-процеси, базиран на ориентираната към услуги архитектура;
- Изследвани са и са анализирани съществуващи среди за проектиране и изпълнение на бизнес-процеси в EGI, на базата на формулираните изисквания;
- Предложен е модел на среда за управление на бизнес-процеси в EGI;
- Предложено е архитектурно решение на среда за управление на бизнес-процеси в EGI;
- Проучени са съществуващите услуги в грид инфраструктурата EGI. Установена е необходимост от създаване на адаптери за достъп до услугите на грид инфраструктурата.
- Демонстрирана е концепция за прилагане на ориентирана към услуги архитектура в грид;

Приложни приноси

- Проектиран и разработен е метод за регистриране на Уеб услуги от информационната система на EGI в модул Регистър на услуги;
- Проектирани и реализирани са адаптери за достъп до услуги на грид инфраструктурата EGI;
- Проектирани и реализирани са услуги за достъп до компоненти от софтуер на Физиката на високите енергии, по специално ROOT;
- Проектирани и реализирани са примерни бизнес-процеси, които се изпълняват в грид инфраструктурата EGI;
- Демонстрирани са възможности за приложение и използване на средата за управление на бизнес-процеси в EGI;
- Демонстрирано е приложение на средата за създаване и изпълнение на примерен бизнес-процес за научен експеримент от Физиката на високите енергии;

Публикации

Основните резултати от настоящият труд са публикувани в рецензирани издания и са докладвани на следните научни форуми:

1. Goranova, R. D., Framework for service composition in g-Lite, American Institute of Physics, Conference Proceedings Volume 1404, ISBN 978-0-7354-0976-7, ISSN 0094-243X, 2011, pp. 218-224.

   Работата е докладвана на международната научна конференция "Application of Mathematics in Technical and Natural Sciences (AMiTaNS'11)", Албена, България, 2011.

2. Goranova R. D., "Service composition tools in g-Lite", Conference Proceedings of the Fifth International Conference ISGT, ISSN 1314-4855, 2011, pp. 228-235.

   Работата е докладвана на международната научна конференция "Information Systems and Grid Technologies (ISGT'11)", София, България, 2011.

3. Goranova, R. D., Dimitrov V.T., An Approach to Developing Business Processes With Web Services in Grid, Conference Proceedings of the 4th International Conference "Distributed Computing and Grid Technologies in Science and Education", ISBN 978-5-9530-0269-1, 2010, pp. 114-120.

   Работата е докладвана на международната научна конференция "Distributed Computing and Grid Technologies in Science and Education (GRID'10)", Дубна, Русия, 2010.

4. Goranova, R. D., Development of ROOT Services for Grid, American Institute of Physics, Conference Proceedings Volume 1301, ISBN 978-0-7354-0856-2, ISSN 0094-243X, 2010, pp. 661-668.

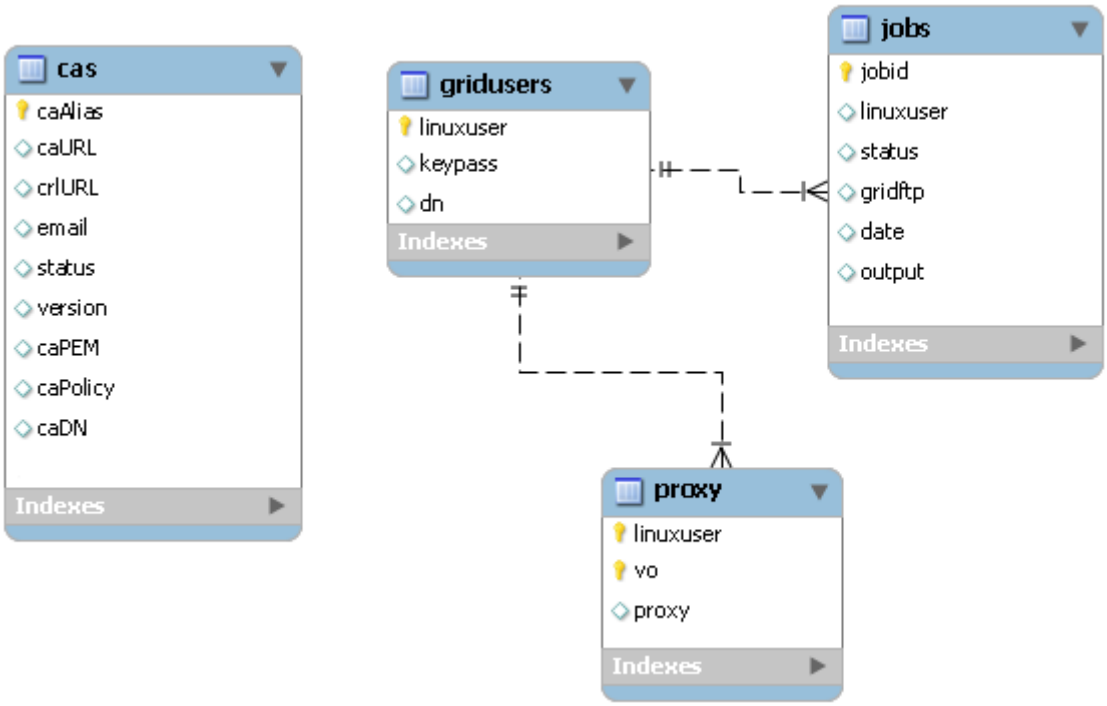
   Работата е докладвана на международната научна конференция "Application of Mathematics in Technical and Natural Sciences (AMiTaNS'10)", Созопол, България, 2010.

5. Горанова, Р. Д., "Обзор върху композиране на процеси от физиката на високите енергии в грид", Списание на Софийския Университет за електронно обучение, ISSN 1314-0086, Бр.4, 2010

Публикацията е обзорна.

Приложения

Приложение 1 - Релационен модел на данни



Име на колона	Тип на колона	Ограничение	Описание
linuxuser	VARCHAR(50)	PK, NOT NULL	Име на потребител - име на потребителски акаунт на операционната система SLC4, на която е инсталиран приложния сървър
keypass	VARCHAR(32)	NOT NULL	Парола за ключа на потребителския грид сертификата. Паролата се съхранява кодирана в базата от данни
dn	VARCHAR(250)	NOT NULL	Отличително име (Distinguished Name). Уникален за всеки грид-потребител. Съдържа се в грид сертификата.

Таблица 17. Описание на таблицата Gridusers

Име на колона	Тип на колона	Ограничение	Описание
linuxuser	VARCHAR(50)	PK, FK, NOT NULL	Име на потребител - име на потребителски акаунт на операционната система SLC4, на която е инсталиран приложния сървър
vo	VARCHAR(50)	PK, NOT NULL	Име на виртуална организация, към която принадлежи потребителя
proxy	BLOB		Прокси сертификат на грид-потребителя

Таблица 18. Описание на таблицата Proxy

Име на колона	Тип на колона	Ограничение	Описание
jobid	VARCHAR(150)	PK, NOT NULL	Уникален идентификатор на грид задача
linuxuser	VARCHAR(50)	FK, NOT NULL	Име на потребител - име на потребителски акаунт на операционната система SLC4, на която е инсталиран приложния сървър
status	VARCHAR(15)	NOT NULL	Състояние на задачата (изпратена, стартирана и т.н.)
gridftp	TEXT		GSIFTP адрес на задачата
date	DATETIME	NOT NULL	Дата на изпращане на задачата
output	TEXT		Адрес на директория от приложния сървър, в която се намират изходните файлове на задачата

Таблица 19. Описание на таблицата Jobs

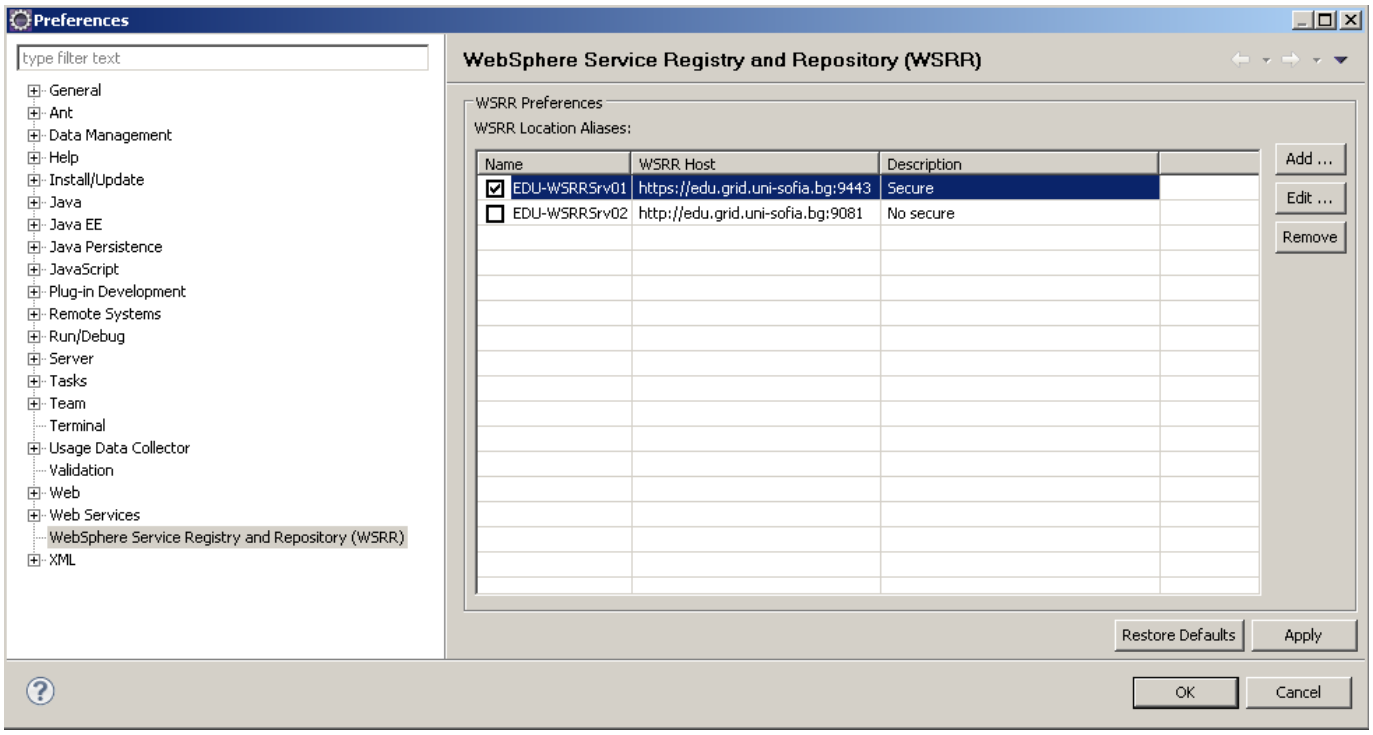
Име на колона	Тип на колона	Ограничение	Описание
caAlias	VARCHAR(50)	PK, NOT NULL	Псевдоним на сертифицираща организация.
caURL	VARCHAR(150)		Адрес на PEM сертификат на сертифициращата организация
crlURL	VARCHAR(150)		Адрес на CRL сертификат на сертифициращата организация
email	VARCHAR(50)		e-mail адрес
status	VARCHAR(50)		Статус
version	DOUBLE		Версия на обновяване
caPEM	LONGBLOB		PEM сертификат на сертифициращата организация
caPolicy	BLOB		Политика за достъп на сертифициращата организация
caDN	VARCHAR(150)		Отличително име (Distinguished Name). Уникален за всяка сертифицираща организация. Съдържа се в сертификата.

Таблица 20. Описание на таблицата CAS

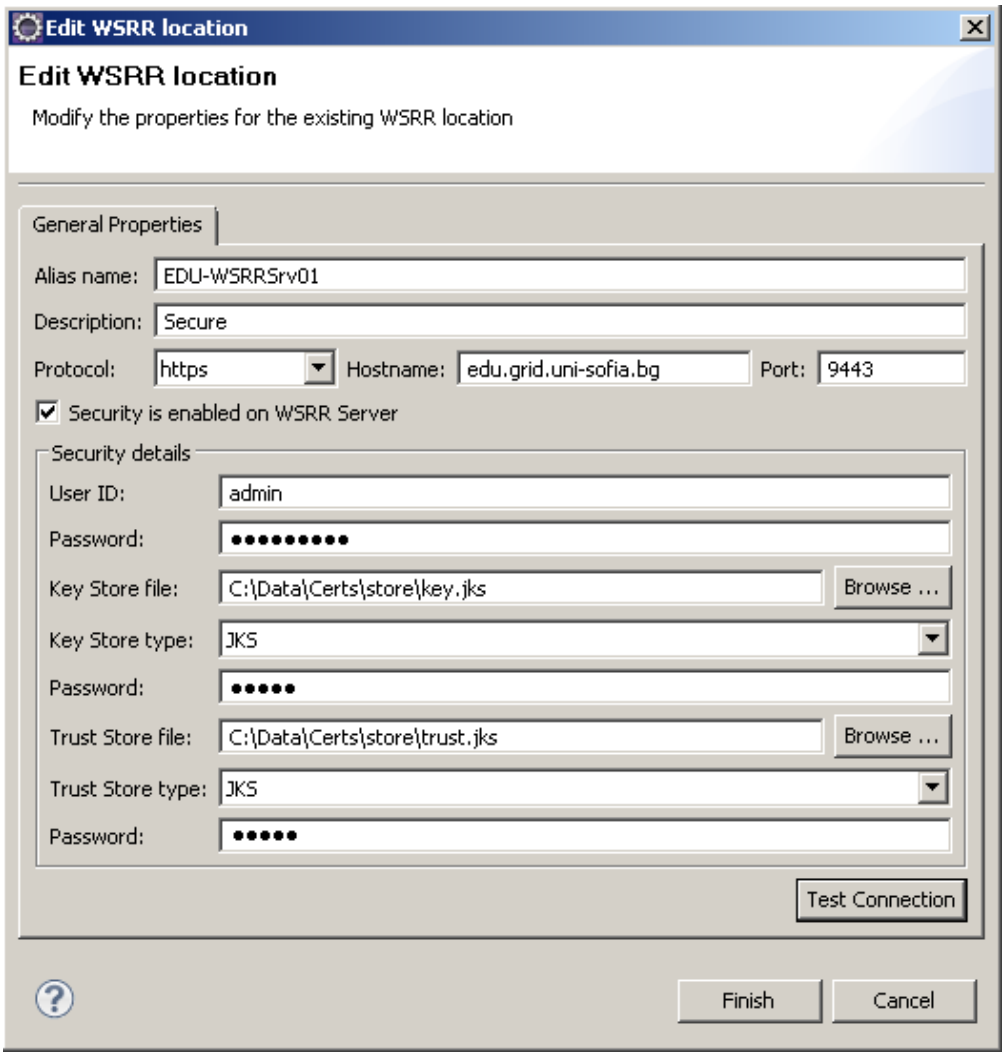
Приложение 2 - Конфигуриране на връзка до WSRR регистър

Конфигуриране на връзката до WSRR регистъра в Eclipse

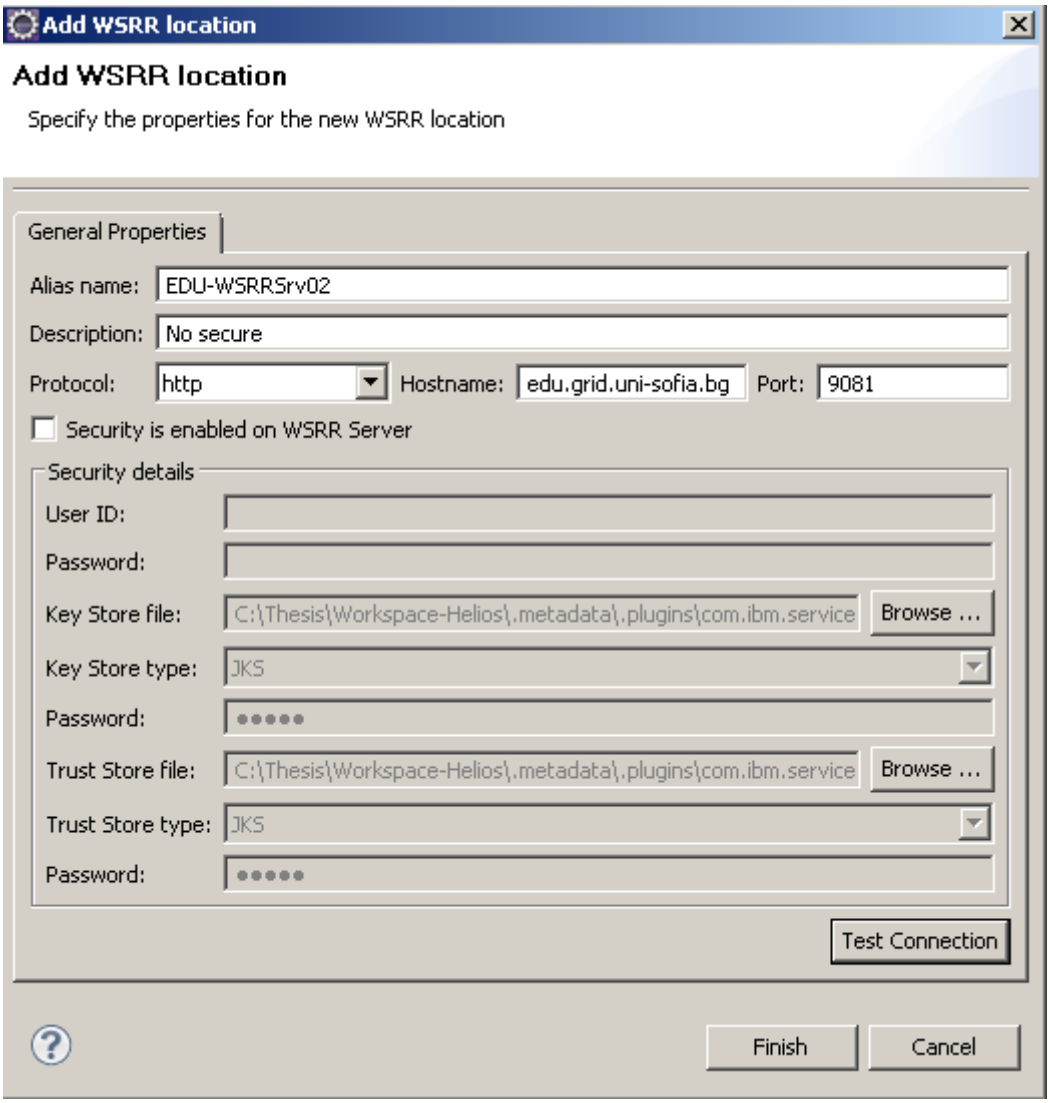
Необходимо условие за конфигуриране на връзката до WSRR регистър е добавката за Eclipse (sa02_non_IBM_JRE.zip) да бъде предварително инсталирана. Добавката е достъпна на адрес: https://www-304.ibm.com/support/docview.wss?uid=swg24013925



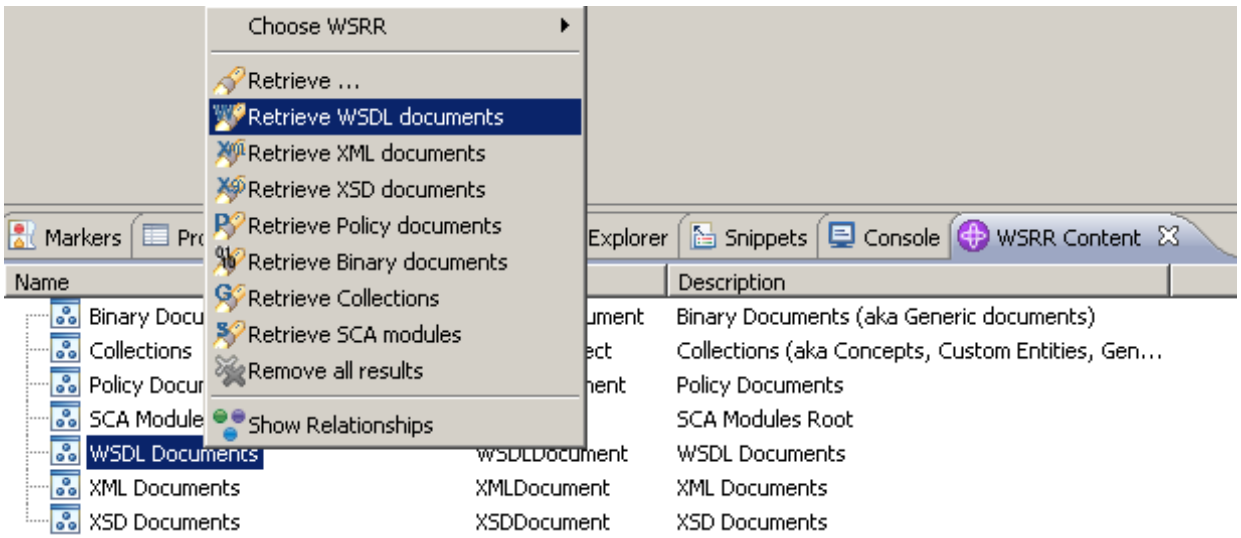
Конфигуриране на сигурна връзката до WSRRSrv01 профил на регистъра в Eclipse



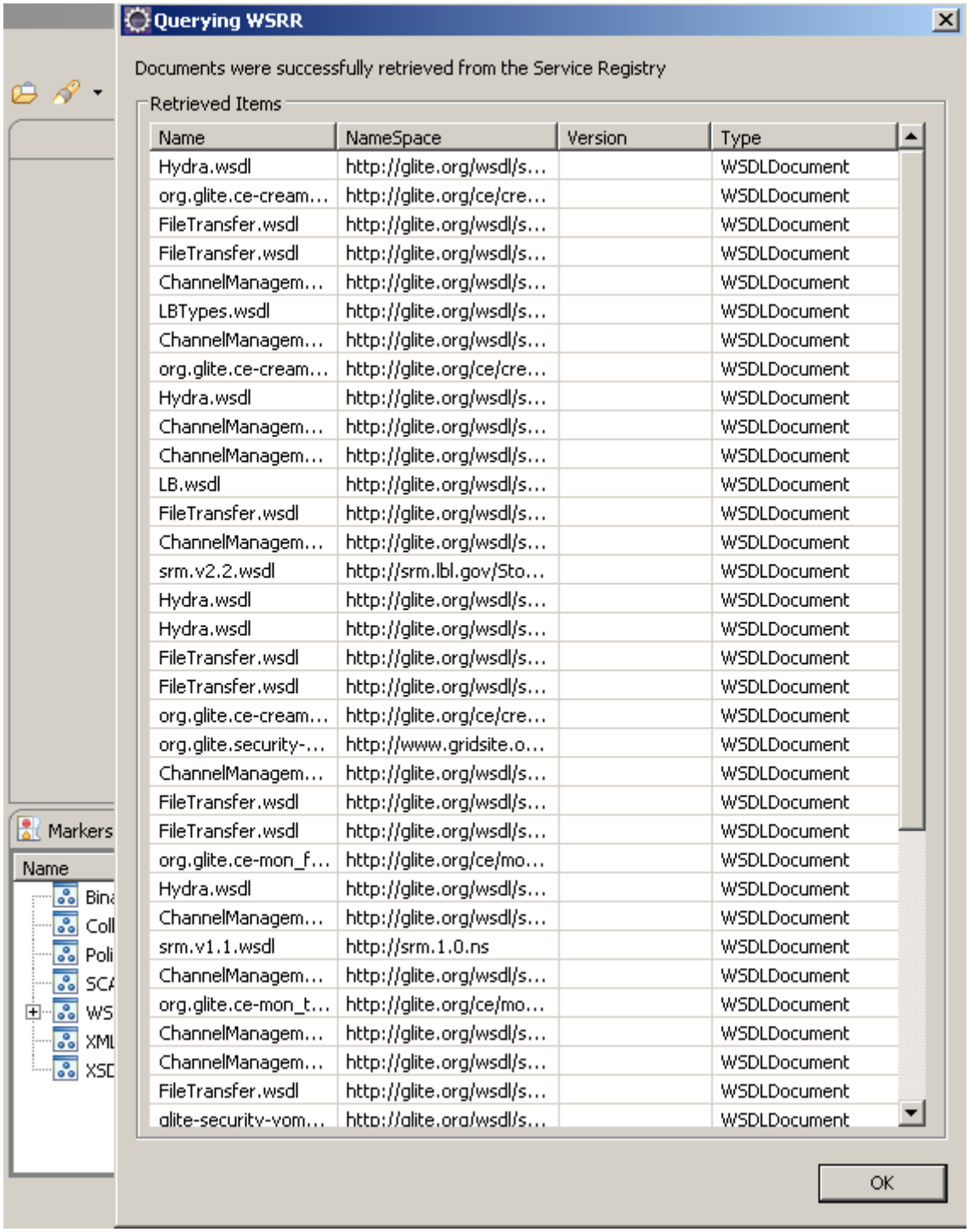
Конфигуриране на връзката до WSRRSrv02 профил на регистъра в Eclipse (без сигурност)



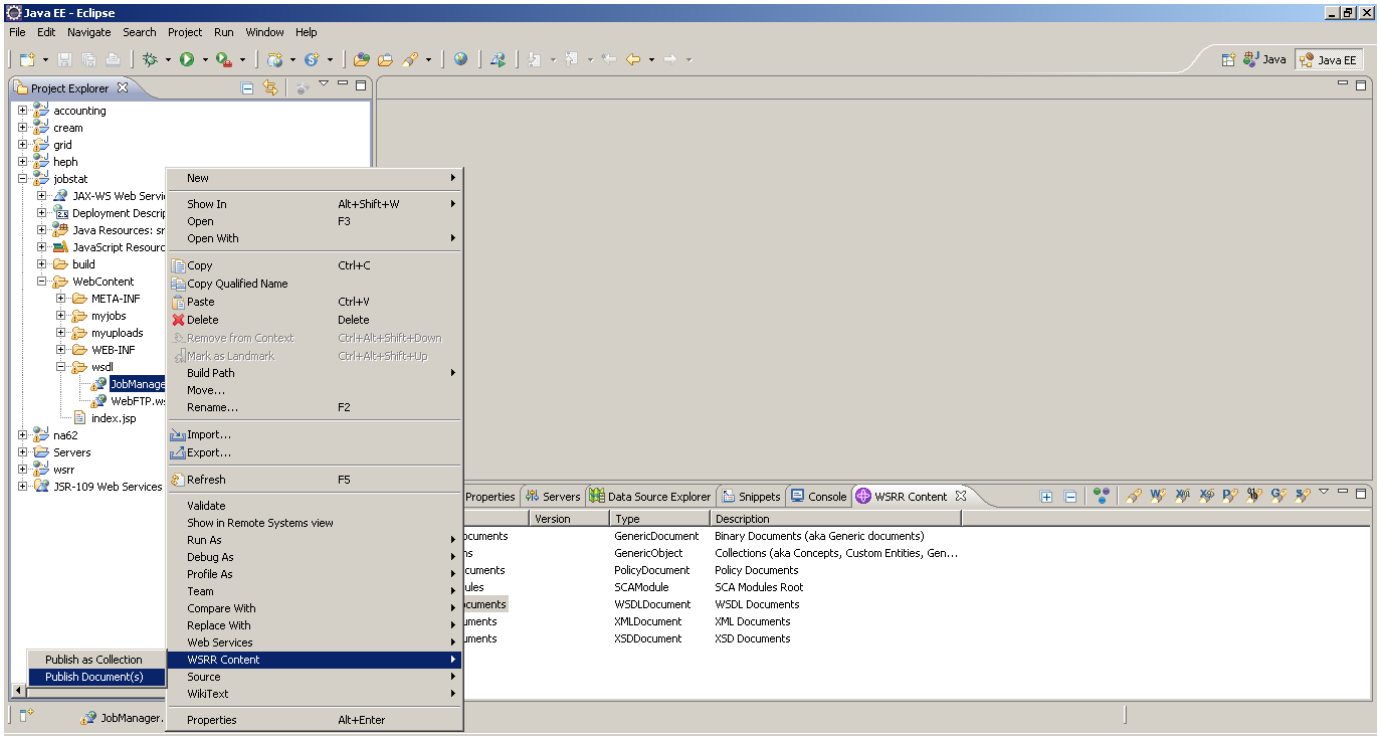
Извличане на наличните WSDL описания на услуги от регистъра в Eclipse

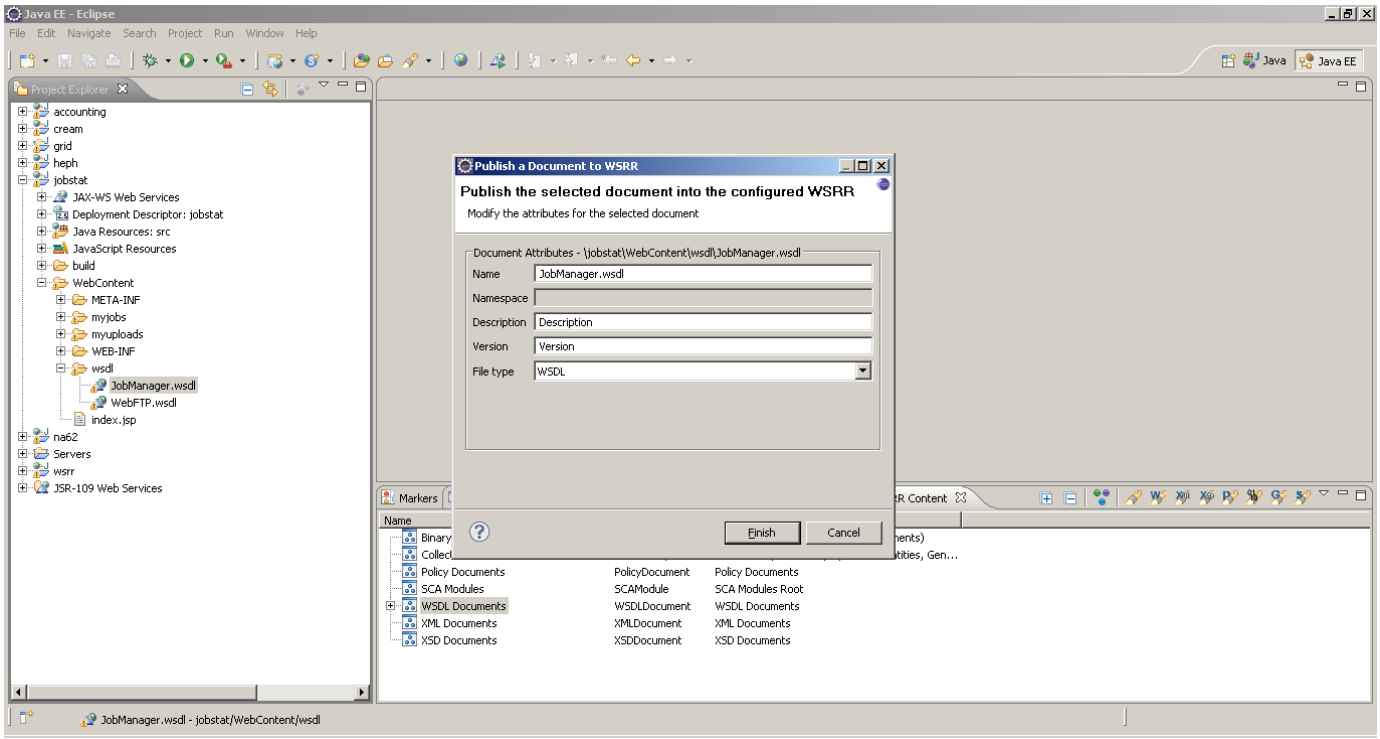


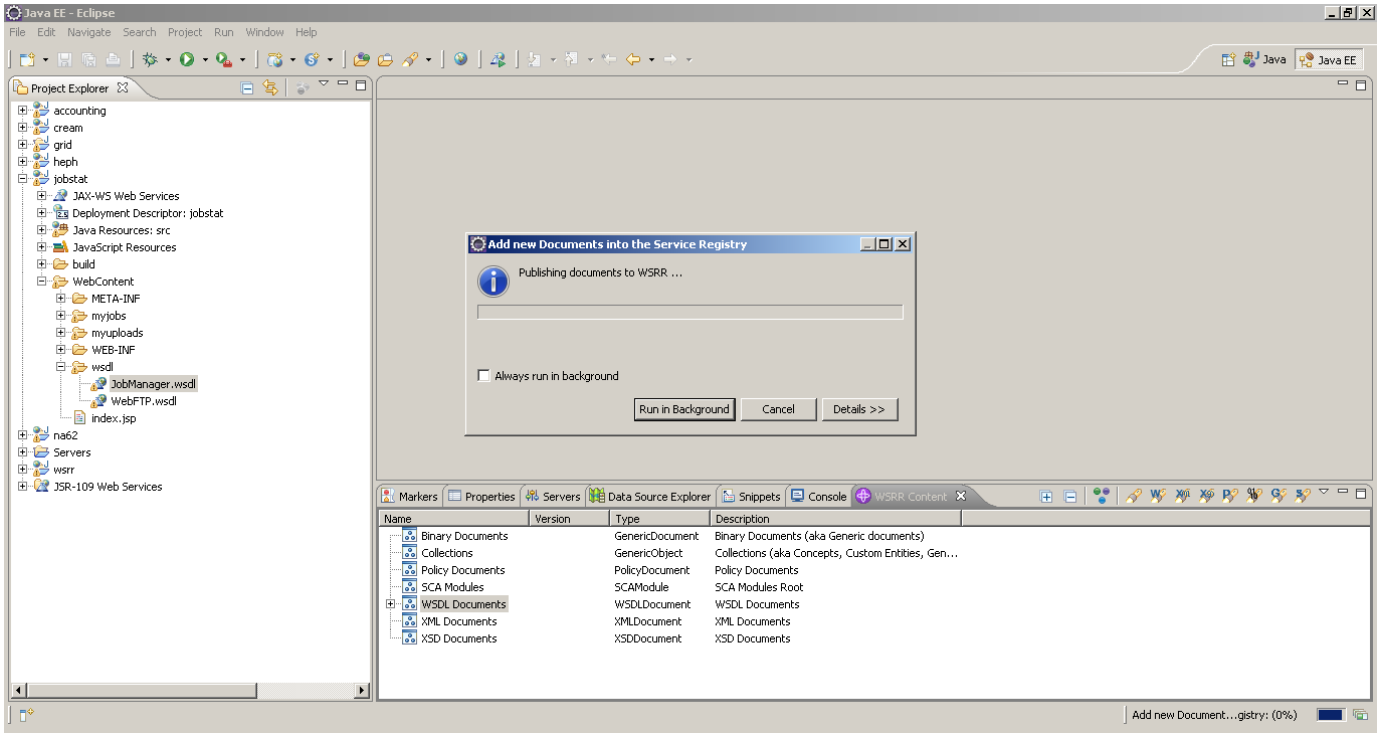
Извлечени WSDL описания на услуги от регистъра в Eclipse

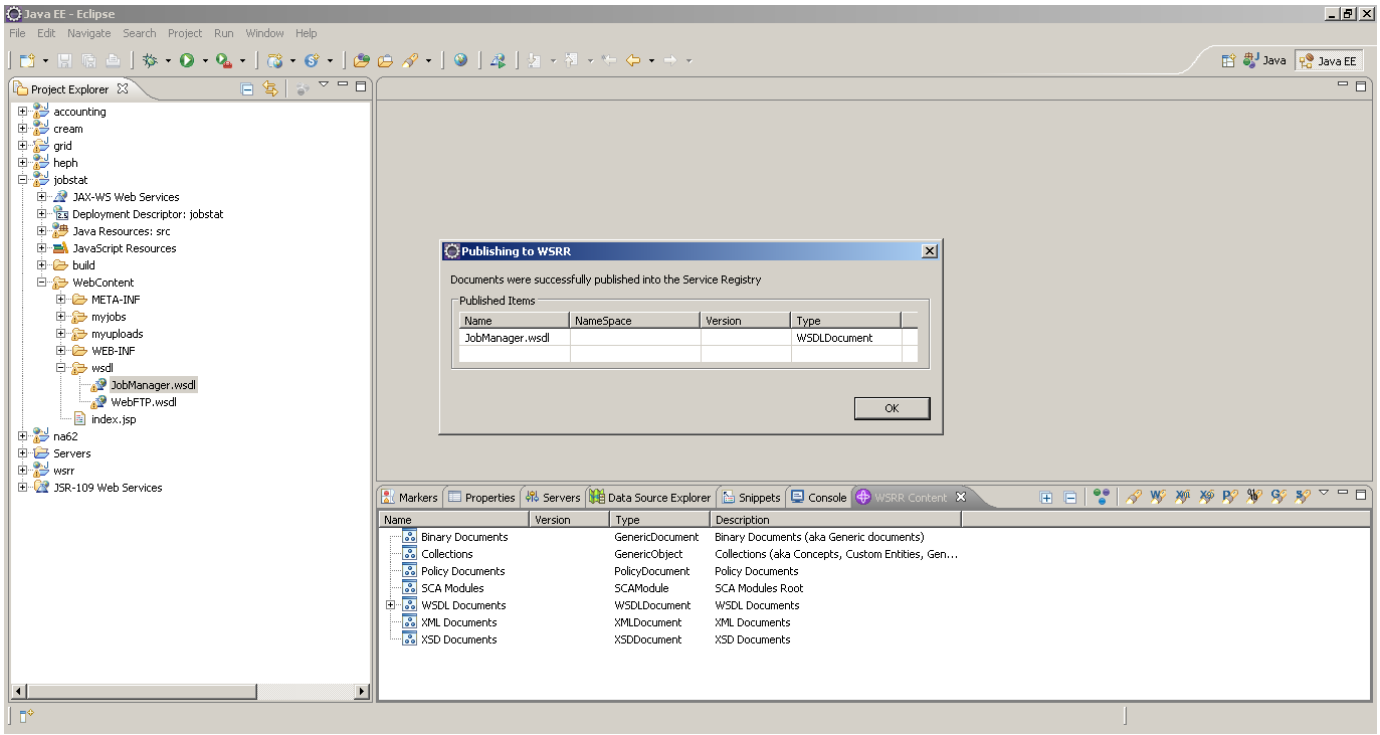


Публикуване на услуга в регистъра от Eclipse



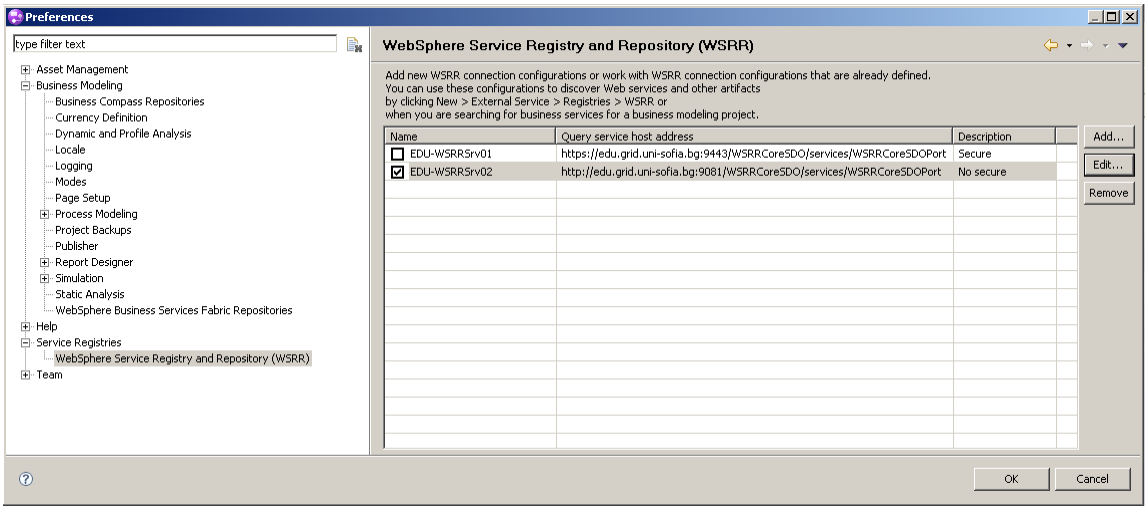



Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer
accounting
cream
grid
heph
jobstat
JAX-WS Web Services
Deployment Descriptor: jobstat
Java Resources: src
JavaScript Resources
build
WebContent
META-INF
myjobs
myuploads
WEB-INF
wsdl
JobManager.wsdl
WebFTP.wsdl
index.jsp
na62
Servers
wsrr
JSR-109 Web Services
Add new Documents into the Service Registry
Publishing documents to WSRR ...
Always run in background
Run in Background
Cancel
Details >>
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
WSRR Content
Name
Version
Type
Description
Binary Documents
GenericDocument
Binary Documents (aka Generic documents)
Collections
GenericObject
Collections (aka Concepts, Custom Entities, Gen...
Policy Documents
PolicyDocument
Policy Documents
SCA Modules
SCAModule
SCA Modules Root
WSDL Documents
WSDLDocument
WSDL Documents
XML Documents
XMLDocument
XML Documents
XSD Documents
XSDDocument
XSD Documents
Add new Document...gistry: (0%)

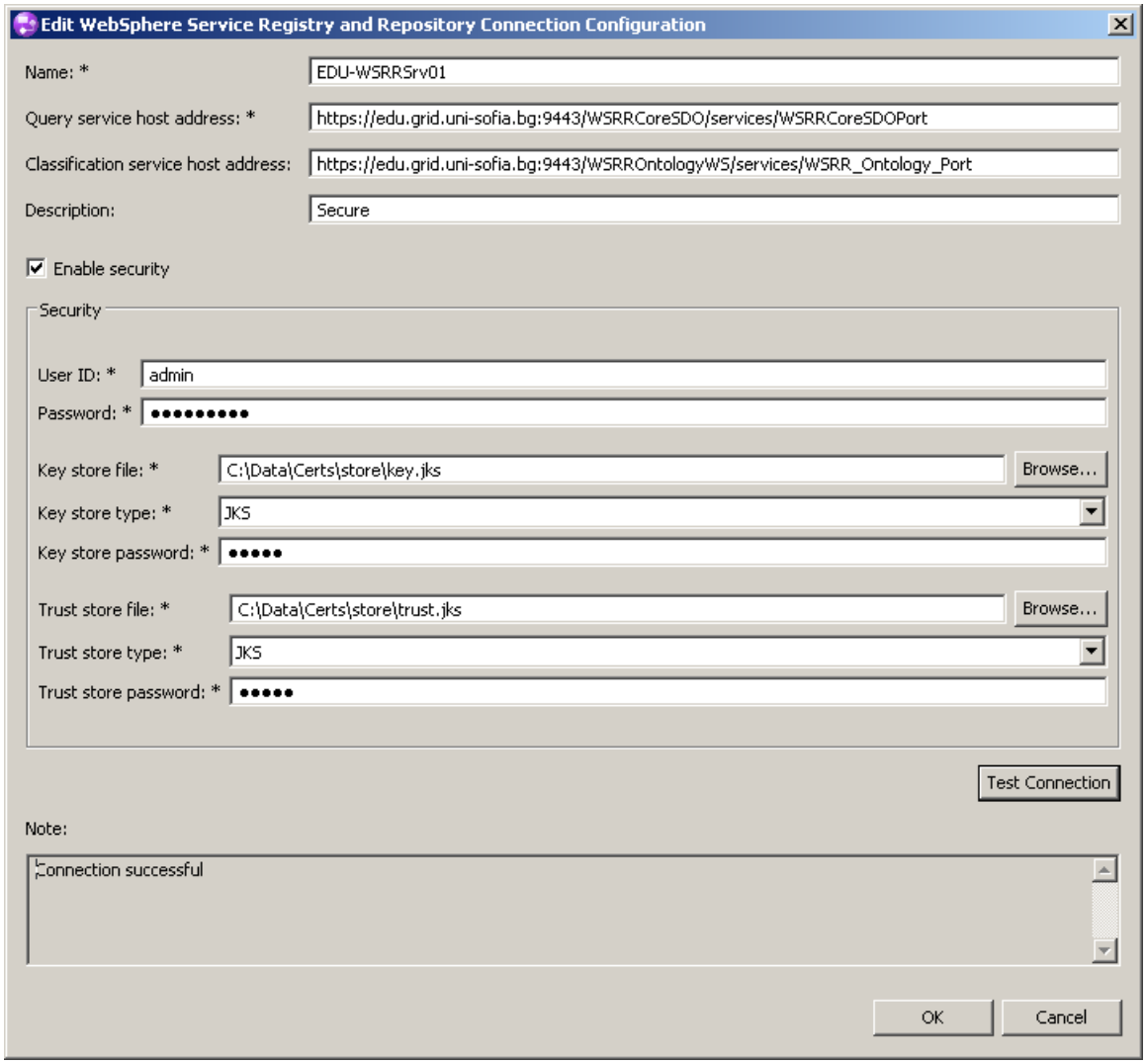

Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer
accounting
cream
grid
heph
jobstat
JAX-WS Web Services
Deployment Descriptor: jobstat
Java Resources: src
JavaScript Resources
build
WebContent
META-INF
myjobs
myuploads
WEB-INF
wsdl
JobManager.wsdl
WebFTP.wsdl
index.jsp
na62
Servers
wsrr
JSR-109 Web Services
Publishing to WSRR
Documents were successfully published into the Service Registry
Published Items
Name
NameSpace
Version
Type
JobManager.wsdl
WSDLDocument
OK
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
WSRR Content
Name
Version
Type
Description
Binary Documents
GenericDocument
Binary Documents (aka Generic documents)
Collections
GenericObject
Collections (aka Concepts, Custom Entities, Gen...
Policy Documents
PolicyDocument
Policy Documents
SCA Modules
SCAModule
SCA Modules Root
WSDL Documents
WSDLDocument
WSDL Documents
XML Documents
XMLDocument
XML Documents
XSD Documents
XSDDocument
XSD Documents

Конфигуриране на връзката до WSRR регистъра в IBM WebSphere Business Modeler

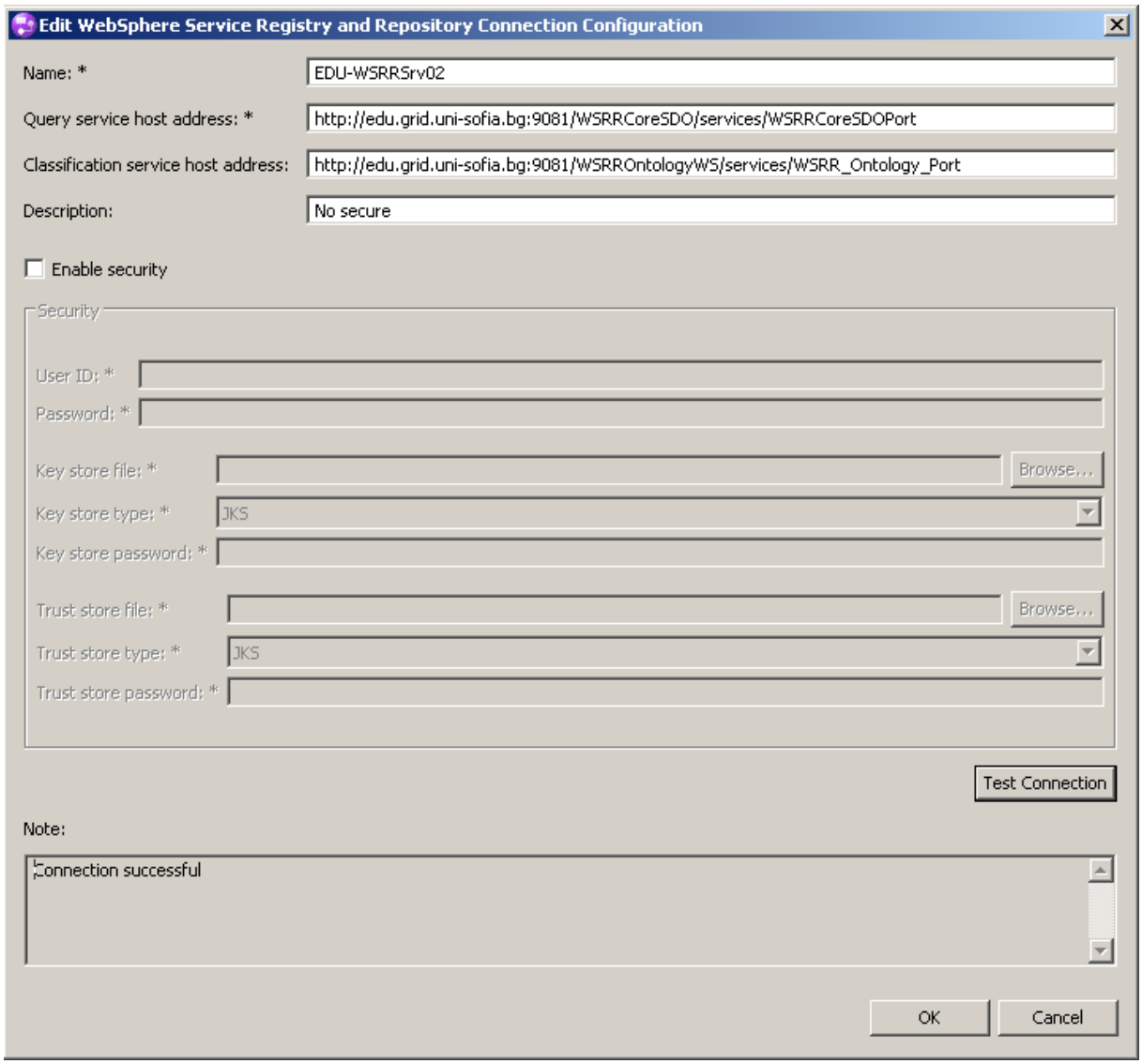
Няма необходимо условие за предварително инсталирана добавка към средата.



Конфигуриране на сигурна връзката до WSRRSrv01 профил на регистъра в WebSphere Business Modeler

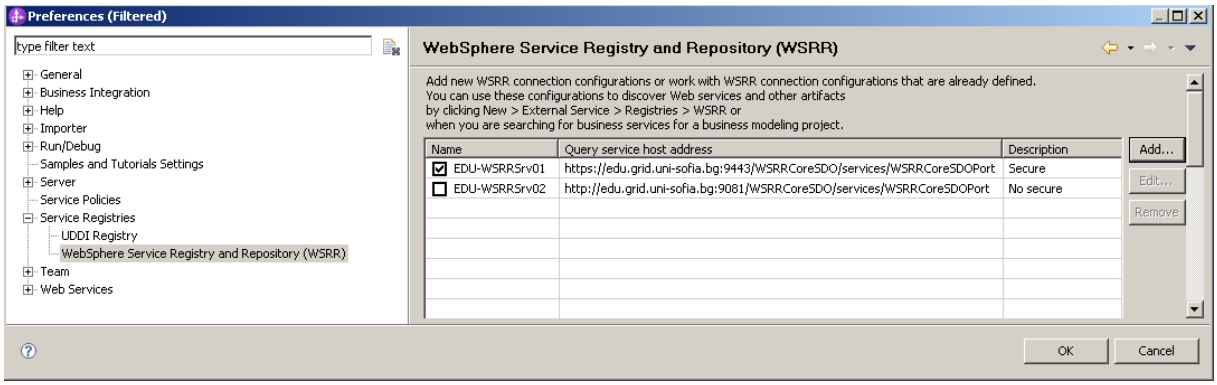


Конфигуриране на връзката до WSRRSrv02 профил на регистъра в WebSphere Business Modeler (без сигурност)

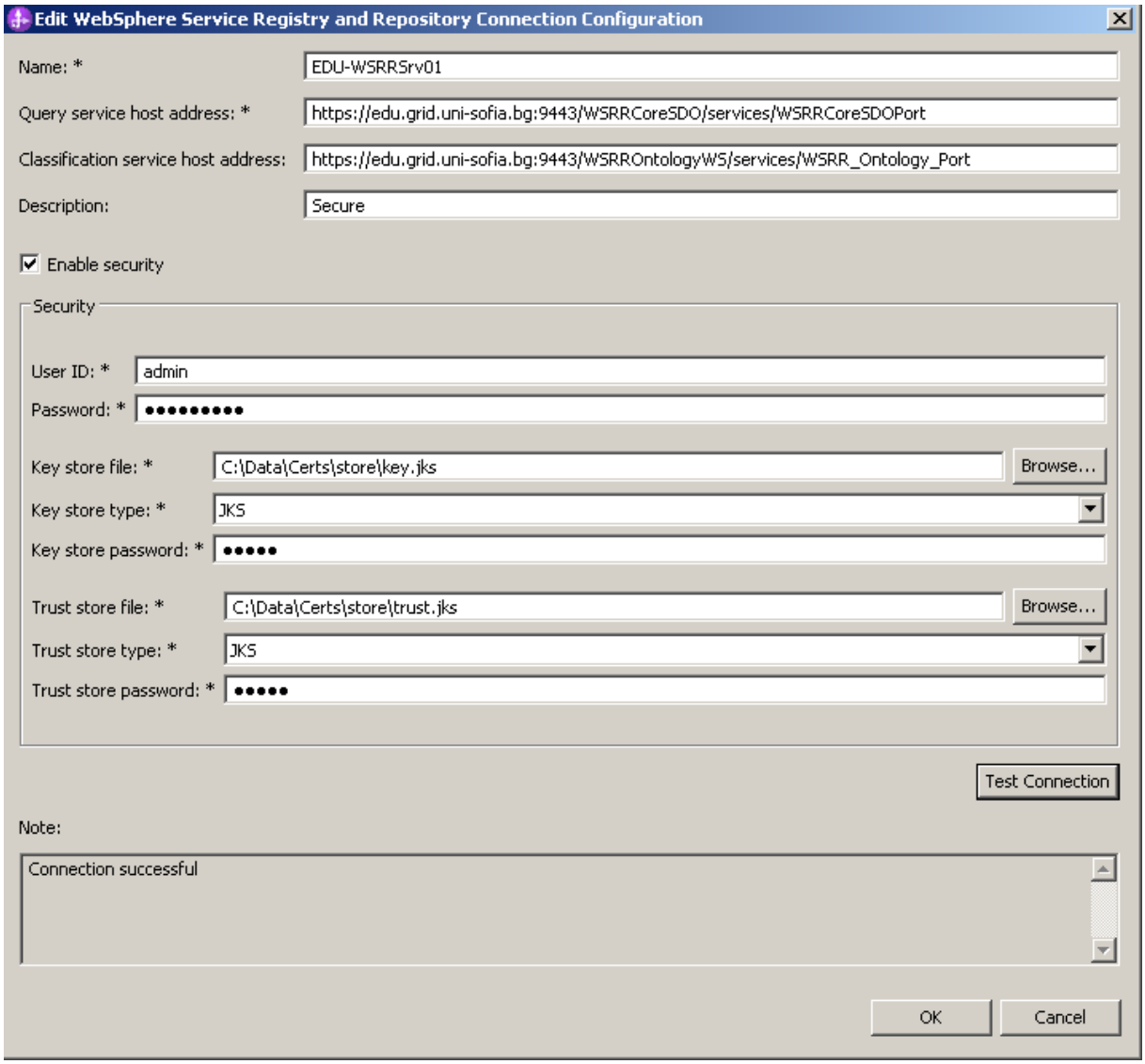


Конфигуриране на връзката до WSRR регистъра в IBM WebSphere Integration Developer

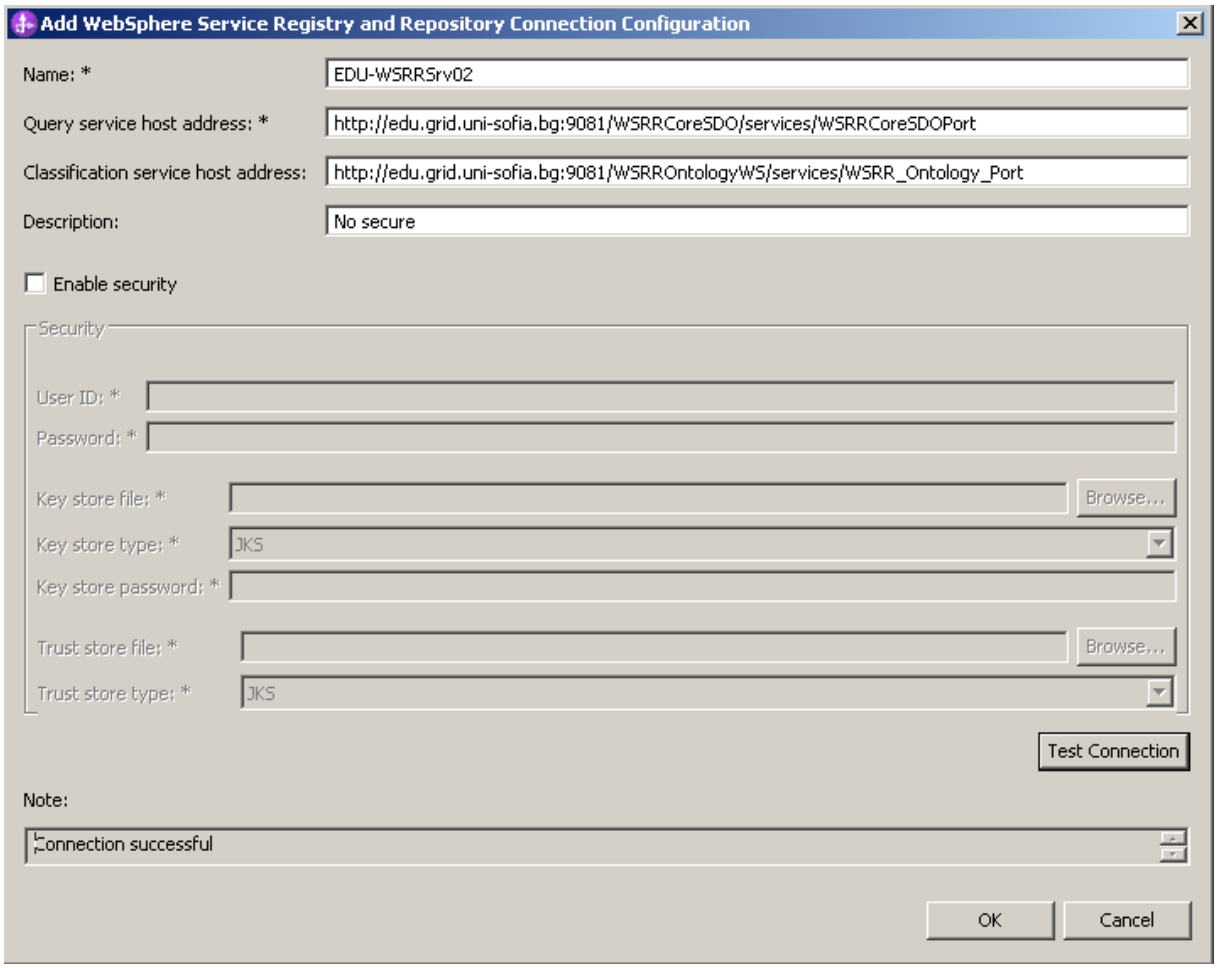
Няма необходимо условие за предварително инсталирана добавка към средата.



Конфигуриране на сигурна връзката до WSRRSrv01 профил на регистъра в WebSphere Integration Developer

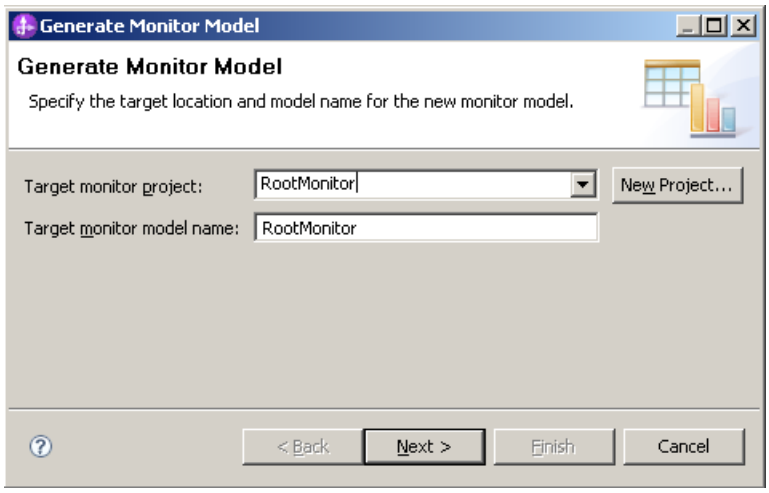


Конфигуриране на връзката до WSRRSrv02 профил на регистъра в WebSphere Integration Developer (без сигурност)

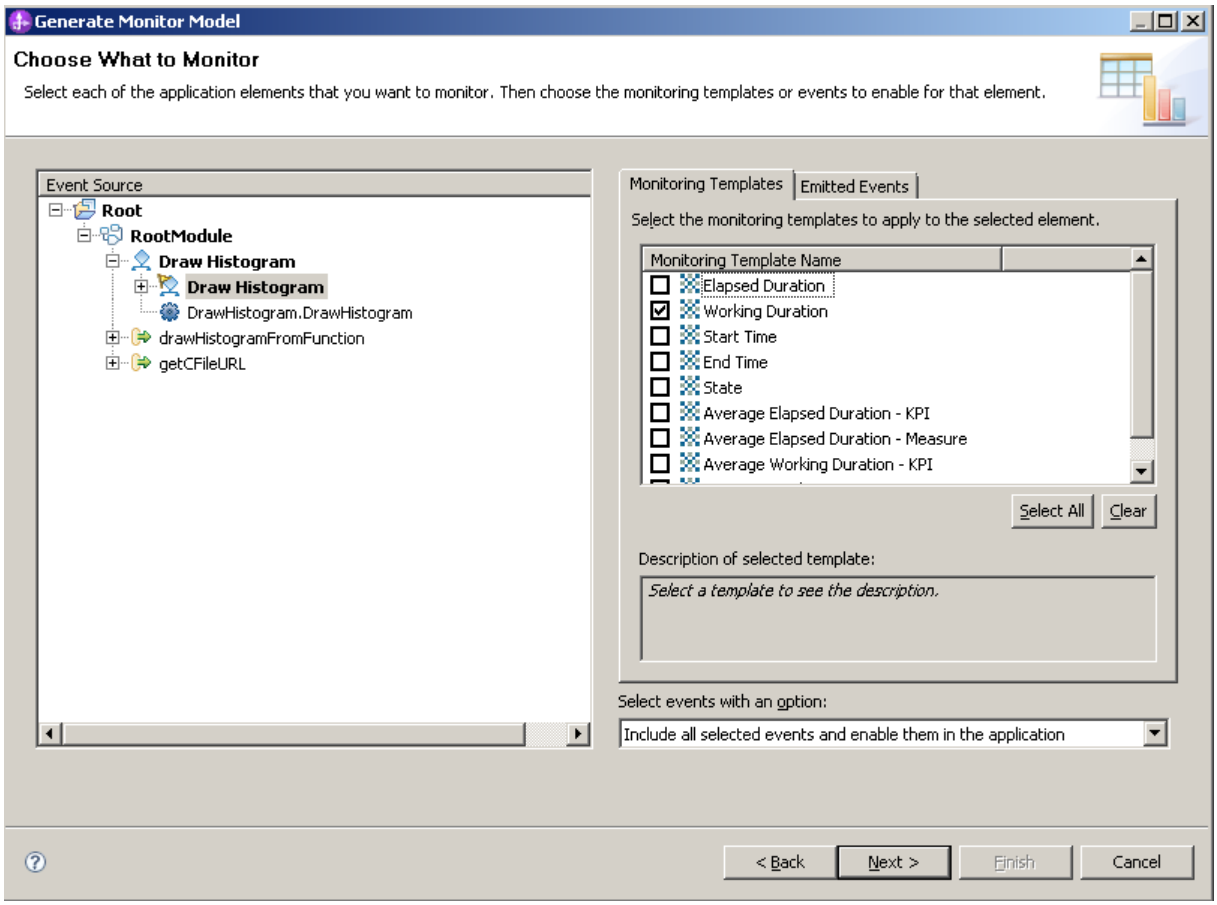


Приложение 3 - Генериране за проект за наблюдение в среда за разгръщане

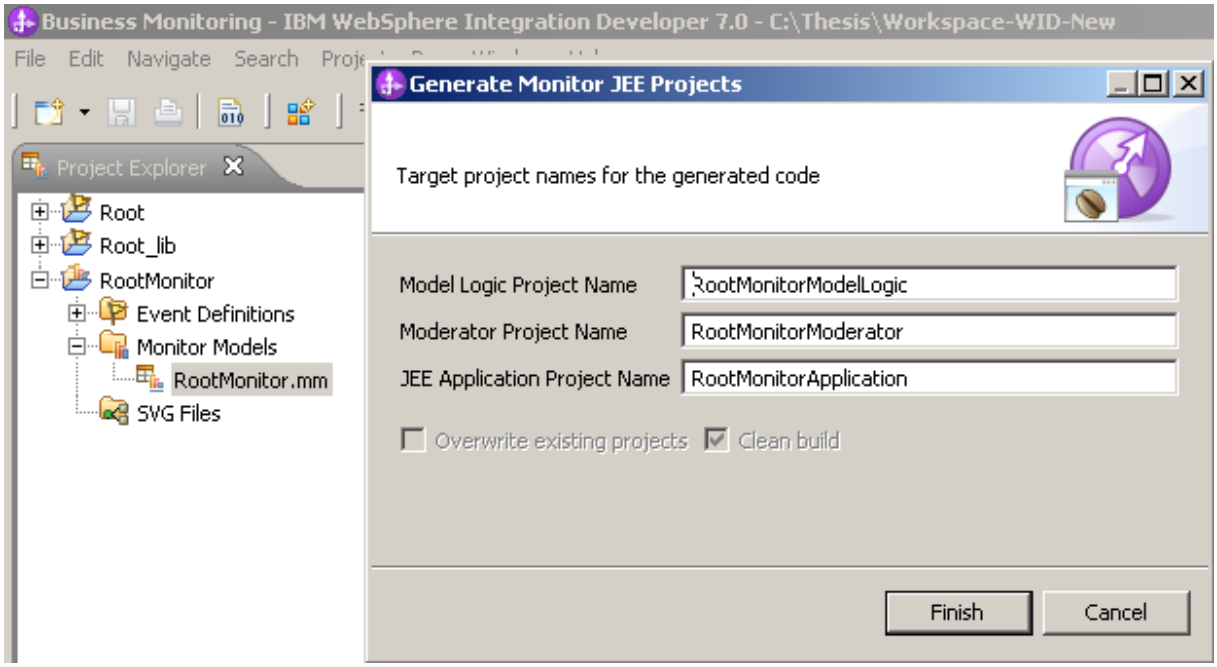
Генериране е на проект за наблюдение



Задаване на елементи за наблюдение



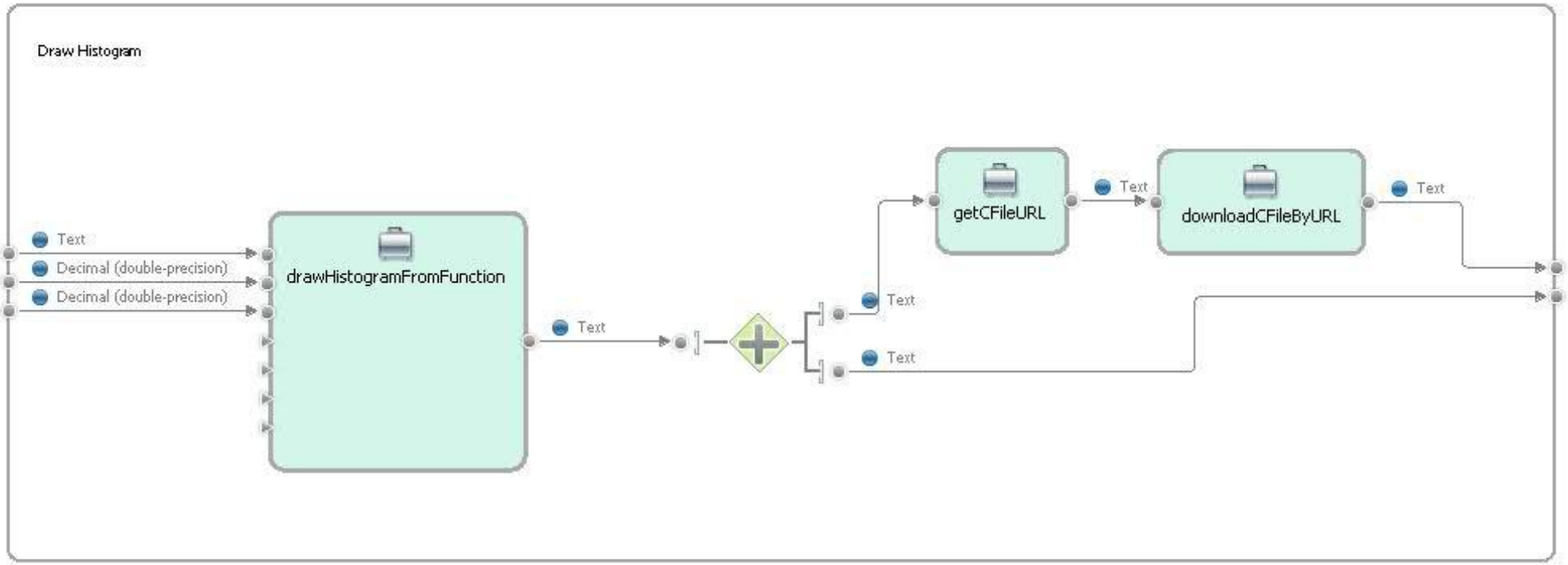
Генериране на Java JEE проект



Приложение 4 - Разработени модели на бизнес-процеси

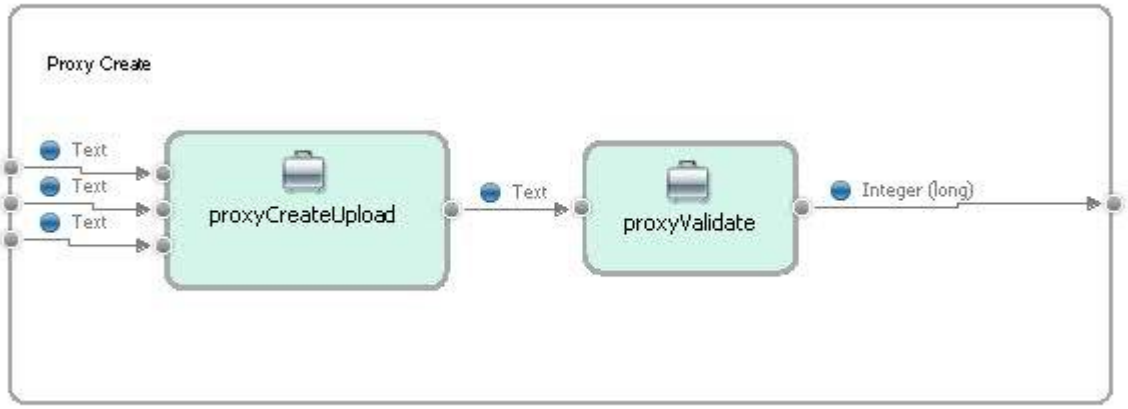
Процес 1: Draw Histograms

Описание: Изчертава хистограма по зададена функция, като резултат извежда URL до картинката и C файл на хистограмата.



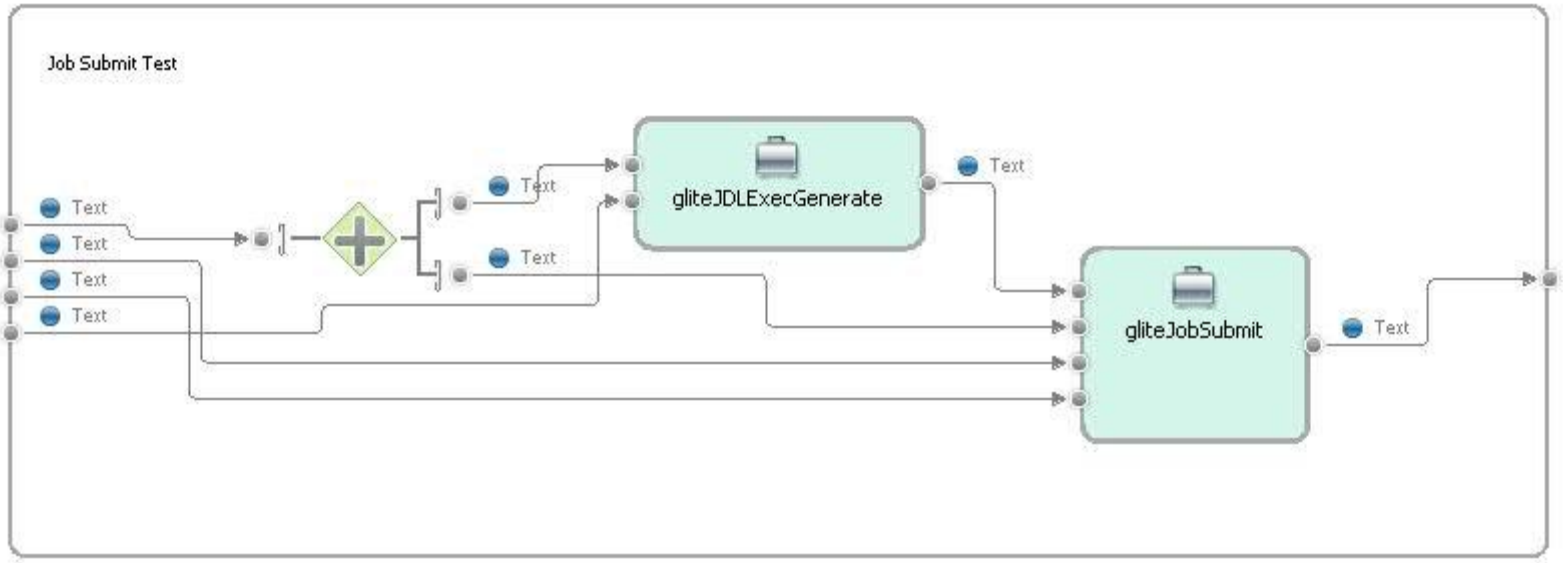
Процес 2: Proxy Create

Описание: По зададени потребителско име, виртуална организация и парола, създава грид прокси за потребителя и го записва в базата от данни. Като резултат връща за колко време е валидно създаденото прокси.



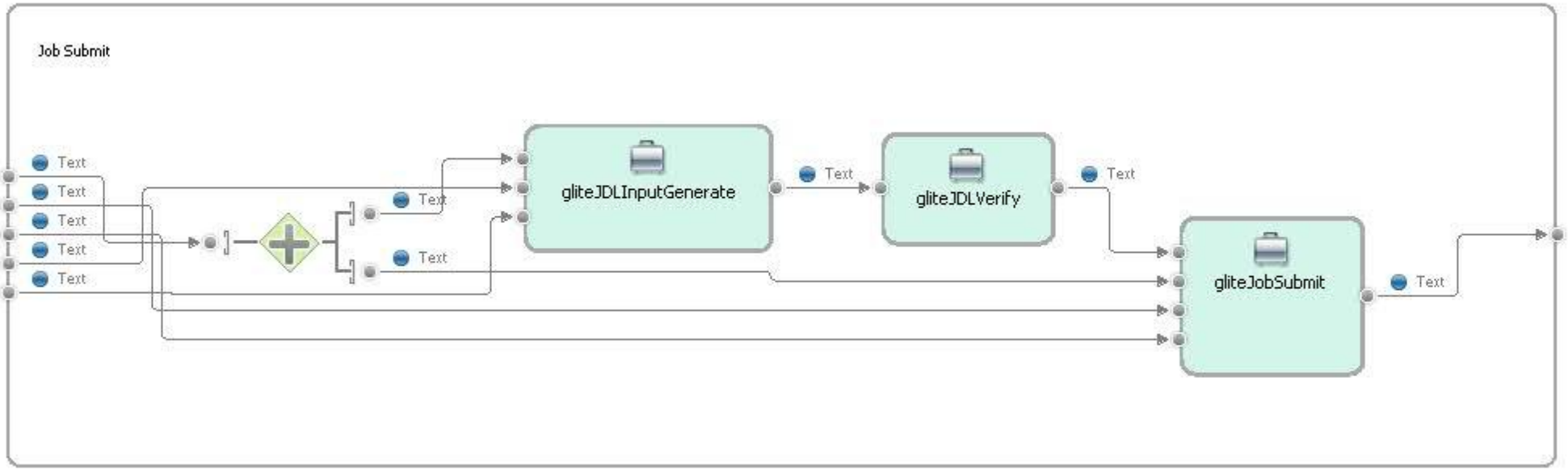
Процес 3: Job Submit Test

Описание: По зададени потребителско име, виртуална организация, парола и изпълним файл, изпраща тестова задача в грид. Като резултат връща уникален идентификатор на задачата (JobID).



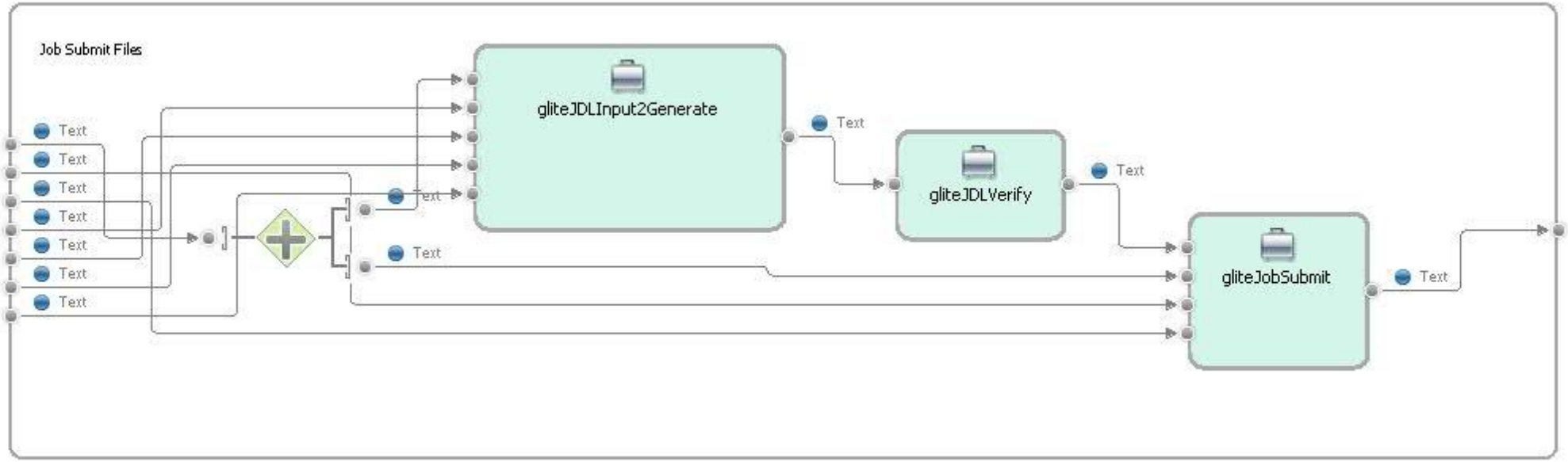
Процес 4: Job Submit

Описание: По зададени потребителско име, виртуална организация, парола, изпълним файл и входни аргументи към изпълнимия файл изпраща задача за изпълнение в грид. Като резултат връща уникален идентификатор на задачата (JobID).



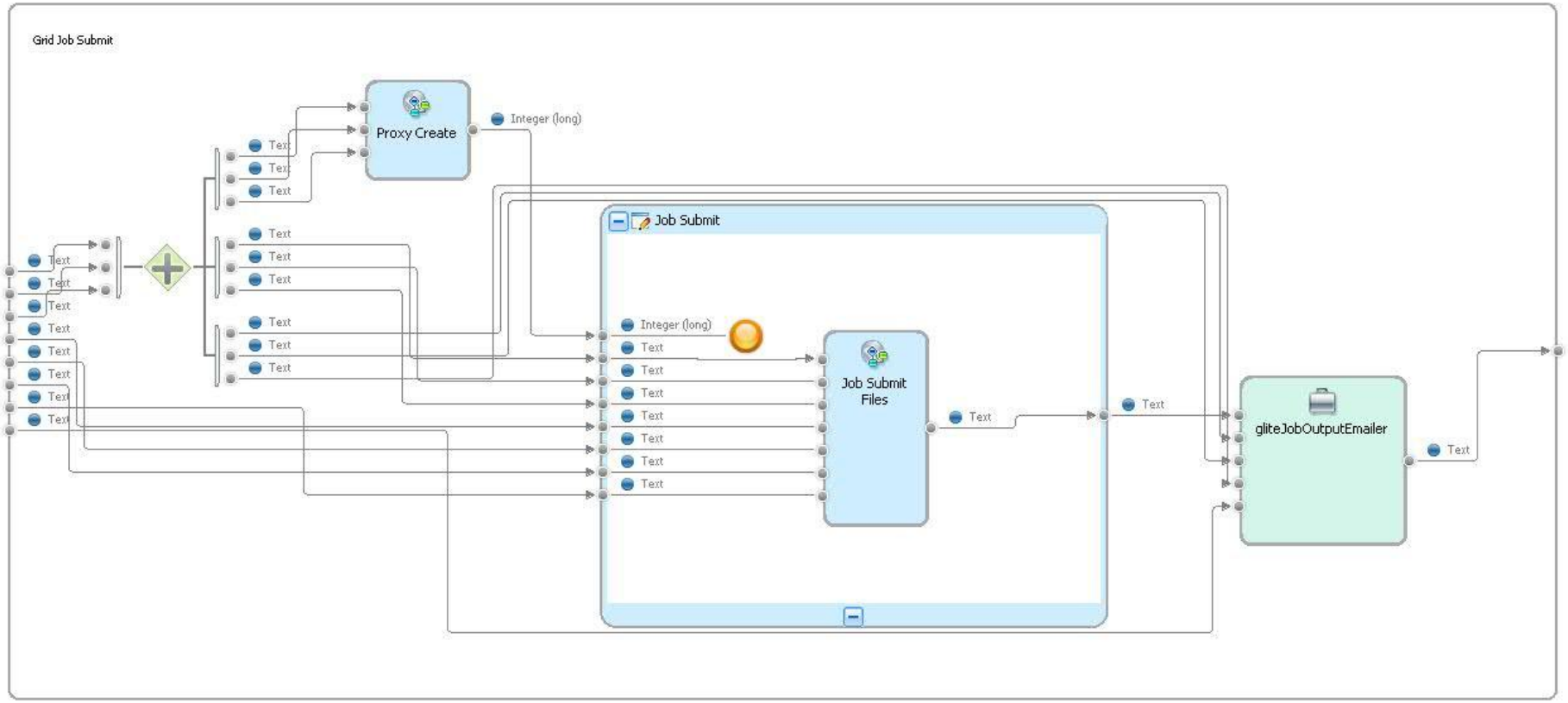
Процес 5: Job Submit Files

Описание: По зададени потребителско име, виртуална организация, парола, изпълним файл, имена на файлове обработвани в изпълнимия файл и входни параметри изпраща задача за изпълнение в грид. Като резултат връща уникален идентификатор на задачата (JobID).



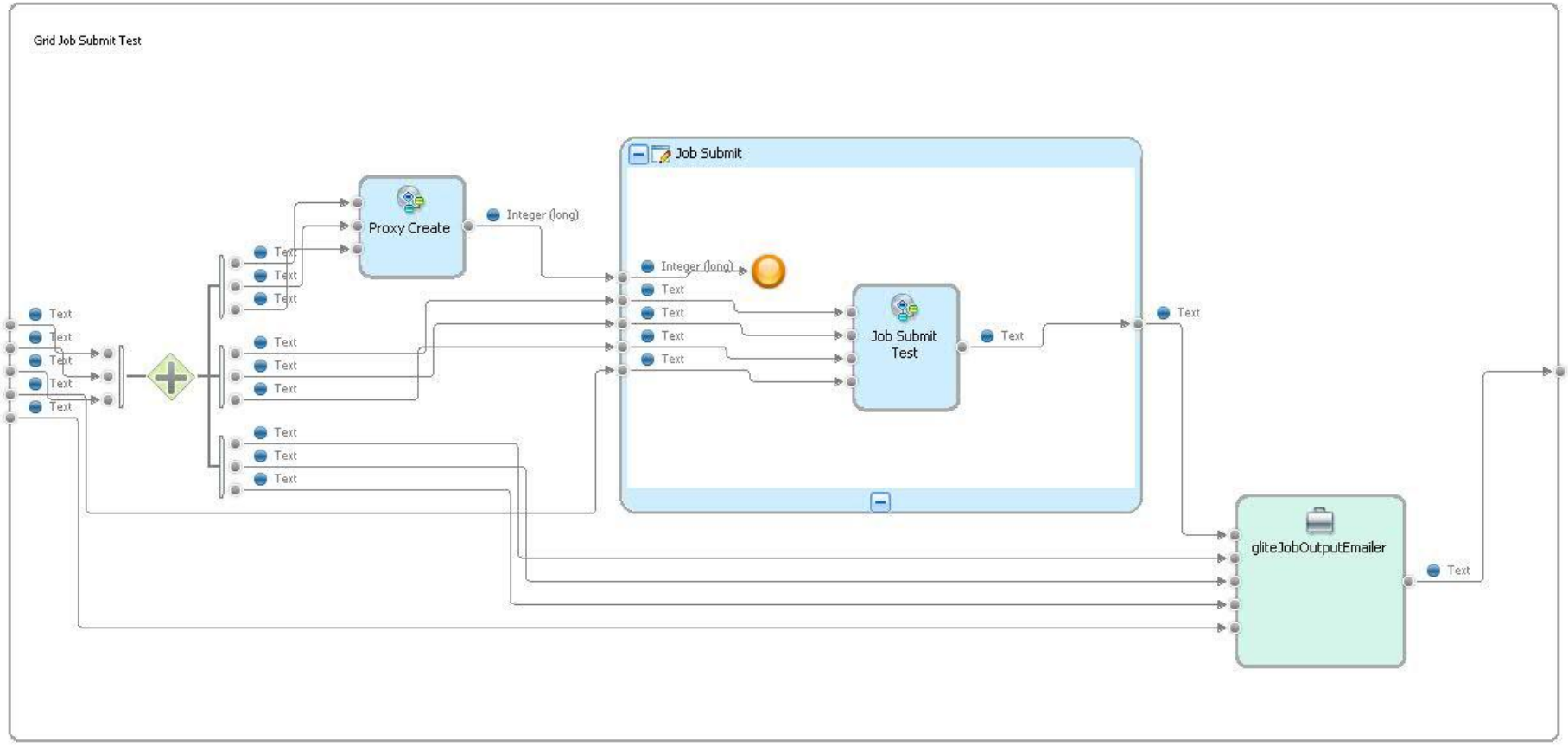
Процес 6: Grid Job Submit

Описание: По зададени потребителско име, виртуална организация, парола, изпълним файл, имена на файлове обработвани в изпълнимия файл, входни параметри и e-mail на потребителя, създава прокси, изпраща задачата с изпълним файл в грид и e-mail на потребителя при успешно завършена задача.



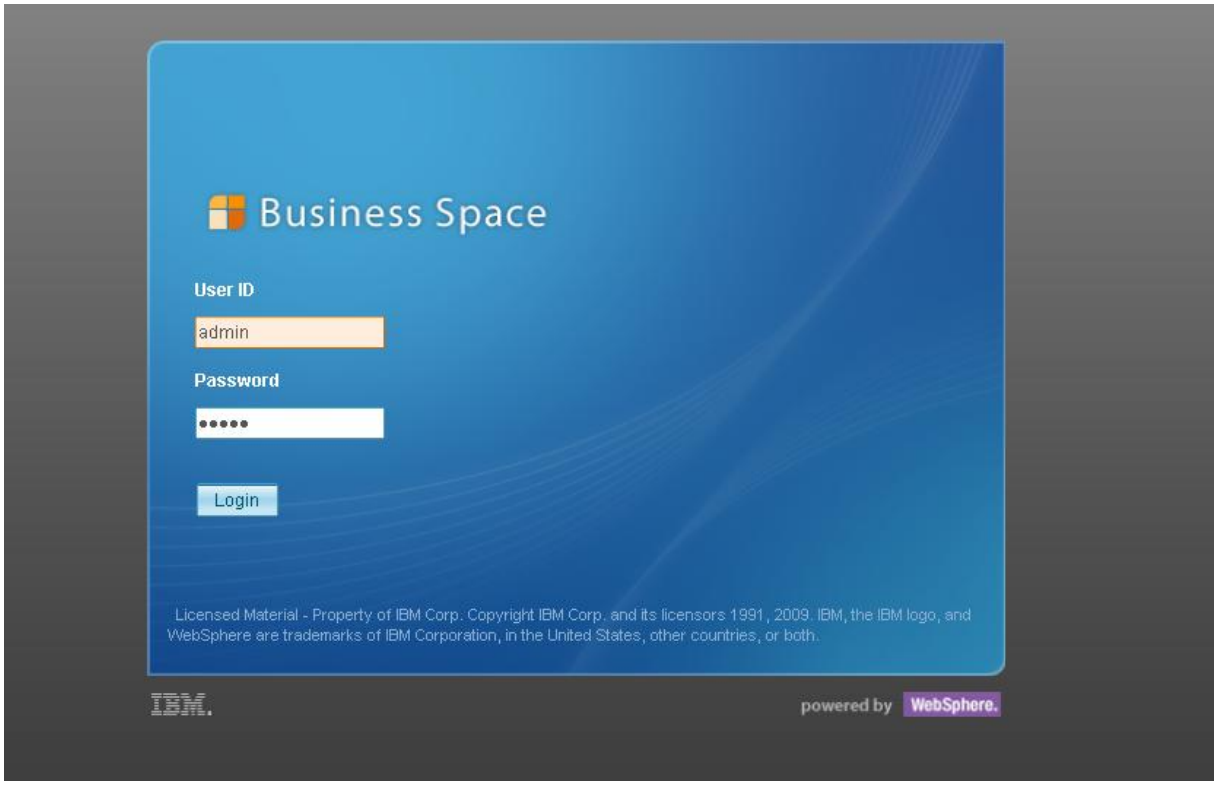
Процес 7: Grid Job Submit Test

Описание: По зададени потребителско име, виртуална организация, изпълним файл и e-mail на потребителя, създава прокси, изпраща задачата с изпълним файл в грид и e-mail на потребителя при успешно завършена задача.

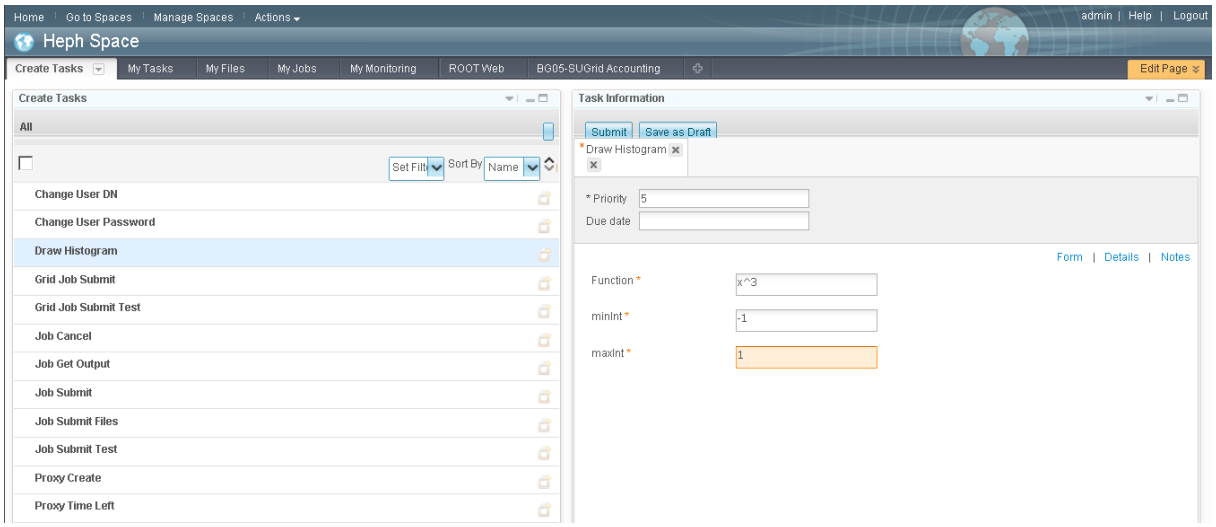


Приложение 5 - Работно пространство на грид-потребител

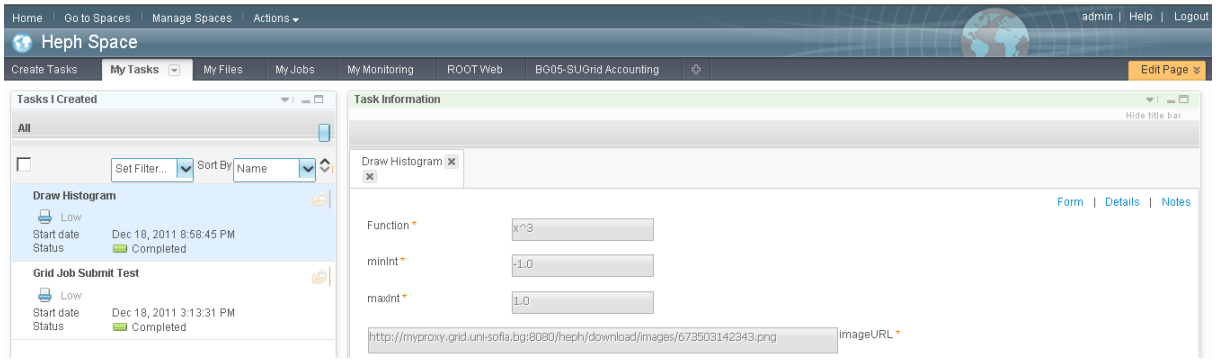
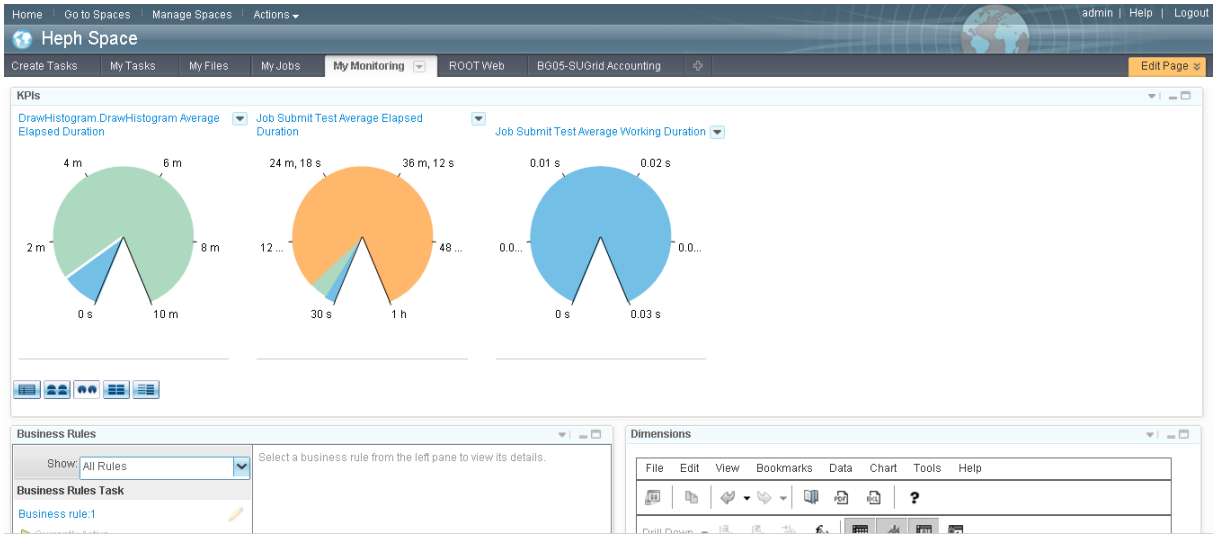
Начална страница на работно пространство към среда за изпълнение



Стартиране на бизнес-процеса *Draw Histogram* от работното пространство



Резултата от изпълнение на бизнес-процеса *Draw Histogram* в работното пространство



Наблюдение на бизнес-процес в среда за изпълнение