

Inżynieria oprogramowania w projekcie UE "CrossGrid"

Marian Bubak^{*}, Maciej Malawski^{**}, Grzegorz Młynarczyk^{***}, Piotr Nowakowski^{*}, Robert Pająk^{*}, Michał Turała^{****}, Katarzyna Zajac^{**}

^{*}*Akademickie Centrum Komputerowe CYFRONET-AGH, Kraków*

^{**}*Katedra Informatyki AGH, Kraków*

^{***}*Websoft sp. z o. o., Kraków*

^{****}*(4) Instytut Fizyki Jądrowej im. H. Niewodniczańskiego, Kraków*

bubak@uci.agh.edu.pl malawski@uci.agh.edu.pl

Grzegorz.Mlynarczyk@websoft.pl ymnowako@cyf-kr.edu.pl

ympajak@cyf-kr.edu.pl michal.turala@cern.ch kzajac@uci.agh.edu.pl

Streszczenie

Niniejszy rozdział opisuje aspekty inżynierii oprogramowania w projekcie CrossGrid. Projekt ten jest jednym z większych projektów informatycznych sponsorowanych przez Unię Europejską w ramach Piątego Programu Ramowego i obejmuje 21 niezależnych instytucji naukowych i komercyjnych w 11 krajach. Znaczne rozmiary projektu w połączeniu z jego geograficznym rozproszeniem wymagają opracowania specjalnych procedur, gwarantujących jednorodność i spójność oprogramowania w obrębie całego Konsorcjum Projektu.

Wprowadzenie

CrossGrid to jedno z największych europejskich przedsięwzięć w zakresie sieci gridowych [CGRE2002], łączące 21 oddzielnych instytucji i finansowane z środków Piątego Programu Ramowego Unii Europejskiej. Główne cele projektu obejmują rozszerzenie zakresu stosowalności oprogramowania gridowego o nową kategorię aplikacji (o dużym znaczeniu praktycznym) oraz rozbudowę europejskiej infrastruktury gridowej, z uwzględnieniem nowych państw członkowskich UE. Wszystkie aplikacje opracowywane w zakresie środowiska CrossGrid charakteryzuje stała interakcja z użytkownikiem na zasadzie sprzężenia zwrotnego. Każda aplikacja wymaga reakcji systemu w przewidzianych ramach czasowych (od bezpośredniej do długoterminowej) - projekt kładzie więc nacisk na aspekty związane z jakością usług (*Quality of Service*) oraz interaktywnością oprogramowania (zwłaszcza jeśli chodzi o generowanie raportów, monitorowanie pracy systemu oraz analizę poprawności i efektywności kodu aplikacji). Aby wypełnić sformułowane wyżej zadania, projekt opracowuje zestaw narzędzi (tzw. *middleware*), stanowiących uzupełnienie bazowego oprogramowania udostępnianego przez siostrzany projekt DataGrid [EDGR2001] o moduły związane z wymienionymi

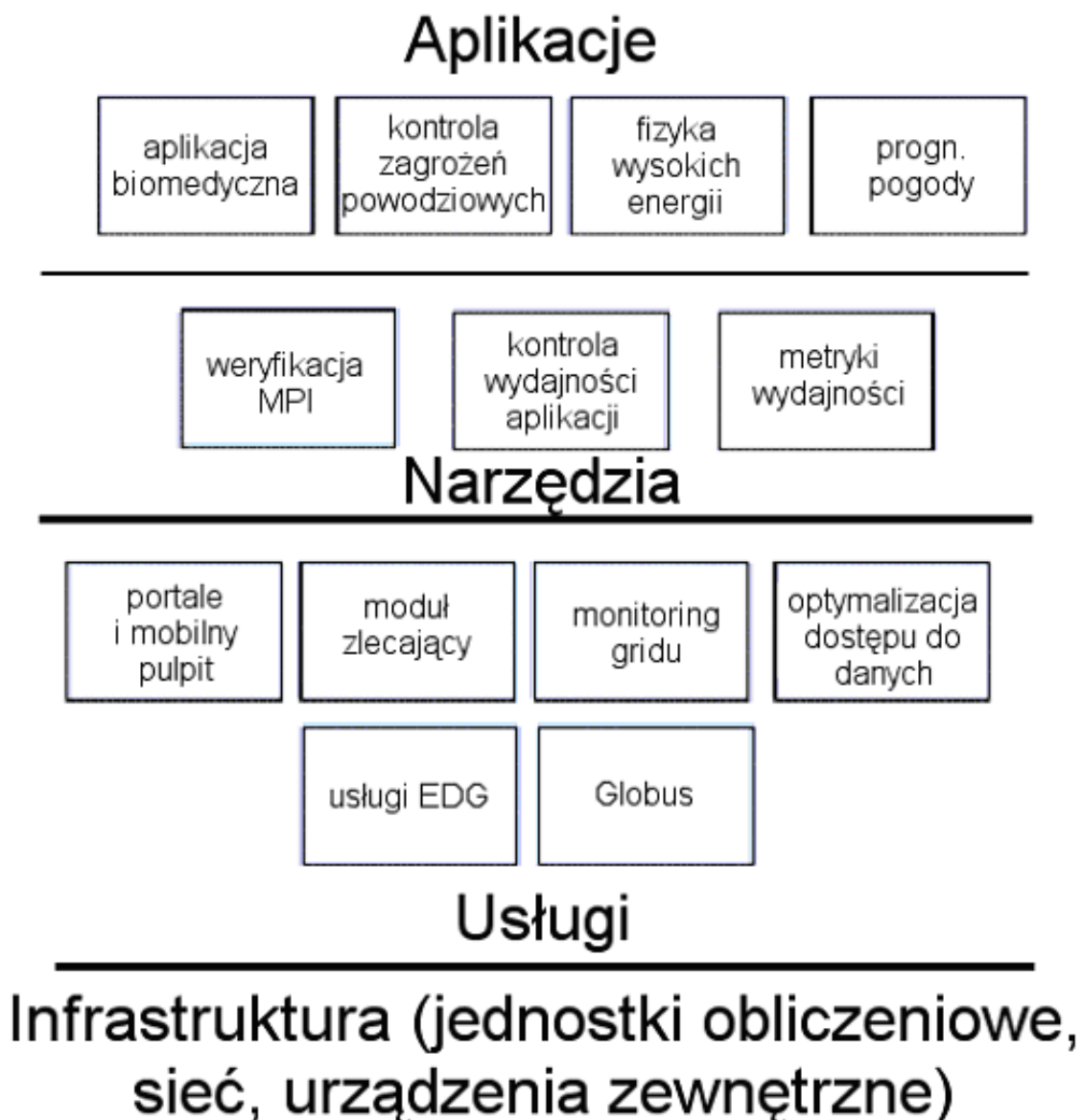
wcześniej zadaniami. W celu ułatwienia potencjalnym użytkownikom kontaktu z systemem, CrossGrid implementuje szereg dedykowanych portali dla poszczególnych aplikacji i narzędzi monitorujących, oraz rozwija koncepcję mobilnego środowiska pracy (*migrating desktop*) - przechowywanego na zewnętrznym serwerze i udostępnianego za pośrednictwem przeglądarek WWW. Wszystkie komponenty opracowywane w zakresie projektu CrossGrid są weryfikowane za pośrednictwem infrastruktury testowej, łączącej znaczące ośrodki obliczeniowe państw uczestniczących w projekcie i stanowiącej rozszerzenie europejskiej infrastruktury gridowej. Warto zauważyć, że nasza infrastruktura jest całkowicie kompatybilna z projektem DataGrid (z którym CrossGrid pozostaje w ścisłej współpracy), i - za pośrednictwem współpracy z ośrodkami w Stanach Zjednoczonych, prowadzonej przez komplementarny wobec DataGridu projekt DataTAG - wchodzi w skład największej obecnie sieci gridowej świata.

Architektura

Architektura projektu CrossGrid opiera się na podejściu hierarchicznym (podobnie jak w przypadku innych znaczących projektów gridowych). Wyróżniamy trzy podstawowe warstwy oprogramowania (aplikacje, narzędzia i usługi), a także jedną warstwę sprzętową (ang. *fabric* [ANAT2001]). Poszczególne warstwy projektu (patrz rysunek 1) znajdują odzwierciedlenie w podziale pracy na tzw. pakiety robocze (*Work Packages*): trzy pakiety techniczne (związane z poszczególnymi warstwami oprogramowania), jeden pakiet poświęcony infrastrukturze oraz pakiet zarządzający. Poniżej znajduje się krótki opis każdego z pakietów.

Aplikacje

Celem pierwszego pakietu roboczego projektu jest zaprojektowanie i implementacja aplikacji gridowych wymagających dostępu do funkcjonalności charakterystycznej dla projektu CrossGrid. Podstawowe problemy związane z tego rodzaju aplikacjami to: rozproszenie danych źródłowych, wizualizacja rezultatów (niezależna od platformy sprzętowej, na której są one odbierane) oraz możliwość cofania operacji i zmiany ich przebiegu w zależności od wyborów podejmowanych przez użytkownika na etapie wykonywania aplikacji. Pierwszy pakiet roboczy opracowuje obecnie cztery aplikacje korzystające z elementów interaktywności CrossGridu. Zapotrzebowanie na tego rodzaju interaktywność stanowi siłę napędową dla rozwoju oprogramowania middleware'owego w pozostałych pakietach roboczych; co więcej - każda z aplikacji wniesie znaczący wkład w obszar tematyczny, z którym jest związana, demonstrując korzyści wynikające ze stosowania sieci gridowych w obliczeniach rozproszonych.



Rys. 1. Warstwowa architektura projektu CrossGrid

Interaktywne symulacje i wizualizacja biomedyczna: Przeznaczeniem tej aplikacji jest obliczanie i wizualizacja przepływu krwi w zadanej strukturze naczyń krwionośnych, projektowanej przez lekarza na podstawie danych angiograficznych pacjentów oczekujących na zabieg wszczepienia przetok (przy stwierdzonej niewydolności serca), lub angioplastyki innych organów wewnętrznych. Możliwość dokonania bieżącej analizy proponowanej struktury umożliwia optymalizację umiejscowienia wszczepianych by-passów - w zależności od indywidualnych potrzeb pacjenta. Wyniki są prezentowane w przyjaznej dla użytkownika postaci graficznej, bezpośrednio na komputerze chirurga (w gabinecie lub sali operacyjnej). Aplikacja biomedyczna korzysta z infrastruktury gridowej w celu przeprowadzenia wymaganych obliczeń oraz uzyskania dostępu do rozproszonej biblioteki angiogramów. **System wspomaganie w planowaniu przeciwpowodziowym:** Jest to system umożliwiający prognozowanie i wizualizację zagrożeń powodziowych, na podstawie bieżących prognoz pogody nad zlewiskami wybranych rzek.

Aplikacja obejmuje zastosowanie zaawansowanych modeli meteorologicznych, hydrologicznych i hydraulicznych, połączonych ze sobą na zasadzie kaskady symulacji, gdzie każdy kolejny etap (prognoza pogody, analiza hydrologiczna, symulacja hydrauliczna) dostarcza danych wejściowych do kolejnej fazy badań. **Analiza danych rozproszonych w fizyce wysokich energii:** Ta aplikacja umożliwia szybki dostęp do rozproszonych baz danych, zawierających wyniki pomiarów przeprowadzanych przy użyciu nowego akceleratora cząstek LHC budowanego obecnie w CERNie. Ilość danych generowanych przez LHC wymaga utworzenia dedykowanej infrastruktury sprzętowej i aplikacyjnej dla gromadzenia, katalogowania i analizy wyników (akcelerator rejestruje miliardy zderzeń, ale tylko jedno zderzenie na milion jest interesujące z punktu widzenia fizyków wysokich energii). Tworzona w ramach projektu CrossGrid aplikacja stanowi odpowiedź na to wyzwanie. **Prognozowanie pogody i modelowanie dystrybucji zanieczyszczeń atmosferycznych:** Czwarta aplikacja stanowi zestaw narzędzi gridowych do wykorzystania przez meteorologów i oceanografów. Nowoczesne algorytmy prognozowania pogody nad wybranymi zbiornikami wodnymi oraz analizy rozkładu zanieczyszczeń atmosferycznych są dostosowywane do wymagań związanych z pracą w środowisku gridowym.

Narzędzia wspomagające

Jednym z kluczowych aspektów sieci gridowych - zarówno dla użytkowników jak i dla programistów - jest ich złożoność. Opracowywanie nowych aplikacji, ich optymalizacja i debugging sprawia nierazko poważne problemy, wynikające z heterogeniczności i niestęłego charakteru środowiska (obejmującego liczne urządzenia rozproszone na rozległym obszarze; w przypadku projektu CrossGrid - na całym kontynencie). Z tych przyczyn podstawowym celem projektu stało się opracowanie narzędzi wspomagających proces tworzenia równoległych aplikacji gridowych; intensywnych obliczeniowo i jednocześnie wymagających obróbki znacznej ilości danych. Automatyczna kontrola zgodności kodu MPI wprowadzanego przez użytkowników z ogólnie przyjętymi kryteriami poprawności zapobiega uruchamianiu aplikacji, które wykorzystywałyby zasoby systemu w nieefektywny sposób - CrossGrid opracowuje w tym celu dedykowane narzędzie (MARMOT). Ponadto, w przygotowaniu są narzędzia kontroli wydajności aplikacji, przydatne na etapie projektowania i implementowania oprogramowania gridowego. Narzędzia te - oparte na koncepcji sond, umieszczanych wewnątrz kodu programu i dokonujących automatycznego pomiaru wybranych wartości (np. czasu, wymaganego do wykonania kluczowego fragmentu kodu lub ilości danych przesyłanych przez sieć) - stanowią pomoc dla programistów z pierwszego pakietu roboczego i mogą być wykorzystywane również w przyszłych aplikacjach, nie tylko w ramach projektu CrossGrid.

Nowe usługi gridowe

Od czasu swoich narodzin, koncepcja sieci gridowych [ANAT2001] obrosła w szereg usług ułatwiających aplikacjom korzystanie z gridów i enkapsulujących kluczowe aspekty funkcjonalności tego rodzaju systemów (warto wymienić takie pakiety oprogra-

mowania middleware'owego jak Globus Toolkit [GLOB2003], Unicore [UNIC2003] czy pakiet DataGrid [EDGR2001]). Projekt CrossGrid zajmuje się opracowywaniem i implementacją usług wysokopoziomowych, które nie były dotychczas dostępne, a które mają szczególne znaczenie w pracy z interaktywnymi aplikacjami w rozproszonym środowisku obliczeniowym. Jedną z tego rodzaju usług jest możliwość kontaktu z systemem za pośrednictwem przyjaznego dla użytkownika systemu portali - koncepcja polega tu na możliwości zapisania środowiska pracy każdego z użytkowników na dedykowanym serwerze, z którego środowisko to byłoby następnie wczytywane i prezentowane na ekranie użytkownika, niezależnie od jego lokalizacji; stąd nazwa modułu: "mobilny pulpit" (ang. *migrating desktop*). Kolejną usługą realizowaną w ramach projektu jest optymalizacja dostępu do danych przechowywanych w archiwach taśmowych i odległych bazach danych. Na podstawie analizy działań użytkownika (na poziomie tzw. schedulera, czyli modułu odpowiedzialnego za kojarzenie zadań zleczanych przez użytkowników z odpowiadającymi im zasobami gridowymi) system automatycznie ocenia które pliki mogą okazać się przydatne dla danej aplikacji w najbliższej przyszłości i importuje je w miejsce, z którego mogą zostać szybko pobrane (np. na serwer, na którym wykonywany jest kod aplikacji). Warstwa usług projektu CrossGrid wykorzystuje komponenty opracowane w innym ważnym projekcie europejskim - DataGrid. Projekt ten, w powiązaniu z amerykańskim zespołem z Argonne National Laboratory, odpowiedzialnym za pakiet podstawowych usług gridowych (Globus Toolkit [GLOB2003]), opracowuje standardowy zestaw usług umożliwiających zlecenie zadań, zarządzanie plikami w środowisku gridowym oraz monitorowanie wybranych aspektów systemu. Zestaw ten (EDG Middleware Suite) składa się obecnie z następujących komponentów:

- moduł zlecający (*scheduler*),
- interfejs JDL (*Job Description Language*) - język, w którym wyrażane są wymagania każdego ze zleczanych zadań - na podstawie plików JDL scheduler może przyporządkować zadanie do konkretnych zasobów gridowych.
- katalogi replik i narzędzia zarządzania replikami (*replica catalogs* i *replica managers*),
- bezpieczeństwo transakcji - system oparty na certyfikatach zaufania i szyfrowaniu RSA,
- logowanie i księgowanie transakcji.

Poniżej znajduje się opis dodatkowych usług implementowanych w projekcie CrossGrid.

Usługi interakcji z użytkownikiem

Zestaw usług interakcji z użytkownikiem (*User Interactive Services* - UIS) obejmuje usługi ułatwiające interaktywną pracę ze złożonymi aplikacjami gridowymi, w tym - symulacje i wizualizacje symulowanych procesów. CrossGrid korzysta z procesów HLA Runtime Infrastructure [HLAR2000], które pozwalają na synchronizację rozproszonych komponentów aplikacji z portalem użytkownika i sterowanie symulacjami w czasie zbliżonym do rzeczywistego. Mechanizm ten znajduje zastosowanie w aplikacji biome-

dycznej, gdzie użytkownik (chirurg) jest w stanie modyfikować parametry aktywnych symulacji bez konieczności uruchamiania w tym celu nowych zadań. Indeksy informacyjne OGSA [OGSA2002] umożliwiają wyszukiwanie zasobów obsługujących interakcję HLA (tzn. z zainstalowanymi odpowiednimi modułami middleware'owymi).

Grid Visualization Kernel

Moduł Grid Visualization Kernel (GVK) służy do łączenia rozproszonych zadań symulacji i urządzeń udostępniających dane graficzne (np. tomografów) z portalami klienckimi odpowiedzialnymi za wizualizację owych danych, stosując w tym celu automatyczną konwersję formatów graficznych do wspólnego formatu, obsługiwanego przez portale aplikacyjne (NetCDF) [NCDF1999].

Portal i Migrujący Pulpit

Portal i mobilny pulpit to dwa interfejsy użytkownika umożliwiające interakcję z położonymi na niższych warstwach usługami gridowymi. Dzięki zastosowaniu tzw. serwerów zdalnego dostępu (ang. *Remote Access Server* - RAS) możliwe jest zapisywanie bieżących sesji użytkownika i odtwarzanie ich na dowolnej stacji roboczej - każdy użytkownik dysponuje więc jednolitym, konfigurowalnym środowiskiem pracy, udostępnianym za pośrednictwem zwykłej przeglądarki WWW - jedynym wymaganie jest połączenie sieciowe i oprogramowanie obsługujące transmisję HTTP. Profile użytkowników przechowywane są w dedykowanej bazie danych LDAP.

Zlecanie zadan

Zlecanie zadań w środowisku CrossGrid odbywa się poprzez specjalne pliki (w języku JDL - *Job Description Language*), zawierające opis wymagań sprzętowych i aplikacyjnych danego zadania. Moduł zlecający wyszukuje optymalny zestaw zasobów gridowych potrzebnych do realizacji danego zadania, w oparciu o żądane kryteria optymalizacyjne (np. duża przepustowość łączy sieciowych lub krótki czas dostępu). Moduł zlecający korzysta z oprogramowania DataGrid [EDGR2001] i Condor-G [CDRG2002].

Infrastruktura monitorowania

Pod wspólną nazwą "monitorowanie" kryje się grupa usług wykorzystywanych na różnych etapach gromadzenia i przetwarzania informacji. CrossGrid rozwija usługi zdalnego śledzenia aktywnych zadań za pomocą oprogramowania Jiro (nieinwazyjne monitorowanie infrastruktury) oraz SANTA-G (monitorowanie sieci). Dane pochodzące z obydwu modułów mogą być bezpośrednio przekazywane użytkownikowi lub zapisywane w bazie danych gridowego systemu informacyjnego (R-GMA).

Zarządzanie danymi

Jest to kolejna usługa rozszerzająca funkcjonalność oprogramowania przejętego z

projektu DataGrid. Zarządzanie danymi w projekcie CrossGrid zostaje rozbudowane o moduł optymalizacji wykorzystania archiwów taśmowych o długich czasach dostępu. Implementowany w tym celu system ekspertowy ma za zadanie analizować aktywność poszczególnych zadań i odnajdywać dane, których w najbliższej kolejności może potrzebować dany użytkownik. System ów zostanie zintegrowany ze standardowymi narzędziami zarządzania danymi (tzn. z katalogami replik i modułem zarządzania replikami) wchodzącymi w skład pakietu EDG.

Pozostałe usługi

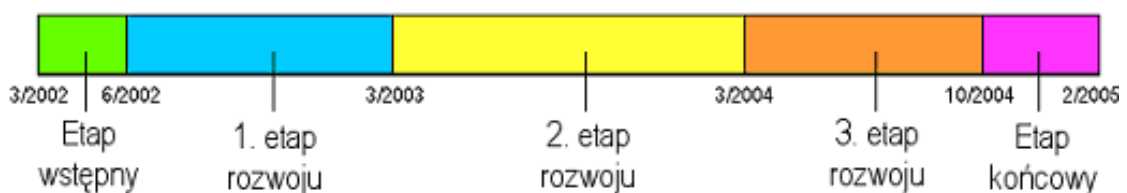
CrossGrid korzysta ze standardowych usług gridowych, zawartych w pakiecie Globus Toolkit. Aktualna wersja tego pakietu wykorzystywana w projekcie CrossGrid nosi numer 2.4. Zespół ds. architektury projektu bada obecnie możliwości zastosowania nowej wersji 3, stanowiącej implementację architektury usług gridowych (OGSA). [PHYS2002]

Infrastruktura testowa (Testbed)

Podstawową rolą infrastruktury testowej projektu CrossGrid jest umożliwienie integracji wszystkich aplikacji, narzędzi i usług opracowywanych w ramach projektu. Wiele istotnych testów systemu może zostać przeprowadzonych tylko w oparciu o rozproszony zbiór jednostek obliczeniowych, połączonych siecią szkieletową o wysokiej przepływności (w naszym przypadku - europejską siecią Geant). Podstawowym zadaniem czwartego pakietu roboczego jest więc rozbudowa i utrzymanie takiej sieci, przygotowywanie kolejnych dystrybucji oprogramowania powstającego w ramach projektu, zapewnienie kompatybilności z innymi projektami gridowymi (zwłaszcza z projektem DataGrid) oraz wspomaganie procesu tworzenia nowych węzłów sieci gridowej we wszystkich krajach uczestniczących w projekcie. Infrastruktura testowa projektu CrossGrid obejmuje obecnie 15 centrów obliczeń dużej skali w dziewięciu krajach.

Etapy projektu

CrossGrid jest projektem obliczonym na trzy lata - oficjalne rozpoczęcie projektu miało miejsce w marcu 2002; jego zakończenie planuje się natomiast na luty 2005 (rysunek 2).



Rys. 2. Etapy projektu

Proces rozwoju oprogramowania CrossGridowego został dostosowany do specyficz-

nych potrzeb rozproszonej kolaboracji i obejmuje następujące etapy: **Etap wstępny**, obejmujący specyfikację i scalanie wymagań (miesiące 1-3), **Pierwszy etap rozwoju** - opracowanie szczegółowego projektu każdego modułu, przegląd wymagań, wstępna implementacja i pierwsze prototypy (miesiące 4-12), **Drugi etap rozwoju** - ciąg dalszy implementacji (w tym uzupełnienie prototypów o funkcjonalność gridową); integracja modułów (miesiące 13-24), **Trzeci etap rozwoju** - ostateczna implementacja wszystkich modułów projektu z funkcjonalnością określoną w dokumentach projektowych (miesiące 25-32), **Etap końcowy** - demonstracje, przygotowanie końcowej dokumentacji i ostateczna kontrola rezultatów projektu (miesiące 33-36).

Standardy, konwencje i metryki

CrossGrid jest dużym, międzynarodowym projektem, zaangażowanym w tworzenie szerokiej gamy modułów aplikacyjnych. Ze względu na złożoność projektu i znaczną ilość platform sprzętowych i software'owych wykorzystywanych w fazie jego rozwoju konieczne stało się opracowanie zestawu wspólnych standardów i konwencji programistycznych - wystarczająco ogólnych, by nie ograniczać rozwoju oprogramowania, ale jednocześnie na tyle szczegółowych, by zagwarantować jego spójność i jednolitość.

Standardowe Procedury Operacyjne

Zbiór procedur stosowanych w trakcie rozwoju oprogramowania w projekcie został opisany w specjalnym dokumencie Standardowych Procedur Operacyjnych (ang. *Standard Operating Procedures* - SOP) [SOPS2002] Dokument ten obejmuje następujące działy: **Centralne repozytorium projektu**: informacje związane z funkcjonowaniem i warunkami użytkowania centralnego repozytorium CVS projektu, obsługiwanego przez FZK (Karlsruhe), oraz repozytorium zapasowe w Walencji. Repozytorium odpowiada za gromadzenie i archiwizację całego oprogramowania wchodzącego w skład projektu. **Narzędzia**: lista programów narzędziowych, które powinny być wykorzystywane przez wszystkich partnerów w projekcie (np. narzędzia Autobuild, wspomagającego proces tworzenia dystrybucji oprogramowania). Lista ta jest obsługiwana przez specjalny zespół ds. technicznych (*Technical Board*) w porozumieniu z kierownictwem poszczególnych grup roboczych. **Mechanizmy zgłaszania problemów i wymaganych zmian**: ta sama procedura obowiązuje wszystkich partnerów w projekcie. Narzędzia umożliwiające zgłaszanie problemów i reagowanie na owe zgłoszenia zostały wbudowane w centralne repozytorium projektu i mają za zadanie wspomagać proces rozwoju oprogramowania. **Konwencje nazewnictwa dla oprogramowania**: zestaw jednolitych konwencji związanych z nazewnictwem modułów, klas i metod oraz komentarzy. **Pozostałe konwencje**: formaty dat/godzin, jednostki miernicze itp.

Procedury kontrolne

Zgodnie z wcześniejszym opisem, projekt dzieli się (pod względem czasowym) na pięć kolejnych etapów. Każdemu etapowi towarzyszy odrębny zestaw procedur kontrol-

nych, mających na celu zagwarantowanie właściwego przebiegu pracy w ramach projektu i kontrolę zgodności tworzonego oprogramowania z wymaganiami wstępnymi. Proces kontroli powinien udzielić odpowiedzi na następujące pytania ([BRAU2001]):

- Czy budujemy to, co trzeba?
- Czy to, co budujemy jest możliwe do zbudowania?
- Czy dostępne zasoby są wykorzystywane w efektywny sposób?

Niniejszy dział opisuje procedury kontrolne projektu CrossGrid w odniesieniu do każdego z powyższych aspektów inżynierii oprogramowania.

Opis wymagań projektowych

Opis wymagań projektowych opiera się na tzw. specyfikacjach wymagań (*Software Requirements Specification* - SRS), opracowywanych według standardu IEEE-830-1993 przez każdy pakiet techniczny. Dokumenty SRS zawierają następujące informacje: **Ogólny opis modułu** (funkcjonalność, operacje, interfejsy, wymagania i ograniczenia), uzupełnione o przypadki użycia w języku UML. **Opis dostępnych technologii** - zawierający charakterystykę porównawczą rozwiązań, którymi można posłużyć się w pracy nad danym modułem. Oprócz dokumentów SRS, zespół do spraw architektury (TAT) przygotowuje wstępny szkic ogólnej architektury projektu - w tym interakcji pomiędzy jego poszczególnymi komponentami.

Dokumentacja projektowa

Po przedłożeniu, recenzji i zatwierdzeniu dokumentów SRS przychodzi czas na opracowanie (na ich podstawie) szczegółowych projektów każdego modułu. Opisy te mają formę dokumentów projektowych, zgodnych ze standardem IEEE-1016-1987 i zawierających następujące dane: **Skład systemu**: diagramy komponentowe UML każdego modułu i opis funkcjonalności modułów, **Opis zależności**: interakcja danego modułu z pozostałymi modułami oraz z oprogramowaniem zewnętrznym (spoza projektu), **Opis interfejsów**: formalny opis wszystkich interfejsów (zarówno użytkownika jak i programistycznych) udostępnianych przez moduł, **Szczegółowy projekt**: diagramy klas dla każdego modułu - niskopoziomowy opis architektury, razem z objaśnieniami dotyczącymi poszczególnych metod i atrybutów.

Implementacja i dystrybucje oprogramowania

Po zakończeniu fazy projektowej (w dziewiątym miesiącu projektu) przychodzi czas na rozpoczęcie procesu implementacji i towarzyszących mu inkrementacyjnych dystrybucji oprogramowania. Aneks Techniczny (dokument opisujący szczegółowo założenia strukturalne projektu oraz podział obowiązków partnerów) przewiduje trzy dziewięciomiesięczne cykle, z których każdy ma zakończyć się opublikowaniem kolejnej dystrybucji oprogramowania. W zależności jednak od bieżących uwarunkowań, Koordynator projektu może zażądać stworzenia dodatkowych, pośrednich dystrybucji - proces rozwoju projektu powinien więc być na tyle elastyczny, by dopuszczać możliwość takich

zmian. Aby osiągnąć ten cel, Centralne Repozytorium projektu zostało wyposażone w aplikację **Autobuild**, pozwalającą na szybką kompilację oprogramowania projektowego zawartego w danej chwili w całej strukturze katalogów repozytorium (patrz [PORT2002]). Proces przygotowania dystrybucji obejmuje następujące etapy: **Spotkanie koordynacyjne**: poświęcone omówieniu poprzedniej dystrybucji i sformułowaniu planu działania. Spotkanie koordynacyjne jest otwarte dla wszystkich uczestników projektu; szczególnie istotny jest jednak udział kierowników pakietów roboczych, zespołu ds. architektury projektu i specjalnego zespołu integracyjnego działającego w ramach czwartego pakietu roboczego. Podstawowym celem zebrania jest sporządzenie opisu funkcjonalności istniejącej dystrybucji oraz hierarchicznej listy problemów, które wymagają rozwiązania, wraz z zaleceniami dotyczącymi stosowanych narzędzi (wersji pakietu Globus Toolkit, kompilatorów itp.). **Spotkania robocze poszczególnych pakietów**: postanowienia spotkania koordynacyjnego powinny być przedyskutowane w ramach poszczególnych pakietów, których kierownicy odpowiadają za przydział działań związanych z przygotowaniem kolejnej dystrybucji konkretnym pracownikom. Osobne zebranie przeprowadza zespół ds. architektury - celem jest tu ocena, czy plan kolejnej dystrybucji wymaga zmian w definicji architektury projektu; jeśli tak - konieczne staje się przygotowanie nowej specyfikacji. **Plan roboczy**: w konsultacji z kierownikami pakietów roboczych, kierownictwo projektu tworzy listę czynności, które powinny poprzedzać kolejną dystrybucję (w tym - listę oprogramowania, które należy zainstalować na każdym z węzłów infrastruktury testowej **Integracja oprogramowania**: każdy z zespołów rozwijających oprogramowanie powinien upewnić się, że Centralne Repozytorium projektu w Karlsruhe zawiera bieżące wersje wszystkich modułów, dostosowane do potrzeb nadchodzącej dystrybucji. Zespół ds. weryfikacji i kontroli jakości czwartego pakietu roboczego testuje całe zgromadzone oprogramowanie i zgłasza wszelkie problemy odpowiednim programistom, za pośrednictwem mechanizmu zgłoszeniowego opisanego w poprzednim rozdziale. **Testy akceptacji**: gotowy zestaw oprogramowania jest instalowany na wszystkich węzłach infrastruktury testowej, gdzie przechodzi dalsze testy (w tym - testy wydajności), zgodnie ze scenariuszami zawartymi w planie roboczym. **Dystrybucja**: Zespół ds. integracji publikuje dokument zawierający szczegółowy opis kolejnej dystrybucji, łącznie ze wszelkimi wprowadzonymi zmianami, testami przeprowadzonymi zgodnie z [TEST2002], dokumentacją wszelkich dostrzeżonych błędów i problemów, opisem ewentualnych różnic między funkcjonalnością dystrybucji a założeniami zawartymi w dokumentach SRS (razem z uzasadnieniami) oraz instrukcją instalacji i obsługi oprogramowania.

Dystrybucje infrastruktury testowej i ich walidacja

Z pojęcia "dystrybucja infrastruktury testowej" (ang. *Testbed Release*) korzystamy w odniesieniu do dobrze zdefiniowanego i funkcjonalnie kompletnego zbioru oprogramowania projektu, instalowanego na dobrze zdefiniowanym zbiorze rozproszonych zasobów komputerowych, wraz z towarzyszącym mu opisem mechanizmów instalacji. Zgodnie ze specyfikacją architektury [ARCH2002], infrastruktura testowa powinna obsługiwać oprogramowanie opracowywane w ramach projektu CrossGrid a także projek-

tów stowarzyszonych - w tym Globus i DataGrid (o zbiorczej nazwie "oprogramowanie EDG"). Infrastruktura testowa projektu CrossGrid dzieli się na trzy rozłączne sieci: infrastrukturę rozwojową (*development testbed*), infrastrukturę kontrolną (*test and validation testbed*) oraz infrastrukturę produkcyjną (*production testbed*). Infrastruktura rozwojowa jest stosunkowo najmniejsza i służy do testowania poszczególnych komponentów przez ich twórców, w ramach tzw. testów wewnętrznych (*unit tests*). Infrastruktura kontrolna służy do przeprowadzania testów integracyjnych i przygotowywania kolejnych dystrybucji, podczas gdy infrastruktura produkcyjna ma za cel wykorzystywanie opracowanych aplikacji w rzeczywistych scenariuszach zastosowania. Pakiety oprogramowania EDG są pobierane na bieżąco z repozytorium projektu DataGrid, choć w ramach udogodnienia można je również znaleźć w repozytorium projektu CrossGrid, wraz z aktualną dystrybucją (patrz [PORT2002]). Repozytorium zawiera komplet oprogramowania niezbędnego do instalacji bieżącej dystrybucji na dowolnym węźle każdej z trzech sieci testowych.

Wskaźniki jakości

W celu zagwarantowania wysokiej jakości oprogramowania tworzonego w ramach projektu CrossGrid niezbędne staje się zastosowanie sformalizowanych metod pomiaru jakości. Idąc w ślad za projektem DataGrid, stojącym wobec podobnych problemów organizacyjnych (21 instytucji - głównie naukowych - w wielu krajach europejskich), CrossGrid stosuje system tzw. wskaźników jakości (*quality indicators* - QI), czyli metryk opisujących różne aspekty rozwoju projektu, gromadzonych i publikowanych w miesięcznych raportach przez kierownictwo. Wskaźniki jakości można podzielić na trzy grupy: wskaźniki związane z rozwojem oprogramowania, wskaźniki związane z rozwojem infrastruktury oraz wskaźniki ogólne. Każdemu wskaźnikowi towarzyszy zakres dopuszczalnych wartości, którego przekroczenie wymusza zastosowanie czynności zaradczych, będących przedmiotem uzgodnień między poszczególnymi zespołami roboczymi a kierownictwem projektu. Sumaryczne raporty stanowią odzwierciedlenie ogólnej jakości projektu i umożliwiają szybką identyfikację potencjalnych problemów.

Wskaźniki ogólne

Ta grupa wskaźników dotyczy przede wszystkim warunków personalnych i finansowych projektu; jako taka ma więc ograniczony wpływ na aspekty techniczne. Każdy partner powinien udostępniać (co miesiąc) następujące dane:

- ilość osobogodzin przeznaczonych na realizację poszczególnych zadań projektowych,
- ilość osób zatrudnionych w pracy nad projektem.

Biuro projektu przygotowuje następujące wskaźniki:

- ilość przewidzianych przez Aneks Techniczny dokumentów w danym okresie rozliczeniowym,
- odsetek dokumentów zatwierdzonych przez wewnętrznych recenzentów projektu,

- odsetek dokumentów zatwierdzonych przez Komisję Europejską,
- całkowita ilość osobogodzin przeznaczonych na pracę nad projektem.

Wskaźniki związane z rozwojem oprogramowania

Jakość tworzonego kodu można wyznaczyć na podstawie dwóch podgrup wskaźników:

- metryk statycznych,
- rezultatów testów wewnętrznych.

Szczególne znaczenie należy przywiązywać do grupy drugiej - wiadomo bowiem, że polityka testowa jest głównym czynnikiem warunkującym jakość tworzonego oprogramowania ([BRAU2001]). Każdy zespół roboczy odpowiada za przygotowanie odpowiedniej ilości tzw. scenariuszy testowych (skryptów testujących tworzone oprogramowania w zgodzie z przypadkami użycia określonymi w dokumentach SRS). Scenariusze te są następnie wykorzystywane w regularnych (comiesięcznych) testach wewnętrznych i na ich podstawie dokonana zostaje ocena efektywności kodu. Oprócz publikowania rezultatów testów, pracownicy odpowiedzialni za kontrolę jakości są również zobowiązani do dokonywania pomiarów metryk statycznych, w tym:

- ilości linii kodu wprowadzonych do repozytorium od czasu ostatniego pomiaru,
- zgodności kodu z konwencjami opisanymi w dokumencie Standardowych Procedur Operacyjnych,
- ilości pakietów, klas i funkcji przypadających na każdą z grup roboczych,
- ilości formalnych komentarzy przypadających na określoną ilość linijek kodu,
- metryk McCabe'a (złożonościowych),
- ilości problemów zgłoszonych przez każdą grupę roboczą za pośrednictwem kanałów formalnych,
- ilości nierozwiązanych problemów o istotności określonej jako "poważna", "krytyczna" lub "uniemożliwiająca dalszą pracę",
- średniego czasu potrzebnego na rozwiązanie pojedynczego problemu.

Metryki statyczne mierzone są za pośrednictwem wyspecjalizowanego oprogramowania (w tym: CODECOUNT, QStudio, JWalk, JavaNCSS oraz CodeWizard), zgodnie z [CGQA2003].

Wskaźniki dotyczące infrastruktury testowej

Poniższe wskaźniki są mierzone co miesiąc:

- ilość zasobów obliczeniowych wchodzących w skład infrastruktury testowej,
- średni miesięczny czas aktywności (dla każdego węzła), wyrażony w procentach,
- ilość użytkowników systemu (zarejestrowanych i aktywnych),
- średni odsetek zadań obliczeniowych zakończonych powodzeniem.

Powyższe wartości zostają uwzględnione w miesięcznych raportach razem z pozostałymi wskaźnikami jakości; są one również udostępniane na witrynie projektu zarejestro-

wanym użytkownikom.

Bieżący stan projektu

Sierpień 2003 jest osiemnastym miesiącem projektu, co oznacza, że CrossGrid znajduje się obecnie na drugim etapie rozwoju oprogramowania (patrz rozdział 3). W ciągu ostatnich siedemnastu miesięcy udało nam się osiągnąć następujące rezultaty (zgodnie z tzw. kamieniami milowymi, wyszczególnionymi w dokumencie [ANNE2002]): Specyfikacje SRS zostały przygotowane i zatwierdzone. Każda grupa robocza przedłożyła szczegółowy projekt rozwijanego przez siebie oprogramowania; projekty te zostały następnie poddane recenzji i zatwierdzone. CrossGrid zorganizował dwa spotkania integracyjne, w Santiago de Compostela (Hiszpania, luty 2003) oraz w Poznaniu (lipiec 2003). Trwają przygotowania do kolejnego spotkania w Nikozji (Cypr). Drugi i trzeci pakiet roboczy opublikowały pierwsze prototypy swojego oprogramowania; w przygotowaniu są również prototypy aplikacji gridowych, opracowywanych przez pierwszy pakiet roboczy. Wszystkie prototypy zostały poddane testom, zgodnie z planem roboczym projektu. W przygotowaniu znajduje się kurs wprowadzający dla przyszłych użytkowników oprogramowania CrossGrid. CrossGrid uczestniczy w międzynarodowej działalności standaryzacyjnej w ramach komitetu Global Grid Forum oraz prowadzi aktywną działalność naukową w zakresie technologii gridowych (kilkadziesiąt opublikowanych artykułów, udział w licznych międzynarodowych konferencjach, organizacja własnej, dorocznej konferencji naukowej Cracow Grid Workshop oraz współorganizacja kilku innych konferencji - zwłaszcza w powiązaniu z projektem DataGrid). **Podziękowania:** pragniemy podziękować M. Garbaczowi, J. Marco, N. Meyerowi, P.M.A. Sloomowi, W. Funice, R. Wismullerowi, D. van Albadzie oraz M. Hardtowi za ich wkład w niniejszą pracę. Opisana działalność jest w części finansowana przez Komisję Europejską w ramach projektu badawczego IST-2001-32243 "CrossGrid".

Bibliografia

- [ANAT2001] I. Foster, C. Kesselman i S. Tuecke, *The Anatomy of the Grid. Enabling Scalable Virtual Organizations.*, 200--222, marzec 2001.
- [ANNE2002] Aneks Techniczny projektu CrossGrid (CrossGrid Technical Annex and Description of Work), URL:
http://www.eu-crossgrid.org/CrossGridAnnex1_v31.pdf.
- [ARCH2002] Dokument D5.2.2 projektu CrossGrid - Wymagania w zakresie architektury I wstępna definicja architektury (CrossGrid Deliverable D5.2.2 - CrossGrid Architecture Requirements and First Definition of Architecture); wieloczęściowy dokument dostępny pod adresem, URL: <http://www.eu-crossgrid.org/M3deliverables.htm>.
- [BRAU2001] E. J. Braude, *Software Engineering: An Object-Oriented Perspective*, 2001, John Wiley & Sons, Inc..

- [CDRG2002] Portal projektu Condor-G, URL: <http://www.cs.wisc.edu/condor/condorg/>.
- [CGQA2003] Plan kontroli jakości projektu CrossGrid (CrossGrid Quality Assurance Plan), w recenzji.
- [CGRE2002] M. Bubak i M. Turala, *CrossGrid and its Relatives in Europe*, 14--15, wrzesień/październik year, D. Kranzlmüller, P. Kacsuk, J. Dongarra, J. Volker (Eds.), Proc. 9th European PVM/MPI Users' Group Meeting, Linz, Austria.
- [EDGR2001] Portal projektu EU DataGrid, URL: <http://www.eu-datagrid.org/>.
- [GLOB2003] Portal projektu Globus, URL: <http://www.globus.org/>.
- [HLAR2000] Portal projektu High Level Architecture, URL: <https://www.dms0.mil/public/transition/hla/>.
- [NCDF1999] Unidata NetCDF, URL: <http://www.unidata.ucar.edu/packages/netcdf/>.
- [OGSA2002] Portal architektury OGSA, URL: <http://www.globus.org/ogsa/>.
- [PHYS2002] I. Foster, C. Kesselman, J. M. Nick i S. Tuecke, *The Physiology of the Grid. An Open Grid Services Architecture for Distributed Systems Integration*, styczeń 2002.
- [PORT2002] Repozytorium oprogramowania i portal projektu CrossGrid, URL: <http://gridportal.fzk.de>.
- [SOPS2002] Dokument D5.2.3 projektu CrossGrid - Standardowe Procedury Operacyjne (CrossGrid Deliverable D5.2.3 - Standard Operating Procedures), URL: <http://www.eu-crossgrid.org/Deliverables/M6pdf/CG5.2-D5.2.3-v1.0-CYF020 StandardOperatingProcedures.pdf>.
- [TEST2002] Dokument D4.1 projektu CrossGrid - Plan Rozwoju Infrastruktury Testowej (CrossGrid Deliverable D4.1 - Testbed Development Plan), dodatek 4.
- [UNIC2003] Forum UNICORE, URL: <http://www.unicore.org/>.