Vědecká rada MFF UK v Praze

Julius Hřivnáč

Vyvoj programového zabezpečení a simulace fyzikálních procesů s kvarkem b pro experiment ATLAS

Číslo vědního oboru: 11-24-9
Název vědního oboru: jaderná a subjaderná fyzika

Autoreferát disertace k získání vědecké hodnosti kandidáta fyzikálně matematický věd

V Praze, 4.března 2001

Disertace byla vypracována v rámci spolupráce Fyzikálního Ústavu AV ČR s CERNem na projektu experimentu ATLAS.

Uchazeč:
RNDr. Julius Hřivnáč, Sekce fyziky elementárních částic, Fyzikální ústav, Akademie věd České republiky

Odborné vedení uchazeče:
Školitel:
doc. RNDr. Vladislav Šimák, DrSc. Sekce fyziky elementárních částic, Fyzikální ústav, Akademie věd České republiky

Oponenti:
Prof. Kors Bos, NIKHEF Amsterdam,
Doc. RNDr. Dušan Bruncko, CSc., UEF SAV Košice,
RNDr. Rupert Leitner, DrSc., MFF UK Praha

Stanovisko k disertaci vypracovala ....

Autoreferát byl rozeslán dne ....

Obhajova disertace se koná dne .... před komisí pro obhajoby kandidátských prací .... na MFF UK, Ke Karlovu 3, Praha 2, v zasedací místnosti, II.patro v .... hod.

S disertací je možno se seznámit v odd. pro vědecký výzkum matematicko-fyzikální fakulty UK v Praze 2, Ke Karlovu 3.

Předseda komise pro obhajoby ve vědním oboru ....

Obsah

1 Úvod **4**

2 Vývoj programového zabezpečeníexperimentu ATLAS **4**
2.1 Obecné otázky vývoje softweru 4
2.2 Grafický softwer ve Fyzice Elementárních částic 6
2.3 Současný stav softweru v experimentu ATLAS 6
2.4 "Software Proccess" v experimentu ATLAS 8
2.5 Vývoj grafického softweru v experimentu ATLAS 9
2.6 Vývoj rekonstrukčního algoritmu OPATREC 13

3 Studium fyziky kvarku b v experimentu ATLAS **13**
3.1 Fyzika kvarku b v experimentu ATLAS 13
3.2 Parametr rozmytípři měřenínarušeníCP symetrie a jeho vliv na konstrukci detektoru ATLAS 14
3.3 Možnosti měřenípolarizace baryonů s kvarkem b v experimentu ATLAS 14

4 Resume v anglickém jazyce **16**

5 Resume ve francouzském jazyce **16**

1 Úvod

Autor je jedním ze zakládajících členů experimentu ATLAS. ATLAS [24] je jedním ze čtyř experimentů, které jsou připravovány na urychlovači LHC [25] v mezinárodní laboratoři CERN [26]. Zahájení experimentu ATLAS je plánováno na roky 2005-2007.

Autor se od počátku aktivně podílí na mnoha stránkách přípravy detektoru a to především v oblasti vývoje softweru a studia možností experimentu ATLAS v oblasti fyziky kvarku b.

2 Vývoj programového zabezpečení experimentu ATLAS

Autor je již více než pět let koordinátorem skupiny zabývající se vývojem grafického softweru experimentu ATLAS. Kromě koordinace skupiny autor vyvinul některé klíčové komponenty grafického softweru (kontrolní strukturu, program GraXML, aj.). V rámci koordinace grafické skupiny autor aktivně spolupracuje na vývoji celého programového zabezpečení ATLASu a také se zabývá studiem možností aplikací standardních procedur vývoje softweru v podmínkách experimentu ATLAS.

2.1 Obecné otázky vývoje softweru

Význam programového zabezpečení ve Fyzice elementárních částic roste nebývalou rychlostí. Je to především stále dokonalejší hardwer výpočetních prostředků, jenž nám umožňuje stavět experimenty dávající množství vysoce komplexních dat, které bylo ještě nedávno nemyslitelné zpracovat. Hardwerové možnosti se zdají být takřka neomezené, nicméně jejich reálné využití neroste zdaleka tak rychle jako jejich možnosti. Kapacita našich strojů se za poslední roky mnohonásobně zvýšila - používáme však programy na úrovni osmdesátých a devadesátých let minulého století. Před mnoha lety byla Fyzika elementárních částic jednou z oblastí s nejkvalitnějším a nejmodernějším softwerem, knihovna CERNLIB [27] byla kupříkladu považována za jednu z nejúspěšnějších programových knihoven Fortranu. Pro uspokojení současných nároků však toto tradiční programové vybavení nestačí. Tento fakt se ukázal zcela jasně u experimentů, které se snažily použít staré postupy (a staré programy) pro řešení svých nových problémů. Velmi brzo zjistily, že toto programové vybavení není schopno ony vyšší nároky uspokojit - tradiční postupy dosáhly své hranice. Programové vybavení plánovaných experimentů se stalo čím dále hůře udržovatelným, většina fyziků v podstatě rezignovala na požadavek přesné opakovatelnosti výsledků, počet lidí skutečně schopných existující softwer vyvíjet rapidně poklesl. A k tomu všemu došlo v době, kdy nároky na programové vybavení (jak jeho kapacitu, tak jeho spolehlivost), dosáhly úplně nových úrovní. Budoucí experimenty budou na softweru závislé více než kdykoliv dříve, tento software je však čím dál méně spolehlivý.

Popsaný problém není vlastní pouze Fyzice elementárních částic, každý, kdo se zabývá softwerem v kterékoliv oblasti, se s ním potýká. Byly identifikovány především dva zdroje tohoto problému:

- Neadekvátní technologie užitá pro vývoj programového zabezpečení. Klasické, procedurální a imperativní programovací jazyky jako Fortran nebo C nejsou schopny vyhovět novým požadavkům. Ve skutečnosti nedošlo pouze k pokroku v existujícím hardweru, ale také v softwerové technologii tento hardwer využívající. Hlavní částí této nové technologie, použitelné ve Fyzice elementárních částic, je *Objektově-Orientované programování* používající deklarativní programovací jazyky jako Smalltalk, Eiffel, C++ nebo Java. Nové aplikace napsané v těchto jazycích prokazují jasně jejich vhodnost pro uspokojení rostoucích požadavků kladených na softwer.

- Přechod k novým programovacím technologiím není možný beze změny způsobu organizace vývoje softweru. Nové nástroje jsou mocnější než nástroje používané doposud, jsou však také složitější a náročnější. Již není možné psát softwer chaoticky, bez přípravy a bez organizace. Všeobecnou reakcí na uvědomění si nutnosti změny organizace vývoje softweru bylo vytvoření Procesu pro vývoj softweru (Softwer Process). A protože problém je velký, je i vytvořený process velký. SEI (Software Engineering Institute) [34] vytvořil svůj CMM (Capability Maturity Model) [33] definující stupně, kterými musí organizace projít pro zlepšení svého procesu (a následně svého softweru). Pro usnadnění této cesty bylo vytvořeno mnoho metod. Nejznámější z nich, OMT [35] a Booch [36], se postupně spojily v "metodu všech metod" - RUP (Rational Unified Process) [37]. To je také metoda, kterou se s větším či menším úspěchem snažila a snaží aplikovat většina nových experimentů Fyziky elementárních částic. Tento postup v mnoha případech selhal, pravidla byla pociťována jako příliš omezující a často nebylo jasné, co je jeho účelem a jaký může být jeho přínos. Bohužel, na tomto bodě se aplikace procesu ve většině experimentů zastavila. Experimenty buď odmítly celý proces a snaží se vrátit k původní organizaci (která je v případě nových programovacích technologií ještě méně adekvátní než dříve), nebo se pokoušejí applikovat RUP znovu a "lépe". Vývoj za hranicemi Fyziky elementárních částic však pokračoval. Jako reakce na příliš těžký (heavy-weight) proces RUP bylo vytvořeno několik nových lehkých (light-weight) procesů.

2.2 Grafický softwer ve Fyzice Elementárních částic

Vývoj programů pro zobrazování fyzikálních případů (Event Display) připravovaných experimentů na urychlovači LHC je v mnoha ohledech náročnější než byla příprava odpovídajících programů pro předchozí experimenty. Dvě hlavní příčiny jsou:

- Extrémní hustota dat určených k zobrazování. Fyzikální případy experimentů ATLAS a CMS (včetně případů pozadí) obsahují řádově tisíce drah a miliony prostorových bodů, případy experimentu ALICE jsou ještě bohatší. Je úplně vyloučeno doufat, že zobrazení všech naměřených nebo nasimulovaných dat přinese uživateli jakoukoliv fyzikální informaci o proběhlých procesech či efektivitě rekonstrukce případu. Pro základní studium konstrukce detektoru a rekonstrukce drah je možné použít jednoduchých výběrů dat pro zobrazení (např. dráha se všemi souvisejícími body interakce). Pro komplexní studium efektivity rekonstručních algoritmů to však v žádném případě nepostačuje. Zde je nutné vyvinout složitější postupy.

- Experimenty na LHC jsou velmi nehomogení a mají pouze přibližnou symetrii. Rozlišení vnitřních detektorů se pohybuje v oblasti mikrometrů, vnější detektory se ale nacházejí ve vzdálenosti mnoha metrů. Válcová symetrie detektoru je pouze přibližná. Např. mionový detektor ATLASu má pouze osmiúhelnou symetrii a navíc ne všechny strany mají stejné vlastnosti hlavně kvůli podpůrnému systému a systémům sběru dat. Jednotlivé části vnitřního detectoru zase často nejsou rovnoběžné s hlavními osami detektoru kvůli větší efektivitě rekonstrukce.

K tomu přistupují obecné problémy spojené s vývojem grafických programů pro experimenty Fyziky elementárních částic. Předchozí experimenty pouze výjimečně dokázaly dodat grafické programy, které by byly shledány užitečnými kolektivem fyziků. Problém byl podrobně rozebrán v klasickém článku H. Drevermanna [28] (H. Drevermann je autorem velmi kvalitního programu zobrazení případů - *Dali* - používaného experimentem Aleph).

2.3 Současný stav softweru v experimentu ATLAS

ATLAS je v současné době ve stádiu přechodu od programového vybavení založeného na klasických principech (procedurální a imperativní softwer s použitím převážně jazyka Fortran77) k softweru využívajícímu moderní metody. Jedná se hlavně o nástup Objektově-Orientovaných programovacích jazyků (C++, Java) a vyspělých metod návrhu a vývoje programů (Software Engineering a Software Process). V tomto ohledu se ATLAS takřka neliší od ostatních experimentů podobného typu (ať již běžících - jako experimenty BaBar, D0 nebo CDF, nebo připravovaných - jako CMS, LHCb nebo ALICE).

V ATLASu nyní existují v podstatě dva více-méně nezávislé softwerové systémy. Počátky původního systému sahají do doby prvních studií plánovaného experimentu ATLAS (tehdy pod názvem Eagle). Tento systém je založený na klasických programových prostředcích Fyziky elementárních částic - knihovně CERNLIB [27], simulačním programu Geant3 [41] a analyzačním komplexu HBook/Paw [42]. Velká většina systému je napsaná v jazyce Fortran77, k zápisu geometrie detektoru je použit dialekt jazyka Mortran [43] - jazyk Age [47]. Hlavními součástmi tohoto programového vybavení jsou programy SLUG [44] (zprostředkuje přístup k datovým strukturám programu Geant3), Dice [45] (obsahuje specifické detektorové funkce pro Geant3), Atrecon [46] (provádí rekonstrukci), Atlsim [48] (je druhem interaktivního Atreconu) a Atrig [49] (obsahuje software specifický pro trigger).

Procedurální a imperativní softwer se v poslední době přiblížil k hranicím svých možností. Velká provázanost jednotlivých programových částí, nedostatek celkové architektury a chaotický přístup k vývoji softweru způsobily, že tento softwer je krajně obtížné udržovat a rozvíjet, a také vlastní opakovatelnost výsledků dosažených pomocí tohoto softweru je velmi pochybná. Ukázalo se jasně, že je nutné přejít k modernějším softwerovým konstrukcím, založeným na Objektově-Orientovaném deklarativním přístupu. V podmínkách komunity experimentu ATLAS byla tato cesta navržena a poprvé studována skupinou *Moose - Methodology for Object-Oriented Software Development* v letech 1994-1997. Výsledky této skupiny, spolu s výsledky dalších podobných skupin v společenství Fyziky elementárních částic (např. Geant4) vedla nakonec k přijetí nových postupů celým experimentem. Autor byl jedním z aktivních členů skupiny Moose, přednesl mimo jiné zprávu o jejím stavu na konferenci CHEP'95 v Rio de Janeiru [14]. Závěrečná zpráva skupiny Moose byla publikována v roce 1997 [3].

Nové programy jsou psány v jazycích podporujících Objektově-Orientované programování - C++ a Java. Základní zkušenosti s vývojem Objektově-Orientovaného softwerového systému byly získany při vývoji několika prototypů (Proto II, Arve [50], Paso [29]). Autor přispěl výrazným způsobem k vývoji prototypu Paso. V současné době se vývoj nového softweru vytváří v rámci (framework) Athena [30], který je vyvíjen především Architektonickou skupinou ATLASu na čele s hlavním architektem. Athena vychází z prostředí vyvíjeného experimentem LHCb - Gaudi [31]. Na Athenu navazují a s ní úzce spolupracují další programové balíky, zabývající se specifickými stránkami softwerového systému - simulací, rekonstrukcí, grafikou či databázemi. Autor se podílí na vývoji grafické části Atheny, podílel se na organizaci školení o používání Atheny, kde spolupracoval také jako školitel. Školení proběhlo zatím dvakrát v CERNu a jednou v LBNL-Berkeley. Autor je členem komise zabývající se vyhodnocením současného stavu programu Athena - *ARC - Architecture Review Committee*. ARC bude publikovat své závěry na jaře roku 2001.

2.4 "Software Proccess" v experimentu ATLAS

Pro vývoj původního programového vybavení ATLASu (tj. softweru založeného na procedurálním jazyce Fortran77) nebyl použit žádný specifický process. Pokud bychom přece jen chtěli způsob vývoje tohoto softweru pojmenovat, hodilo by se pravděpodobně nejlépe označení *Chaotic Software Process*. Tento fakt (společně s problémy zmíněnými v předchozí kapitole), tvořil další zdroj situace, která vedla ke složitému stavu současného softweru a k pochopení nutnosti změny.

Nutnost změnit organizaci softweru (chápaná však bohužel často pouze jako změna programovacího jazyka a již méně jako změna programovacího způsobu) byla společenstvím vývojářů a fyziků přijata, pochopení nutnosti změny vlastní organizace vývoje softweru je obtížnější. První pokus jasně definovat proces vývoje softweru v ATLASu byl *Atlas Software Process - ASP* [32]. Autor se podílel na formulaci jedné části ASP - definici pravidel pro přijímání cizího softweru v ATLASu [2]. ASP byl založen na standardních postupech vyvinutých v softwerovém průmyslu - na metodologii OMT [35] a doporučeních SEI [34]. ASP byl úspěšně aplikován v několika oblastech, také při vývoji grafické softweru. Grafický softwer byl několikrát ohodnocen (Review) podle postupů ASP, a to na úrovni Uživatelských požadavků (User Requirements), Návrhu a Implementace. Část ASP věnovaná přijímání cizího softweru do ATLASu je stále v platnosti a byla několikrát aplikována. Přes nespornou užitečnost byl ASP obecně chápán jako příliš těžkopádný a svazující. To je zcela v souladu s trendy v softwerovém průmyslu, kde jsou tyto procesy v současné době často nahrazovány procesy lehčími, více orientovanými na skutečnou vývojářskou praxi. Dva z nich, *FDD - Feature Driven Development* [39] a hlavně *XP - Extreme Programing* [38] autor navrhuje jako vhodné pro uplatnění v rámci experimentu ATLAS. Mnohé rysy procesu XP (jako například budování softweru v krátkých cyklech, častá refaktorizace, absence nesmyslně dlouhého plánování či snaha po nejjednodušší implementaci) se zdají být pro toto prostředí ideální.

Experiment ATLAS, podobně jako další podobné experimenty, vypracoval podrobná pravidla pro používání programovacích jazyků C++ a Java (Coding Rules). Na přípravě pravidel pro jazyk Java se autor výrazně podílel, pravidla byla publikována jako interní zpráva ATLASu [4]. Nedostatečně jsou zatím definována pravidla pro vlastní návrh (Design) Objektově-Orientovaných programů, přičemž tato část má na kvalitu výsledného programu větší vliv než zmíněná programovací pravidla. Ta je totiž často možno automaticky kontrolovat a relativně snadno opravit i v pozdějších fázích vývoje softweru. Je mnohem těžší opravit chyby v návrhu softweru. Autor vypracoval seznam pravidel pro návrh softweru (Design Rules), který hodlá navrhnout ATLASu a publikovat jako interní zprávu ATLASu.

2.5 Vývoj grafického softweru v experimentu ATLAS

Od jara 1996 je autor koordinátorem grafické skupiny ATLASu. Osobně navrhl a implementoval kontrolní strukturu grafiky a některé důležité komponenty, podílí se na vývoji a připojování dalších komponent, jakožto i na integrování grafického systému do programového prostředí ATLASu. Mnoho grafických komponent vzniká ve spolupráci s ostatními experimenty a vývojáři mimo experiment ATLAS. Od počátku jsou při práci skupiny využívány moderní metody pro vývoj softweru, nejdříve byl aplikován process ASP, v poslední době se experimentuje s některými prvky procesu XP.

První prototyp grafického systému ATLASu byl velmi jednoduchý trojdimensionální zobrazovač založený na jazyce Eiffel a využívající jazyk VRML 1.0 pro popis geometrie. Tento systém byl vyvinut v rámci projektu Moose.

Počátkem roku 1996 byl autor na ustavující schůzi grafické skupiny zvolen koordinátorem.

V prvé fázi projektu autor shromáždil rozsáhlý soubor Uživatelských požadavků (User Requirements) [51]. Během sbírání Uživatelských požadavků probíhala paraleleně práce na vyhodnocování existujících grafických programů a vzniklo několik prototypů pro kontrolní strukturu grafiky. Uživatelské požadavky prošly úspěšně vyhodnocením (Review) [6], jehož se zúčastnili odborníci na grafiku jak z experiemntu ATLAS, tak odjinud. Již na počátku bylo učiněno rozhodnutí implementovat grafický systém v jazyce C++.

Na základě sebraných Uživatelských požadavků autor formuloval dva klíčové architektonické principy grafiky pro ATLAS:

- Grafika nesmí být závislá na žádném konkrétním grafickém programu. Musí být možné grafické programy přidávat a měnit bez nutnosti výrazně zasahovat do kontrolní struktury a mostu k celkovému softweru ATLASu. Tento princip plyne především z poznání, že Uživatelské požadavky jsou velmi široké, a budou se bez pochyby nadále vyvíjet, a že také grafické prostředky se vyvíjejí velmi rychlým tempem.

- Skutečnost, že objekty jsou zobrazované grafickým systémem, nesmí mít žádný vliv na návrh a implementaci vlastních objektů. Pokud uživatel grafiku nepoužívá, nesmí být nucen se s ní potýkat.

Formulované architektonické principy pak vedly autora k návrhu (Design) vlastního kontrolního prostředí grafiky. Tento návrh je založen na schématu, které lze stručně popsat takto:

- Uživatel neinteraguje přímo s objekty softwerového prostředí ATLASu. Každá uživatelská interakce je zprostředkována prostřednictvím různých typů reprezentací. Tyto reprezentace nemusí být nutně pouze grafické v úzkém slova smyslu. Možné reprezentace jsou například také textový obraz objektu nebo jeho obraz v databázi.

- Uživatelský objekt, nazývaný `Plottable`, je doprovázen objektem, který zprostředkovává jeho interakci s grafickým systémem. Tento objekt se nazývá `Model` a obsahuje všechny grafické funkce, které nejsou závislé na konkrétním způsobu zobrazování.

- Pro každý podporovaný způsob zobrazování (*View*) je `Model` doplněn dalším objektem, který se nazývá `Representation`. `Representation` reprezentuje `Plottable` v konkrétním zobrazovacím programu, který je připojen pomocí objektu zvaného `Scene`.

- Celý systém je koordinován objektem `SceneList`, který má informace o všech aktivních objektech `Plottable` a `Scene` a který organizuje jejich spolupráci.

Tento návrh byl opět úspěšně hodnocen týmem odborníků z ATLASu i odjinud [7].

Celá architektura byla implementována v jazyce C++, tato implementace byla opět hodnocena, tentokrát pouze týmem pracovníků ATLASu [8]. Popsaný systém je v současné době funkční a je součástí celkového programového prostředí ATLASu.

V druhé polovině roku 1999 skupina pracovala na způsobu řešení některých problémů celého systému. Tyto problémy jsou většinou spojeny s nedostatky jazyka C++, především s chybějící možností prohlížení objektů běžícího programu (Introspection, Reflection). Bylo navrženo několik řešení, z nichž dvě byla podrobena vyhodnocení koncem roku 1999 [9]. První návrh počítal s využitím sofistikovaného a komplikovaného systému *MultiMethod*, druhý návrh spočíval v přechodu k vhodnějšímu programovacímu jazyku - Javě. Z mnoha důvodů se skupina rozhodla následovat druhý návrh a přepsat celý systém v jazyce Java. K tomu došlo v průběhu roku 2000 a nové prostředí je nyní mnohem bohatší a kvalitnější. Nová implementace je založena na stejných architektonických principech a na stejném globálním návrhu (popsaném výše). Celá implementace nového systému byla provedena autorem. Jediná chybějící část je průhledné spojení mezi tímto systémem a Athenou (která používá jazyk C++). Autor spolupracuje na implementaci tohoto spojení, které bude mít dosah i mimo grafické prostředí. V souvislosti s přechodem grafického systému z jazyka C++ na jazyk Java se autor podílel na návrhu použití jazyka Java v ATLASu, který byl publikován jako vnitřní zpráva ATLASu pod názvem *On the Importance of Java to ATLAS* [5].

V principu jakýkoli grafický program může být připojen ke grafickému systému. Je však pochopitelné, že v každém okamžiku bude připojen jen konečný počet programů, tak aby byly uspokojeny aktuální Uživatelské požadavky. V současné době jsou k dispozici tyto hlavní připojené programy:

Text umožňuje zápis objektů do textového souboru. Je to nejjednodušší komponenta a je vhodná pro ladění jak grafického prostředí tak používaných objektů. Tato komponenta podporuje všechy objekty, nejenom ty, které jsou vybaveny reprezentacemi. Komponenta byla vytvořena autorem.

Test slouží k přímému testování jak grafického prostředí, tak použitých objektů. Vlastní "zobrazení" spočívá v testovací akci aplikované na příslušný objekt. Komponenta byla vytvořena autorem.

XML je podobná komponentě Text s tím rozdílem, že vytváří reprezentaci objektů ve formátu XML. Vytvořené soubory lze pak použít jako vstupní soubory dalších komponent nebo programů (např. Atlantis, WIRED, GraXML nebo JAS). Vytvořené soubory mohou být několika typů: *AGDD* [52] pro zápis geometrie detektoru, *AtlasEvent* či *HepRep* pro zápis vlastních fyzikálních případů nebo *AIDA* (viz níže) pro zápis analyzačních dat (histogramů, n-tuplů, apod.). V této souvislosti je vhodné zmínit, že AGDD je jeden z nejúspěšnějších softwerových projektů v ATLASu a idea použití jazyka XML pro účely zápisu geometrie detektoru byla navržena a prosazena autorem. Autor také aktivně spolupracoval na definici a implementaci AGDD. Tato idea je nyní všeobecně přijímána nejen v experimentu ATLAS, ale také například v experimentech LHCb, GLAST a v projektu Geant4.

Aravis [53] je jediná komponenta napsaná v jazyce C++. Vznikla extrahováním grafické části systému Arve [50]. Její hodnota spočívá v úzké integraci s prostředím C++, což umožňuje snadné ladění algoritmů zapsaných v C++.

Atlantis/Jive [54] je verze programu Dali (viz výše) upravená pro potřeby Atlasu. Atlantis je v současné době beze sporu nejkvalitnější program pro zobrazování a analýzu celých případů ATLASu. Jsou v něm k dispozici sofistikovaná zobrazení umožňující zvýraznit důležité vlastnosti případu (Fish-Eye, V-Plot, Clock-Transformation,...) a provést inteligentní filtraci. Atlantis je zapsán v jazyce Fortran a je v současné době přepisován do jazyka Java (pod jménem Jive).

WIRED [10] (World Wide Web Interactive Remote Event Display) je program v mnohém podobný programu Atlantis. V současné době není tolik funkční, ale má jasnější modulární strukturu a je snazší k použití.

GraXML [11] je program pro realistické trojdimensionální zobrazování detektoru a částí případů (jednotlivých drah a souvisejících interakcí). Je založen na knihovně Java3D a byl kompletně vytvořen autorem. GraXML je schopen uspokojit většinu Uživatelských požadavků kladených na takovýto druh programu. Umožňuje globální i lokální operace na grafických objektech, interakci se skutečnými vzory (takže lze například opravit rekonstrukci dráhy po vyjmutí některých interakcí) a mnoho dalších operací. Unikátní je možnost zapsat obraz vytvořený knihovnou

Java3D do formátu VRML - autorovi není znám žádný jiný program s podobnou možností. Formát VRML lze následně použít v některém programu pro tvorbu foto-realistických obrazů (Ray-Tracing). Program GraXML není specifický pro ATLAS, v současné době ho lze například použít pro zobrazení kompletního detektoru ALICE. Mnoho komponent programu GraXML se může dále použít v rámci jiných prostředí - tyto komponenty byly a jsou přesouvány do knihovny FreeHEP (viz níže). Jde zejména o trojdimensionální representace grafických objektů programu Geant4, podporu interaktivity a vnitřní organizaci korespondence grafických objektů se skutečnými objekty.

Object Browser [12] umožňuje snadno prohlížet všechny objekty, které existují v běžícím programu. V tomto případě opět není nutné aby byl objekt vyzbrojen svou reprezentací, standardní reprezentace je automaticky použita v případě potřeby. Object Browser nabízí zobrazení objektů podobné zobrazení souborů ve standardních souborových prohlížečích (známých například z prostředí MS Windows). Objekty jsou zobrazeny jako soubory, skupiny objektů jako adresáře. Je možné prohlížet a v budoucnu bude možné i měnit hodnoty uvnitř objektů, a také volat jejich metody (Context-Sensitive Actions). Je studována také možnost spolupráce s ostatními programy na principu "Drag-and-Drop", aby uživatel mohl snadno poslat označené objekty do jiné grafické komponenty. Object Browser byl původně vytvořen Lassim Tuurou, ale byl následně přepracován a je dále rozvíjen autorem disertace. Jedná se také o další příspěvek do knihovny FreeHEP.

AIDA [56] (Abstract Interfaces for Data Analysis) je standartní rozhranní (API - Application Programming Interface) pro analýzu dat. Definuje standard pro objekty jako histogram, či n-tuple. Autor stál u zrodu tohoto standartu jako člen organizačního výboru konference HEPVis'99. Grafický soubor ATLASu umožňuje spolupráci s ostatními programy, které používají standard AIDA.

JAS [55] (Java Analysis Studio) je v současné době jeden z nejkvalitnějších programů pro analýzu dat. Je kompletně napsán v jazyce Java, umožňuje analyzovat data zapsaná v mnoha formátech (mj. HBook [42], Root [57] nebo Objectivity) a nabízí některé vlastnosti, které nejsou k dispozici v žádném jiném programu (jako například dynamické rebinování histogramů). Autor spolupracuje na připojení programu JAS do celé grafické struktury ATLASu.

V minulém roce vznikla skupina koordinující spolupráci vývojářů pracujících v Jazyce Java pro Fyziku elementárních částic - FreeHEP [13]. Autor je členem této skupiny. Skupina se stará o knihovnu komponent napsaných v jazyce Java, které mohou být použity dalšími uživateli. Autor přispěl do knihovny FreeHEP některými komponentami programu GraXML a Object Browser. S pomocí knihovny FreeHEP je vytvářen program pro všeobecnou analýzu dat ve Fyzice Elementárních Částic - HEP Studio. Prvními komponentami tohoto programu jsou WIRED a JAS. Další komponenty jsou integrovány. Jde především o části programů GraXML, Object Browser a v budoucnu Atlantis/Jive. Součástí knihovny FreeHEP (a programu HEP Studio) je i podpora pro spolupráci komponent zapsaných v jazycích C++ a Java. Jde o program JACO (Java Access to C++ Objects).

Autor referoval o grafickém systému ATLASu na několika mezinárodních poradách: HEPVis'96 v CERNu [16], HEPVis'98 ve SLACu [17], LHCC'98 v Barceloně [18], LCB'99 v Marseille [19], byl také spoluautorem dalších konferenčních vystoupení souvisejících s jeho prací pro grafiku ATLASu na konferenci CHEP'00 v Padově [15] a Acad'00 ve Fermilabu [20].

2.6 Vývoj rekonstrukčního algoritmu OPATREC

Autor vyvinul rychlý algoritmus pro rekonstrukci drah ve vnitřním detektoru. Algoritmus je založen na transformaci experimentálních dat souřadnic zasažených detektorů na parametry možných drah. Vlastní rekonstrukce se tak převede na problém hledání skupin (clusters) těchto parametrů, pro který existuje podstatně jednodušší a rychlejší algoritmus.

Prototyp algoritmu OPATREC byl autorem implementován v jazycích Fortran a Eiffel.

3 Studium fyziky kvarku b v experimentu ATLAS

V rámci pražské skupiny experimentu ATLAS se autor podílí na přípravě fyzikálního programu experimentu ATLAS v oblasti fyziky kvarku b. Spolu se svými pražskými kolegy autor organizoval simulace případů této fyziky. Z hlediska fyziky kvarku b se autor podílel na studiu rozdílů mezi různými konfiguracemi detektoru ATLAS a na originální práci oceňující možnosti experimentu ATLAS pro studium polarizace těžkých baryonů.

3.1 Fyzika kvarku b v experimentu ATLAS

Velký hadronový účinný průřez tvorby kvarku b na LHC povede k tomu, že ATLAS bude poskytovat velké množství hadronů obsahujících tento kvark (B-hadronů). Přibližně každá stá srážka poskytne jeden pár kvarků b, což je výrazně lepší výsledek než na experimentech Tevatronu. Inkluzivní mionový trigger s prahem $p_T > 6$ GeV povede k 25% případů obsahujících kvark b. V prvním roce tak dostaneme zhruba 10^9 takovýchto případů. To je více než poskytne kterýkoliv jiný experment do té doby.

Základním účelem fyziky B-hadronů na ATLASu je testování Standardního Modelu prostřednictvím přesného měření rozpadů těchto hadronů. Tyto rozpady přeurčí Cabibbo-Kobyashi-Maskawa matici CKM a mohou též poskytnout nepřímé signály nové fyziky.

Plný program fyziky kvarků b na ATLASu obsahuje:

- Přesné měření narušení symetrie CP v rozpadech B-mezonů.
- Přesné měření oscilací B_s^0 a B_d^0 a relativních poměrů rozpadů.
- Hledání zřídkavých rozpadů potlačených ve standardním modelu.
- Přesné měření B-hadronů, včetně polarizace Λ_b. Toto měření bylo navrženo a rozpracováno členy pražské skupiny ATLAS včetně autora.

Mnohá z těchto měření mohou být na experimentu ATLAS velmi přesná díky velké statistice, která bude k dispozici.

Autor se podílel na přípravě a organizaci simulace případů fyziky kvarku b na ATLASu. Během této simulace bylo nagenerováno asi 1.8 milionu případů v různých laboratořích spolupracujících na experimentu ATLAS. Zhruba 25% z nich bylo dále podrobeno plné simulaci detektoru. Nasimulovaná data byla využita ve většině studií fyziky kvarku b v ATLASu. Celý postup byl publikován jako vniřní zpráva ATLASu [1], autor je jedním ze spoluautorů této práce.

3.2 Parametr rozmytí při měření narušení CP symetrie a jeho vliv na konstrukci detektoru ATLAS

Bylo nasimulováno 24 000 případů $B_d^0 B \to J/\psi K^0 \mu X$ a byla spočítána četnost dN/dt rekonstruovaných případů pro různé možné varianty detektoru ATLAS (vzduchový toroid nebo železný toroid, elektronový nebo mionový tracking, plný nebo redukovaný detektor). Dále byla odvozena závislost parametru rozmytí při měření narušení CP symetrie na p_t mionu, která pocházela z jeho chybného označení. Bylo zjištěno, že pro $p_t > 6$ GeV je tento faktor na p_t nezávislý a rovný $D = 0.82 \pm 0.04$. Jako důsledek je chyba parametru narušení CP asymetrie $sin(2\phi)$ závislá na p_t pouze prostřednictvím dN/dt.

Získaných výsledků pro různé varianty detektoru bylo využito při rozhodování a jeho definitivní konstrukci. Práce byla publikována jako vnitřní zpráva ATLASu [21]. Autor je jedním ze dvou spoluautorů této práce.

3.3 Možnosti měření polarizace baryonů s kvarkem b v experimentu ATLAS

Autor byl spolunavrhovatelem možnosti studia polarizace baryonů s kvarkem b v experimentu ATLAS prostřednictvím analýzy úhlového rozdělení při postupném rozpadu $\Lambda^0 J/\psi \to p\pi^- l^+ l^-$.

Měření polarizace Λ_b^0 a Ξ_b^0 může pomoci vyjasnit problémy s vysvětlením velké polarizace $\overline{\Xi^+}$ a $\overline{\Sigma^-}$ naměřené ve Fermilabu. Důležité je zejména měření závislosti polarizace na hmotě těžkého kvarku, kdy do hry vstupuje několik možných efektů a není jasné, jaký je jejich relativní vztah.

Bylo ukázáno, že ATLAS může naměřit polarizaci Λ_b^0 s přesností lepší než 0.016 a polarizaci Ξ_b^0 s přesností lepší než 0.28.

Výsledky byly publikovány v J.Phys G;[22] a v minulém roce byla nová verze celé práce přijata jako jedna kapitola dokumentu *ATLAS Detector and Physics Performance Technical Design Report* [23]. Publikované výsledky byly nyní upřesněny s využitím nejnovějších experimentálních dat.

4 Resume v anglickém jazyce

Theses cover two fields of the preparation of the ATLAS experiment in CERN: the development of the software system and the physics of the B-quark. Naturally, only subsets of all topics are covered. In the software part, the Software Process and the development of the Graphics software are described. A note about the importance of Java to ATLAS is added in the Appendix. The physics part contains the study about the CP Asymmetry Dilution Factor dependency on the ATLAS detector setup and the study of the Beauty Baryon Polarization.

5 Resume ve francouzském jazyce

La thèse couvre deux zones de la préparation de l'expérience ATLAS au CERN: le développement du système logiciel et la physique du B-quark. Naturellement, seulement des sous-ensembles de tous les sujets sont couverts. Dans la partie de logiciel, le procès de logiciel et le développement du logiciel de graphiques sont décrits. La note au sujet de l'importance de Java à l'ATLAS est ajoutée dans l'annexe. La partie de physique contient l'étude au sujet de la dépendance de facteur de dilution d'asymétrie de CP sur l'installation de détecteur d'ATLAS et l'étude de la polarisation de Baryon de beauté.

Autor disertace je autorem či spoluautorem prací [1] až [23].

Reference

[1] **Overview of Monte Carlo simulations for ATLAS B-physics in the period 1996-1999**,
ATL-PHYS-2000-025

[2] **Handling of External Software in ATLAS**
http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/asp/external/

[3] **Final report on MOOSE: Object Oriented Software Development for LHC Experiments**
CERN-LHCC-97-042

[4] **Java in Atlas - coding standards**,
ATL-SOFT-2000-002

[5] **On the Importance of Java to ATLAS**,
ATL-SOFT-2000-004

[6] **ATL-SW-Atlas-Requirements-V1**,
http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/deliverables/ATL-SW-Atlas-Requirements-V1/

[7] **ATL-SW-Graphics-Design-V1**,
http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/deliverables/ATL-SW-Graphics-Design-V1/

[8] **ATL-SW-Graphics-Code-V1**,
http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/deliverables/ATL-SW-Graphics-Code-V1/

[9] **Review of the Graphics Control Design Upgrade**,
http://atlas.web.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Upgrade99/

[10] **WIRED**,
http://atlas.web.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/WIRED/,
J.Hřivnáč, **WIRED (& Atlas)**, 3rd LHC Computing Workshop in Marseilles, 1999

[11] **GraXML**,
http://atlas.web.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/GraXML/

[12] **Object Browser**,
http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/Graphics-/JavaDoc/net/hep/atlas/Graphics/ObjectBrowser/package-summary.html

[13] **FreeHEP**,
http://java.freehep.org/lib/freehep/doc/web/index.html

[14] **Object Oriented Software for the ATLAS experiment**,
CHEP'95 in Rio de Janeiro. by J.Hrivnac et al

[15] **A generic approach to the detector description in Atlas**,
CHEP'00 in Padova, by Ch.Arnault et al

[16] **Event Display at ATLAS**,
HEPVis'96 in CERN, by J.Hrivnac

[17] **ATLAS Graphics**,
HEPVis'98 in SLAC, by J.Hrivnac

[18] **ATLAS Graphics Design**,
LHCC'98 in Barcelona, by J.Hrivnac

[19] **WIRED & ATLAS**,
LCB'99 in Marseille, by J.Hrivnac

[20] **The FreeHEP Java Library**,
Acad'00 in Fermilab, by T.Johnson et al

[21] M.Smižanská, J.Hřivnáč, $\mathrm{B_d^0}$ B $\to \mathrm{J}/\psi$ $\mathrm{K^0}$ μ X **reaction: rates, dilution factor and the error on $\sin(2\phi)$ as a function of trigger p_t threshold**, ATLAS Internal Note, PHYS-NO-022,1993

[22] **Feasibility of beauty baryon polarization measurement in Λ^0 J/ψ decay channel by ATLAS-LHC**
, by Hrivnác, J ; Lednický, R ; Smizanská, M ; (12 p) . 1994 . Publ. Ref: J. Phys. G: 21 (1995) 629-683

[23] **ATLAS Detector and Physics Performance Technical Design Report**, CERN/LHCC/99-15,
http://atlasinfo.cern.ch/Atlas/GROUPS/PHYSICS/TDR/access.html

[24] **A Toroidal LHC ApparatuS**,
http://atlasinfo.cern.ch/Atlas/Welcome.html

[25] **The Large Hadron Collider**,
http://lhc.web.cern.ch/lhc/

[26] **CERN**,
http://cern.web.cern.ch/CERN/

[27] **The CERN Programm Library for Fortran77**,
http://wwwinfo.cern.ch/asd/cernlib/

[28] **Is there a Future for Event Display ?**,
H.Drevermann et al, CSC'92

[29] **Provisional Analysis Skeleton for OO-ATLAS**,
http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/OO/applications/Paso/

[30] **Athena**,
http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/

[31] **Gaudi**,
http://lhcb-comp.web.cern.ch/lhcb-comp/Components/html/GaudiMain.html

[32] **The ATLAS Software Process**,
http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/asp/

[33] **The Capability Maturity Model**,
http://www.sei.cmu.edu/cmm/cmms/cmms.html

[34] **The Carnegie Mellon Software Engineering Institute**,
http://www.sei.cmu.edu/

[35] James Rumbaugh, **Object Oriented Modelling and Design**, ISBN:0-13-630054-5

[36] Grady Booch, **Object Oriented Modelling and Design with Applications**, ISBN:0-8053-5340-2

[37] **The Rational Unified Process**,
http://www.rational.com/products/rup/index.jsp

[38] **The Extreme Programming**,
http://www.xProgramming.com/,
http://www.extremeprogramming.org/

[39] **The Feature Driven Programming**,
Coad, De Luca, Lefebrve, **Java Modeling in Color with UML**, Prentice Hall 1999

[40] **Geant4**,
http://wwwinfo.cern.ch/asd/geant4/geant4.html

[41] **Geant3**,
http://wwwinfo.cern.ch/asd/geant/index.html

[42] **Physics Analysis Workstation**,
http://wwwinfo.cern.ch/asd/paw/

[43] **MORTRAN**,
http://www.slac.stanford.edu/exp/e871/documentation/offline/mortran.html

[44] **SLUG**,
http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS-/SLUG_DOC/slug.html

[45] **DICE**,
http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS-/DICE_320/dice320.html

[46] **ATRECON**,
http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS-/reconstruction.html

[47] **Age**,
http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS-/DICE95/AGE.html,

[48] **ATLSIM**,
http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS-/simulation.html#atlsim

[49] **ATRIG**,
http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/ATRIG/

[50] **The ATLAS Reconstruction and Visualisation Enevironment**,
http://burnett.home.cern.ch/burnett/arve/

[51] **Graphics Users Requirements**,
http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/-Graphics/Requirements/

[52] **The ATLAS Generic Detector Description**,
http://web1.cern.ch/Atlas/GROUPS/DATABASE/detector_description/,

[53] **Aravis**,
http://atlas.web.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/Aravis/

[54] **Atlantis**,
http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/Atlantis/

[55] **The Java Analysis Studio**,
http://atlas.web.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/JAS/

[56] **The Abstract Interfaces for Data Analysis**,
http://wwwinfo.cern.ch/asd/lhc++/theProject/index.html

[57] **ROOT**,
http://root.cern.ch