氏名	Kazakov Artem		
学位（専攻分野）	博士（工学）		
学位記番号	総研大 1279 号		
学位授与の日付	平成２１年９月３０日		
学位授与の要件	高エネルギー加速器科学研究科　加速器科学専攻 学位規則第６条第１項該当		
学位論文題目	Reliable Control System for Future Particle Accelerators		
論文審査委員	主査	准教授	山本　昇
		教授	金子　敏明
		准教授	朴　哲彦
		教授	藤井　啓文
		准教授	中村　典雄（東京大学）
		准教授	古川　和朗

論文内容の要旨

Modern particle accelerator machines are complex and large scale structures. Large projects like Large Hadron Collider and International Linear Collider (ILC) consist of thousands of components that are spread over big distances in underground tunnels. Machines of that scale and complexity raise a set of challenges for all subsystems of the accelerator. With constantly growing size and complexity of particle accelerators the role of the control system becomes more and more important for a successful operation. One of the biggest concerns for large machines is availability. Because of a huge number of components, even very reliable components, final availability of the accelerator might suffer of continuos failures in one of the subsystems.

For example target availability for the International Linear Collider is 75%, but in order to achieve that, the control system has to be available for 99-99.9% of time (15 hours of down time is "allocated" for the control system of the ILC. See Table 1). Design draft specifies that the ILC control system will consist from ~1200 "crates", and that translates into 99.999% availability for each crate. Such availability has not been a requirement for present accelerator control systems. Therefore it sets a new challenge for control system designers, implementors and operators. A multilevel systematic approach should be taken in order to achieve these availability goals.

Lets analyze availability indicators for current accelerators. Typical high energy physics accelerator currently has an availability of 75-85%. Though there are some examples of much better availability: Pohang Light Source (2008) - 97% (with controls responsible for 2% of downtime); SOLEIL light source (2007) - 95.7% (with controls responsible for 2.7% of downtime); KEK Linac (2008) - 98.3% (with controls responsible for 13.3% of downtime). For KEK Linac it means that control system availability was around 99.76%. The KEK Linac control system consists of 30 VME crates, 150 PLCs, 30VXI, 15 CAMAC, 24 intelligent oscilloscopes. ILC control system is 1195 ATCA crates, 8356 network switches and thousands of other lower level control components. ILC control system has 10 ~ 100 times more components than KEK Linac control system. Such tremendous increase in a number of components will dramatically reduce the availability indicators for the control system. Therefore availability issues have to be seriously considered for the future particle accelerator control systems. This research was devoted to that particular goal.

The first chapter provides some introductory information regarding accelerator control systems, historical overview of the control system evolution and a modern view on building control systems.

The second chapter of this work describes different approaches to improve availability of a particle accelerator. The general reliability theory is briefly introduced. Then the applications of that theory to accelerator control system are discussed. An accelerator control system can be roughly separated into four major parts: hardware, software, humans and procedures. Analysis is done for each of these four components. Each of these parts requires different approaches in order to achieve high availability. This work covers improvement of the software and hardware components. Hardware reliability is improved through implementation of redundancy, and software reliability is improved through implementation of a test system, that ensures the software quality. Further chapters provide more detailed explanation on how these goals are met.

Chapters 3,4,5 describe my contribution to improve reliability of accelerator control system. This work is mostly concentrated on improvement of software and hardware components using EPICS software. EPICS stands for Experimental Physics and Industrial Control System. It has more than 15 years history of usage and has been being developed during all these years. It is widely used in many accelerator laboratories all over the world, including KEK, where it is a basis for the KEKB control system.

Chapter 3 describes the EPICS redundant IOC. In order to achieve high availability (such as 99.999%), redundancy is essential (as discussed in Chapter 3, section 3.1, 3.2). Redundancy allows to reduce time needed to recover from a failure to a few seconds or milliseconds, instead of hours and days. The system is not stopped because of the failure and the stand-by component starts to operate immediately after the failure is noticed. The redundancy approach is a common technique used in highly available applications (more than 99.999% availability). The original EPICS software distribution lacked redundancy support. This issue was addressed by developing EPICS redundant IOC. The initial design and development was made by DESY. Unfortunately from the very beginning only vxWorks support was looked for. Later it was realized that other OS support is needed as well. As a part of my research, in collaboration with DESY, I generalized the

redundant EPICS IOC to Linux, Darwin, and other operating systems. The generalization was done using the Operating System Independent (OSI) library, therefore the ported version should work on any platform, where the OSI library is fully implemented. The generalized redundant IOC is an important improvement to the EPICS control system framework. Several serious software bugs were fixed in the original Redundancy software. The result of this work is a very important improvement to the existing RIOC implementation. First, it allowed to use RIOC on many operating systems, such as Linux and MAC OS X, therefore providing much wider application field for the RIOC. Second, it allowed to include the support for the RIOC into the official EPICS distribution from version 3.14.10. Third, working on this project resulted in modification and splitting the original software into several libraries which can be used independently. An example of such usage is provided in the next chapter, describing the implementation of redundant and load-balancing Channel Access gateways. In chapter 6 the generalized version of the RIOC is extended to support the Advanced Telecom Computing Architecture (ATCA) platform. These projects would have been impossible without the generalization of the original RIOC and improvements done during this work.

As mentioned above using the generalized versions of RIOC libraries, Channel Access Gateway was made redundant and load-balancing. This new and original development is described in Chapter 4. Channel Access gateways are in operation in many places. They allow separating control networks into several administrative subnetworks. Also they can be used as a security tool: providing restricted access to the control network, for example read-only access from public networks. Besides this administrative and security aspects gateways also optimize the number of Channel Access (CA) connections to the IOCs, because several CA clients can share one connection to an individual IOC. Due to these important functionalities gateways play a growing role in today's installations. Performance and functionality have been continuously improved over the last years. The availability of this service is key for machine operations in many places. This was the driving force to implement redundancy also for the CA gateways. The redundant and load-balancing Channel Access Gateways were implemented within this research. The development was done using the generalized version of RIOC libraries discussed in the chapter 3. The implementation of the redundant CA gateway allowed to escape the single-point of failure, and by introducing the load-balancing the performance and throughput was improved in the number of 2. Load-balancing version of the CA gateway brings availability improvements as well, due to the fact that half of the connections are handled via the secondary gateway, these connections will not be affected when the failure occurs on the primary gateway.

In chapter 5 new hardware standard ATCA and its application to control system are described. Within this research a support for Advanced Telecommunication Computing Architecture (ATCA) was added to the RIOC software. This addition, called ATCA-driver, allows to monitor the ATCA-hardware and provide better availability. This driver can help predict hardware failures and allows to decrease the Channel Access clients reconnect time from 30 to 2 seconds. Using reliable software in conjunction with reliable hardware can give us even more reliable solutions. Recently Advanced Telecommunication Computing Architecture standard is gaining attention in High Energy Accelerator field as platform for modern controls and Data Acquisition (DAQ) systems. ATCA is an open standard developed by consortium of telecom vendors and users; and from its very early days it is aimed to high reliability, high bandwidth and modularity. Nowadays it is widely used in telecom industry and is widely supported by many big vendors. The ILC control system requirement for a single control shelf is 99.999%. ATCA hardware available on the market provides this level of availability or better, therefore ATCA was suggested to be used in ILC control system. Even though ATCA provides redundant cpu boards, power supplies, interconnections and other facilities, redundant IO boards and software that can work with ATCA hardware and utilize its capabilities must be developed.

For the reasons mentioned above I developed a driver for the Redundant EPICS IOC (RIOC) which provides support for ATCA. Using the Hardware Platform Independent library (HPI) it allows the RIOC to monitor the status of the hardware it's running on. Using this information, fail-over decisions can be made even before the "real" failure happens. For example, if the temperature starts to rise there is some delay until system crashes because of overheating. During that time the fail-over sequence can be triggered. Therefore the fail-over happens in a more stable and controlled environment. An obvious and very important benefit is that client connections can be gracefully closed and clients would reconnect to the stand-by IOC within 2 seconds. In case of a real hardware failure it would take up to 30 seconds (default Channel Access timeout).

This ATCA RIOC driver can be also utilized on any computer which has an HPI support. Therefore providing increased availability for the platforms other than ATCA. For example modern computer severs are usually equipped with the temperature, voltage etc. monitoring hardware. OpenHPI distribution of HPI supports such hardware on Linux operating system.

Chapter 6 presents another approach to improve the reliability - by means of improving the quality of the software. An important part of this process is the software testing and quality assurance. In early years EPICS supported only one operating system for the server side - vxWorks, and one operating system for the client side - Sun OS; and it was well tested at Argonne National Laboratory. But in recent years EPICS has gained support for many operating systems and hardware platforms and now it supports more than 10. Each institution uses its own collection of OS+EPICS running on different hardware. Most of these combinations are not very well tested, due to a lack of convenient, easy to use and reliable system integration testing mechanism. Therefore it leads to a potentially dangerous situation when an untested and unproved software is used for the operation. Obviously a decent automated test system is needed for EPICS software distribution. EPICS has a decent unit-test system included in the base distribution. It has been continuously extended by core-developers as EPICS evolve and new features have been added. Basically a unit-testing is a testing of small pieces (functions) of code, to check that they perform correctly. But it does not mean that these pieces would work when combined together. For that purpose there is another test package, which consists of system tests, when real IOC's are installed on distributed machines and then it is checked if these systems perform correctly altogether.

Originally that system test package for EPICS consisted of several programs/IOCs and text file instructions how to run them. But it is not convenient and takes a lot of time for developers and users to understand how to run these tests, prepare different machines, upload, configure and start the IOCs, preform a test and compare the results. As a part of my research I have developed a system that automates the process of a system testing and system integration testing, and provides a flexible environment to create these tests. This newly developed software supports a wide variety of configurations by default and can be easily configured using simple configuration file. It is developed in the high-level object oriented scripting language Ruby, which makes it easy to extend and add new functionality. Usage of this automated test systems greatly simplifies the testing process. It allows to run a predefined set of tests on a predefined set of computers in a fully automated manner. It requires to create a configuration file, specifying the computers and corresponding test that must be performed. Then only one command must be issued by human to run all the tests. If compared to manual testing, it saves tremendous amount of human time and effort. Due to automation chances for human error are greatly reduced too. After all tests are executed the system provides a detailed report.

博士論文の審査結果の要旨

本研究は大規模化する一方で、安定した運用を求められる高エネルギー加速器制御システムの信頼性向上についての研究である。申請者はこのような信頼性の高い加速器制御の構築の問題に取り組んだ。申請者は、高信頼性（高可用性）システムを構築する上での一般的な議論から、ハードウェア、ソフトウェア、人的要因、手続きの四つの要因について検討を加えている。本研究では特にこれらの四つの要因の内ハードウェア、ソフトウェアにおける高信頼性システム構築のための手法について研究した。高可用性実現のためにはシステムに冗長性を持たせることにより、Sigle point of failure を排し、実質的なシステムの利用不能な時間を減少させることが有効である事を議論した後、具体的に加速器制御システムに冗長性を導入するための基本的ソフトウェアを構築しその有効性を稼働中の加速器制御システムで実証した。

申請者は本研究において、DESY にて開発が進められていた冗長入出力制御装置（Redundant Input Output Controller)のソフトウェア研究に参加し、ソフトウェアの中心的な構成要素である RMT (Redundant Monitoring Task)ライブラリの整備を担当した。申請者によって、RMT はプラットフォーム非依存のライブラリとして完成された。また、このライブラリを用いた Redundant IOC は KEK LINAC 制御システムで実用的に稼働する事が実証されている。また、RMT などのライブラリは Redundant IOC が基礎とする EPICS フレームワークの公式配布ソフトウェアの一部として取り込まれており、今後の加速器等の制御システムの高可用性化の基礎となるものと期待される。

申請者はまた、この RMT が Redundant IOC に限定される事無く、様々な応用アプリケーションに冗長性を持たせることが可能であることに着目し、加速器制御システムの構築フレームワークとして広く採用されている EPICS での、CA gateway ソフトウェアに冗長性を追加する事が、CA gateway のソースコードに変更を加えることなく可能である事を実証した。また、この機構を応用することで CA gateway に負荷分散機能を追加することが可能であることを発見し、これにより単なる冗長性以上の高可用性を実現できることを実証した。

申請者はまた、ATCA(Advanced Telecom computing Architecture)などの高可用性をサポートするハードウェア機器においては、これらの機構と RMT を連携させることによって高可用性の向上が可能であることを指摘し、それを実証した。

高可用性のソフトウェアの開発に置いては、ソフトウェア開発中のテストが重要であるが、EPICS などの制御ソフトウェアフレームワークでは、既存のソフトゥエアテストフレームワークでは十分なテストが実施できないことを申請者は指摘した。制御システムにおいては、複数のプラットフォームとその組み合わせについてテストを行う事が必要であることを指摘し、これらの状況でも柔軟にシステム統合テストを支援するためのツールを開発している。

本論文は以上に述べた様に、加速器制御システムの高可用性の実現について様々な局面から検討を加え、高可用性実現のための手法を具体的に提示しまた実証している。と

くに Redundant CA Gateway の実現およびそれに伴う付加分散機能の追加は申請者の着眼点の独自性を示しており、今後の高信頼性システムの構築に重要なものとなる。

以上の事から判断して、本論文は加速器制御システムの研究および今後の高信頼性加速器制御システムの実現に重要であり、学位論文としてふさわしいものである判断される。審査員会は全員一致で本論文を合格と判定した。