Implementation and evaluation of kinematic mechanism modeling based on ISO 10303 STEP

Yujiang Li

Master Thesis
Royal Institute of Technology
School of Industrial Engineering and Management
Department of Production Engineering
Stockholm, Sweden 2011
Abstract
The kinematic mechanism model is an important part of the representation of e.g. machine tools, robots and fixtures. The basics of kinematic mechanism modeling in classic CAD are about to define motion constraints for components relative other components. This technique for kinematic modeling is common for the majority of CAD applications, but the exchange of kinematic mechanism between different CAD applications have been very limited. The ISO 10303 STEP (STandard for the Exchange of Product data) addresses this problem with the application protocol ISO 10303-214 (AP214) which provides an information model for integration of the kinematics with 3D geometry. Several research projects have tried to, as subtasks, implement the kinematic functionality of AP214. But until recently no one have been able to create a valid dataset to prove the standard’s applicability.
In this research, a framework for integration of the STEP-based kinematic mechanism modeling with existing commercial CAx systems is presented. To evaluate how well the framework is able to be applied for industrial practices, two applications are developed to demonstrate the major outputs of this research. A pilot standalone application, KIBOS (kinematic modeling based on STEP), is developed to prove the feasibility for Java-based STEP implementation on kinematics and to provide comprehensive operation logic to manipulate the kinematic features in STEP models based on ISO/TS 10303-27 Java binding to the standard data access interface (SDAI). And then, an integrated application based on STEP and Siemens NX, KIBOS for NX, is developed to demonstrate the translation and operation of kinematic modeling between Siemens NX and STEP.
The result of this research provides the industrial practitioners with an implementable framework for kinematic modeling exchange between different CAx systems, the IT venders can enhance their products with STEP based kinematic modeling or data exchange in addition to their current support for standard geometric modeling, and the developed application can assist academic researchers to create STEP files with valid kinematic mechanism. The major outputs of this research, KIBOS and KIBOS for NX, can be, and has been, used and extended in the research project Digital Factory Building Blocks at KTH.
# Table of Contents

Abstract .......................................................................................................................... i  
Table of Contents ....................................................................................................... iii  
List of Figure ................................................................................................................ v  
List of table ................................................................................................................... v  
1 Introduction ......................................................................................................... 1  
   1.1 Objective ..................................................................................................... 1  
   1.2 Background ................................................................................................. 1  
      1.2.1 Product data exchange ......................................................................... 1  
      1.2.2 Manufacturing resources modeling .................................................... 3  
2 Literature review ................................................................................................. 7  
   2.1 STEP ............................................................................................................ 7  
   2.2 Kinematic modeling .................................................................................. 9  
3 Research approach ............................................................................................ 11  
4 Kinematic modeling based on STEP ............................................................ 13  
   4.1 Introduction .............................................................................................. 13  
   4.2 Requirement analysis ............................................................................... 13  
      4.2.1 Problem description: objective, scope, and feature ...................... 13  
      4.2.2 Function design ................................................................................ 14  
   4.3 System design ........................................................................................... 15  
   4.4 Implementation ........................................................................................ 17  
5 Kinematic modeling based on STEP and Siemens NX ............................. 21  
   5.1 Introduction .............................................................................................. 21  
   5.2 System design ........................................................................................... 23  
      5.2.1 Architecture ...................................................................................... 23  
      5.2.2 User operation design ....................................................................... 25  
      5.2.3 Data model design ............................................................................. 27  
   5.3 Implementation ........................................................................................ 28  
      5.3.1 Model ................................................................................................. 28
5.3.2  View......................................................................................................................30
5.3.3  Controller ..............................................................................................................31
5.4    Deployment and use ..............................................................................................32
5.5    Discussion ..............................................................................................................33
6     Conclusion ..............................................................................................................37
Appendix I.  Change of coordinate system .................................................................39
References .....................................................................................................................43
List of Figure

Figure 1. A scenario to illustrate exchange of kinematic information (provided by Mikael Hedlind) ................................................................................................................3
Figure 2. Product-Process-Resource (PPR) information model ................... 4
Figure 3. Classified manufacturing resources (Nassehi et al., 2008, Nassehi and Vichare, 2009) ............................................................................................................... 4
Figure 4. STEP structure (Loffredo, 1999) ...................................................... 7
Figure 5. Milestones of STEP kinematic modeling ....................................... 9
Figure 6. Conceptual operation flow (designed by Mikael Hedlind) ........... 11
Figure 7. A-0 diagram of IDEF0 model for the pilot application ................. 14
Figure 8. A0 diagram of IDEF0 model for the pilot application ................. 15
Figure 9. MVC architecture for KIBOS ......................................................... 16
Figure 10. GUI of KIBOS ........................................................................... 17
Figure 11. The DMG 5-axis machine tool model (created by Mikael Hedlind) .................................................................................................................. 18
Figure 12. Dialog to create a new pair ............................................................. 20
Figure 13. Configuration dialog .................................................................... 20
Figure 14. General I/O design of the integrated application ....................... 23
Figure 15. MVC architecture of the integrated application ......................... 24
Figure 16. User operation flow ................................................................. 26
Figure 17. Data model ................................................................................ 27
Figure 18. Model implementation ................................................................. 29
Figure 19. GUI implementation ..................................................................... 30
Figure 20. Controller implementation ......................................................... 32
Figure 21. General mapping between two coordinate systems ................. 40

List of table

Table 1. Suitability for various approaches to perspectives (Nassehi and Vichare, 2009) .................................................................................................................. 6
Table 2. Kinematic instances within a STEP file ......................................... 19
Table 3. Comparison between NX and STEP in kinematic mechanism ....... 35
1 Introduction

1.1 Objective
The main objective is to integrate the translation and operation of the data flow for system neutral kinematic modeling. Numerous commercial CAx software systems have been developed and applied in many fields for product design, prototype testing, life cycle management, etc. At the same time, close diverse partnerships between IT software vendors, industrial practitioners, and academic researchers have been built to enhance the competitiveness of supply chains. Therefore, system neutral solution for product data exchange is required in different perspectives: geometry, kinematics, functions and so on. And modeling and exchanging geometric information have been developed and implemented in a very high level. However, kinematic mechanism there is still no feasible solution even though the relevant standard has been published for nearly 15 years.

1.2 Background
Information technology (IT) has changed the operation of manufacturing industry a lot in the past half century. Different CAx software packages are used widely in all major enterprises in this industry, and college students of engineering always spend a large part of their study time in front of computer screens rather than with pen and paper as the traditional way. In a word, information technology has ruled the development of the modern manufacturing industry. However, the fast development and fierce market competition make this technology gradually become a barrier for the cooperation and communication between stages of the product development process. Julian Fowler summarized the reasons of such barrier in his works (Fowler, 1995):

- New IT systems often cannot be compatible with the existing systems or traditional manual processes;
- The current flexibility of most IT systems is not capable to deal with changing requirements in reality;
- Most IT products cannot fulfill the needs of manufacturing industry;
- All the above reasons can be compounded as the difficulty of sharing and exchanging between systems when information technology is involved.

1.2.1 Product data exchange
In general, product data exchange means the process of transferring data related to the product from one system to another and making it possible to access and store (Kramer and Xu, 2009). The data can be transferred in files,
database, or just paper. The system involved in this process can be geographically or temporally distributed and in any physical forms: it can be a design team in a firm, an NC milling machine, or a small personal data manager.

During the development process of products, usually multiple CAx software tools are used in different stages: idea generation, concept development, system design, business analysis and even testing and implementation. In modern industries, it is common that companies and organizations in different locations using different software systems have to work together due to high level of globalization, multiple forms of out-sourcing, and diverse commercial activities e.g. partnership between equipment providers and manufacturers. Therefore, designers have to face the complex problem that is how to seamlessly exchange and share data with different members in a collaborative development environment. Besides, designers should exchange not only the product geometric data, but also information about processes and resources. STEP, as a system neutral standard for product data exchanging, is introduced to solve this problem.

The purpose of the STEP standard is, as stated in its first part, “to specify a form for the representation and unambiguous exchange of computer-interpretable product information throughout the life of a product” (ISO, 1994). To fulfill such purpose and make it highly extensible and comprehensive, the STEP standard contains an information modeling language (EXPRESS and, the graphical representation, EXPRESS-G), data representation (Part 21 files and XML files), data access interface (SDAI), and categorized information models as application protocols (APs). It is the comprehensive structure of the STEP standard and imperative needs for system neutral product data format that make almost all major vendors related to the product data modeling support it more or less, especially in geometrical representation.

However, supports for exchange of kinematic mechanism data are hardly implemented for any applications in industries. Figure 1 illustrates a real scenario existing today that calls for the system-neutral kinematic modeling. When manufacturers communicate with machine tool and fixture suppliers on the motion modeling and analysis, the two sides have different CAD systems, e.g. Siemens NX is used in machine tool and fixture suppliers but SolidWorks is chosen by manufacturers. Because of a lack of solution based on data exchange standards, two sides with different software have to communicate with e.g. slides, paper, telephone, and fax (see Figure 1.a), and manual re-input become the only option to bridge two systems. Obviously, it takes large cost, time, and human resources, and skilled CAD operators have to be employed only for recreating engineering drawings (Fowler, 1995). This research will use STEP p21 file as the system-neutral data format to provide an alternative
solution for the exchange of kinematic mechanism information (see Figure 1.b).

Figure 1. A scenario to illustrate exchange of kinematic information (provided by Mikael Hedlind)

1.2.2 Manufacturing resources modeling
In any manufacturing enterprises, to make it survive in such a fiercely competitive environment, managers need to make effective decision, designers need to perform innovative tasks, and workers need to execute accurate jobs. To fulfill such requirement, every employee should always have a clear
understanding of the available manufacturing resources in terms of product, process, and resource information.

An important model in this field is the Product-Process-Resource (PPR) information model (see Figure 2) which is proposed in the doctoral thesis of Johan Nielson (Nielsen, 2003) where the PPR model is the fundament concept for the process plan core model. In the part of product, information of geometry, GD&T, functions, structures and so on shall be included. The process plan core contains the information on work plan, sequence, control, and inspection and so on. The resource part contains information on machine tool, fixture, mechatronics, and so on. Between different modules there is a interface, which refers to external data reference library, documentation, and so on (Nielsen, 2003).

![Figure 2. Product-Process-Resource (PPR) information model](image)

Figure 3. Classified manufacturing resources (Nassehi et al., 2008, Nassehi and Vichare, 2009)
The model is one of the major drivers of the development of STEP and has been further extended in several other researches such as the CNC machine operations resources modeling (see Figure 3). Computer-aided solutions which aim to provide reliable framework of computational representation are obviously important solutions to capture manufacturing resources (Rachuri et al., 2006).

Digital Factory Building Block (DFBB) (Sivard, 2009), is an ongoing national collaborative research project in Sweden, which aims to propose comprehensive solution of manufacturing resource modeling. This project focuses on the accurate management of manufacturing lifecycle data via complete and consistent production information library. A large part of the research presented in this master thesis, including the conceptual framework and developed applications has been used and extended in DFBB to demonstrate the effectiveness of neutral data repository in manufacturing resources modeling.

**Approaches for manufacturing resources representation**

Several approaches have been summarized in A. Nassehi and P. Vichare’s research (Nassehi and Vichare, 2009) to represent manufacturing resources, among which the applications of kinematic representations are highlighted. Graphical simulation systems are used to minimize trial errors in the design stage (Ehmann et al., 1997). Examples include DMU Kinematics of CATIA which is able to provide animation to demonstrate the kinematic mechanism in assembly models, and motion simulation of Siemens NX which is another similar solution to provide a simple-to-use integrated method to evaluate the kinematic behavior of designs developed in NX.

During the development in this research, the motion simulation function of NX is utilized to describe kinematic relationships with corresponding components of assemblies and to demonstrate the results of the integrated applications presented in Chapter 5.

Kinematic representation usually plays an important role of the underlying mechanism for most of motion simulation functions and virtual manufacturing systems.

In a research for internet-based virtual machine tools, detailed kinematic representations are modeled for the motion of machine tools collaborated with geometry representation, and internet-based implementation has been presented (Suh et al., 2003). A drawback of this solution is that the operation of the system requires fully support for VRML files which causes a low level of expandability for further development and information sharing and exchange in the real world.

In addition, standardized representations, especially based on ISO 10303, are emphasized more and more due to a lack of a comprehensive model to capture information that is necessary for all stages of operation planning. A model driven approach to represent manufacturing resource data based on
ISO 10303-214 is designed to solve the gap between different stages of process planning (Hedlind et al., 2010). Other approaches include process capability representation (Heusinger et al., 2006), hierarchical classification (Shinno and Ito, 1987), statistics-aided maintenance decision making (Mobley, 2002), and structural machine tool modeling (Seo et al., 2005), but all of them are designed to focus on particular fields and rarely have sufficient capacity to be adapted with more flexible requirements.

**Scopes for manufacturing resources modeling**

Different approaches for manufacturing resource representations often support different scopes respectively. A research for theses scopes will provide clues to identify the importance of approaches. A summary of perspectives for the resource modeling is made corresponding to the suitability of various modeling approaches, which highlights the importation of kinematic representations (see Table 1).

<table>
<thead>
<tr>
<th>0 = Least suitable</th>
<th>5 = Most suitable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process planning</td>
<td>Tool-path generation</td>
</tr>
<tr>
<td>Graphical simulation</td>
<td>1</td>
</tr>
<tr>
<td>Process capability</td>
<td>5</td>
</tr>
<tr>
<td>Hierarchical classification</td>
<td>4</td>
</tr>
<tr>
<td>Kinematic representations</td>
<td>5</td>
</tr>
<tr>
<td>Other approaches</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. Suitability for various approaches to perspectives (Nassehi and Vichare, 2009)

Therefore, an attempt of STEP implementation for kinematic representation will benefit the manufacturing resource modeling to reduce the gaps of sharing and exchanging of product information.
2 Literature review

2.1 STEP
The ISO 10303, also known as STEP, is a set of standards that aim to provide comprehensive solutions for data modeling throughout the lifecycle of products (Kramer and Xu, 2009). As mentioned in the previous section, there are multiple requirements for a neutral modeling solution for the manufacturing resources and bridge the gaps between systems and stages.

The structure of STEP is illustrated in Figure 4. It contains two major parts: infrastructure and information models. The former includes the description methods, implementation methods and conformance testing, and the latter includes application protocols, integrated general resources, and integrated application resources.

Figure 4. STEP structure (Loffredo, 1999)
Nearly all parts of STEP are described based on one single language, EXPRESS. This is a data specification language to describe the data model that can be used in any fields of the manufacturing resources and its application is even outside the STEP standard. According to the introduction of its reference manual (ISO, 2004), it is designed to make both computer and human understand itself, it is not a programming language, and it even seeks to avoid any dependencies with implementation methods to keep its extensibility. EXPRESS focuses on the description of entities and relationships between entities, and with local and global constraints the model developers can define data validity rules. Besides, the separated schemas are important features to make the modularization of data model possible (Fowler, 1995).

The implementation methods of STEP define a system neutral data format in part 21 “Clear text encoding of the exchange structure” (ISO, 2002). Usually, a part-21 file also known as STEP file acts as an exchange medium encoded with ASCII (Pratt, 2001). Its content contains the specified instances following the descriptions and constraints defined by the information models. In this research, the outputs of the outcome applications shall be part 21 files. A Standard Data Access Interface (SDAI) is defined by part 22 (ISO, 1998) as a guide to develop STEP-based implementation, although the physical implementation and the data model are separated. Bindings between SDAI and programming languages such as C, C++, and Java have been specified by different parts of the implementation methods so that programmers can find aids to access STEP files when developing implementations. Particularly, the part 27, Java™ programming language binding to the standard data access interface with Internet/Intranet extensions has been implemented as a development package, JSDAI (LKSoft, 2010), for STEP and Java based applications. The JSDAI provides complete API as defined in part 27 and it is compatible for both late binding and early binding as defined in STEP part 22, SDAI.

The information models for the standard data specifications can be categorized as two major types: application protocols (AP) and integrated resources (IR). APs shall focus on particular application domains and they are designed to be implemented for industrial requirements (Kramer and Xu, 2009). A STEP AP should have four major components (Kemmerer, 1999, Carver and Bloom, 1991):

- An application reference model (ARM) to specify the data types required for a special kind of industrial field;
- An application interpreted model (AIM) to define the way that the data represents the information in such industrial field;
- Documentation to define the context, scope, mapping table, and guidance information;
- Conformance requirements for implementation.
The IRs should be “context-independent standard data specifications” (Kemmerer, 1999). An example that is important for this research is the ISO 10303-105 (ISO, 1996), a specific IR for kinematic data specifications that describe the structure, motion, and analysis for kinematic mechanism which can be used in any industrial fields.

2.2 Kinematic modeling
As mentioned before, in the research examining the relationships between different modeling approaches with different perspectives regarding the manufacturing resources modeling, kinematic representations are seen as an approach that can meet most requirements of most modeling perspectives such as process planning, tool-path generation, CNC controllers etc. (Nassehi and Vichare, 2009).

ISO 10303-105 (ISO, 1996), is a member of integrated resources of STEP standard. This standard was published in year 1996 and with two technical corrigendium published in year 2000. At present, a new edition is under development and first draft of its usage within an AP is planned for year 2011 by ISO.

ISO 10303-105 is an information model providing generic resources that support kinematic information exchange and sharing for computer-aided design and kinematic analysis systems. The major feature of this part is to represent the structure, motion, and analysis related to kinematic mechanism. Typically, the kinematic structure is composed by links, joints, and pairs. As described in the standard document, links represent the rigid parts in kinematic representation, pairs define the geometric aspect for the kinematic motion constraints and joints define the topologic aspect. In the applications presented in Chapter 4 and 5, these concepts will be applied.

So far, there is no known implementation for ISO 10303-105 valid modeling. Almost all found literatures only use it as a conceptual model rather than the standard for real implementation. An important reason for such a blank is that there have been no guides or examples on kinematics in documents on STEP. Figure 5 generally illustrates the milestones in this field.

Figure 5. Milestones of STEP kinematic modeling
An expandable conceptual model for assembly information has been proposed in a project held by National Institute of Standard and Technology (NIST) in USA, named Open Assembly Model (OAM) (Rachuri et al., 2006, Sudarsan et al., 2005), which focuses on representations not only for geometric information, but also for kinematics and tolerances. The model is claimed to use STEP as data structure but it does not specified how and where the standard is used. In the conceptual model of OAM, kinematic pairs are set as a class following the kinematic structure defined in ISO 10303-105 in its UML representation.

A research about machine control software in the context of industrial economy is an early attempt to involve the kinematic mechanism based on ISO 10303-105 (Birla and Kang, 1995), where a modeling process is proposed for lifecycle engineering in the field of agile machining and the kinematic model for the members of machining processes in terms of fixtures, workpieces, and tools are defined based on ISO 10303-105.

A semantic-based machine tool modeling approach is presented for 5-axis machining application by F. Tanaka (Tanaka et al., 2008). This model focuses on the solution for the seamless information exchange between design and manufacturing stages. However, the ISO 10303-105 acts also here as a conceptual fundamental for the model proposed by the research, rather than a STEP-based implementation.

An important attempt to implement STEP-based kinematic standard is IDA-STEP (Integrating Distributed Applications on the Basis of STEP Data Models), which is developed by LKSoftWare GmbH (LKSoft, 2004). An outcome of the IDA-STEP project is a software prototype that can access, view, and edit STEP data which can be store in a STEP database for internet-based exchange and sharing between multiple devices. In this project, “an early prototype of a kinematic editor” is implemented with a relatively complete description of the kinematic structure: joints, links, a limited number of pair types, and range values, and, a VRML format file can be outputted and viewed in a web browser. According to their report, it is the first known implementation based on ISO 10303-105 since it was published, and they were close to achieve an implementation able to model standard conforming data.

Another STEP kinematic implementation that need mentioned is the SKM (Space Kinematic Model) modular within the ongoing STEP-TAS (Thermal analysis for Space) project which aims to build the thermal network and test environment for space mission (European Space Agency, 2007). The SKM module utilizes the kinematic structure of ISO 10303-214 ARM to describe the motion constraints of rigid bodies.
3 Research approach

This research has to face challenges of exploring the right way to apply kinematic features to the existing STEP part 21 file with only geometric instances, because, as mentioned before, there is no guide or valid example about the application of kinematics in existing standard documents AP214 or ISO 10303-105. The only useful resource to guide this research is the DMG model presented by Mikael Hedlind for ISO TC184 SC4 in 2009 (Hedlind et al., 2010), in which the first known standard valid kinematic mechanism becomes one of the most important guide for this research.

Therefore, the approach to process the kinematic structure in the STEP part-21 file should be developed firstly and how to integrate the STEP operation with existing CAx software should be generated.

In this research, the main task is the development of the STEP data integration application to provide feasible solution for extending STEP information exchange functionality provided by CAx system suppliers. The conceptual operation flow is illustrated in Figure 6.

For this work the solution includes two major systems: the Siemens NX and the STEP data integration application, KIBOS for NX. At first, a user should use NX to execute the STEP data integration application developed in this research and both systems will be running at the same time; then, after certain
configuration, the latter will send a request for a STEP part 21 file based on AP 214 and also for the kinematic information; finally, with the received data, the STEP AP214 file with both kinematics and geometrics is exported in a specified path.

During the following process of development, this operation flow is divided into two parts: the manipulation of kinematics in the STEP file is mainly designed and implemented in the pilot application, KIBOS, and the integration with CAD application is developed in the integrated application, KIBOS for NX.
4 Kinematic modeling based on STEP

4.1 Introduction
As presented in the first chapter, the pilot application is a standalone system to evaluate the capability of the STEP implementation for kinematic mechanism based on Java and JSDAI. In this application, part of the operation flow mentioned in the research approach, including writing, editing, and removing the kinematic features in STEP part 21 files, are demonstrated with a graphical user interface. This application is named as KIBOS (Kinematic modeling based on STEP). It is expected to provide complete logic to manipulate and display the data modeling specifications defined in ISO 10303-105 via AP214 within the context of STEP standard. Most part of the data updater, a core controller in the integrated model, will be implemented here.

A similar system has been mentioned in the literature review, which is the kinematic editor and viewer developed within the IDA-STEP project (LKSoft, 2004). Although it is an early prototype, the IDA-STEP application is implemented with the consideration for nearly complete kinematic structure and motion constraint configuration, and it provides proofs that the JSDAI package is sufficient for the STEP implementation for manipulations with kinematic features. However, the application of IDA-STEP only plays a prototype role of an integrated component of its parent system. With such level of the popularization of the IDA-STEP project, it cannot be applied to the wide context for real industrial requirements which is the reason why the integrated application is developed, as described in the next chapter, to provide high extendibility for the kinematic mechanism based on STEP.

4.2 Requirement analysis
The major requirement for this application is to provide a full process to operate kinematic features in a STEP file and to prepare sufficient data manipulation logic for the integrated application. Hence, unlike the single task for the latter, it is necessary for the system to perform several tasks in a STEP file with full supports of human computer interaction, and the requirement analysis should be derived here before design and implementation.

4.2.1 Problem description: objective, scope, and feature
This application is developed as a pilot system which has functionality to create, edit, configure, and remove kinematic features in a STEP files with complete geometric data based on STEP AP214 and to output the updated STEP file when needed.
As a pilot application, it will only be used, tested, and presented within this project and, therefore, the development should focus on the implementation of functionality rather than user experience. Because, as mentioned before, most of the programming code, for data logic here, can be used in the following integrated application. Therefore, the logic for the manipulation of the kinematic-related instances of a STEP file shall be designed and developed relatively independently with the rest part of this application and it should have sufficient extensibility for further re-use.

Then, the feature of this system can be summarized as:
- Identification of required STEP files and the basic geometric structure necessary for kinematics;
- Support for creating, editing, configuring, and removing instances related to kinematics;
- Relatively comprehensive process for human computer interaction for kinematic data operations.

### 4.2.2 Function design

In this section, IDEF0 model is used to define the overall structure of input, output, and functionality. At first, the top layer of the IDEF0 model (A-0 diagram) is show in Figure 7, where the core activity here is to operate the pilot applications. All possible inputs are shown from the left to this activity box. All possible outputs are listed in the arrows on the right-hand. The activity should be constrained by conformance requirement of STEP standard which shown in the top arrow, and for the application there will be a number of STEP files waiting to be updated as the mechanism shown in the bottom arrow. To support the application, Java, JSDAI, and AP214 AIM schema will be used as the mechanism as well.

![Figure 7. A-0 diagram of IDEF0 model for the pilot application](image-url)
Then, the required functionality of this system falls into the following five parts shown as the Figure 8:

- Identifying STEP file
- Displaying available components (can be parts or sub-assembly) of the assembly model and updated kinematic structure;
- Receiving user input in user interface;
- Manipulating STEP (creating a pair with specified components, setting a specific link as base, configuring a pair, and removing a specified pair);
- Exporting updated STEP file with instances for kinematic structure.

### 4.3 System design

As mentioned before, this application has two major functions: to manipulate the kinematic features of a STEP part 21 file and to provide a structural viewer of the updated kinematic entities. Hence, the logic design of data model and user interface are two important aspects of the system design. First of all, the MVC (Model-View-Controller) pattern (Reenskaug, 1979) is applied to describe the whole architecture of this system (see Figure 9). The information flow here is ignored because it would be complex and scrambled due to the multiple-function design shown in Figure 8.
The controller is the core part of the whole system, where the STEP file identification is used to identify the file to be updated, the GUI handler is in charge of processing the data I/O and the generation and disposal of all graphical user interfaces, and the data updater is used to invoke the logic to manipulate entities in a STEP file. As a prototype only for inside testing, functions designed here would be changed or improved more or less during the development of the following integrated application. The STEP file identification is one example: here in KIBOS the STEP file identification is only specified with the hard code in the program rather than a designed viewer, but in KIBOS for NX the file will be specified via the user interface.

The support for multiple tasks in the context of human computer interaction is a big difference of the pilot application from the single task allocation in KIBOS for NX, although most of the tasks are reused in an automated form.
Kinematic modeling based on STEP

in the integrated application. Therefore, the view part, user interface, is a little complex because it shall display the available components in a given assembly and the real-time updated kinematic structure, and that it shall provide sufficient support for manual input of the parameters used to create and configure pairs. Besides, the protection for removing mistakenly is also designed.

The model contains two parts: the STEP file and the working SDAI repository. Developed with SDAI means all the STEP model are stored in a repository temporarily, and any manipulation should only make changes to the instances in repository. When exporting an updated STEP file, the whole necessary information shall be derived from the repository rather than the original STEP file.

4.4 Implementation
As a prototype and pilot serving for evaluation, the implementation of this system here presented will provide a structured view for kinematic mechanism. It also will test the feasibility to output AP214 files with valid kinematic-related instances.

![Figure 10. GUI of KIBOS](image)

This system is developed based on Java, and it utilizes the classes from the API of SDAI for Java binding, JSDAI, the XIM library (LKSoft, 2009) for most STEP APs including AP214, and the graphical user interface toolkit,
Kinematic modeling based on STEP

SWT (Eclipse.org, 2010). The integrated development environment (IDE) here is Eclipse 3.5 (Eclipse.org, 2011), an IDE famous for its extensibility for Java or other programming language development, the WindowBuilder, a plugin for Eclipse, is used for GUI design, and JSDAI plugin for Eclipse is applied to provide necessary classes and interfaces for SDAI operations.

The graphical user interface of this system is shown in Figure 10, where the available components are listed on the left table and updated kinematic structure is displayed in the right table. Besides, in the components list there are checkboxes to specify the needed components to create pairs or to set as the base link.

Buttons are provided for functions selection. The “Set as pair…” button is used to create a new pair with two specified components, and, when it is clicked, a new dialog (see Figure 12) pops out for basic setting: the pair name, placement information, and pair type information. After creation, both the components list and the kinematic structure list are updated to inform new changes. One example is shown in Figure 10, where the component list is the representation for an assembly of a DMG machine tool model (see Figure 11) and a new pair named “1st” is set with two components and the table on the right-hand gives comprehensive information about the kinematic mechanism.

Figure 11. The DMG 5-axis machine tool model (created by Mikael Hedlind)
including the corresponding component of the base link, the organization of one pair and the relative placement data. Moreover, the shown kinematic structure can be exported in a STEP file as new instances in the last part of this file (see Table 2).

```
#3168=KINEMATIC_JOINT(#3171,#3179);
#3169=KINEMATIC_LINK_REPRESENTATION('',#3173,#86);
#3170=KINEMATIC_LINK_REPRESENTATION_ASSOCIATION('',$,#3169,#119);
#3171=KINEMATIC_LINK();
#3172=KINEMATIC_LINK_REPRESENTATION_RELATION(#3171,#3169);
#3173=AXIS2_PLACEMENT_3D('',#3174,#3175,#3176);
#3174=CARTESIAN_POINT('',(0.0,0.0,0.0));
#3175=DIRECTION('',(0.0,0.0,0.0));
#3176=DIRECTION('',(1.0,0.0,0.0));
#3177=KINEMATIC_LINK_REPRESENTATION('',#3181,#86);
#3178=KINEMATIC_LINK_REPRESENTATION_ASSOCIATION('',$,#3177,#149);
#3179=KINEMATIC_LINK();
#3180=KINEMATIC_LINK_REPRESENTATION_RELATION(#3179,#3177);
#3181=AXIS2_PLACEMENT_3D('',#3182,#3183,#3184);
#3182=CARTESIAN_POINT('',(0.0,0.0,0.0));
#3183=DIRECTION('',(0.0,0.0,0.0));
#3184=DIRECTION('',(0.0,0.0,0.0));
#3185=PRISMATIC_PAIR('1st',$,#3173,#3181,#3168);
```

Table 2. Kinematic instances within a STEP file

After creation, the user can configure the actual value, actuator, and range of a selected pair. The configuration dialog is shown in Figure 13.

In addition, KIBOS also provide a “Set as base” button to define the base link, of which the component must have been set as one link of at least one pair. The “Delete pair” button is used to delete any pair you want and, if clicked, a small confirmation dialog will pop out to ask the user to confirm the operation to prevent possible mistake. The “Export” button is used to export the updated STEP file from the working SDAI repository.
Kinematic modeling based on STEP

Figure 12. Dialog to create a new pair

Figure 13. Configuration dialog
5 Kinematic modeling based on STEP and Siemens NX

5.1 Introduction
In this chapter, design and implementation of an integrated STEP implementation based on NX Open and JSDAI, KIBOS for NX, is presented and discussed. The STEP model, with regard to the representations of geometric aspects, has been fully integrated and supported in many kinds of CAD applications such as Siemens NX, Solid Edge, Pro/Engineer, and CATIA. Different channels are used in these kinds of software packages, such as independent translators, importing/exporting commands in user interface, or even directly opening/saving, so that designers and researchers can use STEP files to exchange geometric data of their designs between different software systems. However, there is still a large blank regarding exchange of kinematic mechanism even though the STEP standard for kinematics has been published for 15 years. Most of the applications and researches in the kinematic mechanism based on STEP model exist only in academic institutes or colleges. The barriers for practitioners include (1) the hard tasks for understanding the STEP standards and (2) the difficulty of integration with the existing motion or simulation modules in CAD applications. KIBOS for NX provides a feasible solution for STEP-based kinematic modeling integration with existing CAD system and it is the first known valid STEP implementation for kinematic mechanism.

Information technology applied in the domain of engineering modeling is common in many industries. As an important solution for product data exchange, STEP model has been valued a lot since it came into being. Many data translation applications have been developed between STEP and other different file formats by the application vendors or the third parties. For SolidWorks, the AP214 and AP203 STEP file import and export is supported officially with an integrated translator, where the AP214 is fully supported for both export and import in geometric information (SolidWorks, 2010) but there are many limitations for other conformance options/classes for both AP214 and AP203. SketchUp is an example that is only supported by a third part plugin for STEP importing and exporting, which is provided by Okino’s PolyTrans 3D professional conversion system, although the engineers from Google claimed it was a complex task for translation from/to STEP (Bacus, 2010). Note that Okino also provides PolyTrans|CAD+DCC as standalone applications or plugins for numerous kinds of CAx software in 3D data translation and visualization, of which the vendor actively provide import and export for AP203 and AP214 STEP files in the context of almost all common
CAD platforms (Okino, 2010). Siemens NX provides both standalone and integrated solution for AP203 and AP214 STEP file import and export which can be called by its API for session access. What this chapter presents is an example of a STEP-based integrated application for Siemens NX to implement kinematic mechanism export from NX motion simulation model to STEP AP214 file.

As mentioned in the previous chapter, as a pilot application, KIBOS has been developed to manipulate the kinematic mechanism in a STEP file, but in that application, all the relative instances have to be added, edited, or deleted manually, and there are several rules needed to be obeyed due to the relatively strict propositions defined in the ISO 10303-105. For example, all the links defined must be connected with the base link via one or more joints, which require the user to define the base when there is at least one defined pair. Besides, due to the complexity of 3D implementation in the programming, displaying the image of the models and their components in the user interface is obviously not as easy as it in the existing CAD applications. Therefore, this chapter introduces the design and implementation of an integrated application based on Siemens NX and Java. The reason why NX is chosen is that it provides a relatively open programming interface called NX Open which enables the access to most of the functions (including the kinematic simulation) of Siemens NX via C, Visual Basic, and Java (Siemens, 2009). The general I/O design is shown in Figure 14. In this application, several practical features that are useful for the kinematic data exchange and storage are provided. First, the application can retrieve all the necessary kinematic features from a working .sim file in Siemens NX and translate them to the instances in the STEP file automatically. Second, it is able to utilize the STEP file without kinematic mechanism exported from the STEP AP214 translator provided by the Siemens NX software package. Third, it can be executed within the NX environment directly as a user function. Fourth, the application provides a visualized interface to execute the STEP translator and to set the name of the original STEP file and updated STEP file.

Hence, this application is able to utilize the visualized model and programming interface provided by Siemens NX so that the user can easily make any change of the kinematic mechanism via the graphical user interface in NX and get an STEP AP214 file with kinematic mechanism automatically.
5.2 System design

As illustrated in the previous section, the system performs two major tasks: retrieving the kinematic mechanism from Siemens NX environment and adding it to the existing STEP file. Some auxiliary functions include setting the path of NX, invoking translator to create the original STEP file, looking for the file, checking the errors in the NX model, and setting the name of updated STEP file.

The system design begins with the fundamental architectural description based on MVC (Reenskaug, 1979), which is to clarify the relationship between data logic, user interface, and controller. Then, the user operation flow is designed because the simplification of the manual manipulation is one of the most important objectives of this application and the limitation of NX Open creates some difficulties to such simplification. After that, a model design is presented for data manipulation logic.

5.2.1 Architecture

Since the application is developed with API provided by NX Open and JSDAI, it has to be limited by some constraints within both tools. As for NX Open, it is possible to use classes from the NX Open API and the common Java API, but if an internal application need classes from any other external libraries such as those provided by JSDAI and Eclipse, users have to configure the NX internal classpath manually by using the “Override Java variables” command (Siemens, 2009), which may cause inconveniences for some users. Therefore, in the development of the architecture, special design is applied to skip this configuration. Thus, the possibility to use the common Java API
Kinematic modeling based on STEP and Siemens NX

gives the chance to execute external applications and .jar files which provide the graphical user interface and early binding within the JSDAI context. As mentioned before, the JSDAI, a Java API working closely with STEP standard, also provides the XIM library for further manipulation of most application protocol (AP) in STEP including the support for ISO 10303-105 and AP214. Hence, all the data manipulation can rely on the external library from JSDAI. To adapt the programming to the limitation of NX Open and utilize the benefits of JSDAI, the classic MVC (Model-View-Controller) architectural pattern (Reenskaug, 1979) is chosen to design the whole system, and the basic design based on MVC can be illustrated in Figure 15.

![Figure 15. MVC architecture of the integrated application](image)

The Model Module is the core part of entire architecture, which is used to isolate the manipulation of STEP model from operations in Siemens NX environment and user interaction. As mentioned before, a data buffer which is
a regular ASCII file created and operated by the classes in Java IO package is used to store all the data procured from an NX model. Since the latest version of NX has already provided a stable full-function STEP translator that can be used to import and export STEP models, it is called directly from the program and users can use this translator to get a STEP model in a specified path from an assembly model. But it does not provide any translation on kinematics, which therefore is the major task of the developed KIBOS for NX application. The system is also designed to be used when a STEP model exist so that users can just specify the names of the original file and the updated file, and then combined with the kinematic information stored in the data buffer the updater can create a new updated file with complete kinematic mechanism designed with Siemens NX and represented with STEP.

The View module contains the interfaces to present information and receive user queries. There is a listing window in NX special for presentation of information, which is used here to present the result of data retrieval and storage. It can also handle exceptions such as no work part or assembly in the current model, no kinematic simulation, or no kinematic features and output the exception report. If the top application runs successfully and all necessary information is stored as expectation, the data updater will be invoked and a graphical user interface is presented to users to get more queries about creating STEP files if needed and updating STEP files.

In the Controller module, the application shall begin with executing a Java class file in Siemens NX environment after finishing design of an assembly and its kinematic features, which can be seen as a top application of the whole system. The module also deals with all the operations related to NX: reading the assembly structure, reading the kinematic information, storing all the information in a data buffer, and invoking the listing window in NX. The second part, data updater, shall be executed as a .jar file (Java Archive file, which can help developers to distribute their applications with necessary library and associated resources) if all the information on the assembly and kinematics is read and stored successfully. Data updater is a major part of the application, which maintains a GUI (graphical user interface) to interact with users and is able to receive and execute the queries from users.

5.2.2 User operation design
User operation of KIBOS for NX is designed according to the architecture and optimized to a simple flow expected to be learned and performed easily (see Figure 16).
At first, users are required to install the system manually. The Java class files of KIBOS for NX can be placed in any folder and be used to execute in NX, but a .jar file should be placed in a work path of NX Open that is usually defined in class path or can be found in the support document.

Then, the user should create a normal assembly in a tree structure and define the kinematic mechanism in the Siemens NX environment. It is recommended to use the motion simulation with the analysis type of kinematics. More information and support can be found in the chapter of digital simulation in the NX Help Library (Siemens, 2009). To create a kinematics motion simulation step-by-step, users can obtain lots of know-how knowledge in the CAST (Computer Assisted Self Teach) Library (Siemens, 2010). After creation of models in Siemens NX, the user should check the integrity and validity of the kinematic mechanism such as existence of the base link, the fact that all links are used in joints, and the fact that all joints have correct definitions of links, driver, and orientation.

The execution of the whole system shall begin with execution of the Java class file developed based on NX Open API when the working model is a kinematic motion simulation. Then there is a listing window for notification of successful data retrieval and a user interface where the user can choose to create a new STEP model from the translator or just add the kinematic mechanism to an existing file. Then, the user can get the updated STEP file in a specified path. Besides, this application also provides a textbox where the user can check and change the path of NX, which is very critical for creating the STEP model.
5.2.3 Data model design

There are three types of objects that are emphasized in KIBOS for NX: components, links, and joints (see Figure 17). All components are retrieved from the traversal of the assembly tree structure. A component shall have four properties that are useful in this system: a displayable name to identify the component, a position to present the location and its own local coordinate system, and a list of children components that is useful for the traversal of the assembly tree. Links and joints shall follow the definitions of them in the Standard of ISO 10303-105. A Link should be identified by its own name and be specified to a component that is used in the kinematic model. Notice that not all the components have links and one component should be specified to one link in a kinematic mechanism. Joints have the most complex data structure. One joint shall have one name to identify it, a type such as revolute, prismatic, or fixed, one or two links (if the type of a joint is fixed, it has only one link and not be seen as a kinematic joint in the STEP model), location and orientation information for both links if needed, initial displacement, upper and lower range that should follow strict limitation in both NX model and STEP model, and a specified driver which named as actuator in the STEP context.

![Figure 17. Data model](image)
5.3 Implementation

As discussed previously, this system is implemented based on Java 6 platform, which is the recent version of this most famous open source programming language. The JSDAI library and its associated XIM library are both very useful in the development to manipulate instances based on STEP. Siemens NX is chosen to be the CAD environment for this system because of NX Open, an API providing convenient access to most of its functions, which makes it unique compared with most of other commercial CAx systems. What is more, Eclipse is used as the main development environment which is fully supported by NX Open, and the WindowBuilder, a Java GUI design plug-in to Eclipse, is used in the development of both the KIBOS and KIBOS for NX.

5.3.1 Model

The model part maintains the logic of data manipulation. Information is utilized and integrated with the data stored in the buffer created by the controller and the original STEP file exported from the AP214 translator. The model module also receives and handles the queries from the user interface but no feedback will be provided within this system because all the users need is an updated STEP model file in the specific path.

The first step of the data manipulation is to obtain the original STEP model of which the translator is invoked by a system call from the current running runtime object in this Java application. To identify the right path of the translator, the system may need the right path input by the user in the interface, but if the default path of Siemens NX 7 (C:\Program Files\UGS\NX 7.0) is applied in the running platform, there is no need to provide the path for the user. Thus, the remaining task is to export the required NX model to a STEP AP214 file in a specified location.

Most of the logic of STEP manipulation is developed in the pilot application KIBOS. In order to be adapted to the updating of the kinematic features for the NX model, the previous work is changed in some aspects which are described in the following parts.

First, the location and orientation information of each link need to be transformed before used in the STEP model, because all placement information retrieved from NX is relative to the work coordinate system which is a global coordinate system identical for all components in an assembly. But the coordinate system for the kinematic pair location and orientation is defined as link frame, the local coordinate system of the corresponding component of a link. Therefore, in the controller part, not only the displayable name of each component is recorded, but also the location and orientation of the local coordinate system of the component is stored in the
buffer. A new method to get the pair placement information relative the link coordinate system and the relative matrix calculation method are both added in the class for STEP model updating. And more detailed discussion about the change of coordinate system is given in the Appendix I. Besides, additional information about component placement called “componentInfo” in the form of string array is added as the input parameters in the methods for adding link and adding pair. For the convenience of further development, judgments for whether the string array is null are programmed in both relative methods so that other applications without need of transforming coordinate can utilize this updater as well simply leave this parameter null.

Figure 18. Model implementation
Another improvement to handle the kinematic features in STEP models is integration of the pair parameters when adding pairs. The relative pair parameters include the name referring to a pair actuator, the pair value specifying the relationship between two links of a joint, and the pair range specifying the allowable bounds in terms of the upper limit and the lower limit. As input parameters, both limits should be in the forms of string, which can be "none" when the limitation is not defined. In STEP, the upper limit should be higher than the lower limit and there is no constraint for whether they should be negative or positive, but in Siemens NX the upper limit should be no less than zero, the lower limit should be no higher than zero, and both limits can be same (both are zero), which is inconsistency between NX model and STEP model. The most important operations in the model are illustrated in Figure 18, which include two types of task: adding joints and setting the base link.

5.3.2 View
The graphical user interface of this system was constructed with SWT (the Standard Widget Toolkit) package, an open source widget toolkit for Java designed to provide efficient, portable GUI based on the operation system where it is deployed. SWT designer is a free and powerful Eclipse Plug-in especially for the development of user interface based on SWT, and it has been helpful in the implementation of view module.

![Figure 19. GUI implementation](image)

The main GUI is designed as simple as possible so that users can understand it quickly and accurately. As shown in Figure 19, the major part in the user operation design is implemented in this view. The first line of the GUI is the path of NX. The default installation path of NX 7.0 is provided from start, but it is free for users to change it in case the Siemens NX is not installed default. Then, by clicking the large button, the user can choose to create a
STEP file from an existing NX assembly part file with the STEP AP214 Translator which is a software module included in Siemens NX. After the original STEP file is generated and placed in the proper path, the user should input the original file name and the updated file name and then click “OK” button to execute the update.

Besides, there are several other interfaces to interact with users, such as the warning message dialog, the report of successful data retrieval in Siemens NX, and so on.

5.3.3 Controller

The controller is the top application of the whole system. It includes processing user queries, retrieval of the necessary data, and control of STEP model manipulation. Since it is not allowed to import classes from external library as an internal NX Open application, the controller is implemented in two modules: a Java program (internal NX application) based on NX Open and another Java program (kinematic updater) based on JSDAI. The function model of the controller is illustrated in Figure 20 where the two modules are linked by the user interface.

The system is designed to begin with execution of the Java class file named KinematicNX2STEP.class, which is compiled from a .java file with the same name. The program will open a session from the session factory of NX Open and a listing window to report information to the user at first. Then, it will check whether there is a work part currently, whether the current work part is a simulation file, and whether there is a kinematic model in this file by checking the number of kinematic links. If all these validations are passed, a new file is created in the working directory named “databuffer.txt” by the FileWriter, a Java class to create files and to write data with the platform-default characters encoding. All the necessary information, as shown in the data model design, about the links, joints, and components are store in this file waiting for further use.

What is the most difficult part in this process is to find the name of the corresponding component of a link, because the name of a component is the best way to bridge shape representations in a STEP models and components in an NX model. NX Open provides a special method Component.displayname() to return the displayable name rather than the commonly-used Part.name() which returns a name with certain system-defined suffix. When retrieving the geometrical feature of a link, it is possible to obtain two kinds of features in the NX object list: a body or a component. This difference is due to the different ways of construction of the motion simulation models, and the program should judge whether what it has got is a body or a component. When it is a body, its owning component is used to invoke the displayname() method, otherwise this method can be invoked directly from components obtained from the corresponding link builders.
After all the data is retrieved successfully, a sub-application will be executed in a form of a Java archive in the currently working runtime object. This .jar file is an executable jar turned from a java project by Fat Jar, a useful Eclipse plugin. The reason why the system uses another Java archive file separately is to make it more convenient to isolate the model from the controller so that both parts focus on their own functions during the development, and the databuffer.txt file becomes the bridge between the two parts. Besides, another meaning of the design of the separate executable Java archive project is that the controller can be implemented only with the libraries of Java and NX Open. Other libraries for STEP manipulation and graphical user interface are not needed in this part. Therefore, the system can be deployed without special configuration of overriding the classpath variables manually.

5.4 Deployment and use
At last, a user instruction of this application is presented so that it can be easily deployed and used by any users.
At first, to deploy the application, the user should make sure that there is Siemens NX 7.0 or 7.5 (the tested versions) in the computer. Second, copy the Java archive file readAndUpdater.jar to the UGII folder within the NX folder. Then, to use the application, the below steps should be followed:
• Open a .sim file;
• Press Ctrl + U to execute KinematicNX2STEP.class;
• Change the NX path if needed;
• Create a STEP file if needed;
• Make sure the original STEP file is in UGII;
• Input the names of the original STEP file and the updated STEP file in the user interface;
• Click OK button and find the updated file in UGII.

5.5 Discussion
This chapter has proposed KIBOS for NX which is able to add kinematic-related instances to the existing STEP file exported from commercial NX-STEP AP214 translator. Therefore, this system has the following important features:
• It provides a complete process to export kinematic mechanism from Siemens NX to STEP and the underlying framework can be reused in any relative implementation.
• The data used by KIBOS for NX is from kinematic motion simulation built in Siemens NX and the STEP file exported from NX-STEP AP214 translator, so the user need not any professional knowledge on STEP-based kinematic mechanism.
• Information specified in Siemens NX on kinematic links, joints, pairs, pair values, pair ranges, base links are automatically translated to the kinematic topologic and geometric instances in the STEP files integrated with the geometric model and assembly structures.

With these features, it provides a feasible framework for IT venders to make effort on supports for STEP files importing and exporting with kinematic mechanism so that models of their products can be exchanged by STEP files with not only geometric features but also kinematic mechanism. For industrial practitioners, KIBOS for NX can help them to extend the application of their existing CAx systems so that their kinematic models can be exchanged with their suppliers and customers. For researchers, this application provides a way to create the valid kinematic mechanism in a STEP file with the help of graphical user interface in Siemens NX. This application has been used and extended in DFBB project (Sivard, 2009) to demonstrate using STEP files as neutral data repository to exchange kinematic mechanism between Siemens NX and STEP-NC Machine (a STEP-NC application from STEP Tools Inc.). However, there are a few delimitations in the current version of KIBOS for NX, which can be eliminated in the future development:
• The system only supports identification of two most common pair types: revolute and prismatic. As a prototype, such two types are sufficient for demonstration of the kinematic mechanism manipulation. Other pair
types may encounter unconformity between NX and STEP so that further research is necessary, but of course more pair types can be added in the future development.

- The current KIBOS for NX does not provide functions for import from STEP to NX. To fulfill a complete application sufficient for industrial usage, both import and export are important. Therefore, functions for import will be focused on in the future development of KIBOS for NX.

- The user operation flow is a little complex: users should use predefined path for file manipulation, the program path of NX should be set if not default, and the system should executed within Siemens NX and the specified kinematic motion simulation should be the working part.

- KIBOS for NX works with Siemens NX 7.0 and 7.5. The version 6.0 of Siemens NX has been tested for this application and it did not work. It is highly possible that the lower versions of NX cannot be supported by KIBOS for NX.

In addition, a comparison between Siemens NX and STEP in kinematic mechanism is performed and the results are shown in Table 3.
<table>
<thead>
<tr>
<th>Item</th>
<th>Siemens NX</th>
<th>STEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Several attributes: mass, velocity, name, etc.</td>
<td>No attributes direct on a link. Such data are modeled as properties of the product</td>
</tr>
<tr>
<td>Pair types with same names</td>
<td>Revolute</td>
<td>Revolute</td>
</tr>
<tr>
<td></td>
<td>Screw</td>
<td>Screw</td>
</tr>
<tr>
<td></td>
<td>Cylindrical</td>
<td>Cylindrical</td>
</tr>
<tr>
<td></td>
<td>Spherical</td>
<td>Spherical</td>
</tr>
<tr>
<td></td>
<td>Universal</td>
<td>Universal</td>
</tr>
<tr>
<td></td>
<td>Planar</td>
<td>Planar</td>
</tr>
<tr>
<td></td>
<td>Rack and pinion</td>
<td>Rack and pinion</td>
</tr>
<tr>
<td></td>
<td>Gear</td>
<td>Gear</td>
</tr>
<tr>
<td>Pair types with different names</td>
<td>Slider</td>
<td>Prismatic</td>
</tr>
<tr>
<td></td>
<td>Fixed</td>
<td>Fully constrained</td>
</tr>
<tr>
<td></td>
<td>Inplane</td>
<td>Point on surface</td>
</tr>
<tr>
<td></td>
<td>Constant velocity</td>
<td>Homokinetic</td>
</tr>
<tr>
<td>Pair types only</td>
<td>Unconstrained</td>
<td>Unconstrained</td>
</tr>
<tr>
<td>only in STEP</td>
<td>Sliding surface</td>
<td>Sliding surface</td>
</tr>
<tr>
<td></td>
<td>Rolling surface</td>
<td>Rolling surface</td>
</tr>
<tr>
<td></td>
<td>Point on planar</td>
<td>Point on planar</td>
</tr>
<tr>
<td></td>
<td>Sliding curve</td>
<td>Sliding curve</td>
</tr>
<tr>
<td></td>
<td>Rolling curve</td>
<td>Rolling curve</td>
</tr>
<tr>
<td>Pair types only</td>
<td>Atpoint</td>
<td>Joint: topologic aspect (link relations)</td>
</tr>
<tr>
<td>only in NX</td>
<td>Inline</td>
<td>Pair: geometric aspect (DOF, placement)</td>
</tr>
<tr>
<td></td>
<td>Orientation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parallel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Perpendicular</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cable</td>
<td></td>
</tr>
<tr>
<td>Pair</td>
<td>No such definition</td>
<td>Joint: topologic aspect (link relations)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pair: geometric aspect (DOF, placement)</td>
</tr>
<tr>
<td>Pair range constraint</td>
<td>Lower limit &lt;= 0</td>
<td>Upper limit &gt; Lower limit</td>
</tr>
<tr>
<td></td>
<td>Upper limit &gt;= 0</td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>Defined by a fixed link</td>
<td>Defined by a link, referred to by the base attribute of mechanism</td>
</tr>
<tr>
<td>Placement of links</td>
<td>Defined by an origin and a vector for one link</td>
<td>Defined by an origin and two perpendicular axes for one link</td>
</tr>
<tr>
<td>Coordinate of placement</td>
<td>In the global coordinate system</td>
<td>Described in local coordinate systems of both links</td>
</tr>
<tr>
<td>Similar def.</td>
<td>Driver: cannot be named</td>
<td>Actuator: can be named</td>
</tr>
<tr>
<td>Similar def.</td>
<td>Initial displacement</td>
<td>Actual value</td>
</tr>
</tbody>
</table>

Table 3. Comparison between NX and STEP in kinematic mechanism
6 Conclusion

In this research, the focus is on the STEP compliant integration strategy for kinematic mechanism with existing commercial CAx software systems. The solution is generated with a general framework for system neutral integration and then two applications are developed to implement this framework and demonstrate its industrial significance. A pilot standalone application is developed to prove the feasibility for Java-based STEP implementation on kinematics and to provide comprehensive operation logic to manipulate the kinematic instances in standard models. Relevant classes and methods are provided to handle the instances related to kinematic mechanism and users are able to use a structural viewer to view, update, and export kinematic mechanism based on system neutral standard. Then, an integrated application based on STEP and Siemens NX is developed to demonstrate the translation and operation of kinematic modeling between existing commercial CAx systems and the standard.

As the first STEP valid implementation for kinematic modeling, KIBOS demonstrates the capability of STEP implementation to manipulate the kinematic mechanism based on the framework of ISO 10303-105 standard. The research aims to integrate the translation and operation of the data flow for STEP-based kinematic features. The tasks finished in this research are summarized as follows:

1) A conceptual operation framework is designed to implement the integration of kinematic structure for CAx systems that can provide native kinematic models and the STEP files without kinematic mechanism. This framework helps to merge the data from both and output STEP files with kinematic mechanism that can be used for exchanging with different systems.

2) A pilot standalone application, KIBOS, is developed to evaluate the capability of the STEP-based implementation for kinematic features with Java and JSDAI. In this application, part of operation flow mentioned in the research approach, including writing, editing, and removing the kinematic features in STEP part 21 files, are demonstrated with a graphical user interface.

3) Design and implementation of an integrated application based on NX Open and JSDAI is presented in Chapter 5 as the main task of this research. It is executed as an internal application within Siemens NX. The operation flow illustrated in the research approach is implemented.

During the development of this research, several problems about the standard, the CAD applications, and the research itself arise.
Firstly, there are unconformities between Siemens NX and STEP model in kinematic modeling, which have been discussed and compared in detail in Table 3. Such unconformities in data models may cause unpredictable problems during the future development and usage of KIBOS for NX, but it cannot be fixed in a short term. Therefore, one of tasks for future projects is to identify more of such problems and give feasible suggestions for users and software vendors.

The extensibility of the operation flow defined in this research is questioned more or less due to a lack of open accesses in other CAD applications. As claimed in Siemens NX, the NX Open API for Java is able to provide full accesses for most of the functions that can be performed in the software, so the kinematic information can be obtained from the API and added to the STEP file with SDAI support. However, except for some CAD software systems e.g. CATIA, Pro/ENGINEER, and AutoCAD, several systems cannot provide such API for functionality access as NX Open which means the files with special and proprietary data format to store the kinematic information become the only source for information updating. Hence, more challenges will appear during future extending.

As a prototype, the program can complete the designed task at present, but it obviously not a full job for a comprehensive implementation for kinematic mechanism modeling within AP214. In the future, the current function, which is exporting information from NX to AP214 file, should be refined with more pair type selections, conformances checking, and improvement for data model logic. Besides, more functions can be added, such as importing existing kinematic mechanism from AP214 file and integration with kinematic analysis applications. Although some sample outputs have been validated against the conformance testing of AP214, a lack of real application of the exported kinematic mechanism is still a flaw as a complete research. Therefore, implementations to further explore STEP-based kinematic mechanism modeling are the most important task of future work.
Appendix I. Change of coordinate system

There is a difference with regard of the choice of the coordinate system between the geometric aspects in the ISO 10303-105 and the ones in the Siemens NX model. In the standard, it defines link frame, the local coordinate system of a link, and all relevant geometric definitions are defined with respect to this link frame (ISO, 1996). On the other hand, in Siemens NX, one work coordinate system of the assembly model is used to define the location and orientation when setting a joint, and all coordinates of such geometric aspects are defined from this work coordinate system when they are retrieved from NX Open library or exported to a STEP model.

Therefore, it is necessary to generate the parameter of the geometric aspects of kinematic links from the original work coordinate system to the local coordinate systems relative to the individual kinematic links. In Ibrahim Zeid’s works, changing the description of a point from one coordinate system to another one with its spatial position unaltered is defined as mapping of a point (Zeid, 1991).

**GENERAL MAPPING**

Given a point, a coordinate system A where it is measured originally, the description (a position vector) $P$ of the point in A, and a coordinate system B where it will be mapped, the problem is to get the description $P'$ of the point in coordinate system B, which can be described as (Zeid, 1991, eq. 9.85, p.505):

$$P' = f(P, \text{mapping matrix})$$

The mapping matrix describes the relationship between the two coordinate systems A and B. Generally, the mapping matrix $[T]$ is a combination of a rotational mapping matrix $[R]$ and a translational mapping vector $d$ (Zeid, 1991, eq. 9.88, p. 507):

$$[T] = \begin{bmatrix}
    r_{11} & r_{12} & r_{13} & x_d \\
    r_{21} & r_{22} & r_{23} & y_d \\
    r_{31} & r_{32} & r_{33} & z_d \\
    0 & 0 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
    [R] & d \\
    0 & 1
\end{bmatrix}$$

The basic principle of this equation is illustrated in Figure 21. The three columns of the rotational mapping matrix $[R]$ indicate the unit vectors of the
coordinates of the three axes of coordinate system A expressed in B. For example, \((r_{11}, r_{21}, r_{31})_B\), is a unit vector to express the x axis of coordinate system A in B, and the vector \(d\) describe the position of the origin of the coordinate A relative to B, which is \((x_d, y_d, z_d)_B\), as shown in Figure 21. Thus, the new coordinate of the point can be given by:

\[
P' = [T]P
\]

Note that the mapping matrix \([T]\) is a \(4 \times 4\) matrix. Therefore, the \(P\) and \(P'\) are both \(4 \times 1\) matrices with the three-dimensional Euclidian coordinates as the first three elements and 1 as the elements in the last row.

**Inverse Mapping**

To utilize this mapping equation, the unit vectors of axes of the local coordinate system relative to the work coordinate system can be retrieved from NX Open, and the position of the origin of the former relative to the latter is also possible to obtain in the same way. Hence, the local coordinate system can be seen as A as stated before and the latter is seen as B. However, in this case, the coordinate which is given is expressed in the work coordinate system (\(P'\) measured in the coordinate system B), rather than the local coordinate system (A). Therefore, what is unknown here is \(P\) (measured in A) and the previous equation should be modified further:

\[
P = [T]^{-1}P'
\]
Then, the mapping matrix should be transformed inversely. Because the rotational matrix is orthogonal, its inverse is equal to its transpose and thus the result is still not too complicated (Zeid, 1991, eq. 9.100, p.513):

\[
[T]^{-1} = \begin{bmatrix}
[R]^{-1} & -[R]^{-1}d \\
0 & 1
\end{bmatrix} = \begin{bmatrix}
[R]^T & -[R]^Td \\
0 & 1
\end{bmatrix}
\]

Thus, given \([R]\), \(d\) and \(P'\), it is not difficult to perform a matrix calculation in Java programming and get the result coordinate.

In addition, the geometric aspects of a kinematic link include one or more pair frames that are used by a kinematic pair, and a pair frame contains a vector representing the location of a pair frame on one link and the relative orientation of the pair. For the location, the above equation gives solution, and for the orientation consisting of three unit vectors, the equation can be simplified because the translational mapping vector is a zero vector for a vector and only the rotation mapping is considered. Therefore, for the mapping of a vector, the inverse mapping matrix is modified as:

\[
[T_v]^{-1} = \begin{bmatrix}
[R]^T & 0 \\
0 & 1
\end{bmatrix}
\]

So far, the theoretical background of the mapping of a point and a vector and how to apply it to the kinematic features in STEP model has been discussed. What should be noticed is the mapping of both the location and orientation of a pair relative to a link should be performed at the same time. Therefore, in the controller part of the application, there is a specific method to get the mapped placement information which outputs an array with nine members: three for the location and six for the orientation.
References


FOWLER, J. 1995. STEP for data management, exchange and sharing, Twickenham [England], Technology Appraisals.


LKSOFT 2004. Integrating Distributed Applications on the Basis of STEP Data Models


References

SIEMENS 2009. NX 7 Help documentation
SIVARD, G. 2009. Application within strategic vehicle research and innovation initiative (FFI) -- Digital factory building blocks, DFBB.