KTH Industrial Engineering and Management

Development architecture for industrial data management

Yujiang Li

Licentiate Thesis

Royal Institute of Technology
School of Industrial Engineering and Management
Department of Production Engineering
Stockholm, Sweden 2013

Abstract

Standardized information modeling is important for interoperability of CAx systems. Existing information standards such as ISO 10303 STEP have been proposed and developed for decades for this purpose. Comprehensive data structure and various implementation methodologies make such standards strong in support of different industry domains, information types, and technical requirements. However, this fact also leads to increased implementation complexity and workloads for CAx system developers.
This licentiate proposes the development architecture, STEP Toolbox, to help users implement standards with a simplified development process and minimal knowledge requirements on standards. Implementation difficulties for individuals are identified with analysis on implementation of the information standards in three aspects: tasks, users, and technology. Then the toolbox is introduced with an illustration of design of behavior and structure. Case studies are performed to validate the toolbox with prototypes. Internal and external observation has shown the around two-month learning process skipped and a great amount of workload reduction in implementation with the utilization of this architecture.

Keywords: Information modeling, ISO 10303 STEP, CAx, API

Contents

Abbreviations

AAM	Application Activity Models
AEC	Architecture, Engineering, and Construction
AIM	Application Integrated Models
AP	Application Protocol
API	Application Programming Interface
ARM	Application Reference Models
ASME	American Society of Mechanical Engineers
BIM	Building Information Model
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CAx	Computer-Aided technology
CNC	Computer Numerical Control
DFBB	Digital Factory Building Blocks
DLL	Dynamic Link Library
EUD	End User Development
FBOP	Feature Based Operation Planning
GD&T	Geometric Dimension and Tolerance
GPS	Geometrical Product Specifications
HCI	Human Computer Interaction
IGES	Initial Graphics Exchange Specification
MIM	Module Integrated Models
OCT	Optical Coherence Tomography
OWL	Web Ontology Language
PLib	Parts Library, ISO 13584
PLM	Product Lifecycle Management
SDAI	Standard Data Access Interface
STEP	Standard for the Exchange of Product data
TTM	Time To Market
UML	Unified Modeling Language
UoF	Unit of Functionality
VBA	Visual Basic for Application
XML	eXtensible Markup Language

Publications

This thesis is based on the following publications which are appended at the end of the thesis.

Paper A
Li, Y., Hedlind, M., and Kjellberg, T. (2011) Implementation of kinematic mechanism data exchange based on STEP, *Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology*, pp. 152-159, Athens

Paper B
Li, Y., Hedlind, M., and Kjellberg, T. (2012a) Kinematic error modeling based on STEP AP242, *Proceedings of the 1st CIRP Sponsored Conference on Virtual Machining Process Technology*, Montreal

Paper C
Li, Y., Su, R., Hedlind, M., Ekberg, P., Kjellberg, T., and Mattsson, L. (2012b) Model based in-process monitoring with optical coherence tomography, *Procedia CIRP*, Vol. 2, pp. 70-73

Paper D
Li, Y., Chen, D., Kjellberg, T., and Sivard, G. (2013a) User friendly development architecture for standardised modelling: STEP Toolbox, submitted to *International Journal of Manufacturing Research*

Paper E
Li, Y., Hedlind, M., Kjellberg, T., and Sivard, G. (2013b) Cutting tool data representation and implementation based on STEP AP242, *Smart Production Engineering*, pp. 483-492, Berlin/Heidelberg: Springer-Verlag

Chapter 1
Introduction

Instrumental or mechanical science is the noblest and,

above all others, the most useful.

- Leonardo da Vinci

1.1 Background

In the context of digital factory and digital manufacturing, information of products, production, and manufacturing resources is commonly required for exchange, sharing, and archiving in various systems. Companies and organizations in different locations using different software systems may work together due to e.g. a high level of globalization, multiple forms of out-sourcing and diverse commercial activities. However, most data formats and schemas developed by software vendors are closed and incompatible with each other. Although to a certain degree system-neutral solutions exist for product geometry, requirements of efficient communication are still far from fulfilled. High-level interoperability of manufacturing industry requires not only data exchange of geometry, but also information about processes and resources. This unbalance between reality and requirements may block different types of collaborations, e.g. within an independent company, a virtual enterprise, or an extended enterprise. For some enterprises, it is possible to choose software products from only one vendor to improve communication efficiency, but this is usually not the best choice to optimize each function. In addition, there is a risk of losing supports for all the software if this single vendor is out of business (Lui et al., 2011). Proprietary lock-in is also a potential problem when the existing system needs to be replaced by better software from other vendors (Sharma and Gao, 2006). Hence, disparate systems with different functionalities are preferred, even for a single company. Moreover, an environment with heterogeneous computer-based systems can be easily observed when a virtual enterprise is established (Chryssolouris et al., 2008; Zhang et al., 2000). For a long-term collaboration of companies to share their competence within an extended enterprise or a supply chain, people also have to face the problem to represent and exchange digital data (López-Ortega and Ramírez-Hernández, 2007). Thus, the isolated data sources naturally exist all around the manufacturing industry, and lead to many problems:

- Information quality cannot be guaranteed with risks in data loss or error.
- Time to market (TTM) is prolonged, especially for complex products with large supply networks.
- Legacy systems are difficult to access, especially for long-life products.
- Specialists have to be hired to establish data exchange interface, or to rebuild digital models.
- Overall cost is increased due to the above problems.

Since 1970s, formal information modeling approaches have been identified as fundamental to solve the problem of "information islands", with specifications of data requirements in certain application domains (Lee, 1999). Industry is calling for a solution for product data exchange which means a transferring process of data related to product from one system to another and making it possible to store and access (Kramer and Xu, 2009). The solution is usually presented in a way of information standards for industrial data. The information standards in this research refer to computer-interpretable representation of the general industrial product information for data exchange defined in EXPRESS (ISO 10303-11, 2004; Schenck and Wilson 1994). Such standards, including ISO 10303 STEP (STandard for the Exchange of Product data) and related standards (e.g. ISO 13399, ISO 13584, ISO 15531, and ISO 16739), have become a family for system-neutral data exchange. The standardized solution is noted for unambiguous modeling support for product lifecycle data (SCRA, 2006). This family of the information standards is supported by EXPRESS, a modeling language, and relative implementation methods defined in ISO 10303-2x. Today, there has been a part of STEP utilized as a major data exchange format for CAx (Computer-Aided technology) systems. Especially 3D geometrical modeling within STEP AP214 (ISO 10303-214) and AP203 (ISO 10303-203, 2011) has been widely implemented by almost all major CAD (Computer Aided Design) software (Hardwick, 2010; SCRA, 2006).

However, other than geometry, most parts of the standards have not been populated in designed industrial domains (e.g. shipbuilding and architecture), or with designed implementation methodologies (e.g. ISO 10303-28 STEP-XML and ISO/TS 10303-25 EXPRESS-UML). Even for the well-known application protocol, AP214, there are scopes which are far from commercialization, e.g. kinematics, GD&T (Geometric Dimension and Tolerance), features, and classification. Organizational, technical, and individual issues are blocking the popularity of the family of the information standards.

Gielingh (2008) concluded three organizational reasons of the slow progress of the standardization work in CAx industry: (1) low business motivation, (2) legal and security aspects, and (3) unclear industrial readiness. Similar investigations and conclusions were made by Tassey (1999) for causes of standard-based interoperability failure: (1) economic costs and benefits shared unfairly, (2) risks in technology and marketing globally, and (3) a lack of unbiased expertise. There are also technical issues regarded as potential defects of the standardized data exchange solution, e.g. information loss via translation (Ma, et al. 2006), inconsistent instantiation (von Euler-Chelpin, 2008), and non-neutral definition (Gielingh, 2008).

Nevertheless, the widely-supported 3D geometry translation implementation has demonstrated that the system-neutral solution is possible to achieve, given enough efforts of CAx developers and users. The development experience in this research shows that the 3D geometry modeling is one of the most complicated aspects represented in the information standards. Most of other standardized information types (e.g. kinematics, processes, GD&T, and classifications) are far from commercialization. However, the complexity of these information types is technically much lower than the complexity of 3D geometry. Hence, the success of 3D geometry standardization in CAD industry proves that the technical issues can be solved by the practitioners or developers of standards. An important condition for this success is that the commercial application of the standard is widely motivated and supported by different stakeholders in industry. The solution to organizational issues also depends on the evolvement of CAx systems and clarification of the application requirements of standardized data exchange.

Although the organizational and technical reasons are analyzed a lot previously, this study observes that the issues related to individuals are critical to the low popularity of standardization: the difficulty to understand and implement the standards by the developers and users of the CAx systems. Complicated modeling approaches and documentation methodologies are frequently experienced by the implementation community of the information standards, and this community complains. Ease of individual efforts is important in order to reach a solution for the organizational and technical issues. For instance, an expectation of fast learning and a simplified development procedure for individuals may make a manager agree to invest resources into the implementation.

Based on the above description, the individual issues are fundamental for the popularity of the standard. This research focuses on the solution to optimize implementation experience of the information standards. For individual users, high requirements for time, effort, and other resources blocks the utilization of the standards. The standards are developed by a large number of technical experts who focus on achieving a comprehensive, stable, system-neutral, and computer-interpretable international standard. In order to develop applications for standardized data exchange, thousands of pages of documents are prepared for developers. On the other hand, most of these developers, who are the main readers of documents of the information standards, are not ready to implement the information standards. They do not happen to be the expertise in software engineering, information standards, and the specific industrial domain at the same time, which are required skills for this task. Therefore, it usually takes a long time span to learn and develop the application. To further depict the challenge for individual users of standards, please find detailed description of implementation of the information standards in Chapter 2.

The aim of this research is to simplify the implementation of the information standards and thus spread the use of standards to attain benefits such as interoperability, better data quality and efficiency. The suggested method provides the missing link between a comprehensive information standard and its users.

1.2 Thesis outline

This thesis is a research result based on work of two projects Digital Factory Building Blocks (DFBB) and Feature Based Operation Planning (FBOP). Demonstrators developed for the two projects formulate the five publications appended and construct the organization of this article:

Chapter 1 introduces basic background information and research questions to be solved in this licentiate. The questions come from the development of demonstrators and the discussion with project partners from academia and industry. Hypothesis to be tested in this research is stated. A possible direction to the solution is identified.

Chapter 2 defines a frame of reference for this research. Contexts of use of implementation work based on the information standards are analyzed to identify critical issues during the development process. EXPRESS and SDAI (ISO 10303-22, Standard Data Access Interface) are two important elements required to implement such standards. The similarity in technique makes the developers who implement these standards share a similar procedure in preparation, design, development, and deployment.

Then, in Chapter 3, the major contribution of this thesis, STEP Toolbox, is described in two aspects: external behavior and internal structure. The development architecture simplifies the implementation procedure and makes this procedure similar to normal developers' experiences. It makes knowledge requirements of the standardized information modeling as minimal as possible. Chapter 2 and Chapter 3 are based on the following publication:

- Li, Y., Chen, D., Kjellberg, T., and Sivard, G. (2013a) User friendly development architecture for standardised modelling: STEP Toolbox, submitted to *International Journal of Manufacturing Research*

Chapter 4 uses four case studies to verify this toolbox. Discussion of this new implementation methodology is made in each of the four cases. In the first case study, STEP Toolbox is applied for development of kinematic translation as integrated plugins for CAx systems. Comparison is made with and without STEP Toolbox. Industrial project partners also benefit from the toolbox for their applications. The second case study extends the modeling and data exchange of kinematics to kinematic errors in terms of component errors and location errors. The third case study validates the ability of STEP Toolbox to coordinate multiple information models of ISO 10303, ISO 13399, and ISO 13584. A cutting tool classification application is developed based on a new modeling approach. Various types of information e.g. GD&T, classification, and general features are integrated with STEP AP242. The fourth case study proposes development architecture for in-process monitoring based on point cloud modeling. Interdisciplinary efforts are made to apply new measurement technology into micro-manufacturing industry.

Chapter 4 is based on four publications for the four case studies respectively:

- Li, Y., Hedlind, M., and Kjellberg, T. (2011) Implementation of kinematic mechanism data exchange based on STEP, *Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology*, pp. 152-159, Athens
- Li, Y., Hedlind, M., and Kjellberg, T. (2012a) Kinematic error modeling based on STEP AP242, *Proceedings of the 1st CIRP Sponsored Conference on Virtual Machining Process Technology*, Montreal
- Li, Y., Hedlind, M., Kjellberg, T., and Sivard, G. (2013b) Cutting tool data representation and implementation based on STEP AP242, *Smart Production Engineering*, pp. 483-492, Berlin/Heidelberg: Springer-Verlag

- Li, Y., Su, R., Hedlind, M., Ekberg, P., Kjellberg, T., and Mattsson, L. (2012b) Model based in-process monitoring with optical coherence tomography, *Procedia CIRP*, Vol. 2, pp. 70-73

In Chapter 5, discussion is made from the interdisciplinary view. Information modeling, software engineering, and human computer interaction are considered as three relative areas that would boost the development of this research work. Therefore, the possible directions of the future work are also identified in this chapter. Chapter 6 makes a conclusion of basic features of the result of this research.

1.3 Research questions and hypothesis

The objective of this research can be described in three stepwise research questions. The questions aim at highlighting current issues of implementation of the information standards and leading to a possible procedure to solve the problem.

- How can the information standards be populated?

This first research question is a fundamental subject that has been discussed by many researchers in the standardization community. Most experts in this area focus on the development of new modeling approaches to facilitate different application requirements. However, individual experiences of users who use the technologies and perform the software development are rarely concerned. In Chapter 2, as a contribution of this research, the individual experiences of implementation of the information standards is analyzed from a new point of view, regarding combination of tasks, users, and technologies. This analysis is expected to derive a new vision in activities to implement the information standards.

- How can the need for knowledge about the details of an information standard be reduced in the implementation work process?

Three types of knowledge are often required for developers to implement the information standards: software engineering, a specific industry domain, and standards. Knowledge of CAx systems may also be necessary for integration development. Usually, the developers are experts in software engineering, or in a specific industry domain (e.g. automotive industry, electrical engineering, space technology). Learning process about details of the information standards is a challenge that may be irrelevant to their work nature. In many cases, industrial or academic developers aim at using their applications for their own work. This indicates the implementation of the information standards as being typical End User Development (EUD, Burnett and Scaffidi, 2013). Ko et al. (2011) concluded that the major challenge of EUD is to employ development into users' existing workflow without substantial changes of their work nature or priority. For general developers, the priority is to achieve a feasible data exchange solution between different CAx systems, rather than learning about the information standards. It would be greatly helpful for them to enable implementation without high requirements of knowledge of the information standards. The main contribution of this research is the user friendly development architecture which supports and simplifies the STEP implementation. The architecture, named STEP Toolbox, will be introduced in detail in Chapter 3

- How can such implementation be useful in various contexts of use?

The direct result of the developed architecture is a simplification of implementation which would be useful in various contexts. The new architecture was validated and chapter 4 will share experiences to employ the new solution in different development practices.
These questions state the general issues existing in this domain, and lead to the hypothesis to be tested:

- Implementation of the information standards can be simplified and performed by developers with requirements on their knowledge of the information standards as low as for common users.

Common users of the information standards usually consider standards as data formats for information exchange. They can select proper standards and application protocols for certain usages according to their knowledge. This hypothesis makes the knowledge requirements of these users as a criterion to create the user-friendly development architecture for developers of CAx systems. Such level of knowledge requirements on the information standards is easy to achieve and is already achieved by most developers who prepares to implement the information standards. Therefore, it will be much easier for the developers if knowledge requirements of the information standards can be reduced to such level, which is the objective of the proposed development architecture. Of course, a reasonable level of knowledge on software engineering and a certain industrial domain is still needed to perform such implementation. User groups of the information standards regarding developers and common users are defined further in section 2.2.
The contexts of implementation work are described in the frame of reference and knowledge requirements on different levels of user groups are stated and discussed. Then the new development architecture STEP Toolbox is designed to reduce the workload of the target user group. Prototypes are developed to verify the applicability of the toolbox in various practices. Requirements on

knowledge of programmers and on the implementation procedure are examined and discussed during the prototype implementation. With this solution, the information standards for industrial data exchange can be more easily accepted and utilized by ordinary CAx developers and users.

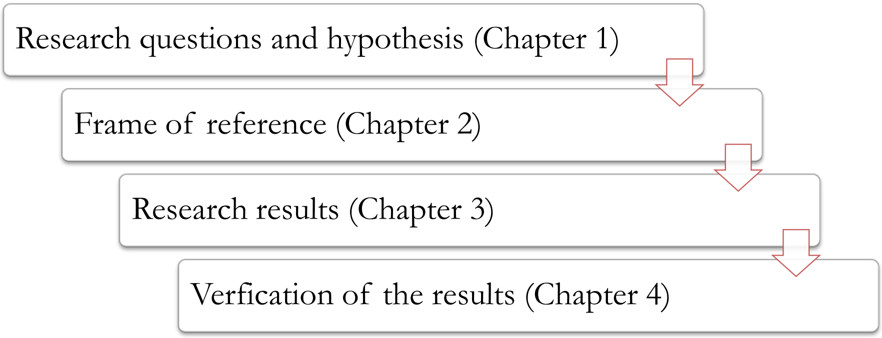

Figure 1.1 Research methodology

Chapter 2

Frame of reference

The more we elaborate our means of communication,

the less we communicate.

- John Boynton Priestley

In general, implementation of the information standards refers to practices to develop software applications with functions to access or manipulate standardized industrial data sets, e.g. in p21 (ISO 10303-21, 2002) or p28 (ISO 10303-28, 2007). Development based on any information models defined with the EXPRESS language (e.g. data sets of ISO 13399 or ISO 13584) shares a similar process with STEP ISO 10303. Developers can utilize different development packages and methods to perform such implementation. This section will treat the implementation procedure as an interaction activity where developers create or extend applications to establish functionality for standardized data exchange. Basic characteristics of interactions between developers, information standards, development tools, and, optionally, other software systems are investigated in different contexts of use. The basic elements of the contexts of use for interaction systems are environment, tasks, users, and technologies (ISO 9241-110, 2006). As the environment hardly influences development activities, this section will investigate the other three aspects: tasks, users, and technologies, respectively.

2.1 Characteristics of implementation tasks

This section will analyze the most significant technical characteristics of implementation practices and difficulties faced by the developers. According to developed programs, such tasks can be divided into two types: standalone and integration. In this research, the former refer to programs which can be deployed and executed independently. The latter mean programs developed as add-ons of existing software or requiring the execution of existing software.

The two choices both lead to special implementation requirements on developers. Because such implementation usually involves standards and another data format or CAx systems, at least two API (Application Programming Interface) systems should to be understood and integrated. The developers have to reason the mapping between different data models in two or more isolated systems. Sometimes, even different types of programming languages need to be coordinated due to the limitations of existing APIs for different data models.

Task complexity of standalone development is usually simpler than the integration development. For example, to enable translation from a CATIA CAD model to a STEP AP242 (ISO 10303-242) p21 model, a standalone program can be developed with input of CATIA data sets and output of AP242 data sets. Developers can take a full control of every component from interface design to data modeling. Similar software commonly exists for audio or video file format translation. On the other hand, the same goal can be accomplished by a program integrated with CATIA CAD software environment. With help of CATIA VBA API (Application Programming Interface), the developers can create an application embedded within user interface of CATIA. Scripts regarding data models and interfaces are highly influenced by the API defined by CATIA. Advantages of integration application are quite obvious: users need not learn new software interface for the new functions, i.e. file format translation. The developers may also benefit from reusing functions in existing systems. However, this may causes troubles for the developers. APIs regarding interfaces and data models need to be coordinated and integrated in programming. Interaction patterns should be carefully designed to make users get used to the new functions without misconception.

Taking kinematic data model translation from CATIA to STEP for example, both standalone and integration applications are possible solutions. Given that kinematic mechanism is very likely to be integrated with geometric models, the translation of geometry becomes a must in this kind of implementation. In this example STEP AP242 with p105 ed2 (ISO 10303-105 ed2, 2013) is considered as the desired standard format as output. The task of the developers should accomplish the transformation shown in Figure 2.1. Of course there may be other types of information that are of interests for further usage of the kinematics data exchange, e.g. machine or robot positions, cutting tool interfaces, and workpiece placement. To simplify the description, this section does not include such additional information.

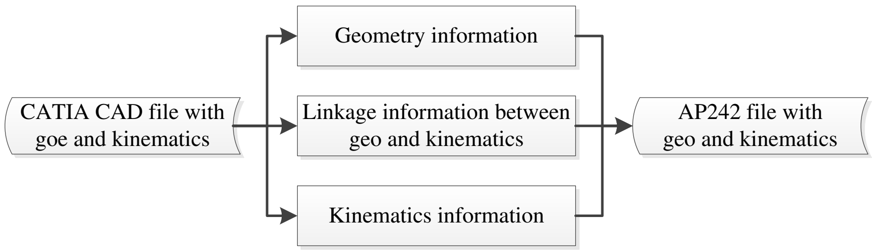

Figure 2.1 Target translation to be accomplished

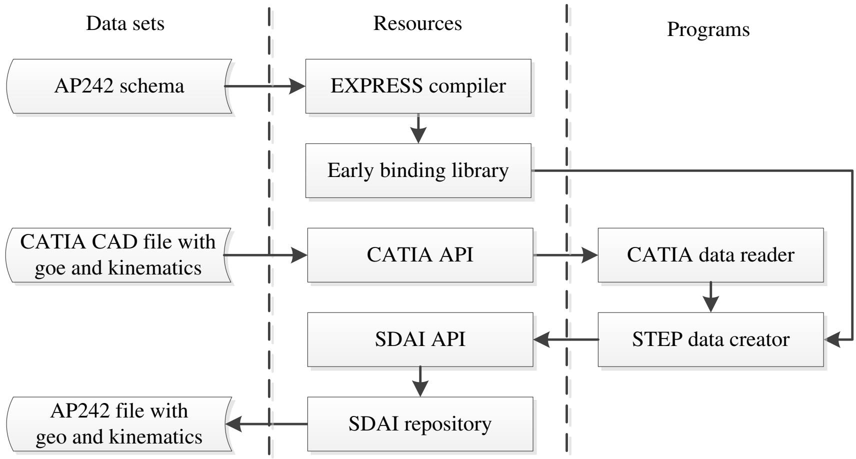

Figure 2.2 System architecture of a standalone application

A direct solution is a standalone application as a file format translator. System architecture can be generally described in Figure 2.2. The AP242 schema is compiled at first as a class library for early binding programming which is suitable for development with a stable schema. CATIA data sets and STEP AP242 data sets should be processed and generated by corresponding APIs. The developers should program a CATIA data reader and a STEP data creator based on such APIs. Although translation of the kinematic mechanism is the core problem, the developers have to deal with geometry translation in such a standalone situation. It can be expected that the complex nature of geometric models makes processing of geometry a tough job. As shown in this example, multiple APIs and programming languages are likely to be used for a

standalone application. In some other cases, the input data formats are not as complex as CATIA, or even designed by the developers, so they only need to deal with the API for STEP data model manipulation (see an example in section 4.2). Therefore, dependency of different programming resources does not always occur for standalone implementation.
On the other hand, the integration development provides a short cut to bypass the geometry translation so that the developers can focus on the kinematics translation. Such solution can be illustrated by interaction between CAx systems and developed applications in Figure 2.3.

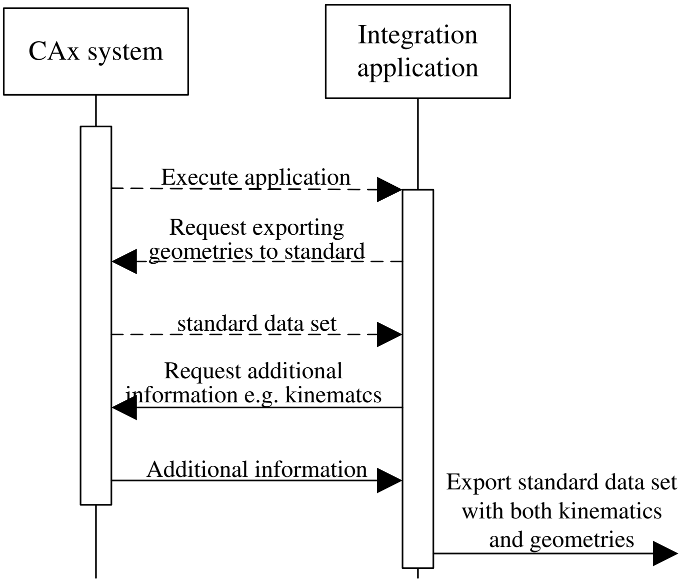

Figure 2.3 Design of integration development (Li et al. 2011)

Figure 2.4 describes the general architecture for the previous objective with an add-on development in CATIA. This is a realization of the design of integration development. With CATIA VBA API, the developed add-on can invoke the AP214 translator natively provided in CATIA for geometry translation, because AP242 is compatible with the geometry modeling of AP214. Then the add-on records the kinematic data separately, and invokes a developed STEP data editor to merge the outcome AP214 model and the kinematic data. Such system will be introduced in detail in section 4.1. The result of such development is usually an add-on or a plugin of existing systems, of which the interface is usually embedded together with other similar built-in functions. In this case, the developers can specify the appearance of the kinematics export/import functions as same as other built-in export/import functions, such as JT, IGES (Initial Graphics Exchange Specification), and STEP AP214/203 translators. The same interaction manners can make the end users easily locate and use this new function as they did similar jobs before. Hence, both developers and end-users can benefit from the integration implementation to a certain degree.

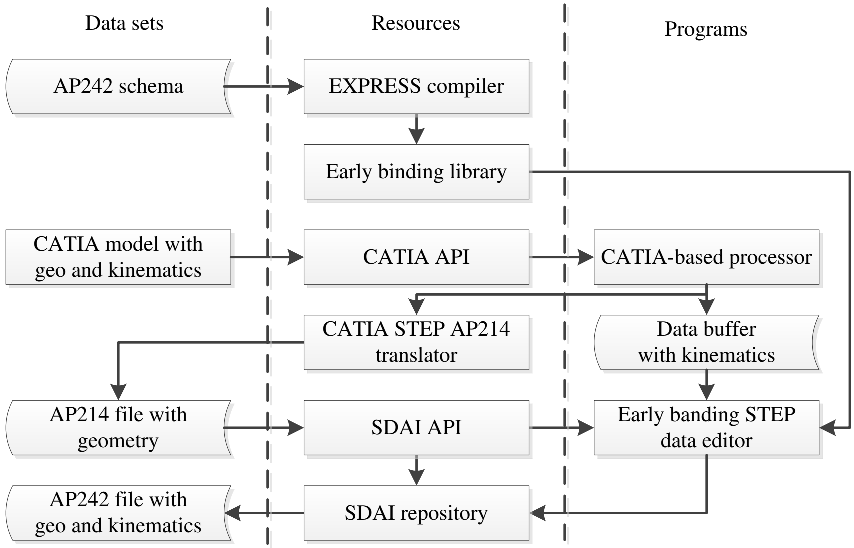

Figure 2.4 System architecture of an integration application

In both cases developers' skills on the implementation of the information standards are required in two aspects: schema processing and data set processing. At first, the developers who use early binding programming need compile the regarding schema of certain APs. Late binding does not require schema compilation but it requires clear understanding about schemas and carefully programming because mistakes of entities, attributes, and relationships cannot be found when compiling the program. Multiple schemas may be involved in one implementation. The developers should make decisions on when and how different schemas interact. On the other hand, data sets should be processed by APIs based on SDAI in a certain programming language with early or late binding. Structure and manipulation methodologies of standard data sets may be very different from other involved data sets (such as CATIA CAD models in the above examples). Knowledge need to be constructed to create or modify the standard data sets precisely to keep its interoperability.

In conclusion, the involved information standards in both standalone and integration developments cause issues for the developers to coordinate different APIs, data sets, schemas, and, probably, programming languages. However, most developers do not happen to have expertise in software engineering, information standards, a specific industrial domain, and CAx systems at the same time. Hence, the development task puts high requirements on individuals in separated technical areas. While software users enjoy the benefits of data exchange of their CAx data, the developers have to face various problems. The more systems are integrated, the more benefits users can enjoy, but the more issues developers have to solve. The following section will introduce such issues from the angle of the developers.

2.2 Potential user of information standards

This section investigates potential users of the information standards and identifies users whose experiences need to be improved. Three levels of users of the information standards are defined according to observation in this research (Figure 2.5). The three user groups have different types of tasks and different levels of knowledge on the information standards. Inclusion relationships of the groups indicate that users with a high level of knowledge are able to perform the tasks of users with a lower level of knowledge.

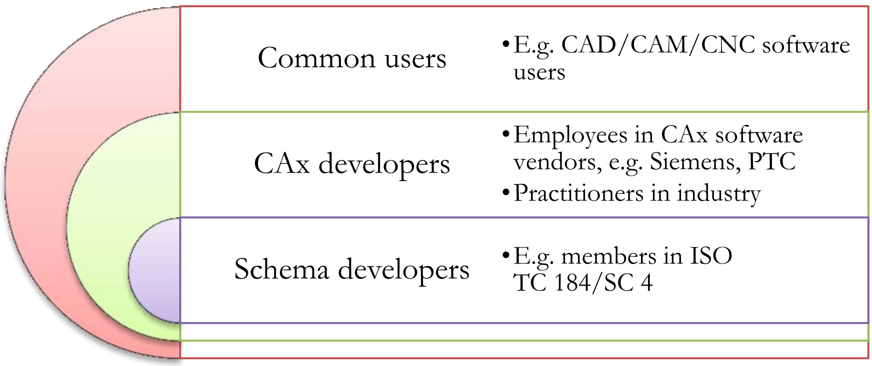

Figure 2.5 Three levels of users

The first level is common CAx users who use standard data sets, e.g. STEP p21 (ISO 10303-21) or p28 (ISO 10303-28) files, for data exchange. These users may only know the major purposes and functions of the standards. The second level is a subset of the first level, with additional knowledge of the standards. They are developers who develop CAx-related applications based on standards for data modeling. Employees in CAx software vendors can be required to program standard-based software to enable translation between data formats. Practitioners in industry need to implement the standards to support their CAx work for internal and external communication. The third level is schema developers who are very familiar with the core structure of standards and able to contribute to schema development based on EXPRESS language. The schema developers are usually most knowledgeable on this huge family of the information standards and they are experts on a particular part depending on their expertise on a particular industrial domain or information type. They should also know how to implement the standards, which makes them the subset of the CAx developers.

The aim of the standards is to solve interoperability issues of the common users (the first level) who are using all kinds of CAx systems. To develop an easy-to-use standard for common users, a plenty of work has been done since its first publish. Various application protocols and strategies are available for different needs of data exchange. On the other hand, complexity for schema developers (the third level) is easily observed by related researchers, which triggers many studies, e.g. Danner et al. (1991), Burkett and Yang (1995), Goh et al. (1996), Feeney (2002), Jardim-Gonçalves et al. (2005), and Maier and Stumptner (2007). However, the CAx developers (the second level), who are struggling to understand the standard specifications, are often ignored.

This study focuses on improvement of experiences of this silent user group, CAx developers. They may be professional programmers who develop standard-related software as their jobs, or end user programmers who develop applications to support their work or hobby (Ko et al., 2011). It is already necessary for such developers to learn knowledge of software engineering, a certain industrial domain, and, optionally, CAx systems. At the same time, his/her working time will be heavily occupied by documents reading and information modeling due to implementation with the information standards. In fact, they are the only practical readers of documents of the information standards, and have to put great effort to conquer relevant knowledge in this area. For a developer of an application for e.g. kinematic translation based on STEP AP242, Required documents are even more than requirements for some schema developers (the third level users). Below is a list of needed documents (not including corrigenda) in such case:

- ISO 10303-1 (1994) for an overview of the documentation structure,
- ISO 10303-11 (2004) for the EXPRESS language,
- ISO 10303-21 (2002) for description of the STEP data sets,
- ISO 10303-22 (1998) for SDAI,
- ISO 10303-27 (2000) for the binding to Java,
- ISO 10303-41 (2005) for fundamental product modeling,
- ISO 10303-43 (2011) for representation structure,
- ISO 10303-105 ed2 (2013) for kinematics,

- And ISO/DIS 10303-242 (2013) for this application protocol.

According to experience of this research, it takes approximately 2 months of full-time learning to enable a beginner to understand and implement the standards. A comprehensive structure makes the standards support data representation in a wide range of industrial domains and lifecycle stages (Pratt, 2001), but it also results in troubles for beginners in preparing and implementation. Besides the heavy workload for learning, inconsistent instantiation is observed as a practical issue regarding information modeling (von Euler-Chelpin, 2008). To achieve precise data modeling, the developers should reason the mapping through AAM-ARM-AIM/MIM (Application Activity Models - Application Reference Models - Application Integrated Models / Module Integrated Models). A lack of experience, knowledge, or carefulness can make the developers interpret information models differently, which damages unambiguousness of the standards.
The information standards for industrial data management are important for enterprises to enable fast and reliable data exchange in internal and external communication. However, the time consumption of preparation and development has become a vital obstacle to make it as popular as designed in its target areas. Both researchers and practitioners are complaining that the "expensive" cost on human resources makes the standards unfriendly for individuals and organizations.

2.3 Technology to implement standards

For most programmers, existing EXPRESS-based development tools are important to reduce manual manipulation and automate validation. Such tools are usually implementations of SDAI in specific programming languages, which provide fundamental operations on EXPRESS-based data sets. For instance, JSDAI of LKSoft is a Java-based programming toolkit for EXPRESS schemas and EXPRESS-based data sets development (Klein, 2000), which is reused frequently in this research. A similar tool in development based on C/C++/Python is STEPcode (SC) which replaces the former product, STEP Class Library (SCL) (Sauder and Morris, 1995). Also for development, Reeper is unique in support of XML-based STEP schemas and data sets conforming to ISO 10303-28. In the commercial software market, ST-Developer has been developed for over two decades by STEP Tools Inc. and adopted in most major CAD applications for STEP translation. It has competition from EDMdeveloperSeat by Jotne EPM Technology, ECCO Toolkit by PDTec, etc. Especially for STEP-NC, Hardwick and Loffredo (2007) developed a high-level programming interface based on a DLL (Dynamic Link Library). This work is promising to enable fast data manipulation for STEP-NC data sets. Table 3.1 makes a simple comparison of some standard development tools.

Table 2.1 Examples of STEP development tools

Name	JSDAI	STEPcode	ST-Developer	Reeper
Creator	LKSoft	BRL-CAD, NIST, and others	STEP Tools	EuroSTEP and NIST
Version	4.3.0	0.6	15.0	0.5
Availability	Open source	Open source	Commercial software	Open source
Language	Java	C, C++, and Python	C, C++, and Java	Ruby
Major features	EXPRESS schema compiler, SDAI library, access of p21, EXPRESS-based validation			XML-based operation on schemas and data sets
Additional features	Plugin in Eclipse, support of p28 XML	EXPRESS formatter	Access of IGES and DXF	

Although such development tools have reduced a lot of preparation work for developers, they are still put within the conventional implementation procedure of the information standards. There is a lack of guideline and data sample in most of the related standards. The developers need to very well understand various related concepts, e.g. EXPRESS schemas, SDAI, AAM-ARM-AIM/MIM mapping, before the first program.

Similar technology can be observed in other areas of software engineering. Flash ActionScript is used to manipulate animation, web interface, mobile applications, etc. MATLAB helps scientists and engineers to perform technical computing and experiments. Microsoft Office VBA can be used to extend and customize the usage of Office components. These kinds of technologies are not only provided for professional programmers, but also for end users of the corresponding software. As same as professional programming languages, such technology often follows the process of scripting, debugging, compiling, and executing, and can be presented in a friendly manner so that even the end-users are willing to create applications to enhance their daily work. The developers to implement the information standards for industrial data management cannot enjoy the same conditions, but the technologies in other areas are good examples to design a next generation of development architecture for the information standards.

Chapter 3
Result: STEP Toolbox

We speak of the "architecture" of a symphony, and call architecture,
in its turn, "frozen music".

- Deryck Cooke

This chapter begins with the design methodology of STEP Toolbox, which describes principles to solve the research questions based on the frame of reference. Then section 3.2 and 3.3 introduce the development architecture in terms of two aspects: external behavior and internal structure. The former refers to the design of the interaction with other systems. The latter is the description of internal modularized components and their relationships.

3.1 Design methodology

As mentioned before, the information standards referred to in this research include standards supporting industrial data management with the modeling approach proposed in STEP. This is the reason why the toolbox is named STEP Toolbox, but it can actually support any data model based on the EXPRESS language.

As discussed in the previous chapter, currently, implementation of information standards usually requires programmers skilled in standardized information modeling. It is important for developers to understand definitions of e.g. EXPRESS, AAM, ARM, AIM/MIM, mapping table, and model validation, while the focus of developers is usually on software engineering or a specific industrial domain. For normal developers, the situation is similar to other kinds of development for integration or tailoring with e.g. Flash ActionScript, MATLAB, and MS Office VBA. Such systems also have special data models and manipulation methodologies, but programming can be accomplished easily only with well documented tutorials and samples for APIs. Such interaction pattern is often ignored during the development of standards. Schema developers usually aim at best design of data models to meet the requirements of the industry. Development of related implementation methodologies usually focuses on applicability in various contexts, e.g. Internet, database, and XML. The interaction with programmers is often missed by researchers and creators of the information standards. This research focuses on the interests of such programmers who really use the standards. The major objective is to simplify implementation of the information standards, where the focus of developers should be shifted from detailed documents of standards to a friendlier interface, STEP Toolbox API.

The architecture should be designed as a friendly, extensible, reusable, and maintainable development platform to simplify the implementation procedure for the information standards. With a designed conceptual model and an object-oriented API, programmers are able to construct their knowledge with a set of representations more similar to other object-oriented API libraries. A new generation of the implementation procedure will be as convenient as a normal software implementation. Hence this architecture can act friendly to its target user group, the CAx developers.

As the development of information standards for industrial data exchange will continue, the architecture to support implementing such standards should be ready to be extended to meet future needs. STEP Toolbox uses modules to group concepts with related information types. Developers can select necessary modules for certain information types. A center manager in the API helps developers to select industrial domains by maintaining data models with certain application protocols of certain standards.

A key feature of STEP Toolbox is reusing existing standard models, rather than inventing representations of concepts again. Most information standards in this area are constructed based on modularized structure with supports from integrated resources and generic resources. The design of most parts of STEP Toolbox can benefit from such modularized structure, and keep the high reusability in implementation that may involve multiple industrial domains. The high reusability is also a basic principle to guide the modularized structure of the API. Representations of concepts can be shared by different modules if needed, and be collected by corresponding module managers to facilitate certain modules.

As requirements for implementing the information standards can be variable and unpredictable, such architecture for implementing standards should be designed as maintainable as possible. The practitioners who use STEP Toolbox may require different levels of simplification for different information types. The structure of modules should be designed to meet various requirements. By maintaining conceptual models, users and creators of the toolbox API can reach agreements on how the relevant concepts and relationships can be simplified.

3.2 External behavior of architecture

The interaction between STEP Toolbox, data sets, and applications is briefly illustrated in Figure 3.1 which uses three layers to represent where and how the toolbox should support development. For developers, the toolbox is presented in a form of API which is more intuitive compared with the documents of the information standards. Software architecture is often presented in terms of the external behavior and the internal structure (Klein and Weiss, 2009). This illustration briefly describes the internal modularized structure of the API and the external behavior with data sets and applications built upon the toolbox. The bottom layer refers to the standardized data sets governed by the standardized schemas. The middle layer is the core part of this architecture, the STEP Toolbox API. It includes the modularized components developed in this research and the SDAI as the basis for data model manipulations. The top layer indicates implementations based on this API. The case studies in Chapter 4 are examples of such implementations which demonstrate the utilization of this new architecture.

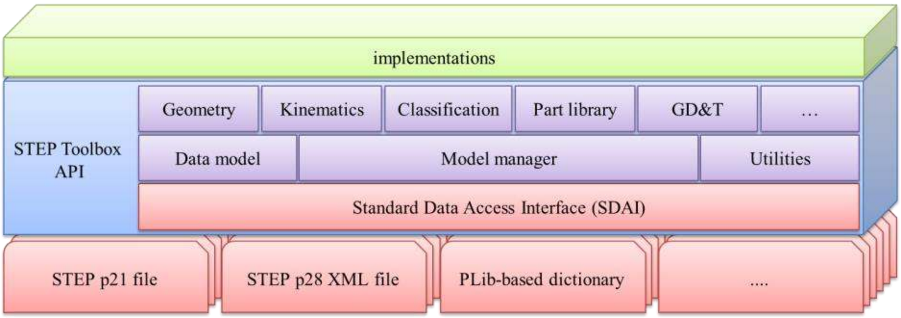

Figure 3.1 The architecture of STEP Toolbox

The bottom layer exemplifies requirements to manipulate various types of file formats, which are fundamental to determine the external behaviors of the toolbox API. In practice, STEP p21 based on EXPRESS is the most commonly used data format by the information standards. An alternative, p28

is introduced to employ XML for the description of both schemas and data sets. Other standards and information models are using EXPRESS as well, which makes STEP Toolbox API able to process their data formats. For instance, the access of ISO 13399 classification dictionary based on PLib (ISO 13584, Parts Library) has been tested with the STEP Toolbox API. These data formats share basic operations based on EXPRESS and SDAI, but different manipulation strategies should be applied for dedicated applications. For example, data sets for exchange of the common STEP application protocols (e.g. AP214, AP203, and AP242) may be accessed and manipulated frequently. Data sets of the PLib standard are usually treated as references of other standardized models, which may not often be altered by developers. Therefore, programming interfaces for different types of data sets should be designed compatibly with the applicable scenarios.
The middle layer is where the STEP Toolbox API is located. It is the only programming interface that the developers need to learn when implementing standards. This API provides simplified interfaces as a compensation for SDAI for the manipulation of EXPRESS-based data sets. To shorten the learning process when interacting with the conventional SDAI, the API is designed programmer-friendly on a level upon which CAx plugins or standalone applications can be built. Many professional concepts of the information standards are hidden from the API users so that the development procedure can be simplified. Advanced users can still use SDAI to perform late or early binding operations. Detailed description of STEP Toolbox API will be presented in the next section.
In the top layer, implementations such as CAx add-ons or standalone applications can be built with the support of STEP Toolbox. There have been successful examples in different areas developed by authors or project partners already. For instance, the kinematics translator for Siemens NX is the initialization of the STEP Toolbox, which demonstrates feasibility to implement standards with commercial CAx software in other areas than geometry. The STEP Toolbox was initially conceived in order to help industrial project partners to support their development based on CATIA. Other examples can be found in Chapter 4.

3.3 Internal structure of architecture

The components of STEP Toolbox API are illustrated in the middle level of Figure 3.1. SDAI is encapsulated within this API, upon which other components are built to support this new implementation approach. The model manager performs static functions on the models. E.g. the central manager provides operations for initializing, exporting and closing p21 or p28 models. A module manager collects related objects of a certain module and provides static operations on the objects to support implementation. The data model is an abstract concept for all classes representing different types of objects in an information model, e.g. components, kinematic links, diameters, linear distances, PLib classes, and properties. Utilities are a group of functions that provides generally reusable operations, e.g. mathematical services for coordinate system transformation.

Grouping the data model and relevant operations into modules is important to simplify the comprehensive standards. The API provides different modules for different types of information, e.g. geometry, kinematics, and classification. These modules are independent of application protocols, which means users can use the data models and operations provided by one module to access data sets of different APs. Different modules are often used together for the development of one application. For instance, the classification application, based on ISO 13399 cutting tool dictionary and AP242, highly relies on data models of GD&T, classification, part library, and a part of geometry. In the GD&T module, a GDTManager is able to collect the available information in the form of the specific data models of GD&T. Developers can simply perform operations from the classification module on the collected GD&T data models. The modifications on the data model can be exported to AP242 or AP214 since both APs support GD&Ts and classifications.

Model managers, data model and the utilities form the basic framework for modules. Besides the central model manger performing general operations, for each module there is a specific manager to access and modify a corresponding group of data models, e.g. GDTManager. The data model of different modules is the key part to make the toolbox support the wide range of domains covered by industrial data standards. Each class of the data model usually represents several entities of EXPRESS schemas for a specific concept. The concepts are usually adopted by the information standards, CAx systems, and industrial domains, which are useful for CAx-based software development. Some classes of the data model belong to specific modules, but some are reused by several modules. Managers for specific modules are used to collect related classes for modules.

As mentioned before, the complicated modeling approaches often introduce problems for the developers. Therefore, the toolbox adopts an object-oriented API for data model presentation and manipulation. Compared with the EXPRESS schema, the API is designed to be easily understand by target readers. Classes are defined for data models and managers. Useful attributes for implementation are collected from the standard schemas. A simpler structure of information modeling is then introduced in the API with the help of UML (Unified Modeling Language) class diagrams.

A bridge is needed for the mapping from the EXPRESS schemas in order to instruct the design of the API. A conceptual model is used to regulate how the concepts in standard models can be simplified, according to the implementation context. It is a result of the agreement between standard models, CAx concepts, and individual knowledge of developers. A vocabulary interpretable by both computers and humans is defined in the conceptual model so that programmers are able to share the same concepts with the standard. CAx users can easily find similar concepts in the standards that are defined differently (semantically or syntactically) in various CAx systems, so a unified and general vocabulary is needed. For instance, in a kinematic mechanism of a machine, one kinematic link should be fixed with the ground or stay static relatively. In an AP242 data set, such link is referred to by the mechanism representation as a “base”. In Siemens NX 7.5, the same concept is expressed with a Boolean attribute of a link. In CATIA, there is no definition for links but the geometric components can be set to compose a joint directly, and a base link is represented by setting the component as a fixed part. In conclusion, the “base” of the mechanism, the “fixed” attribute of a link and the “fixed” part refer to the same concept in CAD area. The conceptual model should unify such terminology and construct reasonable relationship between terms to help the developers focus on their implementation as fast as possible.

As there are several modules included in the toolbox, the conceptual model should be domain-independent so that concepts can be reused by two or more modules if necessary. For instance, a “product” in STEP is a general concept which may have one or more occurrences in a data set. Because occurrences are more useful than products in the CAx implementation, the occurrence, instead of the product, is taken as a class in the vocabulary, which is called the “component”. For one product in STEP, each occurrence is set as a “component” in the conceptual model, which shares the same product ID as an attribute. Hence, the component can be reused easily in other modules than geometry, e.g. in the kinematic mechanism, a link can be associated with one component.

The conceptual model is also a source from which the API users can easily get guidance of the whole structure of the toolbox. Therefore, it should be easily interpreted by humans and mapped to a computer interpretable data model in the API. This mapping relationship is illustrated in Figure 3.2, by which the relatively complicated EXPRESS schemas are hidden from the programmers. The conceptual model is presented in a form of ontology that can be easily understood by common users with basic knowledge of information modeling. Kjellberg et al. (2009) pointed out the importance of the ontology model for manufacturing and products to harmonize concepts between humans, information standards, and CAx systems. This research makes use of this theory and extends it for specific user groups in the cases of real implementation. The mapping between EXPRESS and ontology should be performed according to the general understanding of relative application requirements and information models. Agreement on a certain level of data model simplification should be made by creators and, optionally, users of the

API. In the following chapter, prototypes are used to discuss how such mapping is used in real implementation.

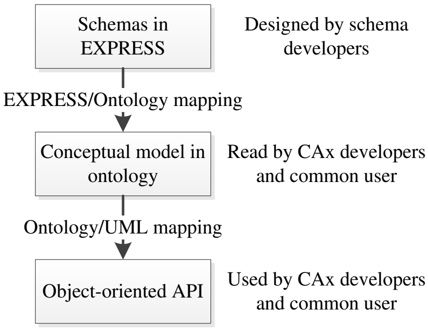

Figure 3.2 EXPRESS/ontology/API mapping

Chapter 4
Verification: case studies

Computers are like Old Testament gods: lots of rules and no mercy.

- Joseph Campbell

Numerous commercial CAx software systems have been developed and applied in various fields of industrial data management, e.g. product design, prototype testing, life cycle management, etc. Meanwhile, diverse types of partnerships between software vendors, industrial practitioners, and academic researchers have been built to enhance competitiveness. Therefore, a system-neutral solution of industrial data exchange is required in many different perspectives: geometry, kinematics, tolerances, classification and so on. STEP Toolbox comes into being to meet requirements of the CAx developers for friendly experience in EXPRESS-based implementation. The programming interface is the core part of the solution.

This chapter validates and discusses the application of STEP Toolbox. Four prototypes for different industrial domains are introduced, which are developed in-house or together with project partners. Comparison can be easily made between conventional experience and a new procedure with STEP Toolbox.

4.1 Kinematic data exchange[1]

Kinematic mechanism is one of the most important aspects of industrial data management. However, applications based on standardized data models to facilitate data exchange between miscellaneous CAx systems are very rare for this information type. The need arises very often for the kinematic mechanism data exchange between information systems through product lifecycle management. As Nassehi et al (2009) states in the field of manufacturing resources modeling, kinematic representation is very important in process planning, tool-path generation, multi-route process planning, intelligent CNC (Computer Numerical Control) controllers, and bi-directional data flow.

STEP p105 (ISO 10303-105, 1996) is a member of the integrated application resources of STEP. It addresses the kinematic mechanism representation. It was published in 1996 and its two technical corrigenda were published in 2000. Currently AP214 is the only application protocol that uses the first edition of p105. At present, a project is ongoing to develop the new application protocol, AP242, of which an important part is p105 ed2 as a new generation of the standardized kinematic mechanism representation (Hedlind et al., 2011). These latest standardization results, i.e. AP242 and p105 ed2, are used in this research.

Support for the kinematic mechanism data exchange is hardly found in related applications in industries. Today, most manufacturing companies have to use slides, fax, telephone, or paper-based documents to describe kinematic motion. Skilled CAx operators' manual re-input usually is the only way to bridge the gap between different systems. The existence of STEP p105 should have made significant improvement for industry, but there has been no known implementation for p105 valid modeling. Almost all found literatures refer to it only as a conceptual model rather than the standard for real implementation, e.g. Sudarsan et al. (2005), Rachuri et al. (2006), and Tanaka et al. (2008). A few early attempts have been made to include STEP p105 partly in implementations in different ways, such as Birla and Kang (1995), LKSoft (2004), and ESA (2007). Therefore the history of STEP-based kinematic modeling is not so long, as shown in Figure 4.1.

[1] This section is mainly based on (Li et al., 2011) and a part of (Li et al., 2013a).

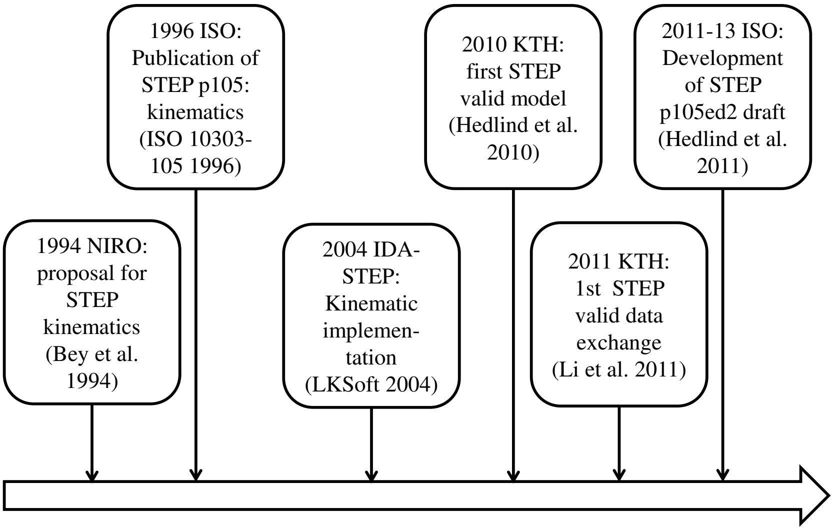

Figure 4.1 Milestone of kinematic modeling using STEP

During the period of this licentiate, the first valid implementation of kinematic mechanism based on STEP is developed. Prototypes are developed and integrated with Siemens NX and CATIA.

The need of kinematic translation initializes STEP Toolbox. Two plugins on Siemens NX with the exactly same functionality are developed in house to make detailed comparison for CAx integrations with and without STEP Toolbox. Besides, with the help of the toolbox, a CATIA plugin is developed by industrial project partners to implement the kinematic translation in their own systems. This implementation validates the expected benefits of this architecture, i.e. skipping the learning process and reducing the development workload.

4.1.1 Kinematics in STEP Toolbox

As stated in Chapter 2, the learning process and the development process are big obstacles to implement the standards. Besides, there is a remarkable lack of knowledge and experience for normal programmers regarding the AP242 and p105 ed2 due to its recent realization. Therefore, a user friendly solution to simplify the development is highly valued for developers from industry.

Information model mapping should be performed, as stated in section 3.3, to create kinematics as a module in STEP Toolbox. The mapping begins with analysis of the data model based on EXPRESS schema. The basic concepts of the kinematic mechanism modeling are common among all major CAD software: links, joints, and pairs are combined to represent the kinematic mechanism. In STEP p105 ed2, links act as rigid or linear flexible objects in topological structure (see Figure 4.2). Each joint is composed of two links to describe the topological aspect of the kinematic mechanism (see Figure 4.3). Kinematic pairs describe motion constraints and associations with geometry (see Figure 4.4). The data models shown here are all samples conforming to the AP242 MIM schema. Due to a lack of detailed documentation of this standardized modeling methodology for developers, it usually takes a long time for beginners to prepare and perform a successful implementation.

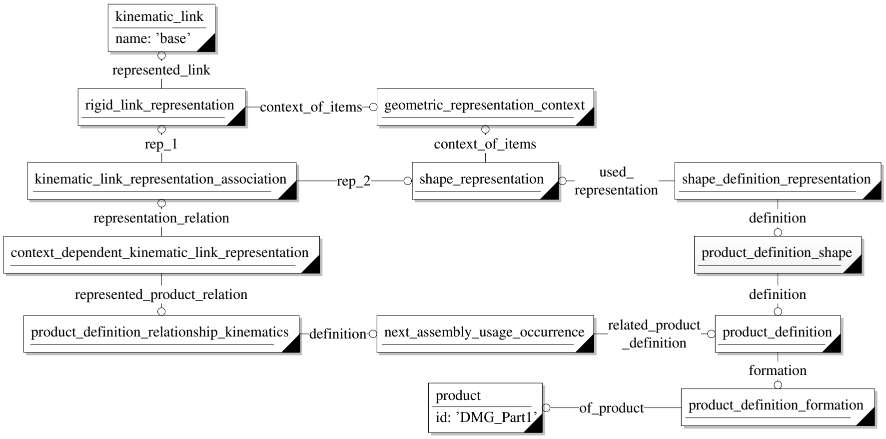

Figure 4.2 Information modeling of a kinematic link in AP242

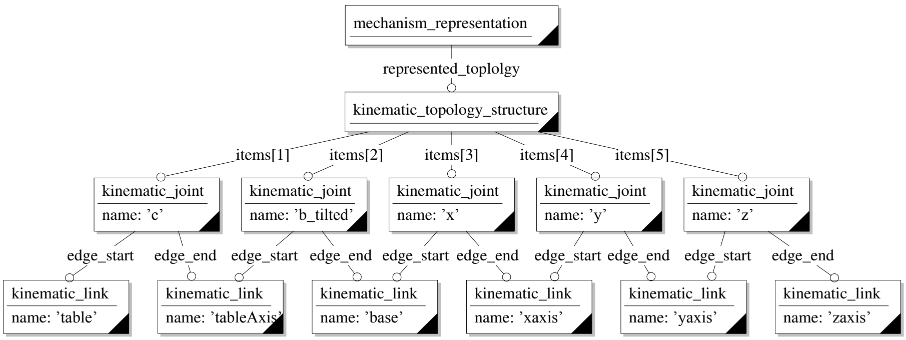

Figure 4.3 Topological structure of a 5-axis machine in AP242

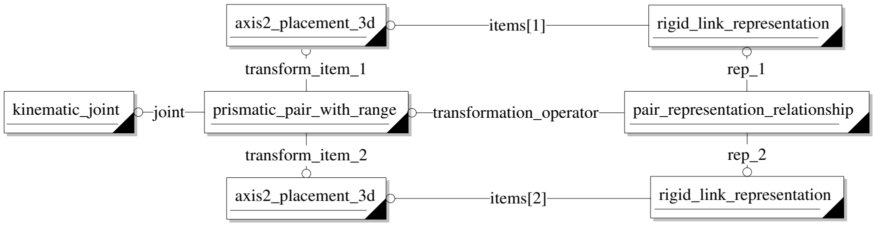

Figure 4.4 Information modeling for geometric aspect of a pair in AP242

The conceptual model of relative classes for kinematics is shown in Figure 4.5 in a simplified ontology. Note that properties and their facets of all the classes in the ontology are not shown in the figure. For instance, a kinematic joint has an attribute to define its motion type such as prismatic, revolution, cylinder, etc. To make it easily interpreted by computer, several simple terms are applied for the relationship, e.g. generalization, aggregation, association, and dependency. The ontology is a combination of kinematics and geometry module. A component is a basic geometric unit to compose an assembly. It is defined as a subtype of geometry which uses the placement to identify the location and the orientation. The self-reference of the component indicates the tree structure of the assembly tree. Each kinematic link is associated with a component. Two links make a joint which uses the placement to define location and orientation as well. The model can be extended further for other purposes. Section 4.2 will show its extension for kinematic error modeling.

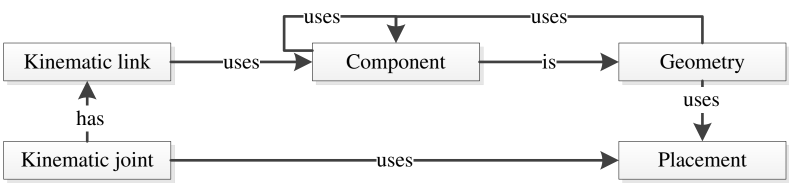

Figure 4.5 Simplified ontology for kinematics

The conceptual model can guide the generation of the UML class diagram of the API which is more helpful for programming. In this case, there are two types of classes: data model classes to represent basic elements and managers to provide static operations to manipulate the data model. One of these managers, the STEP model manager is the central controller to initialize, export, and close a data set. Each module has its own manager for creating, reading, modifying, and removing relative data.
In Figure 4.6, a simplified class diagram for the kinematic module in the object-oriented API is illustrated. Note that only important parts of the

implemented API are presented here. The kinematic manager collects available joints and links as static attributes from data sets so that the developers can easily manipulate them. For the data model, besides the 5 types of classes mapped from the ontology (Figure 4.5), an enumeration set is defined for the kinematic joint type. In STEP p105 ed2, 9 types of low order pairs are defined according to different combinations of degrees of freedom. Two most frequently used types are presented here as examples: prismatic and revolute. Getters, setters and constructors for all the data model classes are not displayed in the diagram to simplify the illustration, but they should be employed properly according to practices. For instance, in an AP242 kinematic model, the placement of each link in each joint is defined in a link frame i.e. the coordinate system of the corresponding component of a link. However, the placements of a joint may be defined in the global coordinate system in some software, e.g. Siemens NX. Therefore, there should be the constructors of the joint defined for local and global coordinate systems respectively. The class diagram provides an instruction to implement and utilize the STEP Toolbox API. The following section will present how to use the API of this example in different cases.

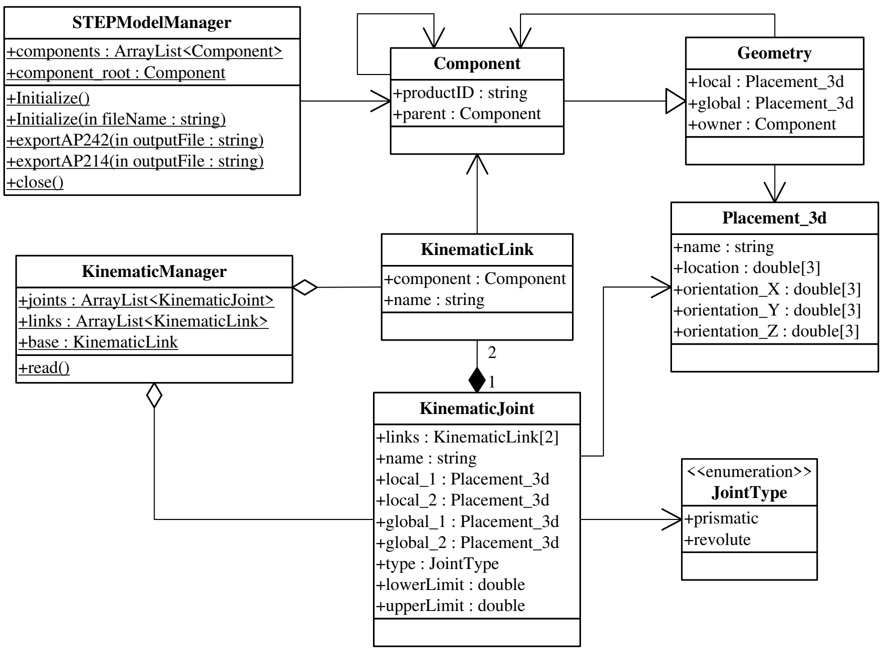

Figure 4.6 A simplified UML class diagram for kinematics in STEP Toolbox

4.1.2 Kinematic translation implementations

The first version of Siemens NX integration development is based on STEP AP214 (Li et al., 2011), from which the focus is changed to STEP AP242 as the new standard evolves. This previous application demonstrates the feasibility to develop the CAx-integrated applications for the kinematic mechanism data representation and exchange with the standards. Afterwards, a Swedish vehicle company, as a project partner, needs a fast and easy way to establish standardized kinematic translations integrated within their own CAD systems CATIA. Therefore, STEP Toolbox is designed, and at first tested in-house with Siemens NX. To make a concrete conclusion of this solution, two versions of kinematic translators are created for Siemens NX based on AP242: without and with STEP Toolbox. The system architecture of the first version (Figure 4.7) is similar to the previous research (Li et al., 2011).

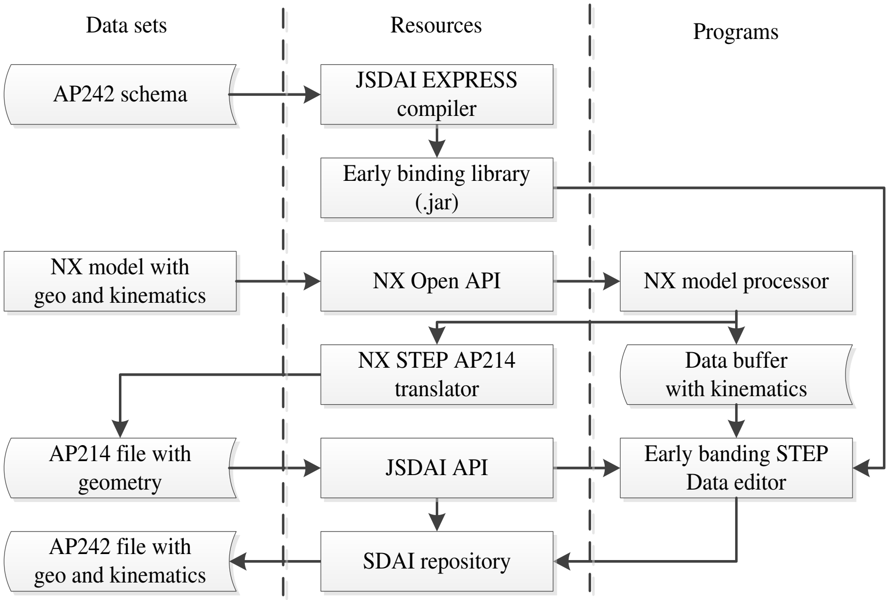

Figure 4.7 System architecture of STEP kinematic translator for Siemens NX without STEP Toolbox

The system integration is performed by combination of NX Open API and JSDAI API. The former is used to generate the AP214 data set of geometry via the NX native translator and to collect the data of kinematic mechanism. JSDAI API can be used to perform data integration of geometry and kinematics. Besides, JSDAI as a software package can compile a schema to a programmable dictionary library for early binding development.

For data integration, mapping between kinematic mechanism of Siemens NX and STEP AP242 should be achieved properly. As the AP214 representation of assembly and geometry is compatible with the AP242 schema, this part of the AP214 data sets exported from CAD software can be directly accessed with the SDAI dictionary of AP242. Hence, this development need not deal with cross-AP schema mapping. In this case, JSDAI is chosen as the API to manipulate the kinematic mechanism with early binding underlying the compiled AP242 SDAI dictionary.

With STEP Toolbox, the integration implementation becomes simpler as shown in Figure 4.8. This system can be developed by programmers without any knowledge of the implementation of the information standards. The schemas of each related APs are compiled into a SDAI-based library and encapsulated within the API so that users even need not know the concepts of APs, SDAI, schemas, early binding, etc. Data sets to be initialized, exported, and closed are controlled by developers with simple interfaces which compress all relative SDAI operations such as sessions, models, repository, and transactions. All manipulations of data sets are performed with the programmer-friendly API so that the knowledge on early or late binding is not important any more. Besides, this integration system separates the program interacting with Siemens NX from the data model editor by the data buffer. With a similar structure, this system design can be easily migrated to other CAx systems with APIs for any programming language.

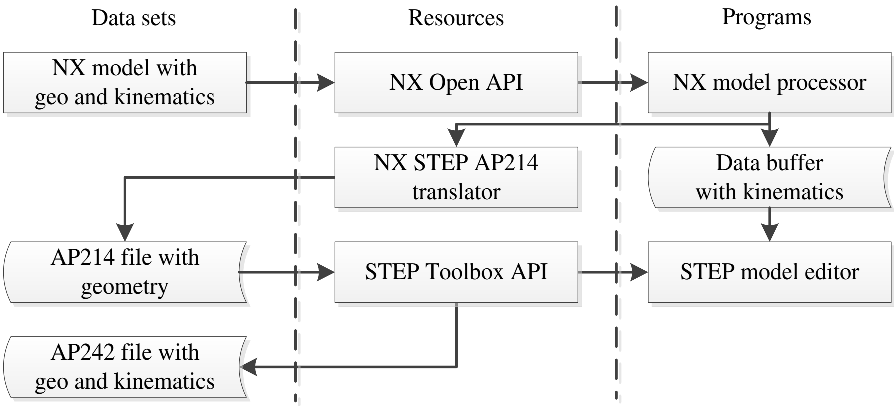

Figure 4.8 System architecture of STEP kinematic translator for Siemens NX with STEP Toolbox

As mentioned earlier in this section, the project partner from vehicle industry needs help to perform standardized kinematic translation for their CAx systems. Figure 4.9 shows how the system architecture can be migrated to a different CAx system, CATIA, and work with the CATIA VBA API. The application is a kinematic translator for CATIA based on AP242, implemented by developers in industry. Similar to the implementation for NX, the CATIA model processor is coded with the CATIA VBA API. Using STEP Toolbox API, a Java program, the STEP model editor, is used to manipulate the STEP data set. The format of the data buffer can be decided and designed by the developers. In this case, the data buffer in the format of CSV (Comma-Separated Values) acts as a bridge between the VBA script and the Java program. The STEP AP214 translator of CATIA performs as same as the translator in Siemens NX. It can be also invoked by the CATIA VBA script to export the AP214 file with geometry. Then, the STEP model editor built upon Toolbox API is able to integrate kinematic data with geometry and generate an AP242 file.

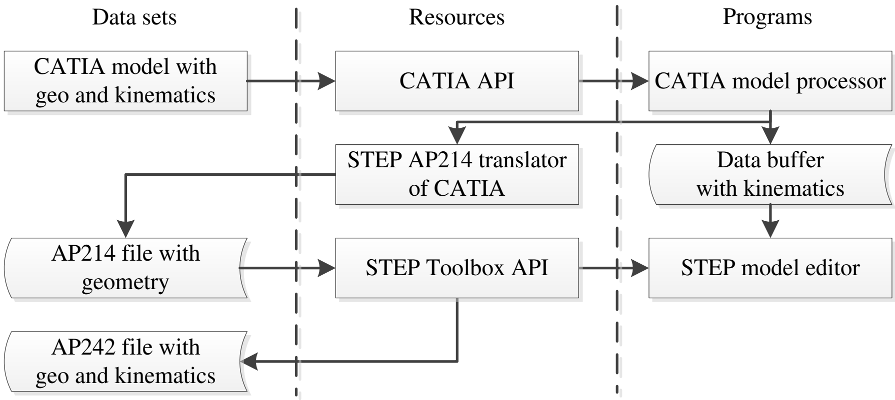

Figure 4.9 System architecture of STEP kinematic translator for CATIA with STEP Toolbox

To share STEP Toolbox API, Javadoc and sample codes are two important documentation methods. For any Java-based API, Javadoc is a primary way to provide a detailed instruction on each element with a structured HTML (Hyper Text Markup Language) framework. However, it is still hard to start programming simply from Javadoc. Therefore, the sample codes are provided as compensation, from which the users can find how to initialize, modify, export, and close a standard data set in a practical way.

Until the end of this implementation, the industrial developers have not learned any detailed information from standard documents. As any other normal programming tasks, Javadoc and sample codes are only references to look up. Compared with the traditional SDAI way, STEP Toolbox is proven as an efficient candidate for implementation of the information standards.

4.2 Kinematic error implementation[1]

An important application of kinematic mechanism modeling in machine tool industry is kinematic error modeling. Geometric accuracy of machine tool operations is highly affected by kinematic errors in terms of component errors and location errors (Schwenke et al., 2008). To achieve high accuracy in machining, calculation methodology (multiplication of homogeneous transform matrices) for kinematic error compensation (Donmez et al., 1986) is commonly used. For the application of virtual machine tool (Altintas et al., 2005) representation of kinematic errors is a key factor to achieve accurate simulation of reality. Although applications for kinematic error utilization and analysis are not as common as for nominal kinematic analysis at present, it can be expected that more and more commercial CAD/CAM/CNC systems will deliver such functionality to support highly precise manufacturing. Figure 4.10 displays a basic scenario in production to integrate the virtual machine in process plan. The representation of kinematic errors can help fast and precise decision-making with virtual machining simulation.

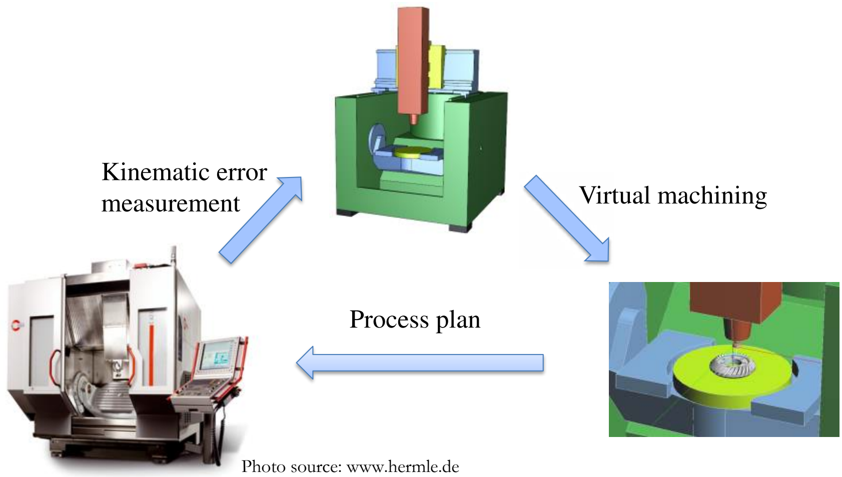

Figure 4.10 Machine tool modeling for realistic virtual machining

As what happened in the area of product design, multiple systems may be used to perform different functionalities with the kinematic errors in manufacturing. This requires the modeling of the kinematic errors should be able to integrate with different information domain to support functionalities in a broad scope, e.g. measurement, analysis, prediction, and compensation. This is the reason why STEP AP242 with p105 ed2 is chosen in this research

[1] This section is mainly based on (Li at al., 2012a).

(Hedlind et al. 2011). A major improvement of this new edition of STEP p105 is reusing the existing general representation construct defined in ISO 10303-43, where the definitions and relationships of representation, items, and contexts are described. Such structure provides possibility to define properties associated with different levels of kinematic elements: mechanism, kinematic pairs, and mechanism states.

The placement of nominal kinematic pair frames, as defined in STEP p105 (see Figure 4.11), is used to relate the kinematic error concepts unambiguously in the context of STEP AP242. Component errors and location errors are basic types of the kinematic errors which have more detailed definitions according to the relative rotational and linear axis. The component error is the pair placement error from the respective nominal placement, and the location error is defined as the average of contact frame placement errors from its nominal placement, against the machine coordinate system.

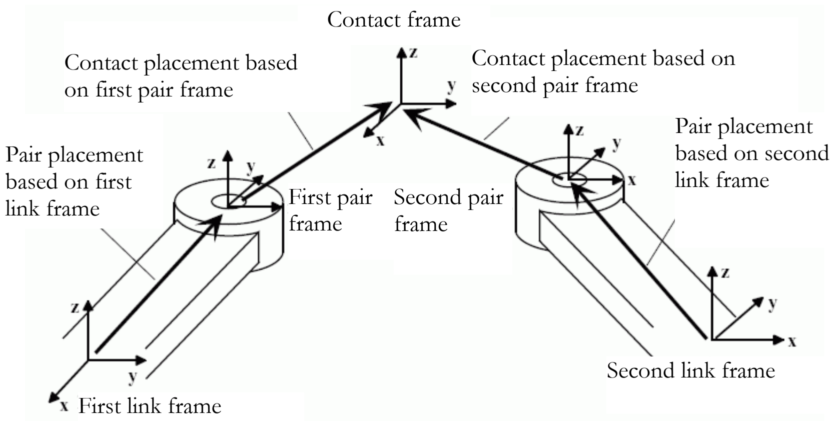

Figure 4.11 Placement relationship of a pair (ISO 10303-105)

This research performs the modeling based on AP242 to represent the kinematic error as a property of kinematic elements. The general property representation structure defined in ISO 10303-41 is used to represent the measured or calculated kinematic error information and the associations to corresponding kinematic instances defined in STEP p105 ed2. The data integration between geometry, kinematics, and kinematic errors is evaluated and demonstrated by a prototype implementation based on STEP Toolbox.

4.2.1 Kinematic error in STEP Toolbox

The concept of kinematic errors highly relies on kinematics and geometry in STEP AP242. A data set example in Figure 4.12 shows the modeling structure for two component errors underlying the kinematic pair value model. A mechanism state is associated with a property named “pair state geometric error”. The representations of component errors are collected and compounded under the context of one representation referred by the property definition. The naming convention defined in ISO 230 is used as the name of each measurement value with unit, combined with the approach direction symbol ↑ or ↓ defined in the ISO 230-2 (2006). The product definition of the assembly model is linked as the owner of the kinematic error property. A general property is assigned to the component error to declare the terminology and concepts defined in ISO 230-1 (2012). The functional point is the location of the tool in the moving component when performing the straight line motion test as defined in ISO 230-1. Therefore this location is represented in the coordinate system of the moving link. The “pair state geometric error” as a derived property definition should be externally classified to the corresponding property defined in ISO 230.

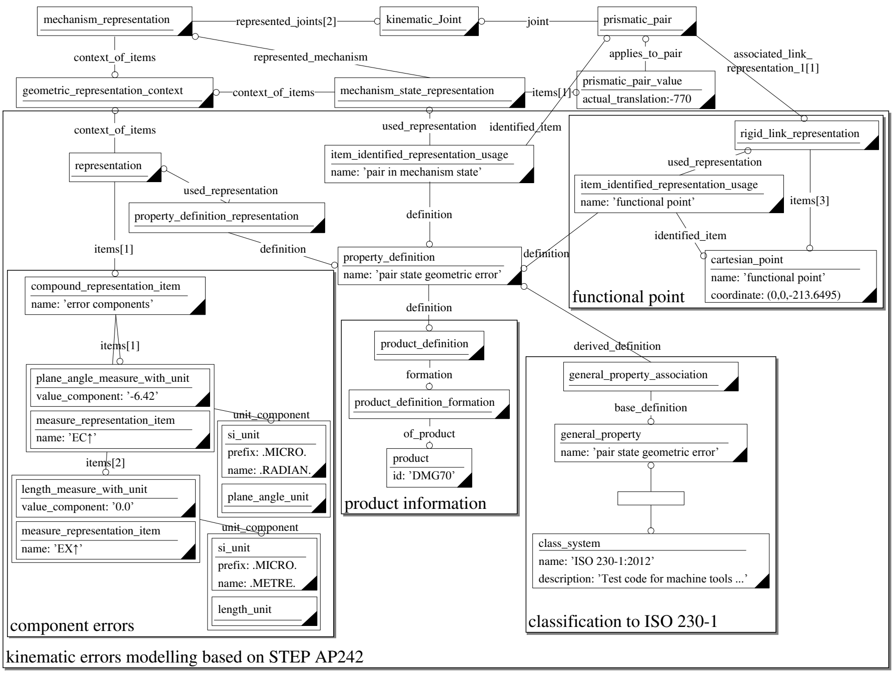

Figure 4.12 Component error modeling in AP242 (Li et al.., 2012a)

As previously mentioned in this section, the location error is an average value that is to be associated to a kinematic pair independent of mechanism state. Hence, instead of the mechanism state as shown in Figure 4.12, the mechanism itself should be referred to as the used representation of the item

identified representation usage. Thus, the modeling for location errors can be achieved in a similar way as component errors.
The conceptual model for kinematic errors is also an extension of kinematics, with a few references to geometry. Figure 4.13 illustrates such extension from the conceptual model of kinematics (see Figure 4.5). The pair value is dependent to the kinematic joint and supports the existence of the component error. The location error is directly associated with the kinematic joint. Both error types are subclasses of the kinematic error, which is associated with the function point.

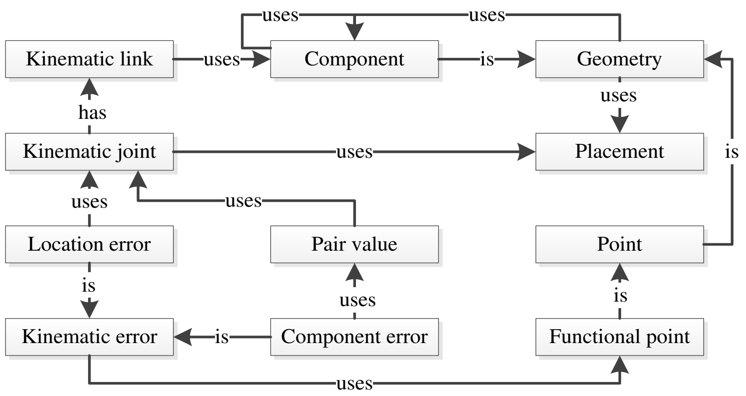

Figure 4.13 Simplified conceptual model of kinematic errors

The ontology can be mapped to a UML class diagram (see Figure 4.14) which is used as a part of the STEP Toolbox API. As illustrated in the ontology model, components and points are both subclasses of geometry which refers to placement information. Objects of both links and joints contain both topological and geometrical information. The property is also a general class that is actually used not only by the kinematic joint, but also as some extensive usages such as dimensions and colors. The kinematic error is a subclass of the property which is aggregated in the joint. Pair value is a pair parameter for special usage, and is a parameter with a collection of component errors. For the location error, the developers can simply define a null value for the pair value which indicates that the error value is independent of the mechanism state.

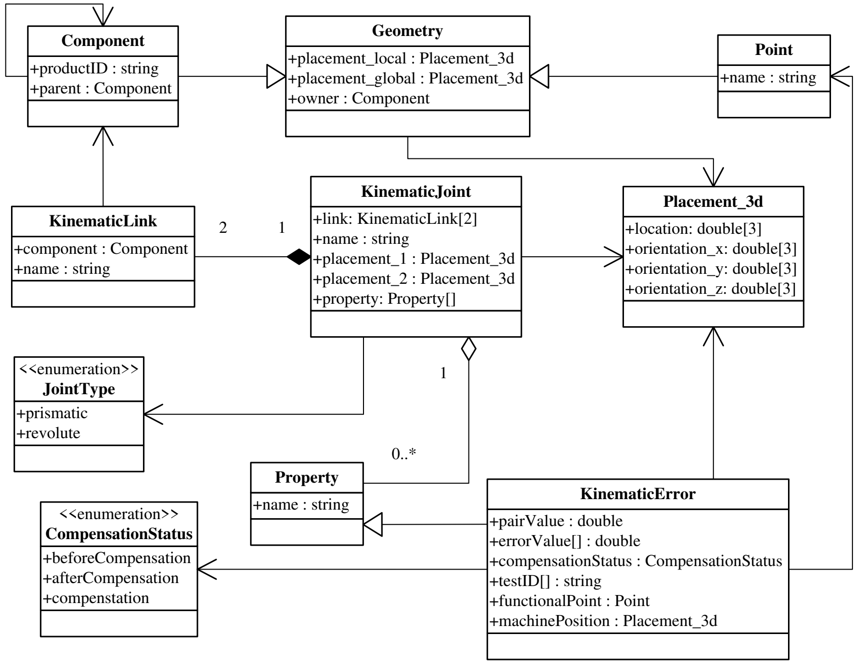

Figure 4.14 Simplified class diagram for kinematic errors

4.2.2 Kinematic error implementation

To evaluate the capability of the proposed modeling approach and implementation methodology, a prototype application is developed to translate an ASME B5.59 (ASME, 2008) XML machine tool data model to a STEP AP242 model. The original data is based on version 13b of the draft standard ASME B5.59-2 (Data Specification for Properties of Machine Tools for Milling and Turning). Performance and capability data of the machining operation is described in the original XML file.

Such application can also help to study the AP242-based data integration between geometry and kinematics in the contexts of machining process. A major benefit of STEP AP242 is the high level of integration between different aspects of the described product, e.g. geometry, kinematics, classification, dimensions, and tolerances. Multiple levels of details and aspects of the machine tool model can be adopted by different applications and integrated as a comprehensive modeling solution. An AP242 model can be read and written by various CAD/CAM systems during its lifecycle without the risk of unwanted information loss. The compatibility with geometry of AP214 also means functionality of STEP translators in existing systems can be reused for standardized extensive exchange of other types of information such as kinematic error data. Such data integration specific for kinematic errors should be evaluated by prototypes implementation. In this research, the development takes the kinematic error model defined by ASME B5.59-2 and its corresponding shape geometry exported from CAD software as an example. As a standalone translator, the system architecture for this implementation is very simple with the help of STEP Toolbox (see Figure 4.15). The input is the XML sample file conforming to ASME 5.59-2 provided by NIST within ISO TC184 SC4 WG3 T24 collaboration, which contains machining capability information of a machine tool. The other input is an AP242 data set with geometry and kinematics of the same machine tool exported from the implementations developed in section 4.1. The output AP242 file will be an integration of the information from the inputs, i.e. geometry, kinematics, and kinematic errors. An ASME XML reader is developed to read the component errors described in the XML file. The STEP Toolbox is used to retrieve geometric and kinematic information from the AP242 file, and to generate required standard models integrating kinematic errors. The generation requires the kinematics module in the toolbox reused and updated with the kinematic error, which embodies the maintainability and reusability of the STEP Toolbox.

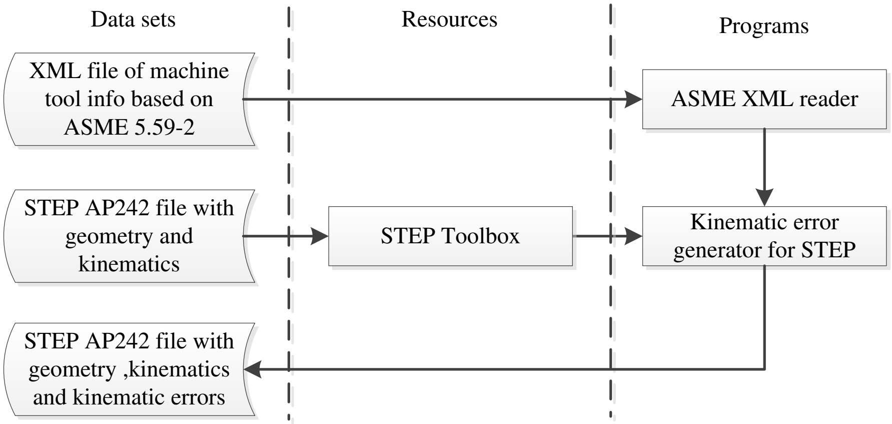

Figure 4.15 System architecture of ASME/STEP kinematic error translator

Currently, more data sets, schemas, and measurement practices are being involved in the research. The presented data model is being improved and expanded during discussion with international partners. As a contribution to the ISO TC184 SC4 standardization work, this modeling approach will be a base for schema-level extension in future development.

4.3 Cutting tool implementation[1]

GD&T is another important aspect of information modeling in CAx systems. With proper classification, semantics of GD&T can be used for many different applications in design and manufacturing. Researchers have done great jobs for the standardization in this area. The joint of ISO 13399, PLib, and STEP are promising to support the data exchange of GD&T with classification regarding cutting tool information (Hedlind, 2013). As several schemas and data sets are involved, the context for implementation can be much more complicated in kinematics and geometry. Therefore, developers can benefit from STEP Toolbox to skip most parts of preparations for the implementation of the information standards.

4.3.1 Introduction of cutting tool modeling

For decades, information management has been regarded as the essence of cutting tool management for efficient computerized manufacturing control in all kinds of CAx systems (Veeramani et al. 1992). With PLM (Product Lifecycle Management), engineering information of cutting tools is integrated to support its development and deployment through the lifecycle. With CAD, cutting tool geometry, assembly structure, and relevant properties are defined in different viewpoints for production and utilization. With CAM, cutting tool requirements and usages are defined for operations with tool paths and cutting parameters. With CNC, operations are executed with input on cutting tool types, main dimensions, and tolerances on tool wear. However, differences in terminology and data format are blocking cross-system interoperability.

Standardization of representation and classification of cutting tool data is fundamental for cutting tool data exchange between such CAx systems. An unambiguous way for the cutting tool data modeling and exchange is ontology based on industrial standards. A standardized hierarchy of classes and properties is the basic structure to describe taxonomies, as for cutting tool ontology. An important contribution in this area is a dictionary of cutting tool classes and property types within ISO 13399. The dictionary is standardized underlying PLib. A data set conforming to ISO 13399 is the cutting tool data representation including its classification referencing the dictionary. On the other hand, shape representation is set outside the scope of ISO 13399. If needed, geometric models in other formats, e.g. STEP AP203/AP214/AP242, can be referred to from the data set. This separation approach is traditional for PLM systems.

[1] This section is mainly based on (Li et al., 2013b).

However, the standardized solution of ISO 13399 still has unsolved issues in representation structure and implementation methodology. ISO 13399 and STEP share the same functionality of GD&T representation but with different modeling approaches. In STEP, the dedicated representation schema for GD&T is a basic functionality that is preferably reused for cutting tools, and the connection to geometry can be easily established. ISO 13399 instead uses general properties without the geometric context for shape dimensions. This does not meet requirements for high data consistency and interpretation precision in future CAx applications. Another problem of the ISO 13399 representation structure is the lack of a common schema for geometry and GD&T. It results in a lack of a complete context and an unambiguous representation. With the representation of e.g. a diameter classified as cutting diameter, the link to a specific geometry shape element is important information required in industry.

The barrier for implementation based on ISO 13399 is the management of multiple data sets and associated data schemas (see Figure 4.16). Note that ISO 13399-1 (2006) is mainly a subset of AP214. It makes two standards with almost equivalent schemas to be synchronized for implementation and maintenance. Besides, the ISO/TS 13399-150 (2008) usage guidelines describe implementation differently from how it is done for STEP AP214. It results in great efforts for the practitioners to develop and coordinate separate implementations. Considering the high possibility of involvement of PLib standard, it makes three different schemas that need to be coordinated. For instance, an ISO 13399 cutting tool data set with PLib-based classification is transferred from a tool supplier to its customer with the corresponding geometry data set of STEP AP214. System developers have to establish and maintain data integration based on the three standards and three data sets (the ISO 13399 cutting tool data, the AP214 geometry data, and the PLib-based cutting tool dictionary).

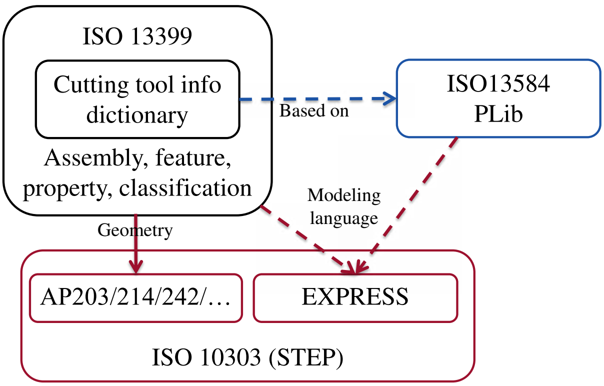

Figure 4.16 Standardized cutting tool modeling solution based on ISO 13399

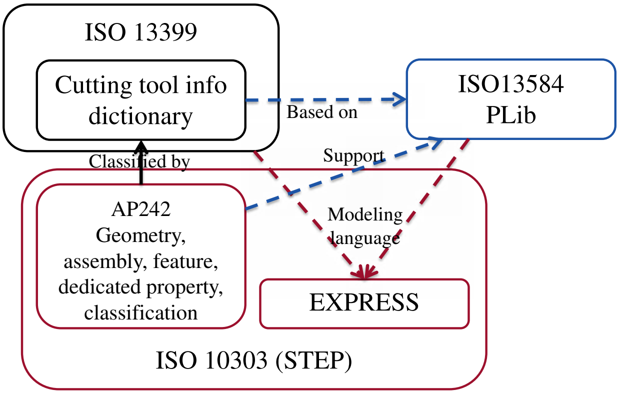

Figure 4.17 Standardized cutting tool modeling solution based on STEP AP242

A standardized approach to represent cutting tool information is important for modern industry. The dictionary provided by ISO 13399 is promising as a system independent cutting tool library, which is able to bridge various suppliers and their customers (Zelinski, 2011). Since it very well fulfills its defined scope for cutting tool data representation, ISO 13399 has contributed to several researches. Kaymakci et al. (2012) adopt concepts from ISO 13399 for a general prediction model of inserted cutters, where they point out the demand for "a unified geometric, kinematic, and mechanics model" cannot be met by ISO 13399 solely. Helgoson and Kalhori (2012) use ISO 13399 for cutting tool data exchange in the context of machining process planning, where the solid model and cutting tool parameters are separately represented with STEP and ISO 13399. Chungoora et al. (2011) highlight the problems for joint usage of standards and present an ontology-based framework to consolidate various information standards for industrial data management, e.g. ISO 10303, ISO 13399, ISO 13584, and ISO 15531. Generally speaking, integrated geometry models are commonly required regarding the adoption of ISO 13399. It becomes a must that multiple schemas are integrated to achieve a complete modeling solution. Therefore, it is necessary to explore a unified modeling approach and extensible development architecture for standardized cutting tool data exchange.

Within the framework of STEP AP214, dimensions and tolerances are defined in a specific UoF (Unit of Functionality). Classification of items based on PLib is supported by AP214 in another UoF, which provides association with any dictionary conforming to PLib, e.g. the ISO 13399 cutting tool library. The draft standard AP242 is going to replace AP214. Added capabilities in AP242 include supports for the Geometrical Product Specifications (GPS) standard.

Dedicated schemas for GD&T and external references are integrated into this protocol and can be used to represent the needed information for CADCAM and CNC cutting tool data exchange as well as PLM applications. Hence, the modeling solution shown in Figure 4.16 can be change into Figure 4.17, where the users only need the AP242 data set to exchange all the information regard cutting tools.

4.3.2 GD&T and classification in STEP Toolbox

Multiple types of information regarding cutting tool data are considered in this research, which requires data integration among schemas. The generic information modeling schemas in STEP are able to support association of classes and properties defined in the ISO 13399 dictionary with products, shape elements, GD&T, features, and other elements. This capability also applies for different levels of details for different user requirements. For instance, tool suppliers use a complete model with a high level of detail for internal data exchange, but tool customers may only need the basic cutting tool parameters. Thus, exact geometry and other detailed design requirements, which in some cases also are confidential or unnecessary, should be trimmed from the complete model for external data exchange.

Figure 4.18 presents a STEP AP242 data excerpt of a diameter representation associated with a face on a solid body representation. In the model, a diameter of 22.0 mm with a tolerance of 0.05 mm and -0.1 mm is defined for the face. All geometric information is represented in the established way of STEP, which is omitted in the empty block. Using the standard modeling concept of the shape aspect, dimension definitions, measure representations and other properties are precisely integrated.

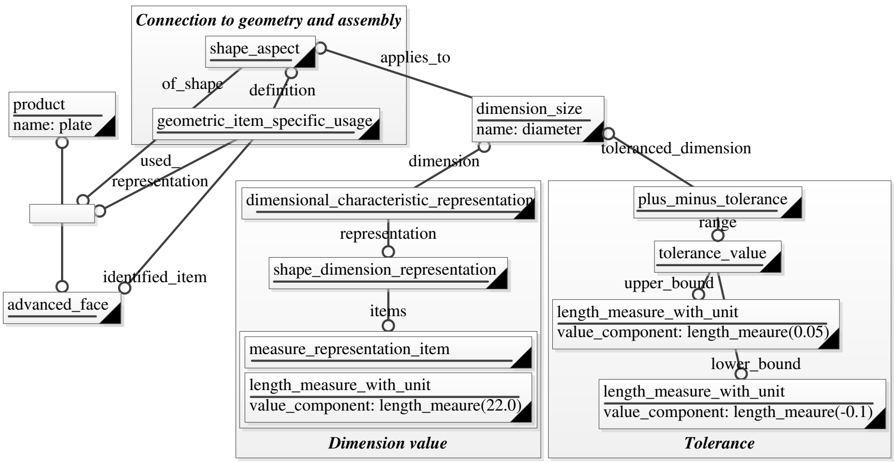

Figure 4.18 Example of GD&T modeling in STEP AP242 (Li et al., 2013b)

Business cases may require hidden shape information, and then there will be no link between the dimension definition and the face. That is, the dimension itself can be represented independent of existence of shape information. Nevertheless, for a complete definition of GD&T, the geometric context is needed to establish coordinate system for the dimensions even without detailed data about geometry. Supplemental geometry can be helpful in such cases, which is also known as help geometry or constructive geometry. Example data types are placements, points, curves and faces. These data types are not used to define shapes, but to support the definition of design requirements. It is a common CAD function also available in STEP. Recommended practices for implementing supplemental geometry are published by the CAx Implementers forum (Bay and Rosché, 2010).
Figure 4.19 exemplifies the classification modeling approach proposed based on AP242 (Li et al., 2013b). The ISO 13399 cutting tool dictionary is referenced in the model. Description of an end mill with a usable length is retrieved from the library and associated with the corresponding product and dimension. A usable length in this data set is specified as a property of the end mill. The definitions and the relationship is defined by the ISO 13399 cutting tool dictionary but stored in the STEP AP242 data sets. Hence, the data set can exchange the classified information with standardization references, but the dictionary (known as ISO 13399 here) is not necessary to be attached for data exchange.

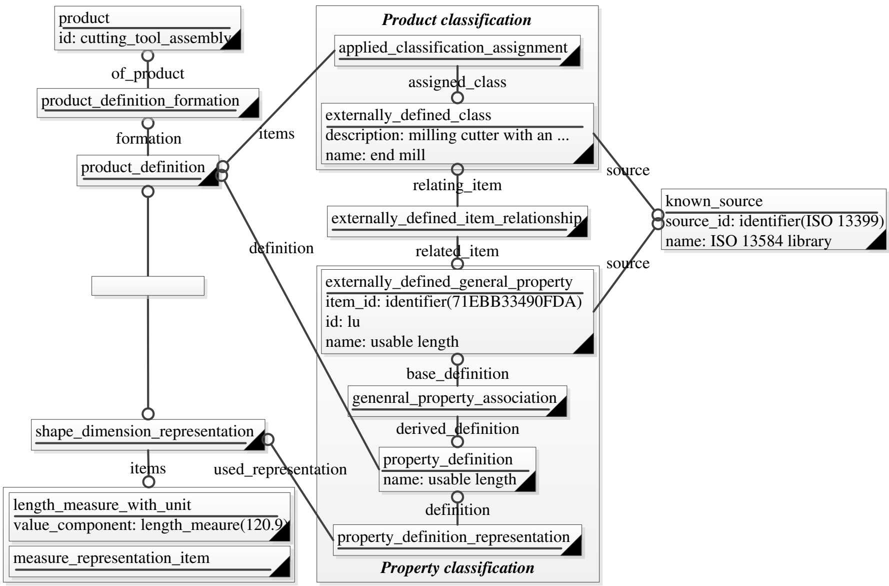

Figure 4.19 Example of classification modeling in STEP AP242 (Li et al., 2013b)

STEP Toolbox aims to provide a friendly programming interface for CAx developers, rather than designers in a specific industrial domain. As three standards based on EXPRESS are involved, the conceptual model should be defined in a domain-independent way to bridge these standards. The simplified ontology defined in Figure 4.20 explains the concepts involved in this type of implementation. The classes for components, geometry, and placement are same as the conceptual model for kinematics. PLib classes and PLib properties are data types with external references to the PLib-based dictionary. The self-reference of the PLib class indicates the tree structure of the hierarchy. In the toolbox, the PLib classes and PLib properties not only can represent the references to a dictionary in a STEP p21 data set, but also can exist independently to represent a dictionary in a PLib data set. Therefore, the references from AP242 to PLib dictionary can be established by simply borrowing the needed objects. Note that the presented ontology model mainly serves the purpose of supporting implementation. It does not necessarily express semantic relationship precisely. For example, a general feature should not be semantically defined as a sub-class of the component. However, it also can be classified and has classifiable properties, which indicates all the requirements of the general feature in implementation can be met by the attributes and operations of the component. To simplify the conceptual model, the general feature is set as the subclass of the component here. A property such as a general property or a dimension is associated with a certain component or a general feature, which can be defined semantically by PLib Property, e.g. a linear distance associated with a component is defined as a usable length in a PLib dictionary.

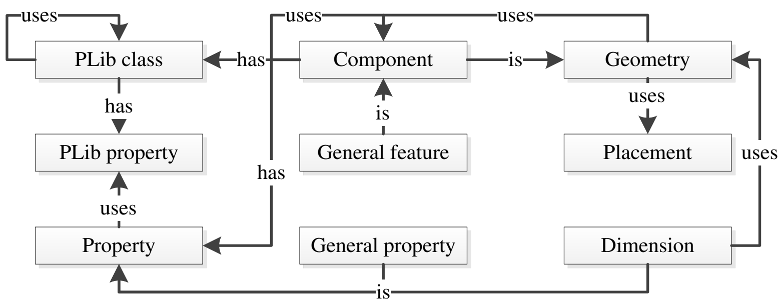

Figure 4.20 Simplified ontology of cutting tool modeling

Programmers may use the ontology model to understand concepts easily, but a well-designed API is fundamental for practical implementation. In Figure

4.21, a simplified UML class diagram of the API for cutting tool is illustrated. Methods of most classes are still not displayed, but they are important for implementation, such as getters, setters, and constructors. As a part of STEP Toolbox, modules are divided and controlled by managers, such as the property manager and the classification manager. Differences from ontology may occur for specific data model implementation. For example, both components and PLib classes have tree structures, but PLib classes also need to record a list of parents, rather than only the direct parent which is the case for component. PLib classes inherit the properties from all the parents. The list of parents is helpful for developers to track the exact parent which supports the inheritance of a certain PLib property and to attach the related parents to the model.

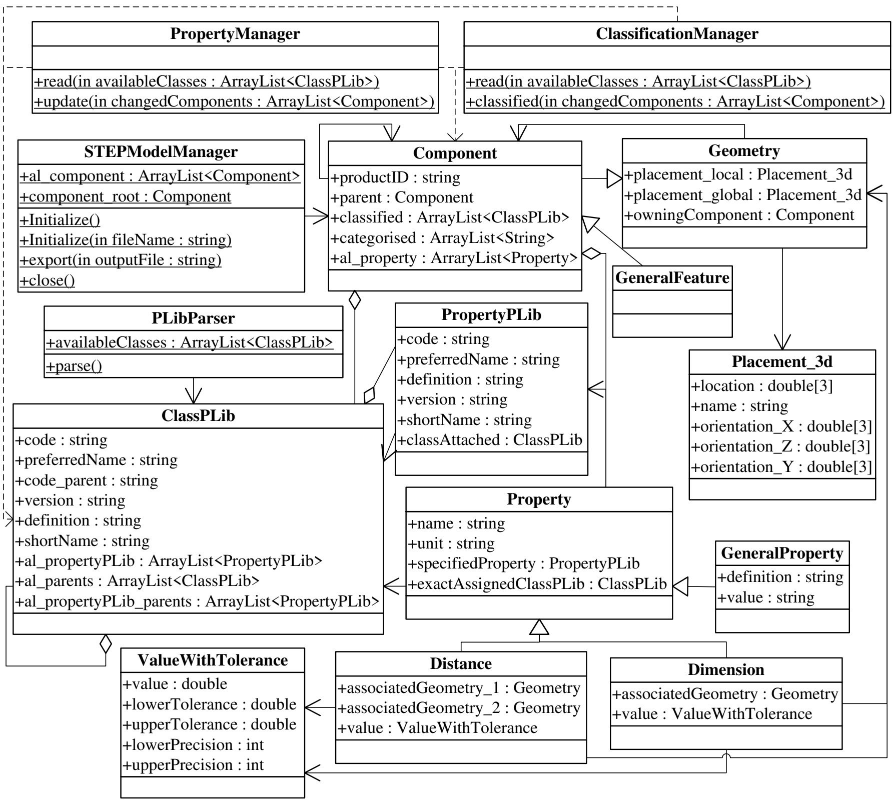

Figure 4.21 Simplified class diagram for cutting tool development

4.3.3 Implementation

The prototype implementation aims to evaluate the feasibility of the proposed modeling approach for cutting tool information mainly based on STEP AP242. Since ISO 13399 and PLib are involved, STEP Toolbox should also be examined with the similar development strategy as previous case studies.
The major function of the prototype software is to classify products and other properties based on AP242 referencing the ISO 13399 cutting tool library based on PLib. The input is a p21 file of AP242 with its assembly structure, shape features, dimensions with tolerances, and other properties. Shape representation or supplementary geometry is optional. The software should be able to identify all the relevant information and present it to users. The input model is created in Siemens NX and exported from a developed plugin which is designed similar to the kinematic exporter in section 4.1.2. The software should also be able to read the ISO 13399 cutting tool dictionary based on PLib and help the users classify products, properties, and features, which requires an interface to present the dictionary and the items to be classified together.
The system architecture of this implementation can be illustrated in Figure 4.22. Three modules of STEP Toolbox are utilized for different purposes. PLib module can parse the ISO 13399 cutting tool dictionary and provide a complete collection of PLib classes of this dictionary to developers. Geometry module is used to get the assembly structure of the model in the AP242 file. GD&T module retrieves the cutting tool information other than geometry, such as general features, dimensions, tolerances, and general properties. The program needs to present the ISO 13399 dictionary and the cutting tool information to users, and collect user requirements to classify the cutting tool information.

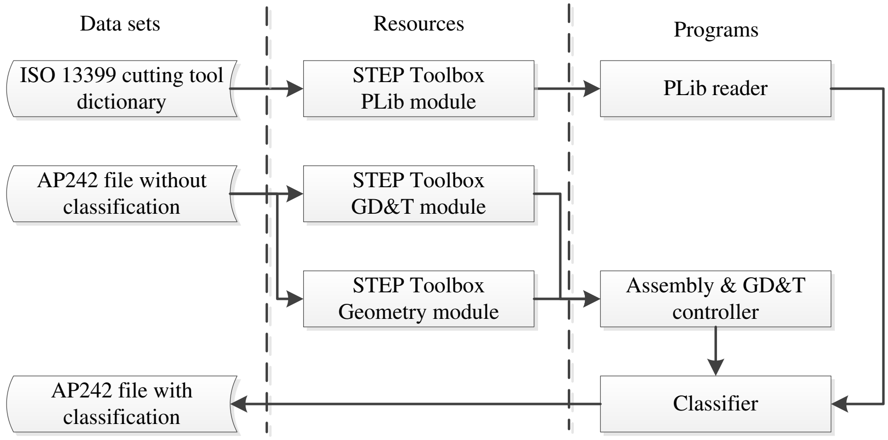

Figure 4.22 System architecture of the classification application based on AP242

The involvement of three standards could make beginners feel difficult to start with. Although the research adopts STEP AP242 as the core for data exchange to skip the schema of ISO 13399, developers still need to coordinate the schemas of AP242 and PLib which have some similarity. Therefore, STEP Toolbox not only helps developers to skip the learning process, but also simplify the development procedure by hiding the coordination of different schemas behind the cooperation of modules.

4.4 Point cloud modeling[1]

This interdisciplinary implementation is cooperation between two projects. Su et al. (2012) developed a dedicated high-precision image processing algorithm for the measurement based on OCT (Optical Coherence Tomography). Further applications of this new technology in industry require system integration architecture to coordinate software for inspection and measurement. Data model based on standards should be adopted to support interoperability within the structure. STEP Toolbox is introduced to assist the establishment of the measurement system with OCT.

Micro-manufacturing industry requires a high-precision, non-destructive, subsurface detectable, rapid and automated inspection technique for in-process 3D monitoring (Bredeau and Federzoni, 2009). Applicable scenarios include measuring the thickness of component layer, determining the shape dimensions of the embedded 3D structures, and detecting de-bonding, cracks, warping and deformation. A promising metrology technique is OCT which has been actively used in many biomedical applications. The industry of ceramic micro processing can benefit from its advantages of non-contact, micrometers resolution and high acquisition rate (Bouma and Tearney, 2002). The data set of the OCT signal is usually point cloud which may be utilized in different contexts. For defects inspection, the designed CAD model should be compared with the measured model composed by the point cloud with special matching methods and algorithms. For dimensional measurement, shape reconstruction and shape recognition may be performed in different applications.

4.4.1 Architecture of measurement system

As a new measurement technology, there is no existing information system that completely supports OCT in manufacturing. Therefore, architecture has to be defined to integrate different applications and data models for different usages. Principles for modeling and exchange should be defined to integrate related CAD/CAM software in this architecture. Requirements on the data

[1] This section is mainly based on (Li et al., 2012b).

schema for a model-based solution can be outlined from the architecture. A data modeling approach should be proposed to enable the OCT measurement in industry and maximize the application of its unique ability.
Most researches on computational system for measurement follow a common stepwise data processing flow. Tang et al. (2010) present a summary of the procedure for creating BIM (Building Information Model) from the laser-scanning point cloud in the context of AEC (Architecture, Engineering, and Construction) industry. Three steps are needed to transform the original measurements to semantic rich BIMs: 1) data collection to collect scanning signals, 2) data preprocessing to filter the data and formulate the coordinate system, and 3) modeling the BIM to construct surface representation and recognize semantic information. In a practice guide of vision system applied in the dimensional measurement (Rodger, 2001), a similar stepwise procedure is defined: 1) image formation and sensing, 2) image processing, and 3) communication.
Based on the previous researches, the architecture of an OCT-based measurement system is presented. Such system can be divided into three layers (see Figure 4.23):

- Collection: producing the image from OCT,
- Processing: translating the image to the computer interpretable model,
- And application: utilizing the model in different contexts.

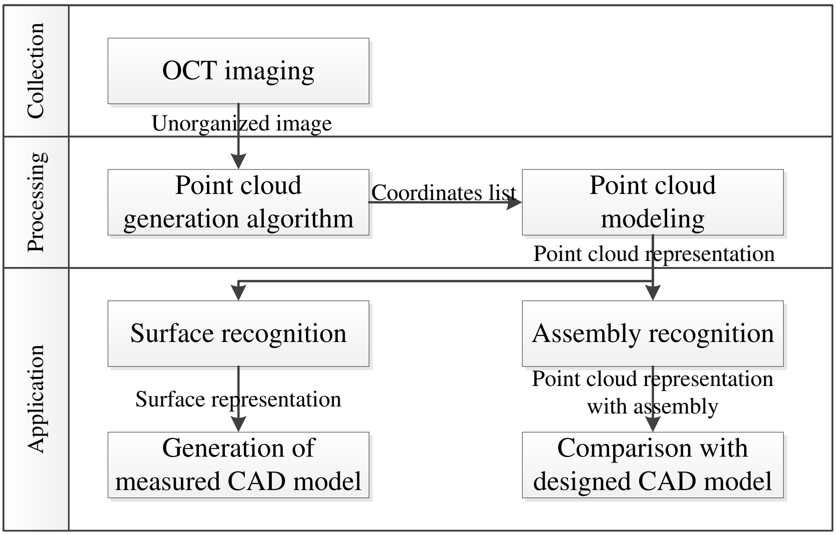

Figure 4.23 System architecture of OCT-based measurement system

At first, project partners perform the OCT imaging to collect subsurface images of the subjects. In the processing layer, the image processing algorithm can translate the images into a list of coordinates as an ASCII data set. To make it interoperable for other types of applications, the data set should be translated to a point cloud representation based on the standardized data modeling approach. To be prepared for computer interpretation, information on geometric representation context is involved: unit, uncertainty, and placement of the coordinate system.

In the application layers, the standardized representation can be applied in different scenarios. For geometrical dimension measurement, with existing commercial reverse engineering software, surface recognition and modeling can be performed. The output surface representation may be used in other applications for final processing, where features, components, assembly, and other types of semantic information are associated to the measured shape model. OCT provides an opportunity to apply the point cloud representation for inspection of subsurface in assembly structures. As mentioned before, an important feature that distinguishes the OCT technology with the others is the rapid detection of subsurface defects and embedded structures, e.g. channels and designed cavities. Such feature enables in-process inspection of both components and assembly. Therefore, a complete point cloud model with an assembly structure is possibly generated with the same recognition methods as surface recognition. If the CAD model of assembly and the comparison algorithms are available, defects inspection can be made for assembly accuracy and internal structures of components. Existing matching methods include area based matching, feature based matching, centroid method etc. (Chen, et al. 2000) The measured point cloud representation and designed CAD model can be employed in separated or merged standardized data sets to support computer-aided matching.

4.4.2 Point cloud in STEP Toolbox

The point cloud modeling is included in the measured data UoF in AP214. However, there is no known implementation or research applying this UoF. At present, the new application protocol AP242 is under development to replace AP214 and AP203. In this research, AP242 is employed to model the point cloud and a modeling approach is introduced to support different aspects of data model requirements: point cloud representation, assembly representation, etc.

Although there is great similarity between AP 214 and AP242, especially in the areas related with geometry, the explicit representation of point cloud has not yet included in the current version of AP242. To facilitate AP242 in this domain, the AP214 ARM can be reused to model point cloud compatibly within existing modeling principle for general product information representation of AP242. The intent use of point cloud representation is explained in AP214 AAM for the measured data UoF, as applicable for prototype test results, conceptual design information, and tool test results. In this research, this data model is reused for specific application areas such as in-process inspection and surface recognition, and still within the defined scope of UoF. In the current version of AP242 under development, the similar functionality has not been specified explicitly, but related concepts can be implemented implicitly by reusing the general representations which are super types of those defined in the AP214 AIM schema. This theory leads to a sample of data model defined with AP242 AIM as shown in Figure 4.24. This model is mainly based on AP214 AIM mapped from the measured data UoF. For specific attributes of instances, a preliminary naming recommendation is introduced, but further discussion is necessary to achieve a modeling convention or update suggestions in the schema level. A shape representation named “point placement” is used to replace point placement shape representation defined in AP214 as a subtype of shape representation. A compound representation item indicates a group of unsorted Cartesian points. The other instantiations such as product, application context, geometric context, and shape, are in line with the AP214.

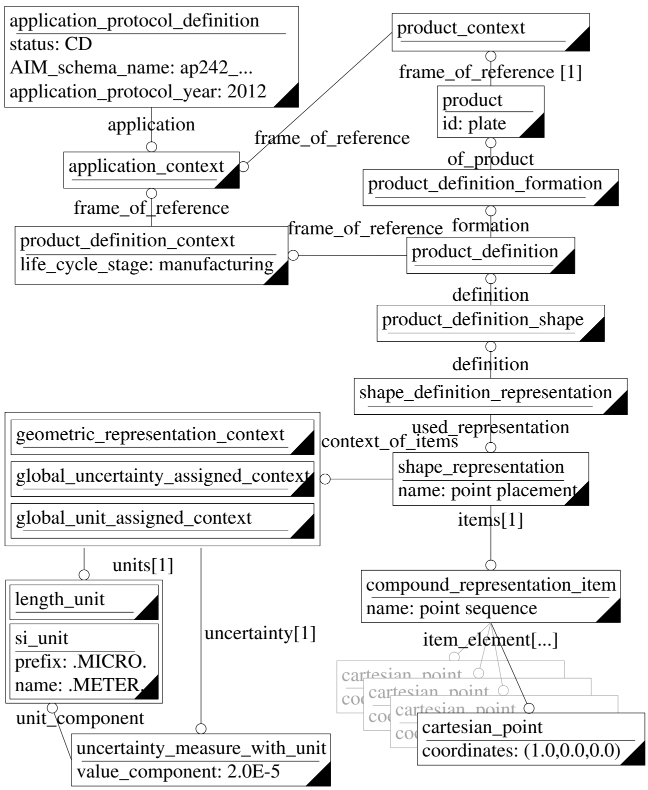

Figure 4.24 Sample data set of point cloud modeling in AP242 (Li et al., 2012b)

Particularly for the utilization of OCT, the focus is on the point cloud in the context of assembly models. The integration between the measured point cloud model and the designed shape model should be developed to support matching algorithm. In Figure 4.25, a product is modeled with the combination of the point cloud and the boundary shape representation. Two product definitions represent two aspects of the product: measurement and design. The product definition formation indicates both definitions are based on the same version of the product. A product definition relationship indicates that two aspects relate to each other.

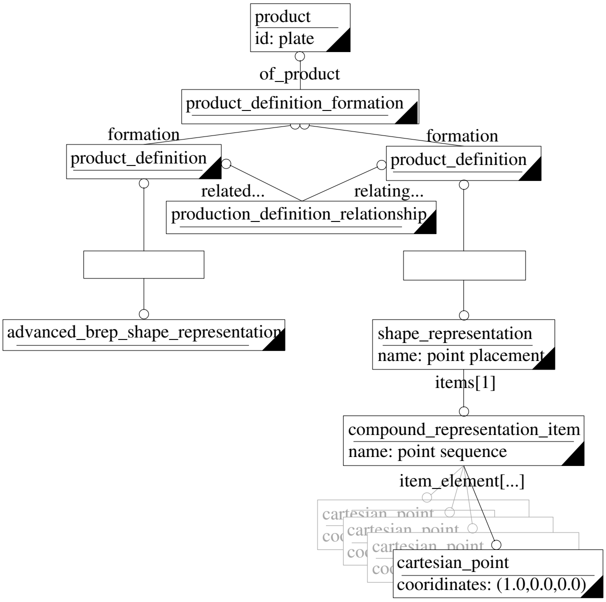

Figure 4.25 Integration of the measured model and the designed model (Li et al., 2012b)

As previous cases, the STEP Toolbox API helps the developers skip the learning process and simplifies the development procedure. A conceptual model (see Figure 4.26) is used to bridge the standard model and the object-oriented API. Still in a simplified form, the conceptual model is expressed as an ontology which partly reuses the models in previous cases. The point sets are defined as a part of component, which represent layers of the scanning result. The point sets can exist together with geometry for a component. As described in previous cases, geometric modeling of standards can be achieved by normal CAD software. This conceptual model is therefore only concerned with the description of the point cloud modeling and provides developers a simplification of EXPRESS models and ARM/AIM mappings.

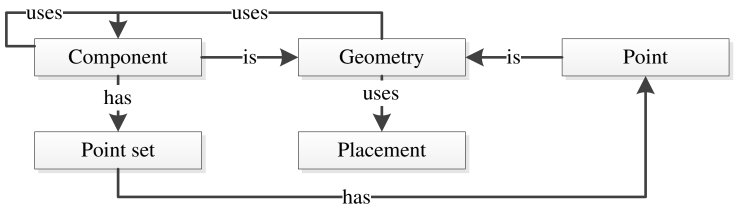

Figure 4.26 Conceptual model of point cloud

Similar to previous cases, the UML class diagram of the API is a fundamental reference for development. In Figure 4.27, a simplified class diagram for point cloud is illustrated, based on the mapping from the conceptual model. The terminology and relationships are kept same through the mapping. Components, point sets and points establish two composition relationships to describe points in groups or layers according to different measurement or data processing methods.

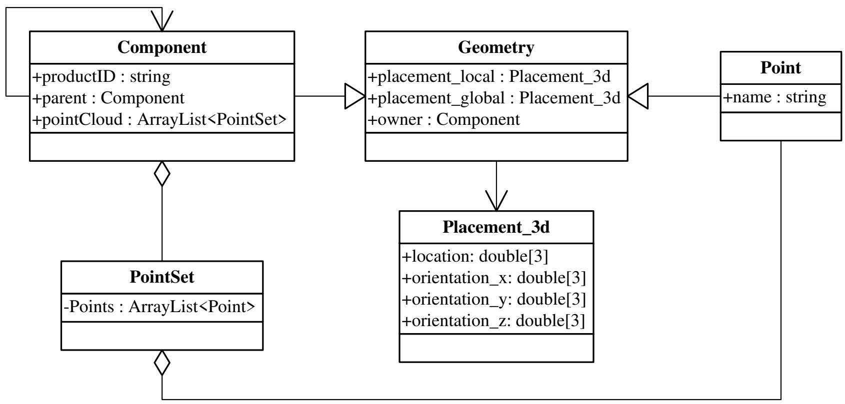

Figure 4.27 Simplified class diagram for point cloud

A prototype has been developed to utilize the point cloud processing algorithm and generate STEP AP242 data sets. The one-to-one translation implementation is similar to previous cases but simpler, so the repetitious details of the design and development are not given here.

The focus of this research is on the system integration principles for in-process monitoring with the potential technology in the future, OCT. The involvements with CAx systems in different possible scenarios are identified, which makes standardized information modeling and exchange solution significant in this promising domain. Especially with the high performance in speed and precision, the achieved system integration and modeling approach is able to support the automated measurement and defines criteria for implementation. In this highly interdisciplinary domain, developers are from either metrology industry or software industry. The support of STEP Toolbox is important for developers to perform implementation for interoperability and achieve system integration with limited knowledge on standards.

Chapter 5
Discussion

I find the great thing in this world is not so much where we stand,
as in what direction we are moving.
- Oliver Wendell Holmes

This licentiate is accomplished within the subject of production engineering, but it is also located in the cross area of information modeling, software engineering, and human computer interaction. This interdisciplinary quality has had the effect that previous researchers have not addressed the aspects of the practitioners of implementation. Thus the presented research has a great potential in different academic directions such as information modeling, software engineering and Human Computer Interaction (HCI).

Information modeling

- Exploration of modeling approaches in multiple industrial domains
- Formalization of conceptual model to bridge EXPRESS and API

In the area of information modeling, the issues of interoperability and data communication are important for most domains as the digitalization is increasing in the design and manufacturing industry. A comprehensive understanding of the target domain is a fundamental requirement to create a toolbox for developers. Hence, the future work will keep the focus on exploration of modeling approaches, especially for the industry with high involvement of CAx technology. For instance, an ongoing project has been

involved in the development of cutting tool catalogues to enable efficient communication between vendors and customers.
The conceptual model is used to bridge the gap between EXPRESS and the object-oriented API and to conduct communication between STEP Toolbox creators and users. Differing from other researches using UML in the standardization area, this research aims to provide the UML much simpler than the EXPRESS of the target data model schema. The Conceptual model is the first step of this simplification. At present it is presented as a simplified ontology model and lacks many elements of ontology – only terminology of classes and relationships are included. The formalization of this model with e.g. OWL (Web Ontology Language) is promising to enable automation between the ontology and the API, and also helping advanced users to understand the mapping more clearly.

Software engineering

- Organization of relevant communities to reach a practical product
- Clarification of roles and procedure to develop and maintain STEP Toolbox

In the area of software engineering, improvement can be made in every aspect of the toolbox as development architecture. This thesis uses several prototypes to validate the proposed architecture, but does not introduce the implementation details of the architecture itself. The analysis of application requirements and design principles are considered most important in this research stage.
Test cases for academia and industry have demonstrated prototype implementations in practice, but further development is necessary to turn the prototype to a mature product. This research has identified who the end users of STEP Toolbox are and how to use it. However, organizational questions are remaining: who should create and maintain the object-oriented API and how can this be performed. The benefit of the toolbox is significant for normal software developers, but the difficulty to develop the API as a formal product is unpredictable. This research develops and validates prototypes academically in the area of production engineering, but the real products should be developed and maintained by professional software engineers.

Human Computer Interaction (HCI)

- Observation and definition of the interaction patterns of developers
- Evaluation of STEP Toolbox with more external users

Information technology is supposed to assist humans to create, optimize, and maintain their prioritized work, which is also the ultimate goal of standardized data exchange solutions. However, the implementation of the standards often makes the practitioners deviate from their priority, with a learning process and development procedure. Current research on standardization mainly focuses on the functionality of information modeling approaches, i.e. the ability to meet users' needs for interoperability. For the implementation tasks, the issues of usability and user experience are rarely addressed in related research.

In Chapter 2, the characteristics of the implementation of information standards have been identified. Research methodologies of HCI and, more specific, EUD (End User Development) can be employed to observe, measure, describe, evaluate, and improve the interactions related with implementation work and the STEP Toolbox. In section 4.1, the work developed in this research has been tested by users from industry and positive results are achieved – the desired application is developed with STEP Toolbox and the designed system integration strategy. Still, more observations and measurements should be performed internally and externally to monitor how the toolbox is utilized and to make adjustment and improvement.

Chapter 6
Conclusion

Everything should be as simple as possible but no simpler.

- Albert Einstein

The information standards for industrial data management are potent in their comprehensive capability of product data modeling and exchange, but this capability is also an obstacle to CAx developers since it makes the documentation hard to understand. This licentiate thesis starts with a detailed description of the present situation of implementations of standards. The analysis of the contexts of use of implementation provides a unique vision of the individual efforts in this area, which leads to the development architecture, STEP Toolbox. The main objectives of this architecture are:

- Simplification of the traditional implementation based on SDAI
- Reduction of knowledge requirements in modeling based on standards

The objectives lead to the major features of STEP Toolbox as development architecture which can be divided into two aspects:

- Behavior
  - Simplified conceptual models to describe the modeling principles
  - UML class diagram to assist implementation
  - A high level object-oriented API convenient to programmers
  - Simplified development procedure and concepts

  - Support of EXPRESS-based data sets required in many standards
  - Support of different application protocols
  - Support of integration or standalone implementations
- Structure
  - Encapsulation of SDAI into simplified controlling managers
  - Modularization of the API to support different domains
  - A centralized model manager for general data set operation
  - Module managers to manipulate different types of information
  - Reusable classes among modules to form an integrated data model

The structure of the toolbox describes the way to encapsulate the standard data models, modeling principles, manipulation interfaces, and to divide it into modules. The behavior design provides an interface presented in a friendly way that the practitioners are able to easily accept and utilize.

Several in-house or collaborative developments were performed in four case studies to validate this solution in the areas of academia and industry. Over 2000 hours have been spent in developing the STEP Toolbox API and validation prototypes. Although modeling approaches and demonstrators are not focused in this licentiate, all case studies with publications are unique and leading practices in respective domains.

With the low knowledge requirements of common users on the information standards, development time span and work load are promising to reduce significantly. According to observation of normal developers so far, a 2-month learning period for STEP-related topics can be skipped by the adoption of STEP Toolbox. The implementation procedure is substantially changed and becomes more simplified and convenient for programmers.

References

Altintas, Y., Brecher, C., Weck, M., and Witt, S. (2005) Virtual Machine Tool, *CIRP Annals - Manufacturing Technology*, Vol. 54, No. 2, pp. 651-673

ASME B5/TC56 (2008) Part 2, Data Specification for Properties of Machine Tools for Milling and Turning, *Information Technology for Machine Tools (ASME B5.59-2)*, version 13b

Bay, J. and Rosché, P. (2010) Recommended Practices for Supplemental Geometry, Release 1.0, CAx Implementor Forum

Birla, S. and Kang, S. (1995) Software engineering of machine control systems: an approach to lifecycle economics, *Robotics and Automation 1995 Proceedings IEEE International Conference*, pp. 1086-1092

Bey, I., D. Ball, H. Bruhm, T. Clausen, W. Jakob, O. Knudsen, G. Schlechtendahl, and T. Sørensen (1994) *Neutral Interfaces in Design, Simulation, and Programming for Robotics*, 1st ed, Berlin: Springer-Verlag.

Bouma, E. and Tearney, G. (2002) *Handbook of Optical Coherence Tomography*, New York: Marcel Dekker

Bredeau, S. and Federzoni, L. (2009) Multilayer: a large scale production of micro devices via new rolled multi material layered 3D shaping technology, 4M/ICOMM 2009 Conference, Karlsruhe, p. 419-422

Burkett, W. and Yang, Y. (1995) The STEP Integration Information Architecture, *Engineering with Computers*, Vol. 11, No. 3, pp. 136 - 144

Burnett, M. and Scaffidi, C. (2013) End-User Development, *The Encyclopedia of Human-Computer Interaction*, 2nd Ed, Aarhus: The Interaction Design Foundation.

Chen, F., Brown, G., and Song, M. (2000) Overview of three-dimensional shape measurement using optical methods, *Optical Engineering*, Vol. 39, No. 1, pp. 10-22

Chryssolouris, G., Makris, S., Mourtzis, D., and Papakostas, N. (2008) Knowledge management in the virtual enterprise: web based systems for electronic manufacturing, *Methods and Tools for Effective Knowledge Life-Cycle-Management*, pp. 107-126, Berlin-Heidelberg: Springer-Verlag.

Chungoora, N., Cutting-Decelle, A., Young, R., Gunendran, G., Usman, Z., Harding, J., and Case, K. (2013) Towards the ontology-based consolidation of production-centric standards, *International Journal of Production Research*, Vol. 51, No. 2, pp. 327-345

Danner, W., Sandford, D., and Yang, Y. (1991) STEP Resource Integration: Semantic & Syntactic Rules (NISTIR 4528), NIST National Institute of Standards and Technology, ISO TC184 SC4 WG5 N10

Donmez, M., Blomquist, D., Hocken, R., Liu, C., and Barash, M. (1986) A general methodology for machine tool accuracy enhancement by error compensation, *Precision Engineering*, Vol. 8, No. 4, pp. 187-196

ESA, (2007) *Thermal control STEP-TAS SKM module*, European Space Agency, http://www.esa.int/TEC/Thermal_control/SEMUCON0LYE_0.html (Accessed 5 Oct. 2013)

Feeney, A. (2002) The STEP Modular Architecture, *Journal of Computing and Information Science in Engineering*, Vol. 2, No. 2, pp.132 - 135

Gielingh, W. (2008) An assessment of the current state of product data technologies, *Computer-Aided Design*, Vol. 40, No. 7, pp. 750-759

Goh, A., Hui, S., and Song, B. (1996) An integrated environment for product development using STEP/EXPRESS, *Computers in Industry*, Vol. 31, No. 3, pp. 305 - 313

Hardwick, M. and Loffredo, D. (2007) STEP-NC: Smart Data for Smart Machining, *Proceedings of the Intl. Conf. on Smart Machining Systems*, Gaithersburg

Hedlind, M. (2013) *Model driven process planning for machining*, Doctoral thesis, Dept. of Production Engineering, Royal Institute of Technology, Stockholm

Hedlind, M., Lundgren, M., Archenti, A., Kjellberg, T., and Nicolescu., M. (2010) Manufacturing Resource Modeling for Model Driven Operation Planning, *Proceeding of the CIRP 2nd International Conference on Process Machine Interactions*, Vancouver

Hedlind, M., Klein, L., Li, Y. and Kjellberg, T. (2011) Kinematic structure representation of products and manufacturing resources, *Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology*, Athens, pp. 340 - 347.

Helgoson M. and Kalhori, V. (2012) A Conceptual Model for Knowledge Integration in Process Planning, *Procedia CIRP*, Vol. 3, pp. 573-578

ISO 10303-1 (1994) Industrial Automation Systems and Integration -- Product data Representation and Exchange -- Part 1: Overview and Fundamental Principles, Geneva: International Standards Organization (ISO)

ISO 10303-105 (1996) Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 105: Integrated application resource: Kinematics, Geneva: International Standards Organization (ISO)

ISO 10303-105 ed2 (2013) Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 105: Integrated application resource: Kinematics, Geneva: International Standards Organization (ISO)

ISO 10303-11 (2004) Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 11: Description

Methods: The EXPRESS Language Reference Manual, Geneva: International Standards Organization (ISO)

ISO 10303-203 (2011) Industrial automation systems and integration -- Product data representation and exchange -- Part 203: Application protocol: Configuration controlled 3D design of mechanical parts and assemblies, Geneva: International Standards Organization (ISO)

ISO 10303-21 (2002) Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure, Geneva: International Standards Organization (ISO)

ISO 10303-214 (2010) Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 214: Application Protocol: Core Data for Automotive Mechanical Design Processes, Geneva: International Standards Organization (ISO)

ISO 10303-22 (1998) Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 22: Implementation Methods: Standard Data Access Interface, Geneva: International Standards Organization (ISO)

ISO 10303-27 (2000) Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 27: Implementation Methods: Java TM Programming Language Binding to the Standard Data Access Interface with Internet/Intranet Extensions. Geneva: International Standards Organization (ISO)

ISO 10303-28 (2007) Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 28: Implementation Methods: XML Representations of EXPRESS Schemas and Data, Using XML Schemas, Geneva: International Standards Organization (ISO)

ISO 10303-41 (2005) Industrial automation systems and integration -- Product data representation and exchange -- Part 41: Integrated generic resource: Fundamentals of product description and support, Geneva: International Standards Organization (ISO)

ISO 10303-43 (2011) Industrial automation systems and integration -- Product data representation and exchange -- Part 43: Integrated generic resource: Representation structures, Geneva: International Standards Organization (ISO)

ISO 13399-1 (2006) Cutting tool data representation and exchange -- Part 1: Overview, fundamental principles and general information model, Geneva: International Standards Organization (ISO)

ISO 230-1 (2012) Test code for machine tools -- Part 1: Geometric accuracy of machines operating under no-load or quasi-static conditions, Geneva: International Standards Organization (ISO)

ISO 230-2 (2006) Test code for machine tools -- Part 2: Determination of accuracy and repeatability of positioning numerically controlled axes, Geneva: International Standards Organization (ISO)

ISO 9241-110 (2006) Ergonomics of human-system interaction -- Part 110: Dialogue principles, Geneva: International Standards Organization (ISO)

ISO/DIS 10303-242 (2013) Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 242: Application protocol: Managed Model Based 3D Engineering, Geneva: International Standards Organization (ISO)

ISO/TS 13399-150 (2008) Cutting tool data representation and exchange -- Part 150: Usage guidelines, Geneva: International Standards Organization (ISO)

Jardim-Gonçalves, R., Olavo, R., and Steiger-Garção, A. (2005) The emerging ISO10303 Modular Architecture: In search of an agile platform for adoption by SMEs, *International Journal of IT Standards and Standardization Research*, Vol. 3, No. 2, pp. 82 - 95

Kaymakci, M., Kilic, Z, and Altintas, Y (2012) Unified Cutting Force Model for Turning, Boring, Drilling and Milling Operations, *International Journal of Machine Tools and Manufacture*, Vol. 54-55, pp. 34-45

Kjellberg, T., von Euler-Chelpin, A., Hedlind, M., Lundgren, M., Sivard, G., and Chen, D. (2009) The machine tool model—A core part of the digital factory, *CIRP Annals - Manufacturing Technology*, Vol. 58, No. 1, pp. 425-428

Klein, L. (2000) *JSDAI -- the Synergy of STEP and JavaTM Technology*, http://doc.jsdai.net/jsdai_doc/tutorial/tutorial.pdf (Accessed 5 Oct. 2013).

Klein, J. and Weiss, D. (2009) What is architecture?, *Beautiful architecture*, pp. 3-24, Sebastopol: O'Reilly

Ko, A., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaf-fidi, C., Lawrence, J., Lieberman, H., Myers, B., Rosson, M., Rothermel, G., Shaw, M., and Wiedenbeck, S. (2011) The state of the art in end-user software engineering, *Journal ACM Computing Surveys*, Vol. 43, No. 3, pp. 21:1-21:44

Kramer, T. and Xu, X. (2009) STEP in a Nutshell, *Advanced Design and Manufacturing Based on STEP*, pp. 1-22, London: Springer

Lee, T. (1999) An overview of information modeling for manufacturing systems integration (NISTIR 6382), Gaithersburg: Manufacturing system integration division, National Institute of Standards and Technology

Li, Y., Hedlind, M., and Kjellberg, T. (2011) Implementation of kinematic mechanism data exchange based on STEP, P*roceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology*, pp. 152-159, Athens

Li, Y., Hedlind, M., and Kjellberg, T. (2012a) Kinematic error modeling based on STEP AP242, *Proceedings of the 1st CIRP Sponsored Conference on Virtual Machining Process Technology*, Montreal

Li, Y., Su, R., Hedlind, M., Ekberg, P., Kjellberg, T., and Mattsson, L. (2012b) Model based in-process monitoring with optical coherence tomography, *Procedia CIRP*, Vol. 2, pp. 70-73

Li, Y., Chen, D., Kjellberg, T., and Sivard, G. (2013a) User friendly development architecture for standardised modelling: STEP Toolbox, submitted to *International Journal of Manufacturing Research*

Li, Y., Hedlind, M., Kjellberg, T., and Sivard, G. (2013b) Cutting tool data representation and implementation based on STEP AP242, *Smart Production Engineering*, pp. 483-492, Berlin/Heidelberg: Springer-Verlag

LKSoft (2004) Integrating Distributed Applications on the Basis of STEP Data Models, Final Report, IDA-STEP, http://www.ida-step.net/sites/ida-step.net/files/IST-2000-30082-Final-Report-v1.pdf (Accessed 28 Jan. 2013)

López-Ortega, O. and Ramírez-Hernández, M. (2007) A formal framework to integrate express data models in an extended enterprise context, *Journal of Intelligent Manufacturing*, Vol. 18, No. 3, pp. 371-381

Lui, M., Gray, M., Chan, A., and Long, J. (2011) Enterprise application integration fundamentals, *Pro Spring Integration*, pp. 1-14, New York: Apress

Ma, H., Ha, K., Chung, C., Amor, R. (2006) Testing semantic interoperability, *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, pp. 1216-1225, Montreal

Maier, F. and Stumptner, M. (2007) Enhancements and Ontological Use of ISO-10303 (STEP) to Support the Exchange of Parameterised Product Data Models, *Proceedings of the 7th International Conference on Intelligent Systems Design and Applications*, pp. 433 - 440, Rio de Janeiro

Nassehi, A. and Vichare, P. (2009) A STEP-NC Compliant Methodology for Modelling Manufacturing Resources, *Advanced Design and Manufacturing Based on STEP*, pp. 261-281, London: Springer

Rachuri, S., Han, Y., Foufou, S., Feng, S., Roy, U., Wang, F., Sriram, R. and Lyons, K. (2006) A Model for Capturing Product Assembly Information, *Journal of Computing and Information Science in Engineering*, Vol. 6, No. 1, pp. 11-21

Rodger, G. (2001) Dimensional measurement using vision systems, National Physical Laboratory, United Kingdom

Sauder, D. and Morris, K. (1995) Design of a C++ Software Library for Implementing EXPRESS: The NIST STEP Class Library, *Proceedings of EUG '95 — The Fifth EXPRESS Users Group Conference*, Grenoble

Schwenke, H., Knapp, W., Haitjema, H., Weckenmann, A., Schmitt, R., and Delbressine, F. (2008) Geometric error measurement and compensation of machines – An update, *CIRP Annals - Manufacturing Technology*, Vol. 57, No. 2, pp. 660-675

SCRA (2006) *STEP application handbook, ISO 10303, version 3*, http://www.uspro.org/documents/STEP_application_hdbk_63006_BF.pdf (Accessed 31 Mar. 2013)

Sharma, R. and Gao, J. (2006) STEP PLCS for design and in-service product data management, *Advances in Design*, pp. 293-301, London: Springer

Schenck, D. and Wilson, P. (1994) *Information Modeling the EXPRESS Way*, New York: Oxford University Press

Su, R., Kirillin, M., Ekberg, P. Roos, A., Sergeeva, E., and Mattsson, L. (2012) Optical coherence tomography for quality assessment of embedded microchannels in alumina ceramic, *Optics Express*, Vol. 20, No. 4, pp. 4603-4618.

Sudarsan, R., Fenves, S., Sriram, R., and Wang, F. (2005) A product information modeling framework for product lifecycle management, *Computer-Aided Design*, Vol. 37, No. 13, pp. 1399-1411

Tanaka, F., Onosato, M., Kishinami, T., Akama, K., Yamada, M., Kondo, T. and Mistui, S. (2008) Modeling and implementation of Digital Semantic Machining Models for 5-axis machining application, *Manufacturing Systems and Technologies for the New Frontier*, pp. 177-182

Tang, P., Huber, D., Akinci, B., Lipman, R., and Lytle, A. (2010) Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques, *Automation in Construction*, Vol. 19, No. 7, pp. 829-843

Tassey, G. (1999) Interoperability cost analysis of the U.S. automotive supply chain—Final report (RTI project number 7007-03) Research Triangle Institute, https://www.rti.org/pubs/US_Automotive.pdf (Accessed 06 Oct. 2013)

Veeramani, D., Upton, D., and Barash, M. (1992) Cutting-Tool Management in Computer-Integrated Manufacturing, *International Journal of Flexible Manufacturing Systems*, Vol. 4, No. 3/4, pp. 237-265

von Euler-Chelpin, A. (2008) *Information modelling for the manufacturing system life cycle*, Doctoral thesis, Dept. of Production Engineering, Royal Institute of Technology, Stockholm

Zelinski, P. (2001) Coming Soon: Universal, CAM-Independent Cutting Tool Library, *Modern Machine Shop*. Vol. 84, No. 4, 22-24

Zhang, Y., Zhang, C., and Wang, H. (2000) An Internet based STEP data exchange framework for virtual enterprises, *Computers in Industry*, Vol. 41, No. 1, pp. 51-63

Paper A

Li, Y., Hedlind, M., and Kjellberg, T. (2011) Implementation of kinematic mechanism data exchange based on STEP, *Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology*, pp. 152-159, Athens

Proceedings of DET2011
7th International Conference on Digital Enterprise Technology
Athens, Greece
28-30 September 2011

IMPLEMENTATION OF KINEMATIC MECHANISM DATA EXCHANGE BASED ON STEP

Yujiang Li
KTH Royal Institute of Technology,
Production Engineering
yujiang@iip.kth.se

Mikael Hedlind
KTH Royal Institute of Technology,
Production Engineering
mikael.hedlind@iip.kth.se

Torsten Kjellberg
KTH Royal Institute of Technology,
Production Engineering
torsten.kjellberg@iip.kth.se

ABSTRACT

In this paper, the first known valid implementation of kinematic mechanism based on STEP (ISO 10303, STandard for the Exchange of Product data) is presented. The result includes a general conceptual framework and two developed prototype applications. The framework is designed for integration of the STEP-based kinematic mechanism modeling with existing commercial CAx systems. The two applications are implemented for kinematic data exchanges between Siemens NX and STEP-NC Machine via STEP AP214 (ISO 10303-214) files. Experiences of design and development of the applications are introduced in this paper, and a valid example of data exchange using the developed applications is shown. As the first valid STEP implementation on kinematics, this research demonstrates the feasibility of STEP-based data exchange for kinematic mechanism. The developed results can motivate a wider application of the STEP standard in industry throughout design and manufacturing.

KEYWORDS

Kinematics, CADCAM, standard, modelling

1. INTRODUCTION

Numerous commercial CAx software systems have been developed and applied in fields for product design, prototype testing, life cycle management, etc. Meanwhile, diverse partnerships between IT software vendors, industrial practitioners, and academic researchers have been built to enhance competitiveness of supply chains. Therefore, the demand for a system neutral solution of product data exchange exists in different perspectives: geometry, kinematics, tolerances, classification and so on. Kinematic mechanism is one of the most important aspects in the field of industrial product data exchange and sharing. The basic conceptual technique to represent kinematic mechanism in CAD is common among majority applications: links and joints are combined to describe topology and geometry. Different types of motion constraints have been defined in similar ways, e.g. revolution, translation, cylinder, etc. However, applications for data exchange, based on a standard, of such mechanism between miscellaneous CAx systems are very rare. At the same time, the need arises very often for the kinematic mechanism data exchange between information systems along the product development and its lifecycle. As Nassehi et al (2009) states in the field of manufacturing resources modeling, kinematic representation is very important in process planning, tool-path generation, multi-route process planning, intelligent CNC controllers, and bi-directional data flow.

The STEP standard addresses a solution for this problem with a particular integrated resource model schema, p105 (ISO 10303-105). Using p105 within the widely-used STEP application protocol AP214 (ISO 10303-214) offers a standardized information model schema for integration of the kinematics with geometry and assembly models. Several research projects have tried to implement the kinematic model of AP214 as subtasks. But until recently neither a valid data set, nor a valid implementation for STEP-based data exchange of kinematic mechanism had been created.

This paper will present the first known valid STEP-based implementation for kinematic mechanism. The result will demonstrate using STEP models to facilitate the integrated data exchange with kinematics, geometry, and assembly.

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 store and access (Kramer and Xu, 2009). In modern industries, it is common that companies and organizations in different locations using different software systems have to work together due to a 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 which is how to seamlessly exchange and share data with different members in a collaborative development environment. Besides, designers should exchange not only the product geometry data, but also information about processes and resources. STEP, as a system neutral standard for product data exchanging, is introduced to solve this problem. It is the comprehensive structure of the STEP standard and imperative needs for system neutral product data format that make almost all major CAx software vendors support it more or less, especially in representation of 3D geometry.

However, supports for exchange of kinematic mechanism data are hardly implemented for any applications in industries. Today, most manufacturing companies have to use slides, fax, telephone, or paper-based documents to describe kinematic motion. Skilled CAx operators' manual re-input usually is the only option to bridge the gap between different systems. This research will use STEP AP214 files as the ISO standardized system-neutral data container to provide an alternative solution for the exchange of kinematic mechanism information.

2. LITERATURE REVIEW

STEP p105, is a member of the integrated application resources of STEP standard. It is an information model providing support for kinematic information exchange and sharing for computer-aided design and kinematic analysis systems. This part of STEP standard was published in 1996 and with two technical corrigenda published in 2000. At present, the second edition is under development and the first draft of its usage within an AP is planned in year 2011 by ISO.

The major features presented in p105 are 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 kinematic structure topology aspect. These concepts are applied during the development in this research.

So far, there is no known implementation for p105 valid modeling. Almost all found literatures document its usages only 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 of STEP standard. Therefore the history of STEP-based kinematic modeling is not so extensive, as shown in Figure-1.

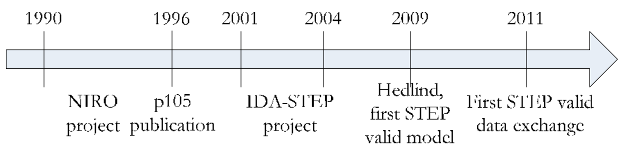

Figure 1 – Milestone of kinematic modeling using STEP

At start of the European research project NIRO (Neutral Interfaces for Robotics) the project partners had developed a proposal for kinematics in STEP. This proposal was accepted by ISO as basis for further STEP integration (Bey et al, 1994).

A research about machine control software in the context of industrial economy is an early attempt to involve the kinematic mechanism based on p105 (Birla and Kang, 1995). The STEP-based kinematic model is defined for the members of machining processes, in terms of fixtures, workpieces, and tools. But the result does not really conform to STEP standard.

An expandable conceptual model for assembly information has been proposed a project held by National Institute of Standard and Technology (NIST) in USA, named Open Assembly Model (OAM) (Sudarsan et al., 2005, Rachuri et al., 2006). This model focuses on representations of geometry, kinematics, and tolerances, and it is claimed to use STEP as the underlying data structure. But it only adopts the concepts defined in p105, rather than the actual schema defined in EXPRESS language.

A small subset of p105 has been used in a semantic-based machine tool modeling approach for 5-axis machining application by (Tanaka et al., 2008). The research extends the subset in its machine tool kinematic model.

An important attempt to implement kinematics of the STEP standard is the IDA-STEP project (Integrating Distributed Applications on the Basis of

STEP Data Models) (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-based file can be outputted and viewed in a web browser. According to their report, it is the first attempt to implement based on p105 since it was published, and they are close to achieve an implementation able to model standard conforming data.

Another STEP related kinematic implementation that need mentioning 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 AP214 ARM (Application Reference Model) to describe the motion constraints of rigid bodies, while AP214 specifies an AIM (Application Interpreted Model) schema for implementation.

3. RESEARCH APPROACH

Two integrations tasks are in focus during the development: integrating a kinematic mechanism with existing geometric model in a STEP AP214 file, and integrating kinematic data converter with existing commercial CAx systems.

At first, this research has to face challenges of exploring a way to merge a kinematic mechanism representation to an existing STEP AP214 file that includes a 3D geometry model. As input to this part of the research, an AP214 valid kinematic model developed by Hedlind et al (Hedlind et al., 2010) was used.

And then, the research focuses on the integration with existing commercial CAx system. The STEP model, regarding the representations of geometric aspects, has been supported in lots of CAD systems, such as Siemens NX, Autodesk CAD, Pro/Engineer, and CATIA. Different channels are used in these systems, such as independent translators, command line, 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.

In order to demonstrate using STEP files to bridge kinematic mechanism between different systems, Siemens NX and STEP-NC Machine are selected. The reason why Siemens NX is chosen is that it provides a relatively open programming interface, NX Open, which enables the accesses to most of its functions including kinematic motion simulation. The choice of the STEP-NC Machine application was for its ability to simulate tool-paths described in AP238 (ISO 10303-238) together with an AP214 machine tool geometry model. The machine tool kinematics is natively defined on a XML format provided by the application provider STEP Tools Inc.

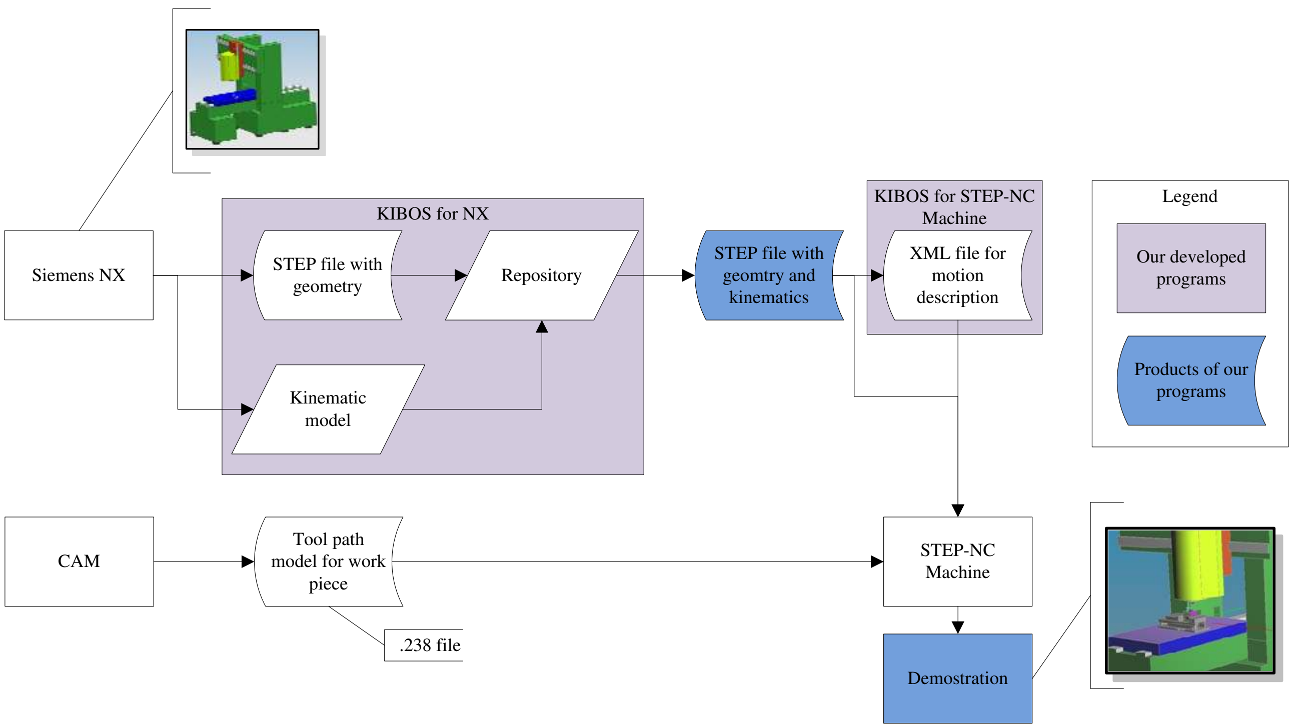

Figure 2 - Operation design

In this research, it is explored how AP214 can be utilized to exchange a kinematic mechanism to support machine tool motion analysis and operation planning.

The essential operations can be illustrated with the work flow diagram shown in Figure-2. The software developed in this research is named KIBOS (KTH Implementation Based On STEP). In this work flow, KIBOS for NX is executed within the session of Siemens NX 7.5. It acquires a STEP file, without kinematics, via Siemens NX native STEP export function. Meanwhile, it also collects kinematic information from the native model. Then, KIBOS for NX merge the kinematic information with the original STEP file in a repository in memory and at last exports a new STEP AP214 file with both geometry and kinematics.

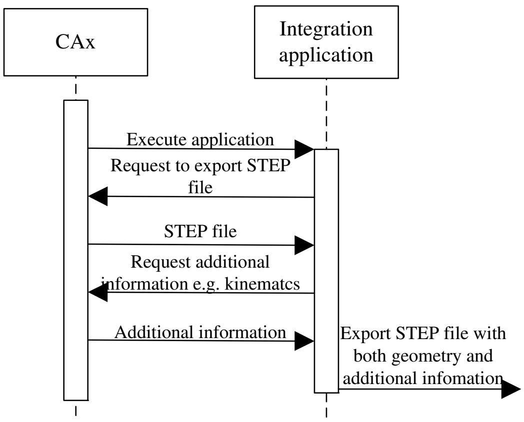

Figure 3 - An application integrated with CAx

This approach applies following conceptual framework illustrated in Figure-3. Using existing CAx software with limited STEP export functionality together with an external integration application to get the requested additional information and merge it with the first exported STEP file, new possibilities for STEP based data exchange is enabled with less effort compared to a completely new developed STEP exporter. This integration procedure requires used CAx software to have an accessible API or specified data format to get requested information (and ability to export the first STEP file to be extended). These requirements are fulfilled by several CAx software. Therefore can this procedure be applied in different contexts when implementing exporter from CAx to STEP.

KIBOS for STEP-NC Machine is an independent application to read the STEP file with geometry and kinematics, and to output an XML file with the kinematic information. The XML file conforms to the format used by STEP-NC Machine. The XML file together with the STEP file of a certain machine can be used to simulate motion of a machining operation in STEP-NC Machine.

In this research, KIBOS for NX is designed to be integrated with Siemens NX. It is able to access functions and to be executed within NX. The aim of this high integration is to make the user's operations as simple as possible. This solution can be easily extended to other CAx systems with similar functionality, i.e. a built-in STEP exporter and an accessible API (Application Program Interface).

4. SYSTEM DESIGN

This research includes developments of two applications, KIBOS for NX and KIBOS for STEP-NC Machine, and in this section they will be described in detail separately. Both applications are similar in technical backgrounds and conceptual design.

KIBOS is a STEP implementation based on AP214 AIM schema, of which the geometric model has been implemented in many kinds of commercial CAx software. KIBOS is developed with Java language, because both Siemens NX and STEP standard have useful programming interfaces for Java implementation. The API for Siemens NX, named NX Open for Java, enables access to all functions for data collecting and exporting required in this work. KIBOS for NX uses NX Open to retrieve information from NX native model and execute STEP translation to produce the STEP file with only geometry and assembly.

ISO 10303-22 SDAI (Standard Data Access Interface) is a common programming interface to access ISO 10303-11 EXPRESS based data, the model schema language of STEP. ISO 10303-27 specifies a Java binding to SDAI and is implemented in JSDAI, an open source development package from LKSoftWare GmbH. Developed applications in this work are done in the JSDAI environment.

In the system design, the integration of three layers is focused on: physical data, resources, and application. From the view of software engineering, KIBOS is simple software with a simple function: data format translation. But it needs to be seamlessly integrated with other resources, e.g. NX Open, STEP, and JSDAI. Hence, these resources are connected in one layer to link the application with physical data. Physical data is the data physically stored in the hard disk, mostly used as input and output. The application layer contains the developed program with data manipulation logic.

4.1 KIBOS FOR NX

The system architecture of KIBOS for NX is illustrated in Figure-4. At first, in development of the application it is needed to compile the AP214 AIM schema with JSDAI so that the schema can be utilized as an external library in Java for early binding to the schema. This library is similar to other external library. It needs compiled only once before development and will be included within the developed application.

KIBOS for NX requires a native NX model with assembly, geometry and kinematics. In order to make the output STEP file able to be used in KIBOS for STEP-NC Machine, the user also needs to use label the faces where the cutting tool and the workpiece should be placed. In this implementation, the PMI (Product and Manufacturing Information) note functionality in NX is used to label the surfaces.

KIBOS for NX generally includes two parts, shown in the application layer: NX model processor and STEP file updater. The former is implemented with NX Open API and used to process the native NX model. It is able to automatically invoke the native STEP translator of Siemens NX to export the STEP AP214 file that includes only assembly and geometry data. The NX model processor also needs to generate a data buffer to store the information that is not included in the exported STEP file: kinematics and PMIs. The STEP file updater is able to parse the data buffer and to read the original STEP file. Further, it translates the kinematic data and PMI data in the data buffer to the kinematic instances and general feature instances in the AP214 file. Besides, it is able to integrate kinematics and PMIs with geometry and assembly instances in the STEP model, and to export a STEP file with all information.

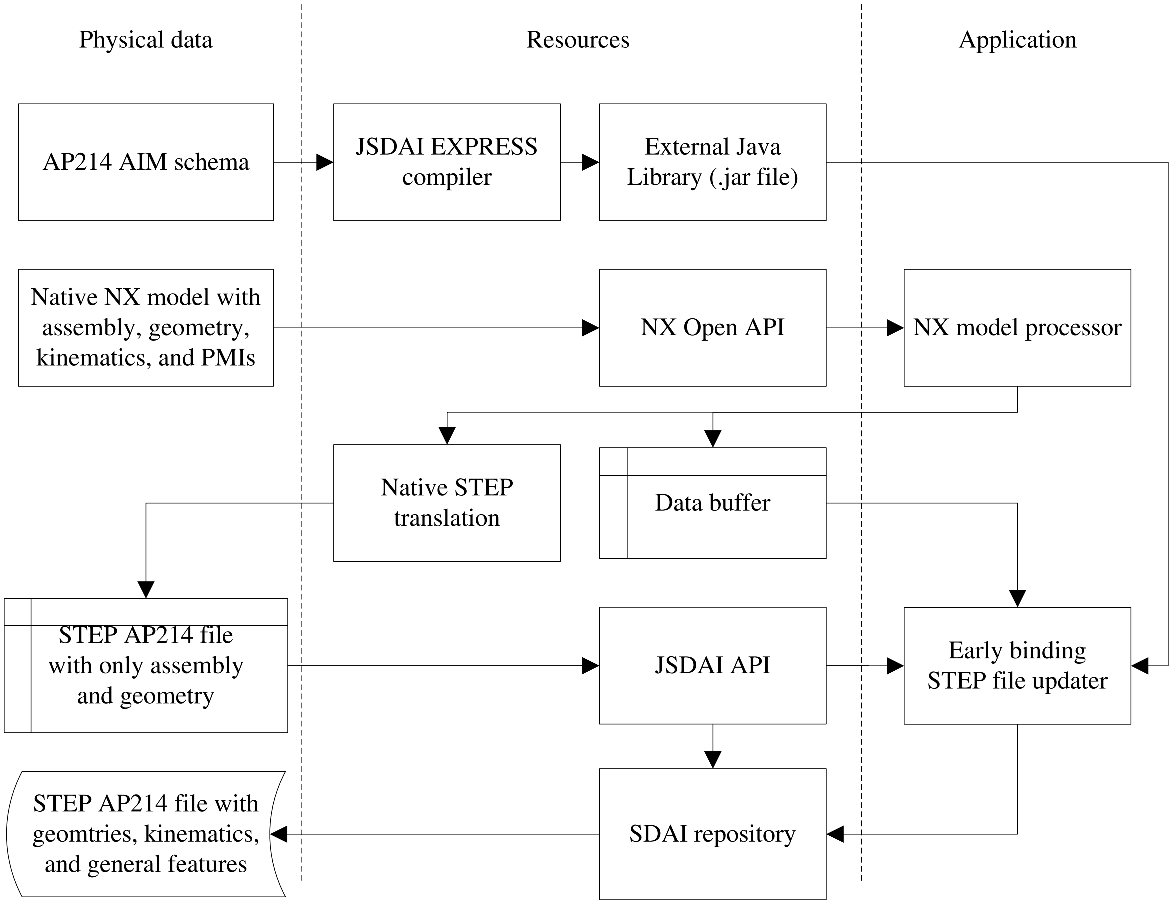

Figure 4 - System architecture of KIBOS for NX

4.2 KIBOS FOR STEP-NC MACHINE

The system architecture of KIBOS for STEP-NC Machine (see Figure-5) is similar to KIBOS for NX. The AP214 AIM schema is also required to read and parse the STEP file by the developed STEP reader. Then, the XML creator creates the XML file according to the data of kinematic mechanism and general features. The XML file is created in a format defined and recognized by STEP-NC Machine. It stores kinematic information for a certain machine, e.g. kinematic chain definitions, axis definitions, axis placements, and motion ranges. It also includes placement data for cutting tool and fixture. Both the STEP file and the XML file of a certain machine should be placed in the "machine" folder of the program folder of STEP-NC Machine.

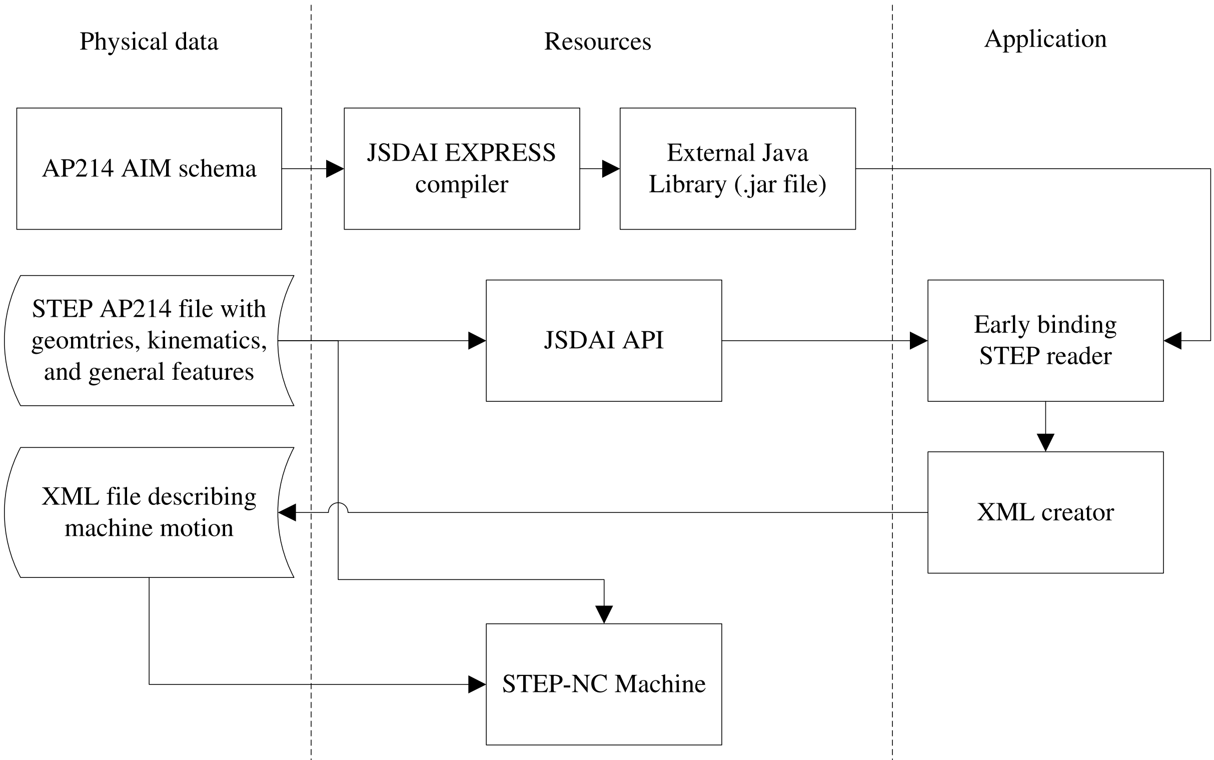

Figure 5 - System architecture of KIBOS for STEP-NC Machine

5. IMPLEMENTATION

Both KIBOS for NX and KIBOS for STEP-NC Machine are typical STEP implementations developed in the same development environment. Used JSDAI and compiled AP214 AIM schema provides full support for all the operations defined in SDAI and the data model of AP214 AIM. The JSDAI is based on Eclipse, an open source IDE (Integrated Development Environment). For this implementation Eclipse 3.5 is used, which is known for its high extensibility for Java programming. The graphical user interface of KIBOS is developed based on SWT (Standard Widget Toolkit). SWT is an open source library for platform-neutral interface implementation.

KIBOS for NX is developed with the focus on highly integration with existing design environment of Siemens NX. The application can be used to export needed STEP model in the same way as other built-in exporters, as shown in Figure-6. The NX Open MenuScript binding Java is applied to customize the menu button. The implementation relies on NX Open for Java API to interact with the NX session.

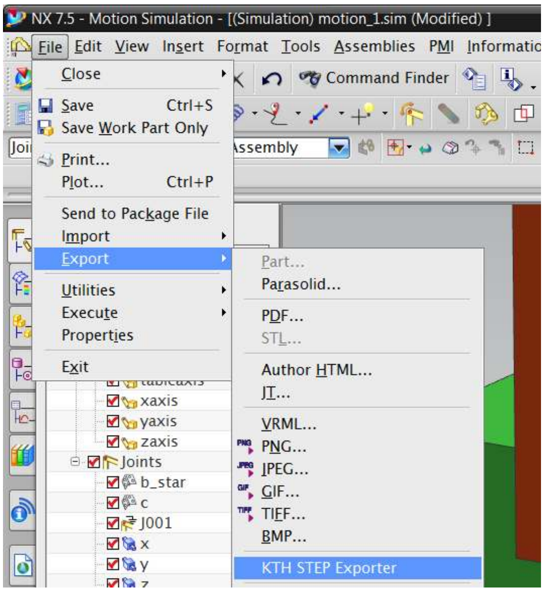

Figure 6 - The integrated menu button

The simple user interface of KIBOS for NX is shown in Figure-7. It is designed similar to other common data format translators. The input model is the current working model in NX, and the file path and name of the output STEP model can be easily defined in the textboxes of the user interface, or selected from the file dialog by clicking the “Browse…” button.

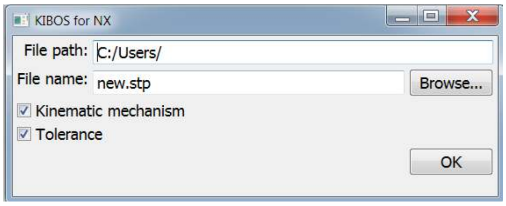

Figure 7 - Interface of KIBOS for NX

The KIBOS for STEP-NC Machine is an independent application. Its interface is shown in Figure-8. The user can set the file target location, the input file, and the output XML file name. Besides, the user can configure the actual orientation of the axis (Z axis) and the reference direction (X axis) of the machine in case they are not set to the default orientation. The user can also specify the kinematic motion solver algorithm by selecting from list of predefined variants defined by STEP-NC Machine. Note that, sometimes, the user maybe would not like to repeat inputting same values in the interface. Therefore, a configuration file is provided to pre-define the default values of all these configurations.

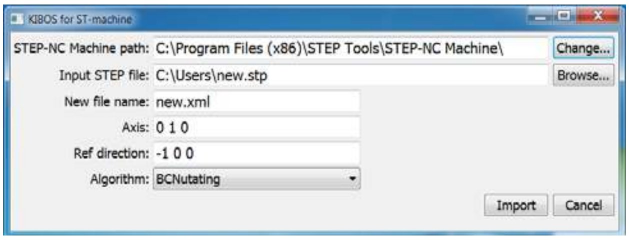

Figure 8 - Interface of KIBOS for STEP-NC Machine

6. CASE STUDY

This case study demonstrates the system neutral solution for kinematic mechanism data exchange and validates the capability of the developed applications.

A CAD model of a 5-axis machine tool is used in this sample, as shown in Figure-9. Although it is a simplified model, it still has full capability to demonstrate the motion of its 5 axes. This sample is used to perform the following tasks:

1. Creating a kinematic model in NX,
2. Exporting a STEP file with geometry and kinematics by KIBOS for NX,
3. Importing the information within the STEP file by KIBOS for STEP-NC Machine,
4. Simulate a machining operation with the machine tool model in STEP-NC Machine.

The first task here is to create a complete kinematic model of this machine with special configuration such as axis definition and motion range. PMI notes are used to label the faces where the tool and the workpiece should be placed. In the component-based motion simulation module, six components or subassemblies are selected as the links to form the five kinematic joints to represent the five axes.

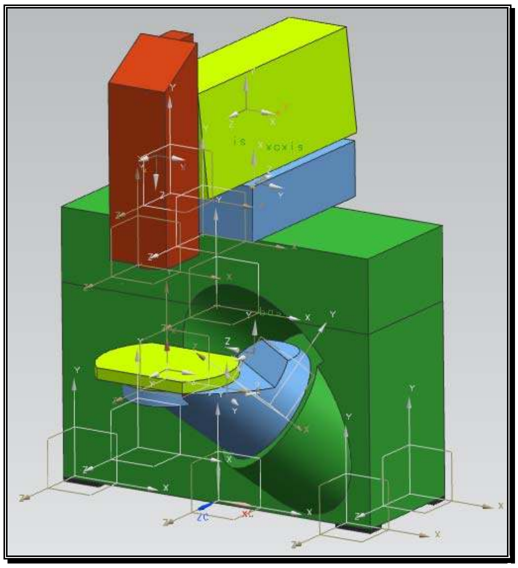

Figure 9 - CAD model of DMG machine in NX

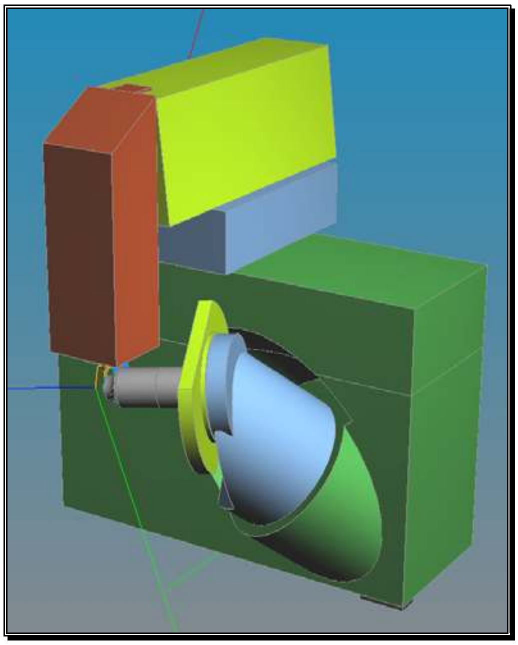

Figure 10 - Demonstration in STEP-NC Machine

Then, using KIBOS for NX and KIBOS for STEP-NC Machine, the STEP file and the XML-file for the machine tool can be produced and imported to STEP-NC Machine.

An AP238 file for machining of an impeller is used for the motion simulation. This AP238 file was downloaded from in the sample data set of the official website of STEP-NC Machine (STEP Tools Inc., 2011). The machining operation described in this file includes fixture definition, tool paths, cutting tools, and operation sequence. During the motion simulation, the five-axis machining can be simulated and displayed in STEP-NC Machine, as shown in Figure-10.

7. CONCLUSIONS

This paper focuses on the strategy of STEP-based integration for kinematic mechanism exchange with existing commercial CAx software systems. The solution is presented with a general framework for system neutral integration and two applications are developed to implement this framework and demonstrate its industrial significance. The major features of this solution include:

- Seamless linkage with existing CAx systems,
- Full integration with exported data on STEP using CAx native export functionality,
- Standardized development environment,

As the first STEP valid implementation for kinematic modeling, KIBOS validates and demonstrates the capability of STEP AP214 to represent and exchange kinematic mechanisms.

The result of this research provides the industrial practitioners with an implementable framework for kinematic modeling exchange between different CAx systems, the IT vendors 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.

In addition to machine tool motion simulation, this result will also improve other engineering domains, e.g. verification of product design, fixture design or factory design with standardised data exchange.

8. ACKNOWLEDGMENTS

We are grateful for the support from Scania, Sandvik Coromant, Volvo, VINNOVA, Siemens PLM Software and for fruitful discussions with members of ISO TC184 SC4 WG3 T24.

REFERENCES

Bey, I., Ball, D., Bruhm, H., Clausen, T., Jakob, W., Knudsen, O., Schlechtendahl, E. G., and Sørensen, T., “Neutral Interfaces in Design, Simulation, and Programming for Robotics”, 1st Edition, Springer-Verlag, Berlin, 1994, p 334, ISBN 3-540-57531-6

Birla, S. and Kang, S., "Software engineering of machine control systems: an approach to lifecycle economics", *Robotics and Automation 1995 Proceedings IEEE International Conference*, 1995, pp 1086-1092, ISBN 0-7803-1965-6

European Space Agency, "thermal control STEP-TAS SKM module", ESA, 2007, Retrieved: 08 01 2011, <http://www.esa.int/TEC/Thermal_control/SEMUCON0LYE_0.html>

Hedlind, M., Lundgren, M., Archenti, A., Kjellberg, T. and Nicolescu, C. M., "Manufacturing resource modeling for model driven operation planning", *CIRP 2nd International Conference on Process Machine Interactions*, 2010, ISBN 978-0-9866331-0-2

Kramer, T. and Xu, X., "STEP in a Nutshell", *Advanced Design and Manufacturing Based on STEP*, 1st Edition, Springer, London, 2009, pp 1-22, DOI 10.1007/978-1-84882-739-4_1

LKSoft, "Integrating Distributed Applications on the Basis of STEP Data Models, Final Report", 1st Edition, LKSoftWare GmbH, Germany, 2004, p 40

Nassehi, A. and Vichare, P., "A STEP-NC Compliant Methodology for Modelling Manufacturing Resources", *Advanced Design and Manufacturing Based on STEP*, 1st Edition, Springer, London, 2009, pp 261-281, DOI 10.1007/978-1-84882-739-4_12

Rachuri, S., Han, Y.-H., Foufou, S., Feng, S. C., Roy, U., Wang, F., Sriram, R. D. and Lyons, K. W., "A Model for Capturing Product Assembly Information", *Journal of Computing and Information Science in Engineering*, vol. 6, No. 1, 2006, pp 11-21, DOI 10.1115/1.2164451

STEP Tools Inc., "STEP-NC Sample Data: Impeller Part", STEP Tools Inc., 2011, Retrieved: 27 05 2011, <http://www.steptools.com/products/stepncmachine/samples/impeller/>

Sudarsan, R., Fenves, S. J., Sriram, R. D. and Wang, F., "A product information modeling framework for product lifecycle management", *Computer-Aided Design*, vol. 37, No. 13, 2005, pp 1399-1411, DOI 10.1016/j.cad.2005.02.010

Tanaka, F., Onosato, M., Kishinami, T., Akama, K., Yamada, M., Kondo, T. and Mistui, S., "Modeling and implementation of Digital Semantic Machining Models for 5-axis machining application", *Manufacturing Systems and Technologies for the New Frontier*, 2008, pp 177-182, DOI 10.1007/978-1-84800-267-8_36

Paper B

Li, Y., Hedlind, M., and Kjellberg, T. (2012a) Kinematic error modeling based on STEP AP242, *Proceedings of the 1st CIRP Sponsored Conference on Virtual Machining Process Technology*, Montreal

Kinematic error modelling based on STEP AP242

Yujiang Li, Mikael Hedlind, Torsten Kjellberg
KTH Royal Institute of Technology, Production Engineering,
Brinellvägen 68, SE-10044 Stockholm, Sweden
yujiang.li@iip.kth.se

Abstract: Kinematic error is one of the major sources affecting geometric accuracy of machine tools motion. Component error and location error are common concepts to characterise kinematic error for both linear and rotation axes. With the second edition of STEP p105 currently under development, kinematic error can be associated directly to corresponding kinematic pairs. Based on the conceptual ontology model, a practical approach is introduced for standardized kinematic error modelling and implementation. Through integration with geometry, assembly, kinematics and classification, a developed application demonstrates and evaluates data representation and exchange for kinematic error. As STEP AP242 is a generic model for any product or manufacturing resource, additional modeling constraints are applied to achieve unambiguous data representation.
Keywords: kinematic error, modelling, data exchange, standardisation

1 INTRODUCTION

This paper presents a kinematic error modelling approach based on standards for data representation and exchange. Geometric accuracy of machine tool motion plays a major role in machining capability. The geometric error is significantly influenced by kinematic error [Schwenke et al., 2008], in terms of component error and location error. To achieve high accuracy in machining, a calculation methodology (multiplication of homogeneous transform matrices) for kinematic error compensation [Donmez et al., 1986] is commonly used. In application of virtual machine tool [Altintas et al., 2005] the representation of kinematic error is a key factor for realistic simulation. Historically, development of kinematic error measurement and analysis has been done with focus on machine tools, but based on general principles applicable for any product mechanism. This enables reuse of data schemas and computation methods for different applications. Although applications for kinematic error analysis are not as common as for nominal kinematic analysis at present, it can be expected that more and more commercial CAD/CAM/CNC systems will support it for more realistic virtual machining (see Figure 1). As what happened in the product design area, multiple systems with different functionalities may be used, often also in different organizations. Therefore, a standard for data representation and exchange of kinematic error will become as necessary as shape geometry to enable efficient information communication and sharing.

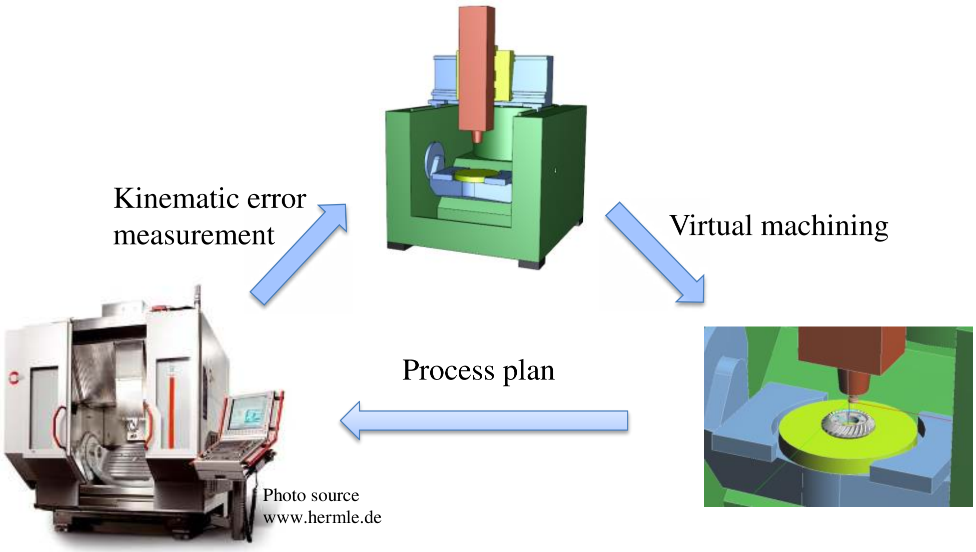

Figure 1; Machine tool modelling for realistic virtual machining

To support different utilisations in different domains, the standard data model for product and manufacturing resource should provide functionalities in a broad scope through the manufacturing industry [Euler-Chelpin, 2008], e.g. the kinematic error model should support measurement, analysis, prediction, and compensation. Therefore, the model should keep sharable concepts that can be interpretable in any domain, and it should be ready for fully data integration with other types of properties defined in different domains, e.g. geometry and kinematics. Therefore, a new standard, ISO/CD 10303-242 (STEP AP242, Managed model-based 3D engineering) [ISO, 2012], is chosen in this research for data modelling.

ISO 10303 STEP (STandard for the Exchange of Product data) is the international standard for industrial product data representation and exchange. Two application protocols of STEP, ISO 10303-203 (AP203, Configuration controlled 3D design of mechanical parts and assemblies) and ISO 10303-214 (AP214, Core data for automotive mechanical design processes), are implemented in all major CAD systems. ISO 10303-105:1996 (STEP p105 ed1, Kinematics) addresses kinematic mechanism representation. Currently AP214 is the only application protocol that uses p105 ed1. Till now the only valid application based on AP214 and p105 ed1 is implemented by Li et al. [2011]. However, due to the structure of p105 ed1, it is not possible to define associated properties, e.g. kinematic error, for existing kinematic elements.

At present, a project is ongoing to develop the new application protocol AP242 which is planned to unify and replace the AP203 and AP214. As an important part of AP242 development, standardised kinematic mechanism representation is being updated as the second edition of ISO 10303-105 (STEP p105 ed2, Kinematics). A major improvement of this edition is reusing the existing general representation construct defined in ISO 10303-43 (STEP p43, Representation structure), where the definitions and relationships of representation, items, and contexts are described [Hedlind et al.

2011]. Such structure provides possibility to define properties associated to different levels of kinematic elements: mechanism, kinematic pairs, and mechanism states.

Based on AP242, this paper presents a practical modelling approach to represent kinematic error as a property of kinematic elements. The general property representation structure defined in ISO 10303-41 (STEP p41, Fundamentals of product description and support) is used to represent the measured or calculated kinematic error information and the associations to corresponding instances defined in STEP p105 ed2. The data integration between geometry, kinematics, and kinematic error is evaluated and demonstrated by a prototype implementation, which indicates a comprehensive solution for data exchange between miscellaneous information systems.

2 KINEMATIC ERROR MODELLING

The kinematic error is divided into component error and location error, and the detailed definitions for both are different for rotational and linear axis [Schwenke et al., 2008]. Nominal kinematic pair frame placement as defined in STEP p105 (see Figure 2), is used to relate kinematic error concepts unambiguously in the context of STEP. Component error is the pair placement error from the respective nominal placement, and location error is defined as the average of contact frame placement error from its nominal placement, relative to the machine coordinate system.

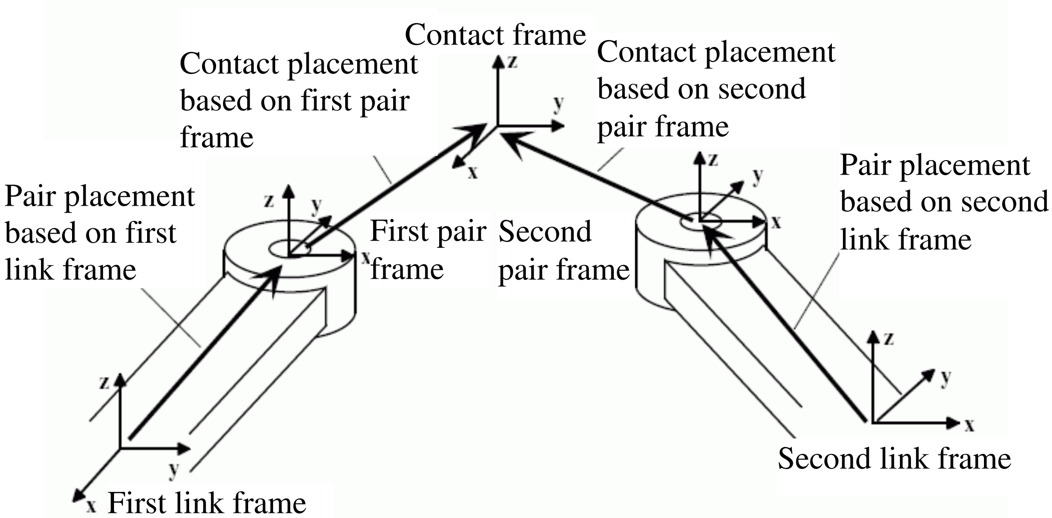

Figure 2; Placement relationship of a pair (ISO 10303-105)

Using the modelling principle defined by Kjellberg et al. [2009], an ontology model for the kinematic error concept is defined and applied on AP242. Figure 3 illustrates the kinematic error combined with geometry and kinematics ontology. Based on a unified modelling approach kinematic error can be modelled as kinematic pair property and represented using AP242 [Hedlind et al. 2010].

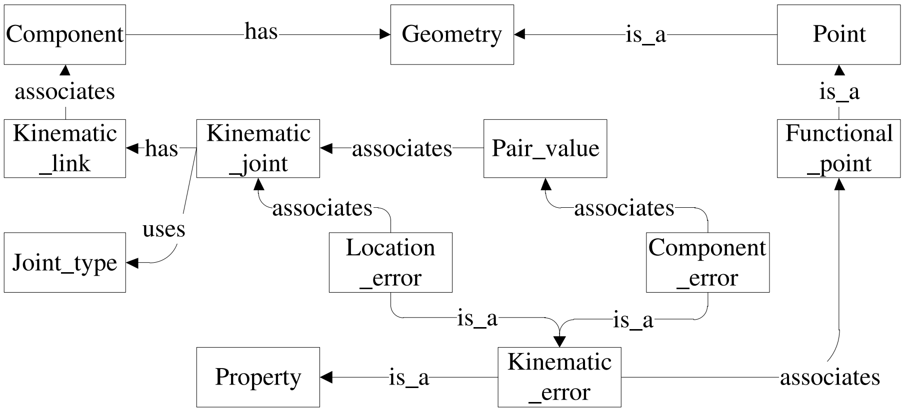

Figure 3; Kinematic error ontology

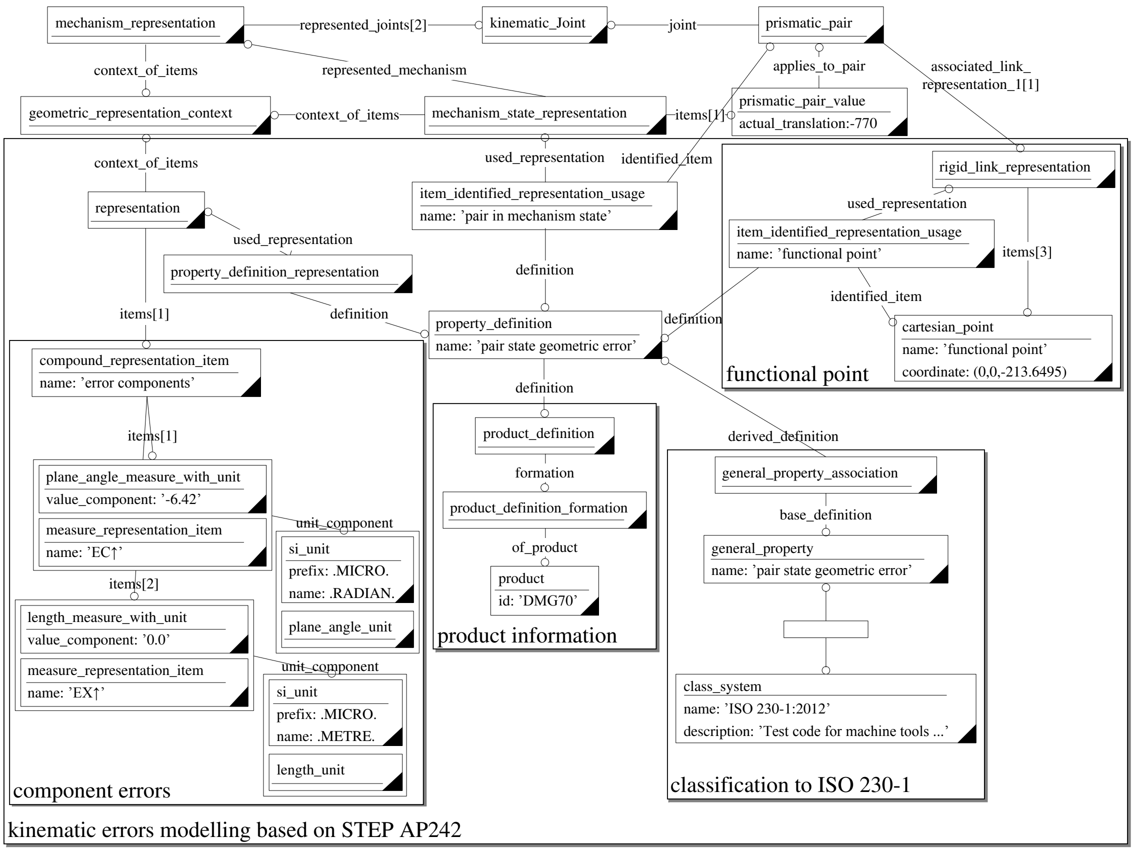

Figure 4; Component error modelling in AP242

With the presented ontology, the data model of kinematic error can be generated as classified properties in AP242. In Figure 4, a complete component error model is defined underlying the kinematic pair value model. A mechanism state is associated with a property named "pair state geometric error". The representations of component error are collected and compounded under the context of one representation referred by the property definition. The naming convention defined in ISO 230 is used as the name of each measurement value with unit, combined with the approach direction symbol ↑ or ↓ defined in the ISO 230-2. The product_definition of the assembly model is linked as the owner of the kinematic error property. A general property is assigned to the component error to declare the terminology and concepts defined in ISO 230-1:2012. The functional point is the location of the tool in the moving component when performing the straight line motion test as defined in ISO 230-1:2012. Therefore this location is represented in coordinate system of moving link.

As previously mentioned in this section, the location error is an average value that is to be associated to a kinematic pair independent of mechanism state. Hence, instead of the mechanism state as shown in Figure 4, the mechanism itself should be referred to as used_representation of the item_identified_representation_usage.

3 IMPLEMENTATION

To evaluate the capability of the proposed modelling approach, a prototype application is developed to translate an ASME B5.59 [ASME, 2008] XML machine tool data model to STEP AP242 model. The original data is based on version 13b of the draft standard ASME B5.59-2 (Data Specification for Properties of Machine Tools for Milling and Turning). Performance and capability data of the machining operation is described in the original XML file.

An important aim of the implementation is to perform and evaluate the AP242-based data integration between geometric model and kinematic error model for one machine tool. A major benefit of STEP AP242 is the high level of integration between different aspects of the described product, e.g. geometry, kinematics, classification, tolerances. Multiple levels of details and aspects of the machine tool model can be modelled by different applications and integrated in a single model. An AP242 model can be read and written by different CAD/CAM systems during its lifecycle without the risk of unwanted information loss. It also means functionality of STEP translators in existing systems can be reused for standardised exchange of kinematic error data. Therefore, the data integration needs to be evaluated by an implementation. In this research, the development takes the kinematic error model defined by ASME B5.59-2 and its shape geometry exported from CAD software as an example.

The application is developed underlying a STEP and Java based integration development project, STEP Toolbox. The project aims to provide a maintainable, reusable, extensible, and flexible programming interface to simplify STEP implementation for CAD/CAM developers. Via ISO 10303-22, SDAI (Standard Data Access Interface), STEP p21 file and other supported file types can be manipulated by several kinds of programming language. The STEP Toolbox provides modularised Java programming interface to process different types of product information in engineering-oriented perspective, e.g. geometry, kinematics, classification. Thus, with little STEP knowledge, developers can produce their own applications or plugins for CAD/CAM software based on the toolbox. During this implementation, the kinematic error data processing for AP242 is integrated within the kinematics module of STEP Toolbox. And the developed ASME/STEP translator performs its functions by using the integrated interface for kinematics within the toolbox.

In order to integrate kinematic error with the existing kinematics module of the STEP Toolbox, an UML (Unified Modelling Language) model is derived from the kinematic error ontology model (Figure 2). In Figure 5, a simplified class diagram for kinematic error is partly displayed underlying design of the STEP Toolbox.

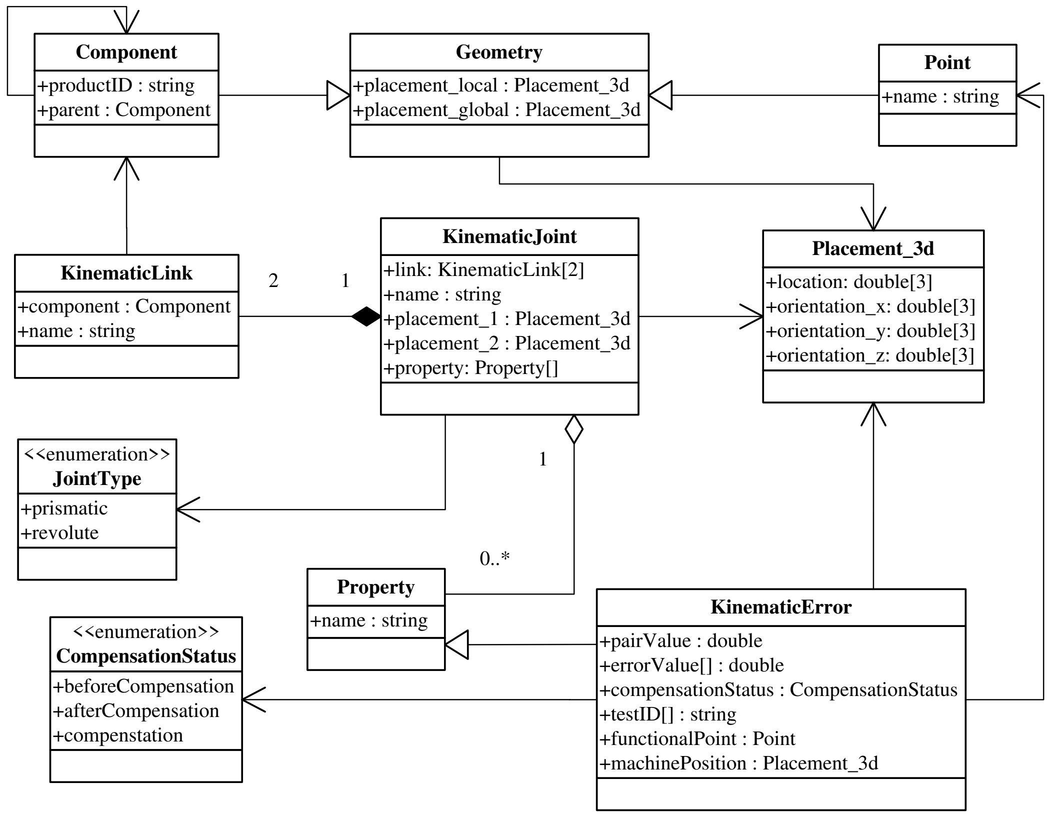

Figure 5; Simplified class diagram for kinematic error

As described in the ontology model, component and point are both subclass of geometry which maintains placement information for general usage. Objects of both links and joints contain both topological and geometrical information. Property is also a general class that is actually used not only by the kinematic joint. Kinematic error is the subclass of the property which is aggregated in the joint. Pair value is a pair parameter for special usage, and is a parameter with a collection of component error. For location error, the developer can simple define a null value for pair value which indicates that the value is independent of the mechanism state.

The system design for the implementation is illustrated in Figure 6. An ASME XML reader is developed to read the component error described in the XML file. The STEP Toolbox is used to retrieve the geometric and kinematic information from the AP242 file. To generate the kinematic error in the STEP model, the kinematics module in the toolbox is reused and updated and new data model of kinematic error is added for this project, which embodies the maintainability and reusability of the STEP Toolbox. The sample input ASME XML file is provided by NIST within ISO TC184 SC4 WG3 T24 collaboration, which contains machining capability information of a DMG70 machine tool. The STEP file with corresponding geometry and kinematics is produced by the KIBOS for NX (Li et al. 2011). A valid AP242 file is exported with the added pair values, functional points, kinematic error data, and classification as described previously.

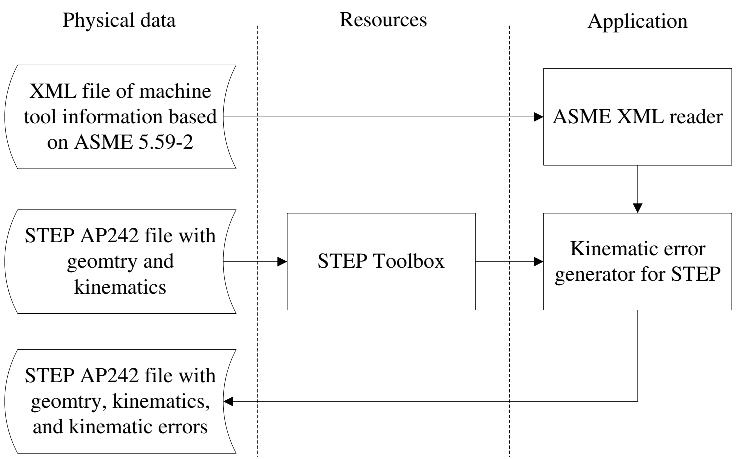

Figure 6; System design for ASME/STEP translator

4 CONCLUSION

This research focuses on the standardised modelling approach for kinematic error. A computer interpretable model for data exchange and integration is proposed and implemented. With the data model and implementation based on STEP AP242, this research utilizes benefits of the representation structure updated in p105 ed2 and demonstrates a modelling methodology applied in practice to reuse the general representation data model in kinematic mechanism. Data representation and exchange between CAD/CAM/CNC systems can be achieved by integration with existing standard translators. Multiple practical functionalities on kinematic error are supported with the proposed data model, e.g. measurement, prediction, and compensation. The solution is promising to be extended to the representation of any other types of properties and classification.

Currently, more data sets, schemas, and measurement practices are being involved in the research. The presented data model is being improved and expanded during discussion with international partners. As a contribution to the ISO TC184 SC4 standardisation work, this modelling approach will be a base for schema level extension in future standard development.

5 ACKNOWLEDGEMENT

We are grateful for the support from VINNOVA and XPRES (Initiative for excellence in production research), and for fruitful discussions with NIST and other members of ISO TC184 SC4 WG3 T24.

REFERENCES

[Donmez et al., 1986] Donmez, M. A.; Blomquist, D. S.; Hocken, R. J.; Liu, C. R.; Barash, M. M.; "A general methodology for machine tool accuracy enhancement by error compensation"; In: *Precision Engineering*, 8(4), 1986, pp. 187-196; ISSN 0141-6359

[Altintas et al., 2005] Altintas, Y.; Brecher, C.; Weck, M.; Witt, S.; "Virtual Machine Tool"; In: *CIRP Annals - Manufacturing Technology*, 54(2), 2005, pp. 651-673; ISSN 0007-8506

[ASME, 2008] ASME B5/TC56; "Part 2, Data Specification for Properties of Machine Tools for Milling and Turning"; In: *Information Technology for Machine Tools*; 2008; ASME B5.59-2, version 13b

[Euler-Chelpin, 2008] Euler-Chelpin, A. von; "Information Modelling for the Manufacturing System Life Cycle"; *Doctoral Thesis*; KTH Royal Institute of Technology, Stockholm, Sweden 2008; ISSN 1650-1888

[Hedlind et al., 2010] Hedlind, M.; Lundgren, M.; Li, Y.; Archenti, A.; Kjellberg, T.; Nicolescu, C. M.; "Kinematic structure representation of products and manufacturing resources"; In: *Proceedings of CIRP 2nd International Conference on Process Machine Interactions*; Vancouver 2010; ISBN 978-0-9866331-0-2

[Hedlind et al., 2011] Hedlind, M.; Klein, L.; Li, Y.; Kjellberg, T.; "Kinematic structure representation of products and manufacturing resources"; In: *Proceedings of CIRP 7th International Conference on Digital Enterprise Technology*, pp. 340-347; Athens 2011; ISBN 978-960-88104-2-6

[ISO, 2012] ISO TC184 SC4; "Managed Model-based 3D Engineering"; Application protocol; Committee draft, ISO/CD 10303-242:2012

[Kjellberg et al., 2009] Kjellberg, T.; Euler-Chelpin, A. von; Hedlind, M.; Lundgren, M.; Sivard, G.; Chen, D.; “The machine tool model—A core part of the digital factory”; In: *CIRP Annals - Manufacturing Technology*, 58(1), 2009, pp. 425-428; DOI 10.1016/j.cirp.2009.03.035

[Li et al., 2011] Li, Y.; Hedlind, M.; Kjellberg, T.; "Implementation of kinematic mechanism data exchange based on STEP"; In: *Proceedings of CIRP 7th International Conference on Digital Enterprise Technology*, pp. 152-159; Athens 2011; ISBN 978-960-88104-2-6

[Schwenke et al., 2008] Schwenke, H.; Knapp, W.; Haitjema, H.; Weckenmann, A.; Schmitt, R.; Delbressine, F.; "Geometric error measurement and compensation of machines – An update"; In: *CIRP Annals - Manufacturing Technology*, 57(2), 2008, pp. 660-675; ISSN 0007-8506

Paper C

Li, Y., Su, R., Hedlind, M., Ekberg, P., Kjellberg, T., and Mattsson, L. (2012b) Model based in-process monitoring with optical coherence tomography, *Procedia CIRP*, Vol. 2, pp. 70-73

Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia CIRP 00 (2012) 000–000
www.elsevier.com/locate/procedia

1st CIRP Global Web Conference: Interdisciplinary Research in Production Engineering

Model based in-process monitoring with optical coherence tomography

Yujiang Li[a]*, Rong Su[a], Mikael Hedlind[a],
Peter Ekberg[a], Torsten Kjellberg[a], Lars Mattsson[a]

[a]*Dept. Production Engineering, KTH Royal Institute of Technology, Brinellvägen 68, Stockholm 10044, Sweden*
*Corresponding author. Tel.: +46-8790-8384; fax: +46-821-0851.*E-mail address*: yujiang.li@iip.kth.se.

Abstract

The demands on in-process 3D monitoring in ceramic micromanufacturing industry require a high-precision, non-destructive, rapid and automated inspection technique for measuring the thickness of component layer, determining the shape and dimensions of the embedded 3D structures, and detecting the de-bonding, cracks, warping and deformation. One of the promising metrology techniques is optical coherence tomography (OCT). With the dedicated image processing algorithm and the industrial product data exchange standard, the model-based integration of OCT as a new metrology tool is demonstrated. As a generic standard for any product or manufacturing information, ISO 10303 STEP AP242 is employed for the measured data model. Unambiguous data representation is achieved by integrating additional modelling constraints. The proposed framework allows fully using the technical advantages of OCT to in-process 3D monitoring.

Keywords: OCT; STEP AP242; in-process monitoring; system integration.

1. Introduction

In order to keep the competitive advantage European manufacturing industry has to face the related R&D (research and development) challenges to achieve cost efficiency, high performance, enhanced robustness, and large-scale production.

In the European project Multilayer, large-scale and cost-effective production of micro devices with advanced ceramic materials is enabled, based on the development of “Roll-to-roll multi-material layered 3D shaping technology”. In this context, the future in-process 3D monitoring will highly rely on a high-precision and non-destructive technique for dimensional measurement and detection of subsurface features and defects. [1]

One of the promising metrology techniques is optical coherence tomography (OCT), which is based on low-coherence interferometry and has been actively used in many biomedical applications. With the advantages of non-contact, micrometers resolution and high signal acquisition rate up to 300 kHz [2], OCT is more suitable to meet the future demands of in-process 3D monitoring than the other candidate techniques such as acoustic microscopy, x-ray micro tomography and laser scanner, in the industry of ceramic micro processing. To enhance the evaluation and allow for accurate geometrical measurements despite poor signal to noise ratios in the OCT images, a dedicated high-precision image processing algorithm has been developed [3]. This method avoids the loss of useful information due to speckle reduction and allows recognising surfaces, embedded boundaries and features in noisy OCT images with high precision.

Generated from the OCT sensor, the data set of unorganised point cloud may be utilised in different contexts. For defect inspection, the as-designed CAD model should be compared with the as-measured model composed by the point cloud with special matching methods and algorithms. For dimensional measurement, shape reconstruction and shape recognition should be performed in specific applications.

In order to automate the inspection and measurement process, this research proposes a framework of system integration and data integration for the application of OCT technology in manufacturing industry.

The system integration is presented with an architecture design to coordinate CAD/CAM software and define the principle for data manipulation and exchange. Requirements on the data schema for a model-based solution can be outlined from the architecture. The design also provides a guideline to perform computer-aided measurement with OCT technology in practice.

Data integration is solved by a set of data models based on a general international standard to support the system in multiple stages. The unique ability of subsurface detection from the OCT technology enables inspection of the assembled product without destructively invading the material. Hence, the proposed data model facilitates the assembly model of point cloud which may be combined with the CAD assembly model.

ISO 10303 STEP (STandard for the Exchange of Product data) is the international standard for industrial product data representation and exchange. Point cloud modelling is included in the *measured data* Unit of Functionality (UoF) in the commonly used application protocol, ISO 10303-214 (AP214, Core data for automotive mechanical design processes). However, there is no known application or research implementing this UoF. At present, the new application protocol, ISO/CD 10303-242 (STEP AP242, Managed model-based 3D engineering), is under development to replace AP214 and AP203. In the following section, AP242 is employed to model the point cloud and a modelling approach will be introduced to support different aspects of data requirements: point cloud representation, surface representation, assembly representation etc.

2. System integration

As a new measurement technology, there is no existing information system that completely supports OCT in manufacturing. Therefore, a framework has to be defined to integrate different applications and data models for different usages.

Most researches on computer aided measurement system follow a similar stepwise data processing flow. Tang et al. [4] present a summary of the process for creating BIM (Building Information Model) from laser-scanning point cloud in the context of AEC (Architecture, Engineering, and Construction) industry. Three steps are needed to transform the original measurements to semantic rich BIMs: 1) data collection to collect scanning signals, 2) data preprocessing to filter the data and formulate coordinate system, and 3) modelling the BIM to construct surface representation and recognise semantic information. In a practice guide of vision system applied in dimensional measurement [5], another stepwise process is defined: 1) image formation and sensing, 2) image processing, and 3) communication.

In this research, the architecture of a model-based system is presented, and standardised data integration is employed to enable the data modelling and exchange for system integration. Three parts constitutes the system:

- Collection: producing the image from OCT,
- Processing: translating the image to the computer interpretable data model,
- And application: utilising the model in different contexts.

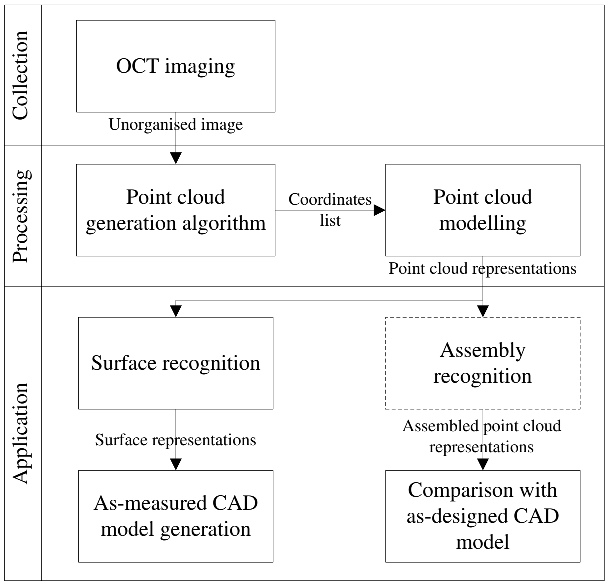

Fig. 1. System architecture

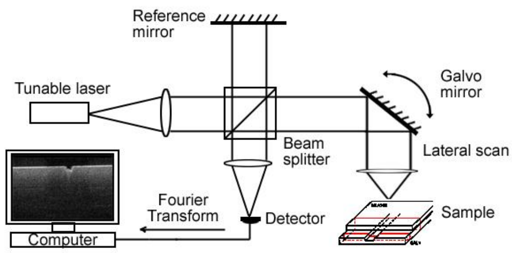

Fig. 2. Schematic setup of swept-source OCT

The detailed principles of OCT can be found elsewhere [2]. A commercial swept-source Fourier-domain OCT (Fig. 2) is used in this study. At the exit of a Michelson interferometer the spatial frequency spectrum is recorded when tuning the wavelength of light source and the power spectrum density is transferred to spatial information in axial direction according to the Wiener-Khinchin theorem. Only lateral scans are performed using galvo mirror, so the signal acquisition rate can be dramatically increased. Moreover, the axial resolution determined by the coherence length of light source is decoupled with the lateral resolution that depends on the numerical aperture (NA) of objective. Thus structures embedded in translucent material or with high respect ratio can be detected without a severe aberration using a small NA.

The second part performs data translation and prepares computer interpretable data model for further application. A developed image processing algorithm [3] reduces noise in the image using Gaussian filter, and then generates a 2D map containing all pixels that are probable candidates for contributing to the continuous backscattering peak in the original image. In the final step fine-tuning algorithm are used for a better estimation of the boundary location in sub-pixel precision.

From a list of coordinates, point cloud representation can be generated based on a standard data model schema. To be prepared for computer interpretation, information on geometric representation context is involved: unit, uncertainty, and placement of coordinate system.

The standardised representation can be applied in different scenarios in the application part. For geometrical dimension measurement, with existing commercial reverse engineering (RE) software, surface recognition and modelling can be performed. The output surface representation may be used in other applications for final processing, where features, components, assembly, and other types of semantic information are associated to the as-measured shape model.

As mentioned before, an important feature that distinguishes the OCT technology with the others is the detection of subsurface defects and embedded structures, e.g. channels and designed cavities. Such feature enables the rapid inspection for the product quality of both components and assembly. Therefore, a complete point cloud model of an assembled product is possibly generated with recognition methods, if the CAD model of assembly and relevant comparison algorithms is available for defects inspection. Existing matching methods include area based matching, feature based matching, centroid method etc. [6] The standardised point cloud representation and as-designed CAD model can be employed based on the same standard data access interface to support computer-aided matching.

3. Data integration

To facilitate the proposed system integration, the data model design focuses on two aspects of data integration: 1) integration with concepts defined in the reference model of AP214, and 2) integration with existing modelling principle for general product information representation. Because AP242 has not yet included the explicit representation of point cloud, the Application Reference Model (ARM) in AP214 can be reused to facilitate reasonable mapping by practitioners. The integration with existing modelling principle for product information of STEP will help to reuse and enhance STEP translators in existing CAD/CAM software.

The intent use of point cloud representation is explained in AP214 AAM (Application Activity Model) for the *measured data* UoF, as applicable for prototype test results, conceptual design information, and tool test results. In this research, this data model is reused for in-process inspection and surface recognition, still within the defined scope of UoF. In the current version of AP242 under development, the similar functionality has not been specified explicitly, but it can still be implemented by reusing the general shape representation which is the super type of point placement shape representation defined in the AP214 AIM schema.

The integration with the existing data schema of STEP AP242 is another focus in this research. At present, the general representation structure of product information based AP214 for data exchange is very commonly implemented among all major CAD software. Therefore, the data integration aims to support the system integration to retain the "legacy applications" in a reasonable way [7]. In a previous research [8], reusing existing product information model of AP214 with additional properties has shown benefits in system integration with existing CAD system. Although AP242 extends and updates representation capabilities from AP214, the fundamental structure will be kept and legacy data will be valid with the new schema. Therefore, it is possible to transform the existing AP214 model to the AP242 while the legacy application can be extended easily to manipulate the additional properties defined in the new schema.

A complete AP242 data model of point cloud structure is shown is Fig. 3. In this initial stage, the assembly structure is not considered. The model focuses on representation of product, application context, geometric context, and shape, in line with the AP214.

The proposed model is mainly based on the pre-defined structure of the *measured data* UoF in AP214. For specific attributes of instances, a preliminary naming recommendation is introduced, but further discussion is necessary to achieve a modelling convention or update suggestions in the schema level. A shape representation named "point placement" is used to replace point placement shape representation defined in AP214 as a subtype of shape representation. The compound representation item indicates a group of unsorted Cartesian points.

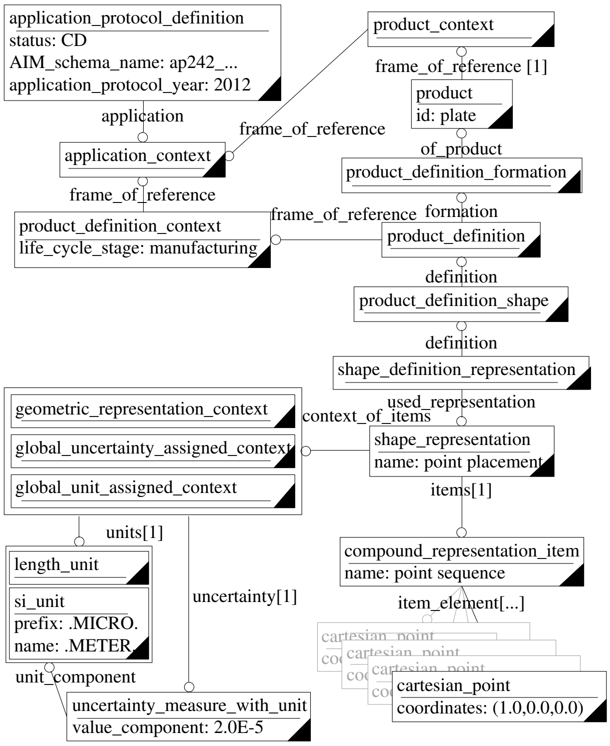

Fig. 3. Point cloud data model based on AP242

Particularly for the utilisation of the OCT technology, the focus is on the assembly model of the point cloud. The integration with the designed shape model should be developed to support comparison algorithm. In Fig. 4, a component is modelled with the combination of the point cloud and the boundary shape representation. Two product definitions represent two aspects of the product: as-measured data and as-designed data. The product definition formation indicates both definitions are based on the same version of the product. A product definition relationship indicates that two aspects relate to each other, as defined by general item definition relationship in AP214.

4. Conclusion

In this research an integration framework is proposed, which utilises the benefits of OCT technology and realises the STEP standard enabling rapid in-process 3D monitoring in micromanufacturing industry. The achieved system integration supports automated measurement process and defines criteria for the information modelling. The computer interpretable data model based on the STEP AP242 is proposed to ensure the implementation of the system integration.

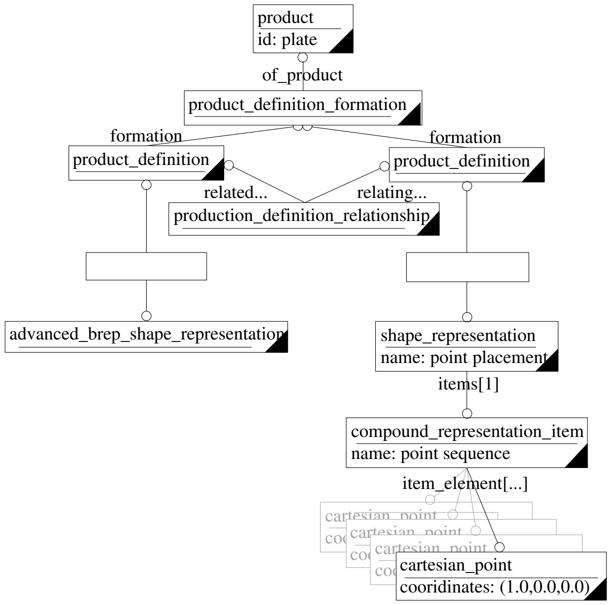

Fig. 4. Integration with CAD model based on AP242

Acknowledgements

We are grateful for the support from VINNOVA, XPRES (Initiative for excellence in production research), and the Multilayer project (FP7-NMP4-2007-214122).

References

[1] Bredeau, S. and Federzoni, L., 2009. "Multilayer: a large scale production of micro devices via new rolled multi material layered 3D shaping technology," 4M/ICOMM 2009 Conference. Karlsruhe, Germany, p. 419-422.

[2] Bouma, E., Tearney, G., 2002. Handbook of Optical Coherence Tomography. Marcel Dekker, New York.

[3] Su, R., Kirillin, M., Ekberg, P. Roos, A., Sergeeva, E., Mattsson, L., 2012. Optical coherence tomography for quality assessment of embedded microchannels in alumina ceramic, Optics Express 20(4), p. 4603-4618.

[4] Tang, P., Huber, D., Akinci, B., Lipman, R., Lytle, A., 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques, Automation in Construction 19(7), p. 829-843.

[5] Rodger, G., 2001. Dimensional measurement using vision systems. National Physical Laboratory, United Kingdom.

[6] Chen, F., Brown, G., Song, M., 2000. Overview of three-dimensional shape measurement using optical methods, Optical Engineering 39(1), p. 10-22.

[7] Hasselbring W., 2000. Information system integration, Communications of the ACM 43(6), p. 33-38.

[8] Li, Y., Hedlind, M., Kjellberg, T., 2011. "Implementation of kinematic mechanism data exchange based on STEP," CIRP 7th International Conference on Digital Enterprise Technology. Athens, Greece, p. 152-159.

Paper D

Li, Y., Chen, D., Kjellberg, T., and Sivard, G. (2013a) User friendly development architecture for standardised modelling: STEP Toolbox, submitted to *International Journal of Manufacturing Research*

User friendly development architecture for standardised modelling: STEP Toolbox

Abstract: Currently one of the methods to overcome system interoperability within manufacturing is to use existing information standards. Some of issues related to this method are information and implementation complexity. This paper presents the development architecture STEP Toolbox to tackle these two issues. It supports implementations on product manufacturing information based on the standard ISO 10303 (STEP). It is designed friendly for industrial practitioners to develop STEP based applications with little knowledge of this standard. The toolbox utilises ontology and UML to represent the necessary components for development, which is more familiar for common developers compared with traditional standard documents. The developers' task is therefore simplified and becomes more similar to ordinary implementations. A two-month learning process to become advanced STEP developers is expected to be skipped according to test cases with three Java-based developments.

Keywords: User friendly; STEP; standard for the exchange of product data; development architecture; implementations; standardised modelling; interoperability; ontology; UML; Java.

1 Introduction

Currently industry is using various kinds of digital manufacturing and digital factory software tools to design and verify the models, concepts and details related to manufacturing processes and manufacturing systems developments. Within an enterprise, it is strategic and historical to have specialised software for each business function (Lui et al., 2011). Considering the collaborations among supply chains, a large number of interoperability issues can be addressed to deal with the high diversity of devices and applications (Tolio et al., 2010). The needs for information exchange and sharing are proposed frequently to bridge data models specialised in disparate digital manufacturing and digital factory tools. A neutral data format is regarded as an important approach to realise interoperability for CAx (Computer Aided technologies) systems (Mokhtar and Houshmand, 2010).

The international standard, ISO 10303 STEP (STandard for the Exchange of Product data), has been proposed as an important system neutral solution for industrial data representation. Two application protocols of STEP, AP203 (ISO 10303-203) and AP214 (ISO 10303-214), are commonly implemented in most major CAx systems for the data exchange of 3D geometry data (Xu et al., 2011). Based on the same modelling language EXPRESS (ISO 10303-11), several other international standards in production engineering, e.g. ISO 13399 for cutting tool representation, PLib (ISO 13584), and STEP-NC (ISO 14649), can share the implementation principle with STEP. A complete and seamless product data exchange cycle can be enabled from the requirements analysis to manufacturing with these standards (Xu et al., 2005). The applications of this standards family can be found in lots of areas (Lau and Jiang, 1998; Pratt, 2005; Campos and Xu, 2010; Khaled et al., 2010; Valilai and Houshmand, 2010). However, compared with the large scope of the standard family, the practical implementations in industry are still a small fraction of the total.

The lack of practical usage of STEP standard is observed mainly due to its relatively complicated implementation methodology. Conventional development process based on SDAI (Standard Data Access Interface, ISO 10303-22) has been summarised by Loffredo (1999), including (1) selection of APs, (2) selection of development tools, (3) selection of data model binding methodology, (4) system design, (5) development, (6) model validation, and (7) deployment. In this process, developers need solid background knowledge to maintain the conceptual mapping between their domain and the standard. The generated or modified data models should be kept conforming to the standard. This requires great efforts, e.g. time and human resources, for industry. This development process is not convenient for normal programmers. Additional reading time is needed to go through a large number of standard documents, compared with common software development. Careful model validation with special technology is required to maintain consistent instantiation.

The objective of this research is to simplify the traditional development process and to save the learning time for developers, with the adoption of the user friendly development architecture, so called STEP Toolbox. It will provide a platform on which developers can access and manipulate STEP data sets with a friendly way. Unlike the conventional STEP development process, an API (Application Programming Interface), STEP Toolbox API, is provided in a way that programmers are familiar with. The complicated SDAI operations and STEP data models are encapsulated within the API, and only needed concepts are revealed for developers. UML (Unified Modelling Language) is used to present the concepts to the end users of this API, and it will be easily implemented by any object oriented programming language. Ontology is used to control how the STEP data models can be simplified and how the UML model should be constructed. Therefore the ontology is a mapping bridge between STEP and UML. With the solution of this toolbox, developers from academy and industry can focus on their own working priorities, instead of spending lots of time in learning STEP related information. Another benefit is that some STEP-related development stages can be skipped, such as selection of data model binding methodology and model validation.

In Section 2, different levels of STEP user groups are defined to clarify the target beneficial owners of this architecture. Then, related projects and products for these users are introduced in Section 3. The structure of the toolbox and its development principle is described in Section 4. A data model mapping strategy is developed to make the information model of STEP friendly for human reading and utilisation, especially for developers in industry. A prototype API is proposed based on the data model mapping, and documented with a programmer-intuitive way. In Section 5, kinematics is taken to exemplify this mapping strategy. In Section 6, an example kinematics data model API is used in test cases performed in industry to validate this new architecture.

2 Various levels of STEP users

STEP Toolbox aims to simplify the utilisation of STEP standard in practical implementation. Therefore, it is necessary to investigate the potential users and identify the user group who needs the help of the toolbox. Three levels of STEP users are defined according to observation in this study (Figure 1). The three user groups have different types of tasks and different levels of knowledge about this standard. The first level is common CAx users who use STEP p21 (ISO 10303-21) or p28 (ISO 10303-28) files to exchange the data between systems. These users may only know the major purposes and functions of STEP. The second level is a subset of the first level, with additional demands for knowledge of the standard. They are developers that develop CAx-related applications based on STEP for data modelling. Employees in CAx software vendors can be required to program STEP-based software to enable data exchange and communication with different data formats. Practitioners in industry need to implement STEP to support their CAx work for internal and external collaborations. The third level is schema developers who are familiar with the core structure of STEP and able to contribute to schema development based on EXPRESS language. Schema developers are usually most knowledgeable on this huge standard or on a particular part of the STEP. They should also know how to implement STEP, which makes them the subset of the CAx developers.

The aim of STEP standards is to solve interoperability issues of the common users (the first level) who are using all kinds of CAx systems. To develop an easy-to-use standard for common users, a plenty of work has been done since its first publish. Various application protocols and strategies are available for different needs of data exchange. On the other hand, the complexity for schema developers (the third level) is easily observed by related researchers, which triggers many studies, e.g. (Danner et al., 1991; Burkett and Yang, 1995; Goh et al., 1996; Feeney, 2002; Jardim-Gonçalves et al., 2005; Maier and Stumptner, 2007). However, the CAx developers (the second level), who are struggling to understand the standard specifications, are often ignored.

This study focuses on the practices performed by the CAx developers. They may be professional programmers who develop STEP-related software for their careers, or end user programmers who develop applications to support their work or hobby (Ko et al., 2011). They are the only frequent readers of standard documents in practices, and have to devote great effort to conquer relevant knowledge to develop STEP implementations. For a developer of a small Java application for e.g. kinematic translation based on STEP AP242, his/her working time will be heavily occupied by learning the standard documents, besides learning general production engineering knowledge. The required documents are even more than the requirement for some schema developers (the third level users). Below is a list of needed documents (not including corrigenda):

- ISO 10303-1 for an overview of the documentation structure,
- ISO 10303-11 for EXPRESS language,
- ISO 10303-21 for STEP file description,
- ISO 10303-22 for SDAI,
- ISO 10303-27 for the binding to Java,
- ISO 10303-41 for fundamental product description,
- ISO 10303-43 for representation structure,
- ISO 10303-105 ed2 for kinematics,
- And ISO/DIS 10303-242 for this application protocol.

According to implementation experience of this research, it takes approximately 2 months of full-time learning to enable a common developer to understand and implement STEP. A comprehensive structure makes STEP to support the data representation in a wide range of industrial sectors and lifecycle stages (Pratt, 2001), but it also results in great complexity for developers to learn it from beginning. Besides the heavy learning procedure, inconsistent instantiation is observed as a practical problem regarding information modelling (von Euler-Chelpin, 2008). A lack of experience, knowledge, or carefulness can make developers interpret the information models differently, which damages the unambiguous usage of standard. As the only ISO standard in this field, STEP is an important choice for industrial enterprises to enable fast and reliable data exchange in internal and external communication. However, the time consumption of preparation and development has become a vital obstacle to make it as popular as designed in its target areas. Both researchers and practitioners are complaining that the "expensive" cost on human resources makes STEP an unfriendly standard for developers.

3 State of the art STEP implementation

As stated before, STEP is an important candidate for system neutral product data representation and exchange, which is noted for the unambiguous modelling support of product data during its lifecycle (SCRA, 2006). For geometry data exchange, AP203 and AP214 have been implemented in almost all major CAD software, e.g. Siemens NX, CATIA, AutoCAD, and Pro/ENGINEER. However, at present, there are still a large number of parts far from practices, in many other implementation domains (e.g. shipbuilding and architecture) and implementation methods (e.g. EXPRESS-XML and EXPRESS-UML). Tassey (1999) has concluded three causes of STEP interoperability development failure: (1) economic costs and benefits shared unfairly, (2) risks in technology and marketing globally, and (3) a lack of unbiased expertise. The interoperability development is performed by the second level of STEP users, as stated before. All the causes of the failure are related with the considerable requirements which are put on this user group. An important solution to this failure is to lower the technical complexity of STEP in the implementation level so that relevant users are able to implement it easily.

In general, a STEP implementation means the development process of applications with functions to access or manipulate STEP-based data sets (e.g. p21 files or p28 XML files). The process also fits for developments based on any other information models defined with the EXPRESS language (e.g. data sets based on ISO 13399 or ISO 13584). As listed before, a heavy workload is required for developers to go through this process, especially during the preparation stage for learning the standard comprehensively. This is the major reason why the new development architecture, STEP Toolbox, is proposed in this article to considerably simplify the conventional development process.

For most programmers, existing EXPRESS-based development tools are regarded important to save human resources and cost from manual creation of data models and operations library according to the standard. Such tools are usually implementations of SDAI in specific programming languages, which provide fundamental operations to access, manipulate, and validate EXPRESS-based data sets. For instance, JSDAI provided by LKSoft is a Java programming toolkit for EXPRESS schemas and EXPRESS-based data sets development (Klein, 2000), which is reused within STEP Toolbox. A similar tool in development based on

C/C++/Python is STEPcode (SC) which replaces the former product, STEP Class Library (SCL) (Sauder and Morris, 1995). Also for development, Reeper is unique in the support of XML-based STEP schemas and data sets conforming to ISO 10303-28. In the commercial software market, ST-Developer has been developed for over two decades and adopted in several major CAD applications for STEP translation. It has competition from EDMdeveloperSeat by Jotne EPM Technology, ECCO Toolkit by PDTec, etc. Table 1 makes a simple comparison of some example STEP development tools.

Although such development tools have reduced a lot of preparation work for developers, they are still put themselves within the conventional STEP development process. Developers need to very well understand data models, EXPRESS schemas, SDAI, and so forth before a “Hello world” program. In this research, STEP Toolbox is proposed as the development architecture with a substantial change of the STEP development process. With the low knowledge requirement of common STEP users, development time span and work load are promising to reduce significantly.

4 STEP Toolbox architecture design

STEP Toolbox aims to provide a friendly, maintainable, reusable, and extensible programming platform to simplify the STEP implementation process. With the designed ontology and UML model, programmers are able to construct their knowledge with a set of documentations more similar to other object-oriented API libraries. A new generation of the STEP implementation process will be as intuitive as a normal software implementation, compared with the conventional one.

4.1 Illustration of STEP Toolbox

The architecture of STEP Toolbox is briefly illustrated in the following Figure 2, which uses three layers to represent where and how the toolbox API should be applied. The figure briefly describes the internal modularised structure of the API and the external behaviours to interact with data sets and applications built upon the toolbox. The standardised data sets in the bottom layer are defined in the STEP standard and STEP-related standards e.g. ISO 13584. The middle layer is the core part of this architecture, STEP Toolbox API. It includes modularised components developed in this research and the SDAI as the basis for data model manipulations. The top layer indicates implementations based on STEP Toolbox API, which demonstrates the usability of this new architecture.

In the bottom, the requirements to manipulate various types of standard file formats are fundamental to determine the external behaviours of the toolbox API. In practices, STEP p21 based on EXPRESS is the most commonly used file format in the family of STEP. An alternative, p28 known as STEP-XML, is introduced to employ XML in the description of both schemas and data sets. Other standards and information models are using EXPRESS as well, which makes STEP Toolbox API able to process their data formats, e.g. the access of ISO 13399 classification dictionary based on PLib (ISO 13584) has been tested with the STEP Toolbox API.

The middle layer is the place where the STEP Toolbox API is located. It is the only programming interface that the developers need to learn when implementing STEP. This API will provide the same functionality as SDAI for exporting, importing, and modifying data sets, but in a much simpler way. SDAI is still the fundamental interface to manipulate the data sets in the format of p21 or p28 file. JSDAI, which implements SDAI with Java, is involved within STEP Toolbox and reused in many operations. To shorten the learning process interacting with the conventional SDAI, the toolbox API acts as a high-level programmer-friendly API upon which CAx plugins or standalone applications can be built. Detailed description of STEP Toolbox API will be presented in section 4.2.

In the top layer, STEP implementations such as plugins with CAx software or standalone applications can be built with the support of STEP Toolbox. There have been successful examples in different areas developed by authors or project partners already. For instance, kinematics translator for Siemens NX is the initialisation of the STEP Toolbox, which demonstrates the feasibility to implement STEP with commercial CAx software in other areas than geometry. In order to help project partners from industry, the STEP Toolbox comes into being to support their development based on CATIA. Other examples include an OCT (Optical Coherence Tomography) based point cloud translator for in-process monitoring, a cutting tool classification based on ISO 13399 cutting tool dictionary and STEP AP242, and an ASME/STEP translator for geometrical error data exchange.

4.2 Internal design of STEP Toolbox API

The components of STEP Toolbox API are illustrated in the middle level of Figure 2. Except the SDAI layer, other components are built to support this new generation of STEP implementation process. STEP model manager performs general functions to initialise, export and close a model from a p21 or p28 file. Data model is an abstract concept for all classes representing different types of objects in a STEP model, e.g. components, kinematic links, diameters, linear distances, classes, and properties. Utilities are a group of functions that provides some generally reusable operations, e.g. mathematical service for coordinate system transformation. Grouping the data model and relevant operations is important to simplify this comprehensive standard so that developers are able to access useful operations easily. The toolbox API provides different modules for different types of information, e.g. geometry, kinematics, and classification. For each module, there is a specific manager to view and modify the data sets, and a group of data models. For instances, the classification application, based on ISO 13399 cutting tool dictionary and AP242, highly relies on data models of GD&T (Geometric Dimensioning and Tolerancing), classification, part library, and a part of geometry. In one of the mentioned modules, GD&T, a GDTManager is able to collect the available information in the form of the specific data models of GD&T. Developers can simply perform operations on the collected data models, and the changes will be directly made in the STEP data set.

The data model of different modules is the key part to make the toolbox support the wide range of domains covered by STEP. The toolbox adopts UML to describe data models of the modules. An ontology model is mapped from the EXPRESS schemas in order to instruct the design of the UML model. The ontology is also a source from which the API users will easily get the guidance of the whole structure of the toolbox. Therefore, the concepts defined in the ontology should be easily interpreted by human and able to be mapped to a computer interpretable UML model. This mapping relationship is illustrated in Figure 3, by which the relatively complicated EXPRESS model is hidden from the programmers eventually. The mapping between EXPRESS and ontology should be performed according to the general understanding of relative domain specific knowledge. In the next section, a detailed description of a prototype implementation of this mapping process will be presented with kinematics as an example.

5 Data model generation

Even in this prototype stage, a large amount of information in several modules has been included in STEP Toolbox. Therefore, it is not necessary to introduce the modules one by one. The remaining part of this paper will use kinematics as an example to describe the data model mapping process stated before.

STEP p105 (ISO 10303-105) addresses the methodology of kinematic mechanism representation. Currently AP214 is the only application protocol that uses the first edition of p105. At present, a project is ongoing to develop the new application protocol, AP242, of which an important part is p105 ed2 for a new generation of the standardised kinematic mechanism representation (Hedlind et al., 2011). These latest standardisation results, i.e. AP242 and p105 ed2, are used in this research.

Implementation requirements of data exchange from both academy and industry are the reason why kinematics is chosen as a representative example among the different modules (see more in Section 6). As stated in Section 1, the long learning process and the abnormal development process are big obstacles to implement STEP. Besides, there is a remarkable lack of knowledge and experience regarding the AP242 and p105 ed2 due to its recent realisation. Therefore, a user friendly solution to simplify the development with STEP is highly valued for developers from industry.

5.1 Kinematic model based on STEP p105 ed2

Because STEP Toolbox is one of the first applications based on these new parts of STEP (AP242 and p105 ed2), it is necessary to introduce the related standardised modelling approach briefly.

The basic concepts of the kinematic mechanism modelling are common among all major CAD software: links, joints, and pairs are combined to represent the kinematic mechanism. Links act as rigid or linear flexible objects in the topological structure. Each joint is composed of two links to describe the topological aspect of the kinematic mechanism. Pairs describe motion constraints and associations with geometries.

Figure 4 illustrates a sample of kinematic link modelled within an AP242 data set based on EXPRESS language. The kinematic link instance is used to describe the topological aspect (see more in Figure 5). The geometrical aspect of the link is represented by a rigid link representation which defines the geometric placements for each user pairs. As a property, a link should be associated with a product definition by a context dependent kinematic link representation.

An example of the kinematic topological structure is illustrated for a 5-axis machine tool in Figure 5. Applying graph theory, a kinematic link is treated as a vertex used by one or more joints which are treated as edges. In this example, the five joints represent the five axes respectively. An instance of the kinematic topology structure is used to collect all the relative joints and to associate them with a mechanism.

The geometrical aspect of a prismatic pair and its links are displayed in Figure 6. The prismatic pair with range defines translation motion DOF (Degrees Of Freedom) between two links along an axis. There are range attributes in the pair to define the lower and upper limit values if necessary.

This section briefly introduces the kinematic mechanism based on STEP p105 ed2. Note that only necessary attributes and relationships of the instances are revealed here. Due to a lack of detailed documentation of this standardised modelling methodology for developers, it takes a long time for beginners to capture enough knowledge for a successful implementation.

5.2 Ontology modelling

The ontology of STEP Toolbox is a bridge between the EXPRESS schema of STEP and the UML class model for programmers. It regulates how the concepts in STEP models can be simplified, according to the implementation environment. It is a result of an agreement among the STEP models, the CAx concepts, and the developer knowledge.

A vocabulary interpretable by both computer and human is defined in the ontology so that programmers are able to share the same concepts with the standard. A CAx user can easily find concepts in STEP that are defined differently in various CAx systems. Therefore, a unified and general vocabulary is needed. For instance, in a kinematic mechanism of a machine, one kinematic link is the base which is fixed with the ground or stays static relatively. In an AP242 data set, such link is referred to by the mechanism representation as a base. In Siemens NX 7.5, the same concept is expressed with a Boolean attribute of a link to set if it is fixed. In CATIA, there is no definition for links but the geometric component can be set to compose a joint directly, and a base link is represented by setting the component as a fix part.

As there are several modules included in the toolbox now, a domain-independent ontology model is needed. It means concepts should be designed reusable when used in two or more modules. For instance, a “product” in STEP standard is a general concept which may have one or more occurrences in a data set. Because occurrences are more useful than products in CAx implementation, the occurrence, instead of the product, is taken as a class in the ontology, which is called the “component”. For the same product in STEP, each occurrence is set as a “component” in the ontology, which shares the same product ID as an attribute. Hence, the component can be reused easily in other modules than the geometry, e.g. in the kinematic mechanism, a link can be associated with one component.

Figure 7 exemplifies an ontology model for kinematics in a simplified way. Note that the properties and their facets of all the classes in the ontology are not shown in the figure. For instance, a kinematic joint has a property to define its motion type such as prismatic, revolution, cylinder, etc. To make it easily interpreted by computer, several simple terms are applied for the relationship, e.g. generalisation, aggregation, association, and dependency. The ontology is a combination of kinematics and geometry module. A component is defined as a subtype of the geometry which uses the placement to identify the location and the orientation. Each kinematic link is associated with a component. Two links make a joint which uses the placement for location and orientation as well. The exemplified ontology model can be extended further for other purposes, such as kinematic errors modelling (Li et al., 2012).

5.3 UML modelling

For programmers, an API well designed and documented is important to implement STEP by the toolbox. Therefore, a UML class diagram is generated from the ontology and implemented. In this example, there are two types of classes: data model classes to represent basic elements in a STEP data set, and managers to provide static attributes and operations to collect information in the data set and to perform necessary functionalities. One of these managers, the STEP model manager is the central controller to initialise, export, and close a data set. Each module has its own manager for creating, reading, modifying, and removing relative data.

In Figure 8, a simplified class diagram for kinematic mechanism manipulation is illustrated. Note that only important parts of the implemented API are presented here. The kinematic manager collects available joints and links as static attributes from a data set so that the developers can easily manipulate them and commit all the changes immediately. For the fundamental data model, besides the 5 types of classes mapped from the ontology (Figure 7), an enumeration set is defined for kinematic joint type. In STEP p105 ed2, 9 types of low order pairs are defined according to different combinations of degrees of freedom. Two most frequently used types are presented here as examples: prismatic and revolute. Getters, setters and constructors for all the data model classes are not displayed in the diagram to simplify the illustration, but they should be applied properly according to the real environment of the CAx development. For instance, in a standard STEP kinematic model, the placement of each link in each joint is defined in a link frame i.e. the coordinate system of the corresponding component of a link. However, the placements of a joint may be defined in the global coordinate system in some software, e.g. Siemens NX. Therefore, there should be constructors of the joint defined for both local and global coordinate systems.

The class diagram provides an instruction to implement and utilise the API. The following section will present how to use the API of this example in different cases.

6 Test cases

STEP Toolbox comes into being to meet the requirement of common CAx developers for friendly experience in STEP implementation. A programming interface is the core part of the solution. In this section, kinematic translations are developed and integrated with Siemens NX and CATIA. Two NX integrations are developed in-house to make a comparison for CAx integration development between with and without STEP Toolbox. The CATIA integration is developed externally by industrial project partners with the toolbox API. It validates the demanded benefits of this architecture, i.e. skipping the learning process and shortening the development process.

6.1 Kinematic translation for Siemens NX

The initialisation of Siemens NX integration development is based on STEP AP214 (Li et al., 2011), from which the focus changes to STEP AP242 as the new standard evolves. This previous application demonstrates the feasibility to develop the CAx-integrated applications for kinematic mechanism data representation and exchange with STEP standard. After that, a project partner, a Swedish vehicle company, need a fast and easy way to establish standardised kinematic translations integrated within their own CAx systems. Therefore, STEP Toolbox is designed and tested in-house with Siemens NX. To make a concrete conclusion of this solution, two versions of kinematic translators are created and compared for Siemens NX based on AP242: without and with STEP Toolbox. The system architecture of the first version (Figure 9) is similar to the previous research (Li et al., 2011).

The system integration is performed by the combination of NX Open API and JSDAI API. The former is used to generate the AP214 data set of geometry via the NX native translator and to collect the data set of kinematic mechanism. JSDAI API can be used to perform data integration of geometry and kinematics. Besides, JSDAI as a software package provides the service to compile a schema to a programmable dictionary library.

For data integration, mapping between kinematic mechanism of Siemens NX and STEP AP242 should be achieved properly. As the AP214 representation of assembly and geometry is compatible with the AP242 schema, this part of the AP214 data sets exported from CAD software can be directly processed by the SDAI dictionary of AP242. Hence, this development need not deal with cross-AP schema mapping. To accomplish data integration, the knowledge requirement on STEP for the developers is divided into two parts: schema processing and data set processing. At first, developers who use early banding programming need compile the regarding schema of certain APs, e.g. AP242 in this case. If great differences exist between two or more APs, more consideration about the data mapping and integration should be made by developers for compiling schemas and other interaction processes. Late binding does not require schema compilation but it requires clearer understanding about schemas and more carefully programming because entities and attributes are referred to with hard-coding. On the other hand, API based on SDAI for certain programming language should be used for early or late banding development to process the data set. In this case, JSDAI is chosen as the API to manipulate the kinematic mechanism with early binding underlying the compiled AP242 SDAI dictionary.

With STEP Toolbox, the integration implementation becomes simpler as shown in Figure 10. This system can be developed by programmers without any knowledge on the STEP standard. The schema of each relevant AP is compiled into dictionary and encapsulated within the toolbox API so that users even need not know the existence of APs, SDAI, schemas, etc. Data sets to be initialised, exported, and closed are controlled by developers with simple interfaces which compress all relative SDAI operations about sessions, models, repository, and transactions. All manipulations of data sets are performed with programmer-friendly data models and managers so that the choice on early or late binding is eliminated as well.

Besides, as this integration system separates the program interacting with Siemens NX from the STEP model updater by the data butter, it will be easily migrated to other CAx systems with APIs for any programming language.

6.2 Kinematic translation for CATIA

This section shows how the industrial developers utilise the STEP Toolbox in their own CAx environment. The simplified system design in Figure 10 is migrated to a different CAx system, CATIA, and work with a different CAx API, CATIA VBA.

To share the STEP Toolbox API, Javadoc and sample codes are two important documentation methods. For any Java-based API, Javadoc is an important tool to provide a detailed instruction on each element with a structured HTML (Hyper Text Markup Language) framework. However, it is still hard to start a “Hello world” programming just from Javadoc. Therefore, the sample codes are provided as compensation, from which the users can find how to initialise, modify, export, and close a STEP data set in a practical way.

The kinematic translator for CATIA based AP242 is implemented by developers in industry with the system architecture as below (Figure 11). Similar to the implementation for NX, the CATIA model processor is coded with the CATIA VBA API. Using STEP Toolbox API, a Java program, the STEP model editor, is used to manipulate the STEP data set. A data buffer in the format of CSV (Comma-Separated Values) is designed as a bridge between the VBA script and the Java program. The STEP AP214 translator of CATIA can be invoked by the CATIA VBA script to output the AP214 file with geometry. Then, STEP Toolbox API is able to integrate kinematic data with geometry and generate an AP242 file.

Until the end of this implementation, the industrial developers have not learned any detailed information from the STEP standard documents. As any other normal programming tasks, Javadoc and sample codes are only things to look up. Compared with traditional SDAI way, STEP Toolbox is proved as a better tool for STEP implementations.

7 Conclusion

STEP is potential in its comprehensive capability of product data modelling and exchange, but this capability is also an obstacle to make its documentation understandable by CAx developers. STEP Toolbox aims to provide a friendly solution for developers in this area. The architecture design of STEP Toolbox focuses on two aspects: structure and behaviour. The structure of the toolbox describes the way to encapsulate the STEP data model and to divide it into modules. The behaviour design provides an interface documented in a programmer-acceptable way that industrial practitioners are able to easily accept and utilise. Two in-house developments and a collaborative implementation with industry are performed to validate this solution in the areas of academy and industry. According to preliminary observation of normal developers, a 2-month learning period for STEP-related topics can be easily skipped by the adoption of STEP Toolbox. The implementation process is changed and becomes more simplified and familiar for programmers.

The successful test of STEP Toolbox in industry is a motivation for the standardisation of industrial product data modelling and exchange based on the STEP standards family.

Technically, it can avoid inconsistent instantiation because the instantiation is only done by the expert producers of the STEP Toolbox API. The tasks of programmers are simplified so that they can focus on other issues with more priority. For industry, the resources to be spent on STEP implementation are reduced and easier to predict, since the development process becomes as same as normal software developments.

The STEP Toolbox proposed in this paper is still in the prototype stage. The test cases have demonstrated its practical application, but further development is necessary to turn this prototype to a mature product in the future. This paper describes what the STEP Toolbox is, who its end users are, and how it can be used. However, there are still other practical questions remained: who should produce the STEP Toolbox and how the STEP Toolbox should be developed. Generally, it should be developed according to various practical requirements from the CAx system designers, developers, and common users. This will require more precise understanding of the lifecycle characters of different types of products. Developer-oriented usability analysis is helpful for its further improvement. Tasks of different communities in this area, i.e. academy and industry, should be clarified to achieve mutual benefits in the future.

References

Burkett, W. and Yang, Y. (1995) 'The STEP Integration Information Architecture', *Engineering with Computers*, Vol. 11 No. 3, pp.136 - 144.

Campos, J.G. and Xu, X.W. (2010) 'STEP-NC-compliant machine automation to support sawblade stone-cutting machining', *International Journal of Manufacturing Research*, Vol. 5 No. 1, pp.58 - 73.

Danner, W.F., Sandford, D.T. and Yang, Y. (1991) *STEP (Standard for the Exchange of Product Model Data) Resource Integration: Semantic & Syntactic Rules*. NIST National Institute of Standards and Technology. NISTIR 4528. ISO TC184 SC4 WG5 N10.

Feeney, A.B. (2002). 'The STEP Modular Architecture', *Journal of Computing and Information Science in Engineering,* Vol. 2 No. 2, pp.132 - 135.

Goh, A., Hui, S.C. and Song, B. (1996) 'An integrated environment for product development using STEP/EXPRESS', *Computers in Industry*, Vol. 31 No. 3, pp. 305 - 313.

Hedlind, M., Klein, L., Li, Y. and Kjellberg, T. (2011) 'Kinematic structure representation of products and manufacturing resources', in *Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology*, Athens, Greece, pp. 340 - 347.

Jardim-Gonçalves, R., Olavo, R. and Steiger-Garção, A. (2005) 'The emerging ISO10303 Modular Architecture: In search of an agile platform for adoption by SMEs', *International Journal of IT Standards and Standardization Research*, Vol. 3 No. 2, pp. 82 - 95.

Khaled, A., Ma, Y-S. and Miller, J. (2010) 'Feature and Product Markup Languages in service-oriented CAX collaboration', *International Journal of Manufacturing Research*, Vol. 5 No. 1, pp. 87 - 101.

Klein, L. (2000) *JSDAI -- the Synergy of STEP and Java™ Technology*. http://doc.jsdai.net/jsdai_doc/tutorial/tutorial.pdf (Accessed 31 March 2013).

Ko, A.J. Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M.B., Rothermel, G., Shaw, M. and Wiedenbeck, S. (2011) 'The state of the art in end-user software engineering', *ACM Computing Surveys*, Vol. 43 No. 3, pp. 21:1 - 21:44.

Lau, L. and Jiang B. (1998) 'A generic integrated system from CAD to CAPP: a neutral file-cum-GT approach', *Computer Integrated Manufacturing Systems*, Vol. 11 No. 1-2, pp. 67 - 75.

Li, Y., Hedlind, M. and Kjellberg, T. (2011) 'Implementation of kinematic mechanism data exchange based on STEP', in *Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology*, Athens, Greece, pp. 152 - 159.

Li, Y., Hedlind, M. and Kjellberg, T. (2012) 'Kinematic error modeling based on STEP AP242'. Paper Presented at the *1st CIRP Sponsored Conference on Virtual Machining Process Technology*. 28 May - 1 June 2012. Montreal, Canada.

Loffredo, D. (1999) *Fundamentals of STEP implementation.* http://www.steptools.com/library/fundimpl.pdf (Accessed 31 March 2013).

Lui, M., Gray, M., Chan, A. and Long, J. (2011) 'Enterprise Application Integration Fundamentals', in Lui, M. et al, *Pro Spring Integration*, Apress, New York, pp. 1 - 14.

Maier, F. and Stumptner, M. (2007) 'Enhancements and Ontological Use of ISO-10303 (STEP) to Support the Exchange of Parameterised Product Data Models', in *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*, Rio de Janeiro, Brazil, pp. 433 - 440.

Mokhtar, A. and Houshmand, M. (2010) 'Introducing a roadmap to implement the Universal Manufacturing Platform using Axiomatic Design theory', *International Journal of Manufacturing Research*, Vol. 5 No. 2, pp.252 - 269.

Pratt, M.J. (2001) 'Introduction to ISO 10303—the STEP standard for product data exchange', *Journal of Computing and Information Science in Engineering,* Vol. 1 No. 1, pp. 102 - 103.

Pratt, M.J. (2005) 'ISO 10303, the STEP standard for product data exchange, and its PLM capabilities', *International Journal of Product Lifecycle Management*, Vol. 1 No. 1, pp 86 - 94.

Sauder, D. and Morris K.C. (1995) 'Design of a C++ Software Library for Implementing EXPRESS: The NIST STEP Class Library', in *Proceedings of EUG '95 — The Fifth EXPRESS Users Group Conference*, Grenoble, France.

SCRA (2006) *STEP application handbook, ISO 10303, version 3* [online]. http://www.uspro.org/documents/STEP_application_hdbk_63006_BF.pdf. (Accessed 31 March 2013).

Tassey, G. (1999) *Interoperability cost analysis of the U.S. automotive supply chain—Final report*. [online] RTI project number 7007-03, Research Triangle Institute. https://www.rti.org/pubs/US_Automotive.pdf. (Accessed 31 March 2013).

Tolio, T., Ceglarek, D., ElMaraghy, H.A., Fischer, A., Hu, S.J., Laperrière, L., Newman, S.T. and Váncza, J. (2010) 'SPECIES—Co-evolution of products, processes and production systems', *CIRP Annals - Manufacturing Technology*, Vol. 59 No. 2, pp. 672 - 693.

Valilai, O.F. and Houshmand, M. (2010) 'INFELT STEP: An integrated and interoperable platform for collaborative CAD/CAPP/CAM/CNC machining systems based on STEP standard', *International Journal of Computer Integrated Manufacturing*, Vol. 23 No. 12, pp. 1095 - 1117.

von Euler-Chelpin, A. (2008) *Information modelling for the manufacturing system life cycle*. PhD Thesis, Department of Production Engineering, Royal Institute of Technology, Stockholm.

Xu, X.W., Wang, H., Mao, J., Newman, S.T., Kramer, T.R., Proctor, F.M. and Michaloski, J.L. (2005) 'STEP-compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration', *International Journal of Production Research*, Vol. 43 No. 17, pp. 3703 - 3743.

Xu, X.W., Wang, L. and Newman, S.T. (2011) 'Computer-aided process planning – A critical review of recent developments and future trends', *International Journal of Computer Integrated Manufacturing*, Vol. 24 No. 1, pp. 1 - 31.

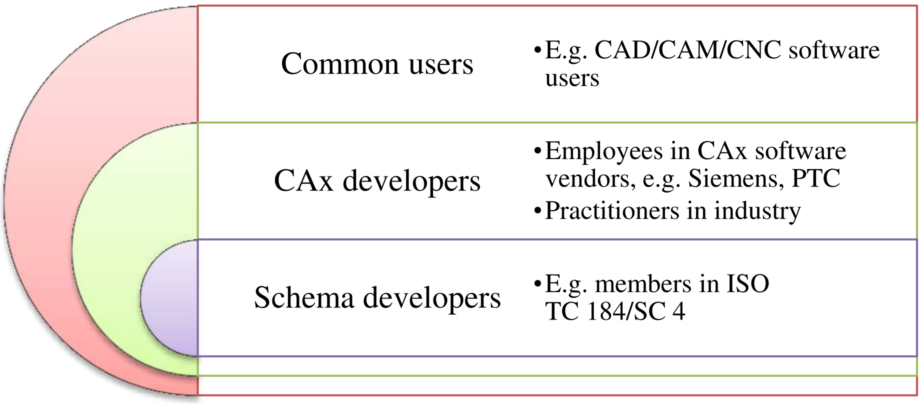

Figure 1. Three levels of STEP users

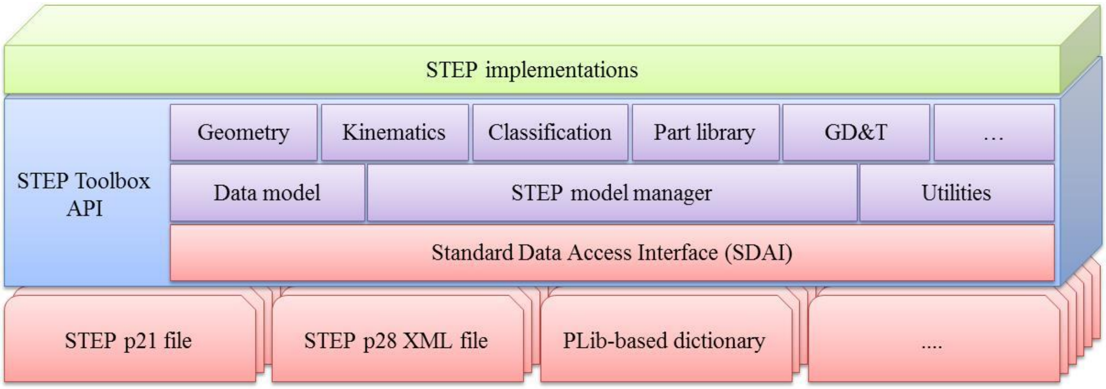

Figure 2. The architecture of STEP Toolbox

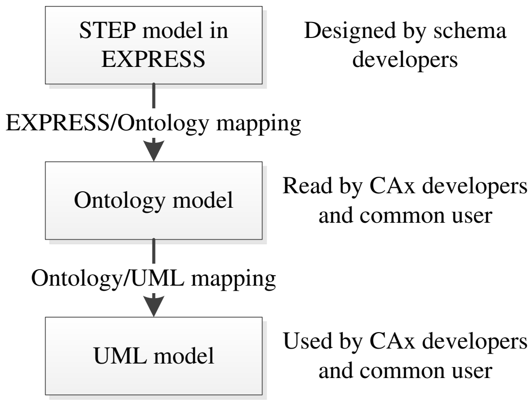

Figure 3. EXPRESS/Ontology/UML mapping

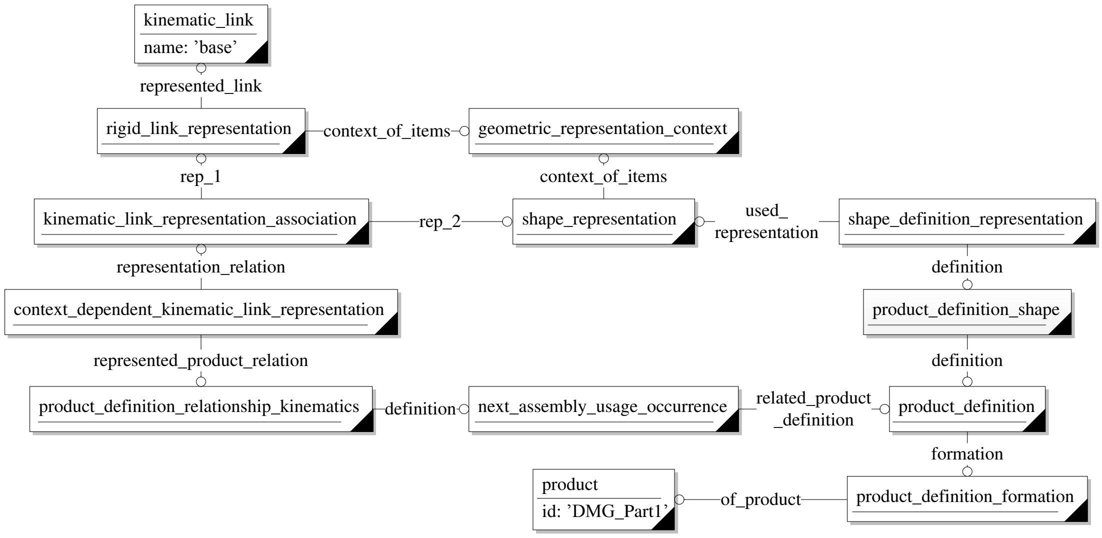

Figure 4. Information modelling of a kinematic link in AP242

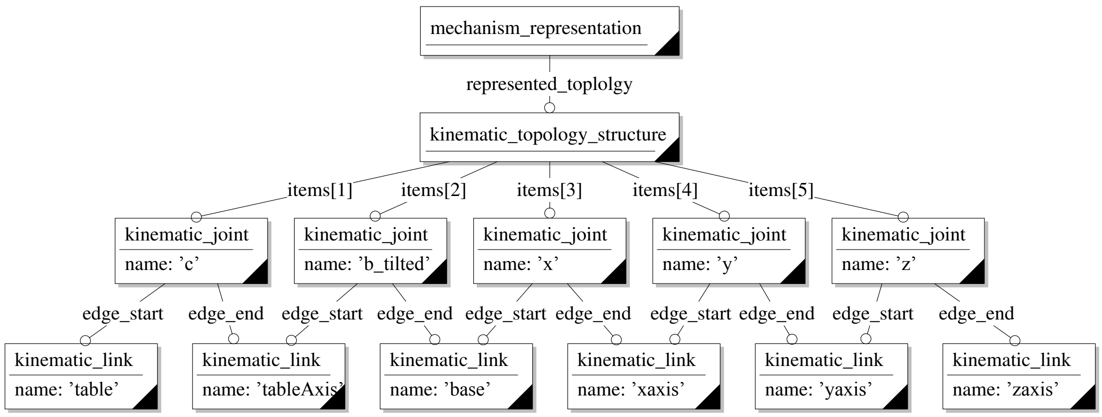

Figure 5. Kinematic topological structure of a 5-axis DMG70 machine in AP242

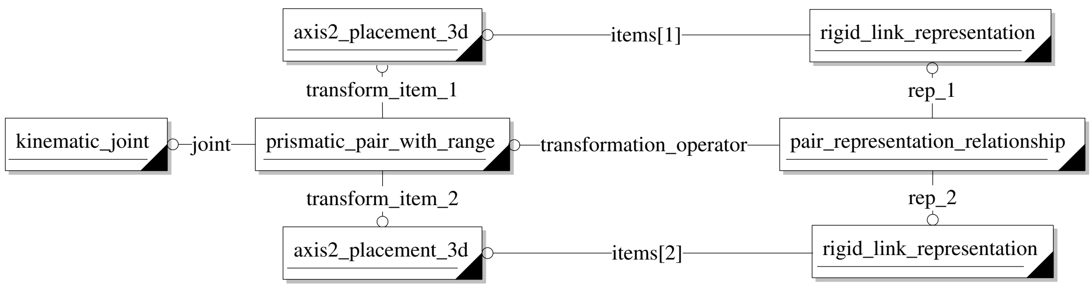

Figure 6. Information modelling for geometric aspect of a prismatic pair

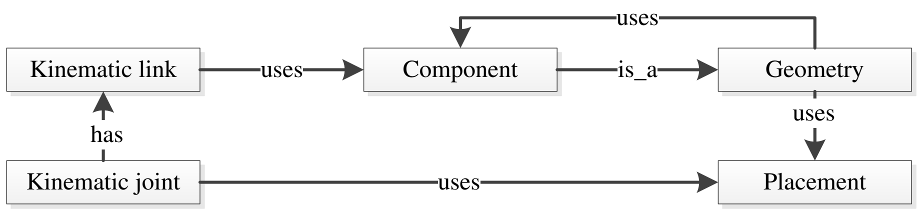

Figure 7. A simplified ontology model for kinematics

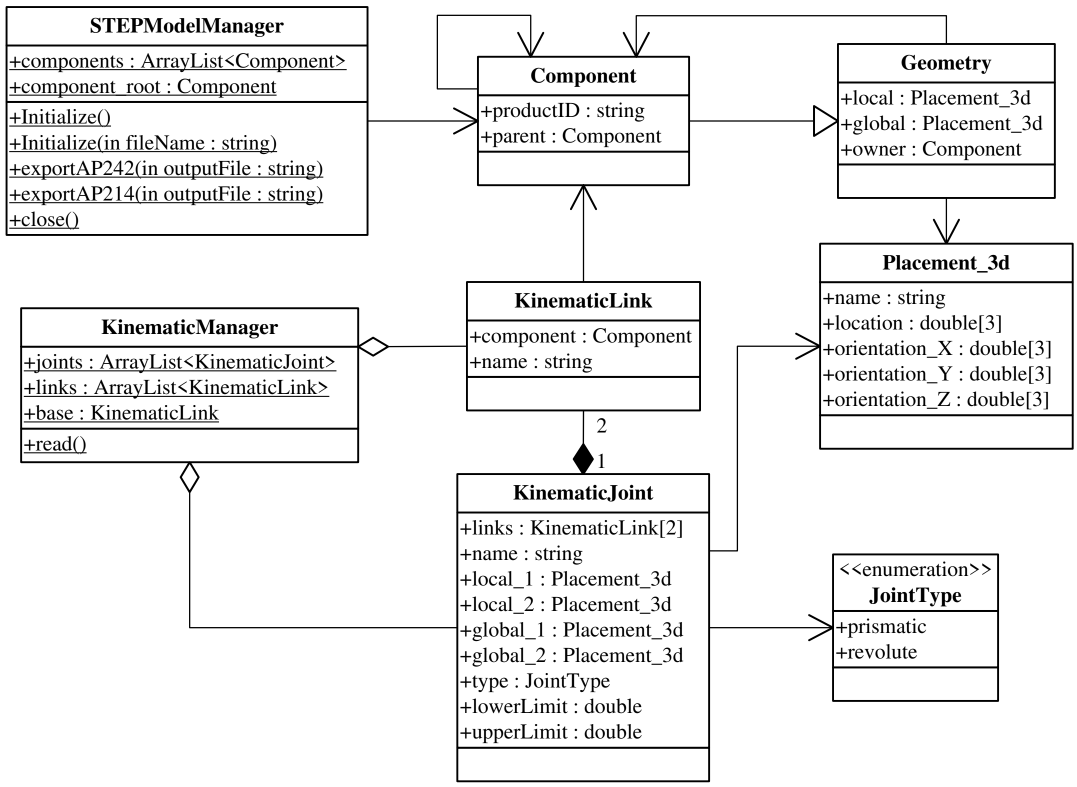

Figure 8. A simplified UML class diagram for kinematics in STEP Toolbox

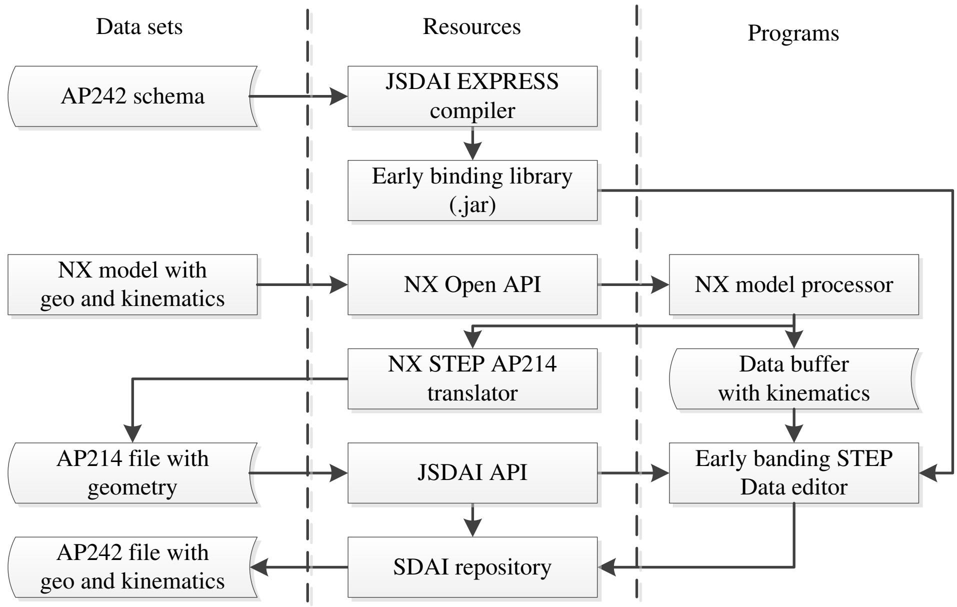

Figure 9. System architecture of STEP kinematic translator for NX without STEP Toolbox

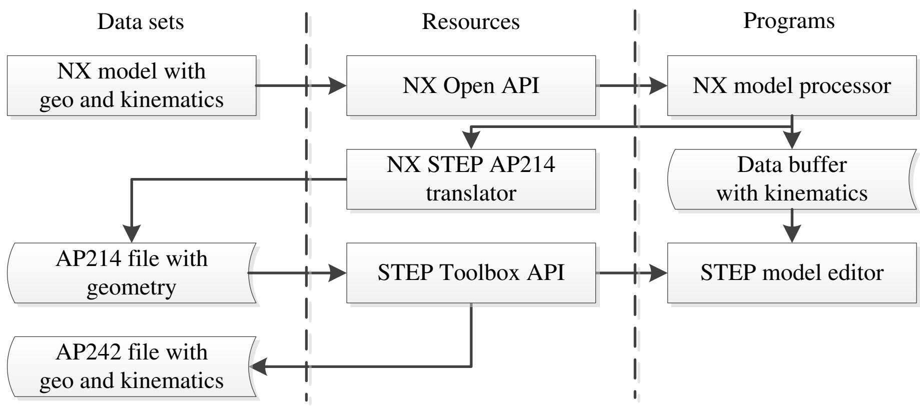

Figure 10. System architecture of STEP kinematic translator for NX with STEP Toolbox

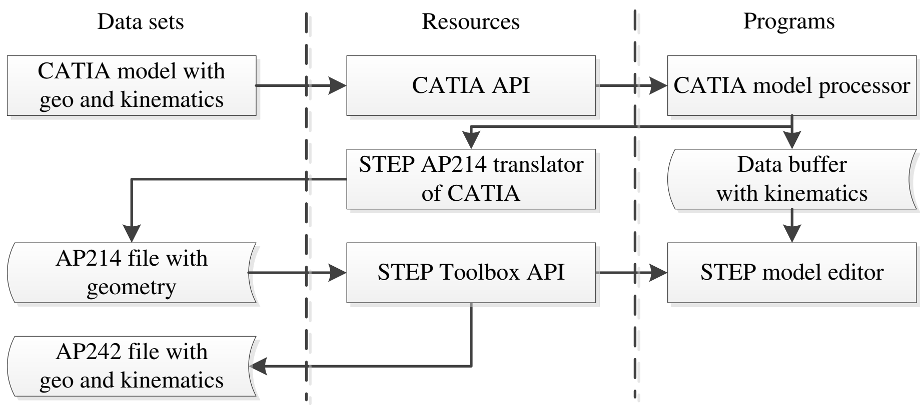

Figure 11. System architecture of STEP kinematic translator for CATIA with STEP Toolbox

Table 1. Examples of STEP development tools

<table>
<tr><th>Name</th><th>JSDAI</th><th>STEPcode</th><th>ST-Developer</th><th>Reeper</th></tr>
<tr><td>Creator</td><td>LKSoft</td><td>BRL-CAD, NIST, and others</td><td>STEP Tools</td><td>EuroSTEP and NIST</td></tr>
<tr><td>Version</td><td>4.3.0</td><td>0.6</td><td>15.0</td><td>0.5</td></tr>
<tr><td>Availability</td><td>Free for non-commercial use</td><td>Open source</td><td>Commercial software</td><td>Open source</td></tr>
<tr><td>Language</td><td>Java</td><td>C, C++, and Python</td><td>C, C++, and Java</td><td>Ruby</td></tr>
<tr><td>Major features</td><td colspan="3">EXPRESS schema compiler, SDAI library, access of p21, EXPRESS-based validation</td><td rowspan="2">XML-based operation on schemas and data sets</td></tr>
<tr><td>Additional features</td><td>Plugin in Eclipse, access of p28 XML file</td><td>EXPRESS formatter</td><td>Access of IGES and DXF</td></tr>
</table>

Paper E

Li, Y., Hedlind, M., Kjellberg, T., and Sivard, G. (2013b) Cutting tool data representation and implementation based on STEP AP242, *Smart Production Engineering*, pp. 483-492, Berlin/Heidelberg: Springer-Verlag

Cutting Tool Data Representation and Implementation Based on STEP AP242

Yujiang Li[1], Mikael Hedlind[1], Torsten Kjellberg[1], Gunilla Sivard[1]

[1] Production Engineering, KTH Royal Institute of Technology, Stockholm, 10044, Sweden

{yujiang.li, mikael.hedlind, torsten.kjellberg, gunilla.sivard}@iip.kth.se

Abstract. For cutting tool data exchange in manufacturing CAx (Computer-Aided technologies), standardized representation and classification of items and properties is important. ISO 13399 (Cutting tool data representation and exchange) provides a solution to represent cutting tool data classified with an ISO 13584 (Parts Library, PLib) based dictionary. However, ISO 13399 does not support classification of shape geometry directly, which limits its use. Another limitation is representing GD&T (Geometric Dimensioning and Tolerancing) as simplified general properties, which does not fulfill high semantic precision and validation rules. This research provides a unified solution to represent cutting tool parameters integrated with geometry and dedicated properties based on STEP AP242 (ISO 10303-242 Managed model-based 3D engineering). Standardized libraries such as the ISO 13399 dictionary can be reused with the modeling approach for AP242 cutting tool representation. Software is developed to validate and demonstrate how this solution facilitates the data integration process to support CAx applications.

Keywords: Modeling, STEP AP242, Cutting tool, Classification

1 Introduction

Collaborations between enterprises put high requirements on seamless digital information exchange and sharing between software systems for industry. Standardized representation and classification of items and properties is fundamental for cutting tool data exchange between applications of PLM, CADCAM and CNC (Product Lifecycle Management, Computer-Aided Design, Computer-Aided Manufacturing, and Computer Numerical Control). With PLM applications, engineering information of cutting tools is integrated to support its development and deployment through the lifecycle. With CAD applications, cutting tool geometry, assembly structure, and relevant properties are defined in different viewpoints for production and utilization. With CAM applications, cutting tool requirements and usages are defined for operations with tool paths and cutting parameters. With CNC applications, operations are executed with input on cutting tool types, main dimensions, and tolerances on tool wear. For decades, information management has been regarded as the essence of cut-

adfa, p. 1, 2011.

ting tool management for efficient computerized manufacturing control [1]. However, differences in terminology and data format between different computer-aided software are blocking cross-system interoperability.

An unambiguous way for the cutting tool data modeling and exchange is ontology based on industrial standards. A standardized hierarchy of classes and properties is the basic structure to describe taxonomies, as for cutting tool ontology. An important contribution in this area is the dictionary of cutting tool classes and property types within ISO 13399 (Cutting tool data representation and exchange). The dictionary is standardized underlying PLib (ISO 13584, Parts Library). A data set conforming to ISO 13399 is cutting tool data representation including its classification referencing the dictionary. On the other hand, shape representation is set outside the scope of ISO 13399. If needed, geometric models in other formats, e.g. STEP (ISO 10303, STandard for the Exchange of Product data), can be referenced from the data set. This separation approach is traditional for PLM systems.

A barrier for implementation based on ISO 13399 is the multiple data sets and associated data schemas that must be managed. For instance, an ISO 13399 cutting tool data set is transferred from a tool supplier to its customer with a corresponding geometry data set of STEP AP214 (ISO 10303-214, Application protocol: Core data for automotive mechanical design processes). System developers have to establish and maintain data integration based on the two standards and find a convenient way to convey both data sets without data loss. ISO 13399-1 is mainly a subset of AP214, which makes the practical situation even worse. As a result, there are two standards with mainly equivalent schemas to be synchronized for implementation and maintenance. Besides, the ISO 13399-150 usage guidelines describe implementation differently from how it is done for STEP AP214. It results in great efforts for implementers to develop and coordinate separate implementations.

The design of schemas also makes ISO 13399 differ from STEP in representation structure. It can be observed that both standards share the same functionality of GD&T (Geometric Dimensioning and Tolerancing) representation but with different modeling approaches. In STEP, the dedicated representation schema for GD&T is a basic functionality that is preferably reused for cutting tools, and the connection to geometry can be easily established. ISO 13399 instead uses general properties without a geometric context for shape dimensions. This does not meet high requirements for data consistency and interpretation precision in future CADCAM and CNC applications. A lack of a common schema for the geometry and GD&T results in a lack of a complete context and an unambiguous representation. With the representation of e.g. diameter classified as cutting diameter, the link to a specific geometry shape element is important information required in industry.

A standardized approach to represent cutting tool information is important for modern industry. The dictionary provided by ISO 13399 is promising for a system independent cutting tool library of CAM systems, which is able to establish interfaces between various tool makers and their customers [2]. Since it fulfills its defined scope for cutting tool data representation, ISO 13399 has contributed to several researches. Kaymakci et al. [3] adopt concepts from ISO 13399 for a general prediction model of inserted cutters, where the demands for "a unified geometric, kinematic, and mechanics model" are not what ISO 13399 is able to meet solely. Helgoson and Kalhori [4] use ISO 13399 for cutting tool data exchange in the context of machining process planning, where the solid model and cutting tool parameters are separately represented with STEP format and ISO 13399. Chungoora et al. [5] highlight the problems for joints usage of standard and present an ontology-based framework to consolidate various production information standards, e.g. ISO 10303, ISO 13399, ISO 13584, and ISO 15531. Generally speaking, integrated geometry models are commonly required regarding the adoption of ISO 13399. It becomes a must that multiple standards are integrated to achieve a complete modeling solution. Therefore, a modeling approach with a unified standard architecture such as STEP will be helpful for cutting tool data exchange.

Within the framework of STEP AP214, dimensions and tolerances are defined in a specific UoF (unit of functionality). Classification of items based on PLib is supported by AP214 in another UoF, which provides association with any dictionary conforming to PLib, e.g. the ISO 13399 cutting tool library. The draft standard AP242 (ISO 10303-242 Managed model-based 3D engineering) is going to replace AP214. Added capabilities in AP242 include supports for the Geometrical Product Specifications (GPS) standard. Dedicated schemas for GD&T and external references are integrated into this protocol and can be used to represent the needed information for CADCAM and CNC cutting tool data exchange as well as PLM applications.

In this research, STEP AP242 and ISO 13399 cutting tool library are combined for the comprehensive cutting tool modeling. Products, GD&T, features, general properties, and classes are basic elements in the model. The following section introduces the standardized modeling approach based on AP242, and the mapping strategy to a UML data model for implementation. The third section presents a prototype implementation which is able to classify AP242 cutting tool data sets with the ISO 13399 PLib-based dictionary.

2 Cutting Tool Modeling

Multiple types of information regarding cutting tool data are considered in this research, which requires data integration among schemas. The standardized product generic information modeling schema in STEP is able to support association of classes and properties defined in the ISO 13399 dictionary with products, shape elements, GD&T, features, and other elements. This capability also applies for different levels of details for different user requirements e.g. geometric level of details. For instance, tool suppliers use a complete model with a high level of detail for internal data exchange, but tool customers may only need the basic cutting tool parameters. Thus, exact geometry and other detailed design requirements, which in many cases also are confidential or unnecessary, should be trimmed from the complete model for external data exchange.

Figure 1 presents a STEP AP242 data excerpt of a diameter representation associated with a face on a solid body representation. In the model, a diameter of 22.0 mm with a tolerance of 0.05 mm and -0.1 mm is associated with a face. All geometric information is represented in the established way for STEP, which is omitted in the empty block. Using the standard modeling concept of shape aspect, dimension definition, measure representation and other properties are precisely integrated.

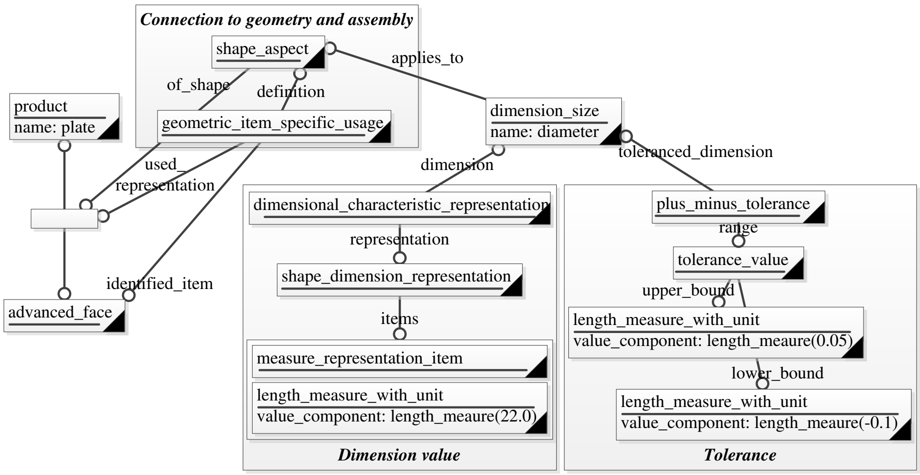

Fig. 1. Example of geometric dimension modeling in STEP AP242

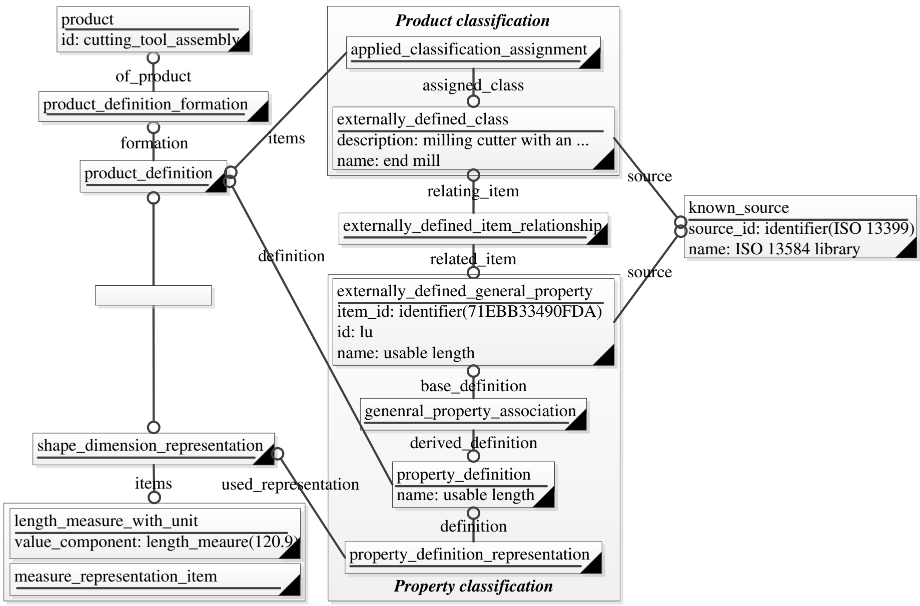

Fig. 2. Example of classification modeling in STEP AP242

Business cases may require hidden shape information, and then there will be no link between the dimension definition and the face. Nevertheless, the dimension itself is represented in a fixed way independent of existence of shape information. For a complete definition of GD&T, a geometric context is needed to establish coordinate system for the dimensions, e.g. supplemental geometry. Supplemental geometry is also known as help geometry or constructive geometry. Example data types are placements, points, curves and faces. These data types are not used to define shapes, but to support the definition of design requirements. It is a common CAD functionality also available in STEP. Recommended practices for implementing supplemental geometry are published by the CAx Implementers forum [6].

Figure 2 exemplifies the classification modeling approach within AP242. The ISO 13399 cutting tool dictionary is referenced in the model. The description of an end mill with a usable length is retrieved from the library and associated with the corresponding product and dimension.

3 Implementation

As an information modeling approach for computational applications, the result of this research should be validated with software implementation. The presented prototype software aims to evaluate the feasibility of the proposed solution and to provide a development strategy for industrial applications in the future.

The major function of the software is to classify products and other properties based on AP242 referencing the ISO 13399 cutting tool library based on PLib. The input is a p21 (ISO 10303-21, clear text encoding of the exchange structure) file of AP242 with its assembly structure, shape features, dimensions with tolerances, and other properties. Shape representation or supplementary geometry is optional. In a case study, the input STEP file is exported from a plug-in to Siemens NX 8.0 that is developed in this research project, of which the integration strategy is demonstrated in [7]. Assembly tree of cutting tool with the identification of each occurrence is displayed as a product part list breakdown. Thus, product parts, shape features, GD&Ts, and general properties can be classified according to PLib in an unambiguous way.

The development of this software reuses and contributes to a Java-based development project, STEP Toolbox. The scope of the STEP Toolbox project has a larger scope than product classification, e.g. kinematics and geometric errors [8]. In general, this project aims to significantly improve the usability of STEP standard for its major readers, CAx system developers. A complete solution to simplify STEP implementation is presented as a programming interface with high maintainability, reusability, and extensibility. The toolbox provides modularized functions to process different types of product information in engineering oriented perspective, i.e. to manage integrated geometry, kinematics, classification and other kinds of product data. Without requiring STEP knowledge, developers can produce their own standalone applications or plugins for CAD/CAM software based on the toolbox. The following data model design is a result of the research presented in this paper as well as a contribution of STEP Toolbox.

3.1 Data Model

STEP Toolbox aims to provide a friendly programming interface for CAx developers, rather than designers in a specific industrial domain. Therefore, concepts should be defined in a domain-independent way so that elements are reusable in multiple programming modules in the toolbox, e.g. the component is a widely used concept in geometry, kinematics, and GD&T. Thus, the toolbox solution adopts ontology model mapping from EXPRESS (ISO 10303-11) schemas in order to generate a proper design of computer interpretable UML model for programming. The ontology also helps advanced programmers to understand the conceptual relationship between EXPRESS models and UML models. Using the modeling principle outlined by Kjellberg et al. [9], an ontology model for cutting tool implementation has been used for mapping from AP242 in EXPRESS to a UML model (see Figure 3).

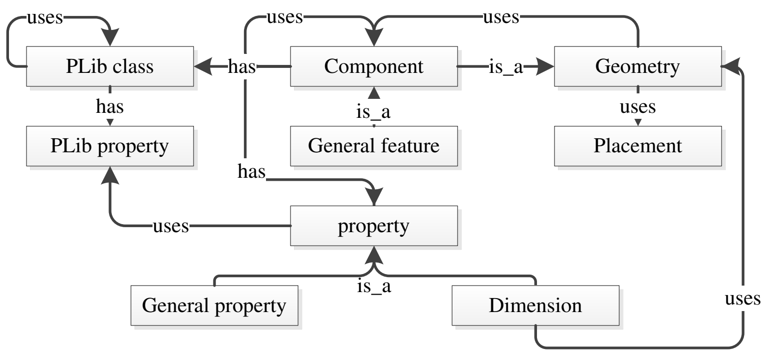

Fig. 3. Cutting tool implementation ontology

Note that the presented ontology model mainly serves the purpose of supporting computational implementation going from EXPRESS to UML. It does not necessarily express semantic relationship precisely. For example, a component actually does not own PLib class, but it can be multiply defined by several classes, which leads to such a composition relationship, e.g. a multi-functional cutting tool can be classified as an end mill and a drill. Another example is that semantically the general feature should not be defined as a sub-class of the component. However, it also can be classified and has classifiable properties, which indicates all the requirements of the general feature implementation can be met by the attributes and operations of the component. As a result, the general feature is set as the subclass of the component here. Geometry is a general concept, of which the subclasses include bodies, faces, edges, and points, besides the displayed components in the figure. Both the PLib class and the component have self-references to indicate the tree structure for presentation. Properties such as general properties and dimensions are associated with certain components or general features, which are defined semantically by PLib Property, e.g. a linear distance associated with a component is defined as usable length in a PLib dictionary (see Figure 2). PLib properties have strict ownership from certain PLib classes, which can only classify the properties of the components classified with corresponding owner PLib classes.

Programmers can use the ontology model to understand concepts easily, but a well-designed UML (Unified Modelling Language) model is fundamental for practical implementation. As a core part of UML, a class diagram is important of such a Java-based programming interface. In Figure 4, a simplified class diagram for cutting tool is illustrated. Methods of most classes are not displayed, but they are important for implementation, such as getters, setters, and constructors. As a part of STEP Toolbox, modules are divided and controlled by managers, such as the property manager and the classification manager. The STEP model manager provides basic operations for STEP data set, such as initialize, export, and close. Differences from ontology may occur for specific data model implementation. For example, both components and PLib classes have tree structures, but PLib classes also need to record a list of parents, rather than only the direct parent which is the case for component. This is caused by the inheritance of PLib properties from all relative parents.

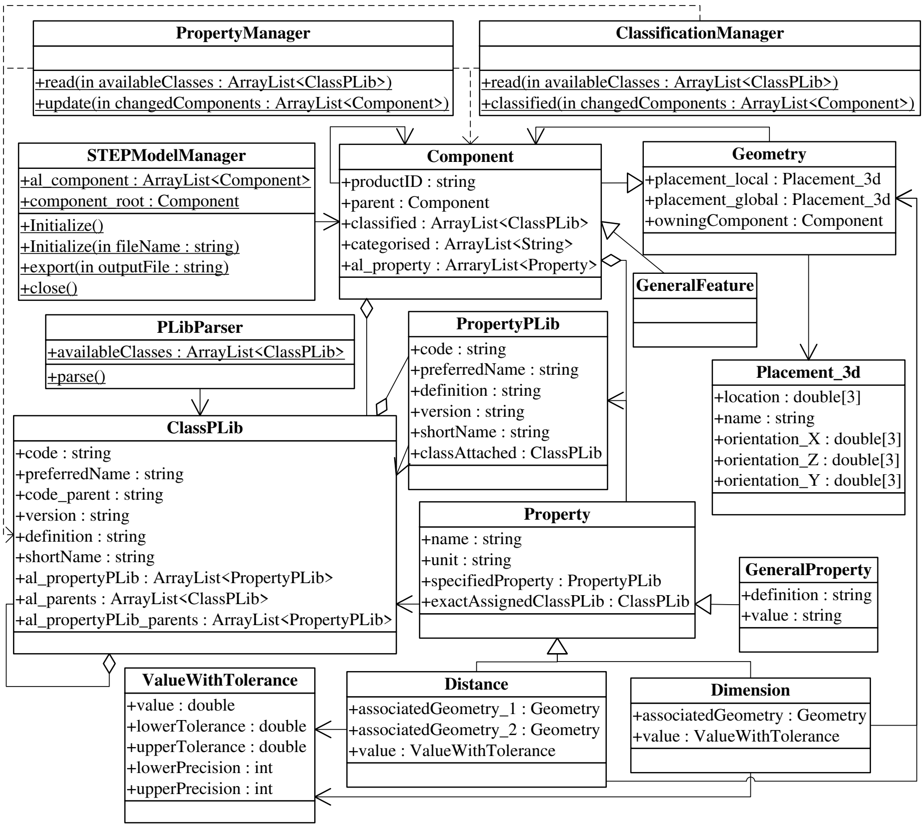

Fig. 4. Simplified class diagram for cutting tool development

3.2 System Development

An MVC (Model-View-Controller) structure design is illustrated in the Figure 5. The relative part of STEP Toolbox is integrated as the Model layer. The system starts with a file opening dialog. If the file exists and is acceptable, the Initializer triggers the STEP model manager to read the file and invokes the PLib parser to generate the standard definitions of classes and properties. The GD&T manager and the classification manager need to analyse the STEP model, read dimensions, general properties, and classification information. Then the assembly controller collects all the data and invokes the assembly viewer to display the assembly with GD&T in the graphical user interface (e.g. see Figure 6). The operation of classification is started by a selected component from assembly. A classification dialog is used to organise the PLib classes and properties in a displayable and selectable way (e.g. see Figure 7). The classifier in the controller classifies the products and properties only in the level of Java data model rather than the STEP model, and records the updated component whenever there is a user operation. Then, the exporter triggered by interface performs all the changes to the data set and exports it with specific configurations.

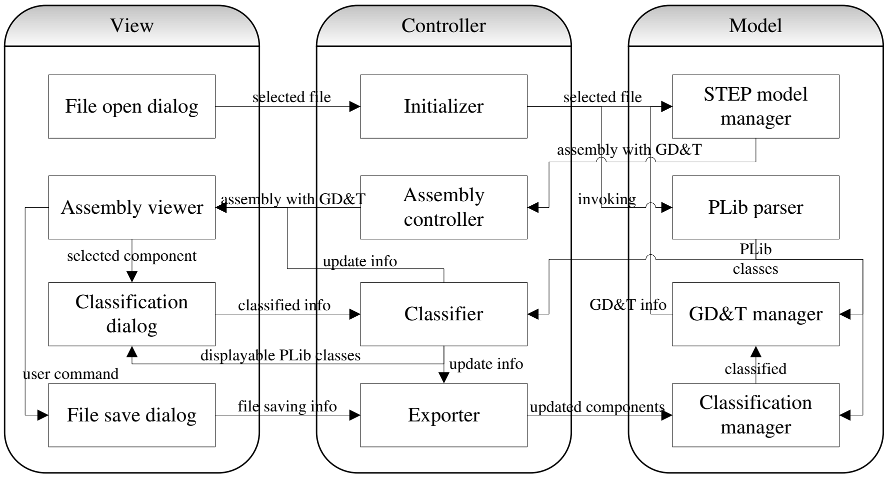

Fig. 5. MVC system design

3.3 Interface Design

The interface is designed to present and manage different elements in the data set with a clear structure. Figure 6 illustrates the basic design of the main user interface. Components are presented in a tree structure as commonly used in CAx systems. Features are displayed with a darker background color and a special category name. Both components and features can be selected to classify. The bottom table displays related properties of the selected item in the above tree table.

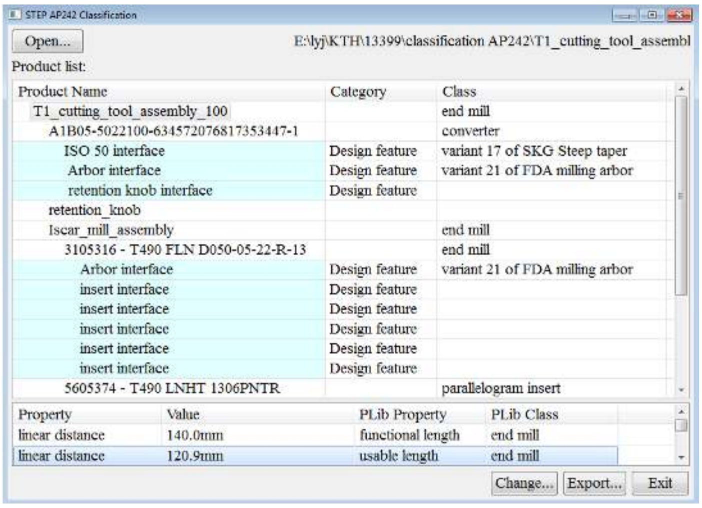

Fig. 6. Main user interface

The major function of this software is performed after the click of the "Change…" button. Then the classification dialog (see Figure 7) pops up with two lists of selection: a tree list of PLib classes that can be multiply selected, and a table list of PLib properties of the selected class. The combo list contains all available properties owned by the selected component/feature to be specified, e.g. the linear distance is specified as the function length of an end mill in Figure 7.

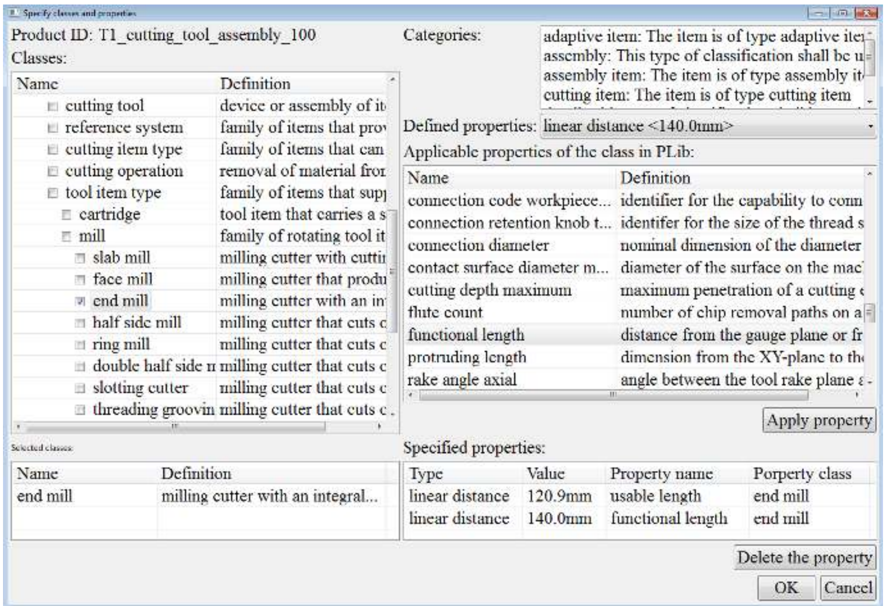

Fig. 7. Classification dialog

4 Conclusion

This research focuses on the description and evaluation of a standardized cutting tool data modeling approach. A unified computer interpretable model for data exchange and data integration is proposed and implemented. Benefits of the comprehensive representation structure of STEP AP242 are utilized and demonstrated in developed application. External standardized library, such as the ISO 13399 PLib-based dictionary, can be easily reused and associated with the shape representation. Practical data exchange is promising by integration developments with current commercial CAx systems. With the unified model, industrial practitioners can skip the barrier to coordinate multiple standards and achieve integration development smoothly.

Acknowledgments. We are grateful for the support from VINNOVA and XPRES (Initiative for excellence in production research), and for fruitful discussions with members of ISO TC184 SC4 WG3 T24 and industrial practitioners from Sandvik Coromant.

References

1. Veeramani, D., Upton, D.M., Barash, M.M.: Cutting-Tool Management in Computer-Integrated Manufacturing. International Journal of Flexible Manufacturing Systems. Vol. 4, no. 3/4, 237-265 (1992)
2. Zelinski, P.: Coming Soon: Universal, CAM-Independent Cutting Tool Library. Modern Machine Shop. Vol. 84, no. 4, 22-24 (2011)
3. Kaymakci M., Kilic Z.M., Altintas Y.: Unified Cutting Force Model for Turning, Boring, Drilling and Milling Operations. International Journal of Machine Tools and Manufacture. Vol. 54-55, 34-45 (2012)
4. Helgoson M., Kalhori V.: A Conceptual Model for Knowledge Integration in Process Planning. Procedia CIRP. Vol. 3, 573-578 (2012)
5. Chungoora N., Cutting-Decelle A.-F., Young R.I.M., Gunendran G., Usman Z., Harding J.A., Case K.: Towards the ontology-based consolidation of production-centric standards. International Journal of Production Research. Vol. 0, 1-19 (2011)
6. Bay, J., Rosché, P.: Recommended Practices for Supplemental Geometry, Release 1.0. CAx Implementor Forum (2010)
7. Li, Y., Hedlind, M., Kjellberg, T.: Implementation of Kinematic Mechanism Data Exchange Based on STEP. In: 7th CIRP-Sponsored International Conference on Digital Enterprise Technology, pp. 152-159. Athens (2011)
8. Li, Y., Hedlind, M., Kjellberg, T.: Kinematic Error modeling Based on STEP AP242. In: 1st CIRP Sponsored Conference on Virtual Machining Process Technology. Montreal (2012)
9. Kjellberg, T., Euler-Chelpin, A.V., Hedlind, M., Lundgren, M., Sivard, G., Chen, D.: The Machine Tool Model — A Core Part of the Digital Factory. CIRP Annals - Manufacturing Technology. Vol. 58, no. 1, 425-428 (2009)