IEIS

A PROPOSAL FOR A RESEARCH PROGRAM ON

INTEGRATED ENGINEERING INFORMATION SYSTEMS

TORE RISCH, PROFESSOR

INTEGRATED ENGINEERING INFORMATION SYSTEMS

TORE RISCH, PROFESSOR
Laboratory of Engineering Databases and Systems
Department of Computer Science
Linköping University
581 83 Linköping
Sweden
Phone: +46 13282541
e-mail: torri@ida.liu.se
www: http://www.ida.liu.se/~torri

1 Summary

Research area

The intention of the IEIS program is to support the education of researchers within Engineering Information Systems (EIS), an area of research which has not been established previously in Sweden. The program aims at developing national competence within the area of EIS with special emphasis on data and information management for the manufacturing industry.

Research tasks

The research area includes research on *management of information in a distributed and heterogeneous environment with many interacting data sources, systems, and users.* Methods and tools need to be developed for management of data from the various systems and tools that are used in the production process. This includes techniques to manage distributed data that need to be shared, transferred, transformed, and converted.

Positioning

The IEIS program is intended to be a *generic* program that supplies basic information management technology to support other application oriented programs for the manufacturing industry. Examples of existing and planned research programs that can be supported by IEIS include ENDREA, PROPER, and IVS.

2 Background

2.1 Distributed, heterogeneous, and dynamic information

The technological evolution has made available an ever increasing number of information systems both within and between enterprises. A trend has evolved away from central information systems towards *distributed* systems with increasing *heterogeneity* where many different kinds of information systems are used, and where an increasing amount of more or less well structured information is attainable. Not least has the development of inter and intra-nets radically increased the supply of information. However, it has also increased the heterogeneity, and the information is increasingly more *dynamic* and continuously evolving.

2.2 Relevance for information systems for the manufacturing industry

Future information systems for the manufacturing industry will work in a heterogeneous computing environment where computers are connected by high speed computer networks, and where both data and computational programs are *distributed, heterogeneous, and dynamic*. The sources of the information can be of *many kinds*, including data from conventional databases as well as data produced by a variety of tools, such as data from design and analysis systems (CAD, CAM, FEA, MSA, CDF, etc.), material production systems (MPS), product data management (PDM) systems, and purchasing systems. In addition infra-structures for future industrial information systems must support coordination of various kinds of more or less interdependent activities (often called ‘concurrent engineering’).

2.3 Need for education

Since the area of IEIS is getting increasingly important for the industry there is also a rapidly growing need for people with a substantial knowledge of the area. The development is moving very fast and the specialists within the area must be able to quickly understand new commercial and academic developments in order to know what implication these developments have on the organizations. Furthermore, new products within the IT area require substantial knowledge contents to be internationally competitive.

Swedish industry thus needs much more people with knowledge of IEIS related areas. It is expected that the IEIS program will significantly raise the Swedish competence within the area both through its educational program, its production of people with advance degrees, and by doing the state-of-the-art research which is a pre-requisite for good research education.

2.4 Need for new software technology

There are several subareas where software technology needs to be developed in order to fulfill the requirements of IEIS. For example, in a dynamic and heterogeneous environment it will be increasingly important to develop methods and systems that *combine* relevant data from several information sources and then *transform* and *present* them in a format suitable and understandable by various kinds of users and applications. There is furthermore need for methods and standards for *sharing* and *exchanging* data between systems and tools.

2.5 Information areas

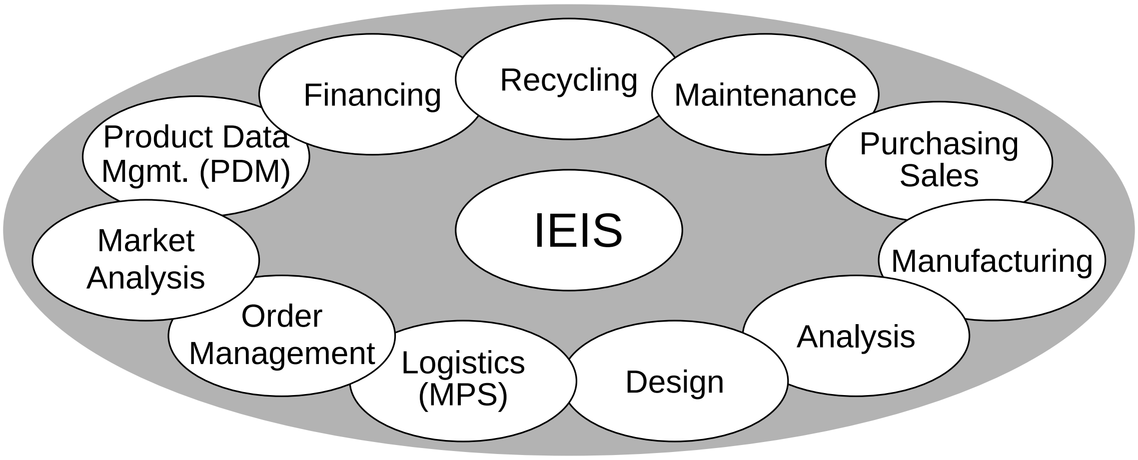

Figure 1. Examples of information areas that are supported by IEIS.

Various *information areas* in Engineering Information Systems are illustrated by the above *Figure 1*. During the product development process the company needs access to various information sources, such as marketing data, purchasing systems, design systems, analysis systems, MPS systems, and PDM systems. Different systems and information areas often have many different ways of managing data. In order to facilitate the information exchange, a number of communication and data exchange standards have been developed, from protocols for standardized interfaces between system components (e.g. CORBA, DCOM, and RMI) to standardized domain specific data exchange descriptions (e.g. STEP/EXPRESS, IGES, SGML, VRML, EDI, and CALS). These standards are important steps on the road towards simplified information management. However, we can definitely not expect all descriptions of all information to become standardized and there will therefore be increased needs for systems and methods to handle *heterogeneous information*.

3 Technological Approach

The infrastructure for IEIS should support a distributed way of working where data and processing is made available to, and coordinated for, several concurrent users. Therefore there must be mechanisms to maintain data consistency and to optimize data operations. *Modern database technology* [17][9] offers many possibilities to fulfill these requirements.

3.1 Modern database technology

A central technology area for IEIS is the use of modern database technology for the information management. Database management systems (DBMS) [3][16] provide facilities to support data modeling (meta-data management), data storage, indexing, query processing, and consistency maintenance. These facilities include a general data representation method (data model), indexed data types, query optimization, and transactions. Database technology has evolved strongly during the last decade. Today’s state-of-the-art in the area is relational DBMS that completely dominate business data processing. Recently object-oriented DBMS [2][6] have made a niche for engineering systems, mostly within the areas of CAD/CAM and CASE, where the traditional relational databases have not had the functionality and performance required. However, relational databases still have an extremely strong commercial position. The DBMS now evolve in the direction of *object-relational databases* [18] which combine the advantages of the object-oriented and relational databases. Modern DBMS furthermore permit not only data to be stored in the databases, but also programs that manipulate data (stored procedures, data blades, data cartridges, etc.), which can substantially improve performance and enable new ways to build IEIS.

3.2 Distributed and heterogeneous databases

A very important new requirement is the management of increasingly distributed [14] and heterogeneous databases [1] where applications need to access, search, and combine many different kinds of data from various data sources distributed over the computer networks. The management of distributed and heterogeneous data has received increased attention as the computer networks make it *easy to access* new data sources while *difficult to utilize* data in a productive way. New research efforts are needed for such *information management in heterogeneous environments*. This is not least the case for IEIS where data from many different kinds of systems, tools, and data representations need to be combined. A number of research projects in the US [21], but also in Linköping [7][8], Aberdeen [4], and Karlsruhe [10] in Europe, work on this problem area.

3.3 Mediators

Our approach to integrate heterogeneous data is to introduce *mediators* [20][21] between data sources and the applications and tools using them, as illustrated in *Figure 2*. The figure also illustrates that the applications are given access to several different kinds of mediators with different kinds of functionality. The enabling technology in our approach is an open and light weight DBMS that is extensible in several different ways [17] and that permits data to be stored and retrieved from several underlying data sources. We call this a *mediator database system*. The mediators can be seen a kind of a virtual database, a *mediator database,* which content is completely or partially retrieved from other mediator databases or from data sources. The rectangles at the bottom of the figure illustrate various kinds of data sources while the rectangles at the top illustrate various applications. Furthermore, in many cases an application can also be a data source for another layer of mediators, etc.

3.3.1 Mediators classes

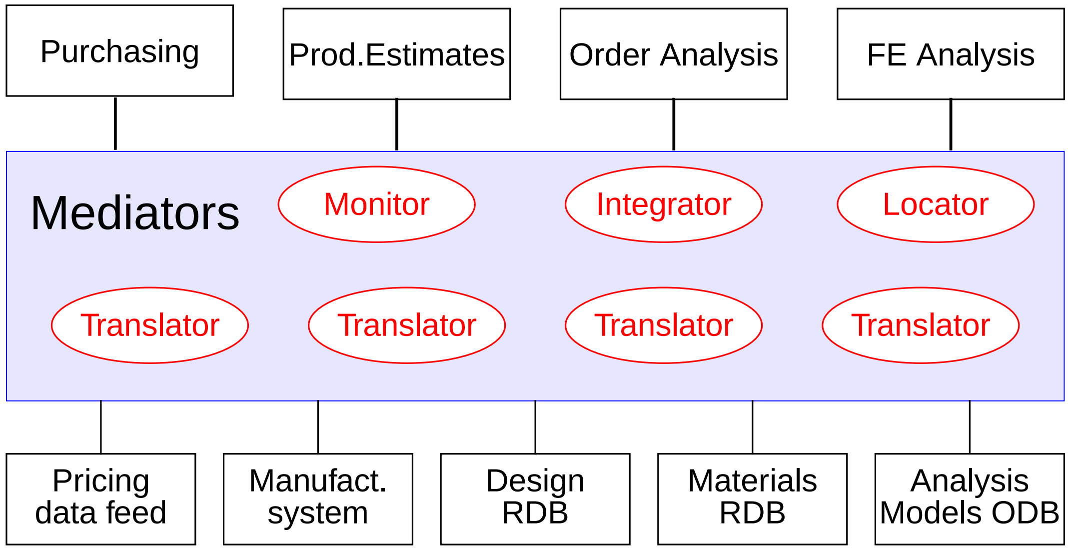

Figure 2. Various kinds of data sources, mediators, and applications

The following kinds of mediators are illustrated by *Figure 2*:

- For each data source a **translator** needs to be developed. A translator is a mediator database system that processes data from data sources of a particular class, such as relational databases (RDBs), object databases (ODBs), files, exchange formats, etc. It translates data from the class of data sources into a common format that permits processing by other mediators and applications. The translators define *views* that hide details of the data sources, that increase the abstraction level, and that permit changes in the data sources without requiring changes to mediator interfaces of the application programs and the other mediators. Since the data sources can have different abstraction levels, the translators, as well as other mediators, must have a data model that is general enough for many types of data sources. For this purpose the translators and the other mediators can be defined using an extensible object-oriented query language that permits object-oriented views to be defined and that calls user defined translation primitives.

- Data from different data sources often need to be combined. For this purpose **integrators** need to be developed. An integrator is a mediator that combines data from translators and other mediators, and that delivers the integrated data in a homogeneous way to other mediators and applications. Integrators define *integrated views* that combine and transform data from other mediators. For this purpose an extensible object oriented query language can be used for customized integration and transformation functions, and for defining high level integrated views.

- Different applications process information in different ways. **Domain databases** are mediators that are *specialized* for a particular application domain. The domain databases therefore often need domain oriented methods and algorithms to represent and process data. For example, a domain database can be specialized for representation and processing of numerical and geometrical data for engineering applications [11][13]. Domain databases can be combined with other data by defining views that integrate them with other mediators.

- Some data sources are *active* meaning that data not only can be retrieved from them, but that they can inform on their own the mediator system that data has been changed, deleted or added. **Monitor mediators** provide *active rules* that can monitor data changes, e.g. to notify when a particular part needs to be ordered, or when two designers try to do conflicting design changes.

- Many different mediators can be available in a computer network. When the number grows it may become difficult to know, e.g., what mediators provide a given service. **Locators** are mediator databases that provide *meta-information* about other mediators, such as their names, their location, and what services they provide. General *meta-queries* about the mediator databases can be sent to the locators.

3.4 Communication between subsystems

The IEIS environment should use the standard computer communication protocols for communication between subsystems and between applications and data sources. The proposed research is thus not intended to improve these infrastructures, but to utilize the existing ones.

Standard communication protocols

Examples of standard protocols for communication between systems are TCP/IP, OLE (Microsoft), CORBA, and DCOM (Microsoft) [5]. Furthermore, for WWW and Intranets the text based HTTP protocol running on top of TCP/IP has become very important. These protocols have made it substantially simpler to build distributed systems [19]. For accessing relational databases the ODBC standard [15] (which is an SQL-based interface) has become very popular. In addition each relational database vendor provides its own high-performing database interface. Some of these and other protocols should be used to communicate between subsystems in the IEIS environment.

Embedded databases

In some cases very high performance requirements may require that some mediators are run in the same address space, without using any computer communication. This is needed, e.g., for extremely fast communication between an embedded domain database and its application. For example, an embedded domain database can provide fast access to specialized data storage structures for the representation of numerical matrixes in an FEA system [13].

3.5 Interfaces to programming languages

In order to define interfaces to different types of systems, application program interfaces (APIs) need to be developed for various programming languages, such as C++, C, and Fortran.

The new programming language Java has become very popular by making it simpler to develop platform independent application programs in inter and intranet environments. Java is becoming a standard programming language for network applications [5]. It is therefore important that Java based applications can be well integrated in the IEIS environment. Java can also be an important tool for providing customized mediators. The Java world has developed its own standards and terminology, e.g. for communication between distributed systems, RMI [5], and with relational databases. JDBC [15].

3.6 References

1 O.A.Bukhres, A.K.Elmagarmid: *Object-Oriented Multidatabase Systems - A Solution for Advanced Applications*, Prentice-Hall, 1996.

2 R.G.G. Cattell: *Object Data Management - Object-Oriented and Extended Relational Database Systems*, Addison-Wesley, 1994.

3 R.Elmasri, S.B.Navathe: *Fundamentals of Database Systems*, Benjamin/Cummings, 1994.

4 P.M.D. Gray, et al: KRAFT: Knowledge Fusion from Distributed Databases and Knowledge Bases, Database and Expert System Applications, Proc. *Intl. Workshop on Database and Expert Systems Applications, DEXA' 97* (IEEE), Toulouse, France, 1997.

5 R.Orfali, D.Harkey: *Client/Server Programming with JAVA and CORBA*, John Wiley & Sons, 1997.

6 D.W.Embley: *Object Database Development - Concepts and Principles*, Addison-Wesley, 1998.

7 G.Fahl, T.Risch, M.Sköld: AMOS - An architecture for active mediators, Proc. *Intl. Workshop on Next Generation Information Technologies and Systems, NGITS'93*, Haifa, Israel, 1993.

8 G.Fahl, T.Risch: Query processing over object views of relational data, *The VLDB Journal*, Springer, No. 6, Nov. 1996.

9 W.Kim: *Modern Database Systems*, Addison-Wesley, 1995.

10 P.C.Lockemann, et al: The Network as a Global Database: Challenges of Interoperability, Proactivity, Interactiveness, Legacy, Proc. *23rd Intl. Conf. on Very Large Databases, VLDB'97,* Athens, Greece, Aug. 1997.

11 D.Maier, D.M.Hanson: Bambi Meets Godzilla: Object Databases for Scientific Computing, *Proc. 7th Intl. Working Conf. on Scientific and Statistical Database Management,* Charlottesville, Virginia, USA, Sept. 28-30, 1994.

12 K.North: *Windows Multi-DBMS Programming*, John Wiley & Sons, 1995.

13 K.Orsborn: *On Extensible and Object-Relational Database Technology for Finite Element Analysis Applications*, Linköping Studies in Science and Technology Dissertation No. 452, ISBN 91-7881-827-9, 1996.

14 M.T.Özsu, P.Valduriez: *Principles of Distributed Database Systems*, Prentice-Hall, 1991.

15 G.Reese: *Database Programming with JDBC and JAVA*, O'Reilly, 1997.

16 A.Silberschatz, H.Korth, S.Sudarshan: *Database System Concepts*, McGraw-Hill, 1996.

17 A.Silberschatz, S.Zdonik: Database Systems - Breaking Out of the Box, *SIGMOD Record*, No. 3, Sept. 1997.

18 M.Stonebraker: *Object-Relational Databases*, Morgan Kaufmann, 1996.

19 A.S.Tanenbaum: *Distributed Operating Systems*, Prentice Hall, 1995.

20 G.Wiederhold: Mediators in the Architecture of Future Information Systems, *IEEE Computer*, March 1992.

21 G.Wiederhold (guest ed.): Special Issue on Intelligent Integration of Information, *Journal of Intelligent Information Systems*, Kluwer, June 1996.

4 An IEIS Scenario

For clarifying how mediator technology can be used in an IEIS environment a possible scenario for a future IEIS environment is presented below. As previously mentioned, various IEIS applications need to produce, analyze, and change more or less overlapping sets of production information. A distributed IEIS environment therefore must permit efficient management and delivery of production information while keeping it consistent. Modern database technology plays a key role for this. However, special mechanisms must be developed for supporting engineering applications in the IEIS mediator environment. The facilities in modern DBMS to represent domain specific data and processing must also be enhanced.

4.1 Types of mediators in the scenario

Figure 3 illustrates a possible IEIS environment where various types of mediators are used for supporting a number of distributed applications acting together over a computer network. The scenario includes IEIS applications for finite element analysis FEA [1], computational fluid dynamics CFD [2], computer aided design CAD [2], product data management PDM[3], and other possible IEIS applications CAx[4]. The scenario has the data sources object-oriented database OODB[11], relational database RDB[14], and STEP-file[15]. There are domain databases for various application areas, some of which are embedded in the applications[1][3][5], and some of which are separate domain database servers[6][7][12][13]. The domain database servers can sometimes also act as data sources[12][13]. The scenario includes data access services through a number of mediators, including localization[10], translation[11][14][15], and integration[8][9]. These mediators make it possible to share, combine and exchange data having different representations and physical locations.

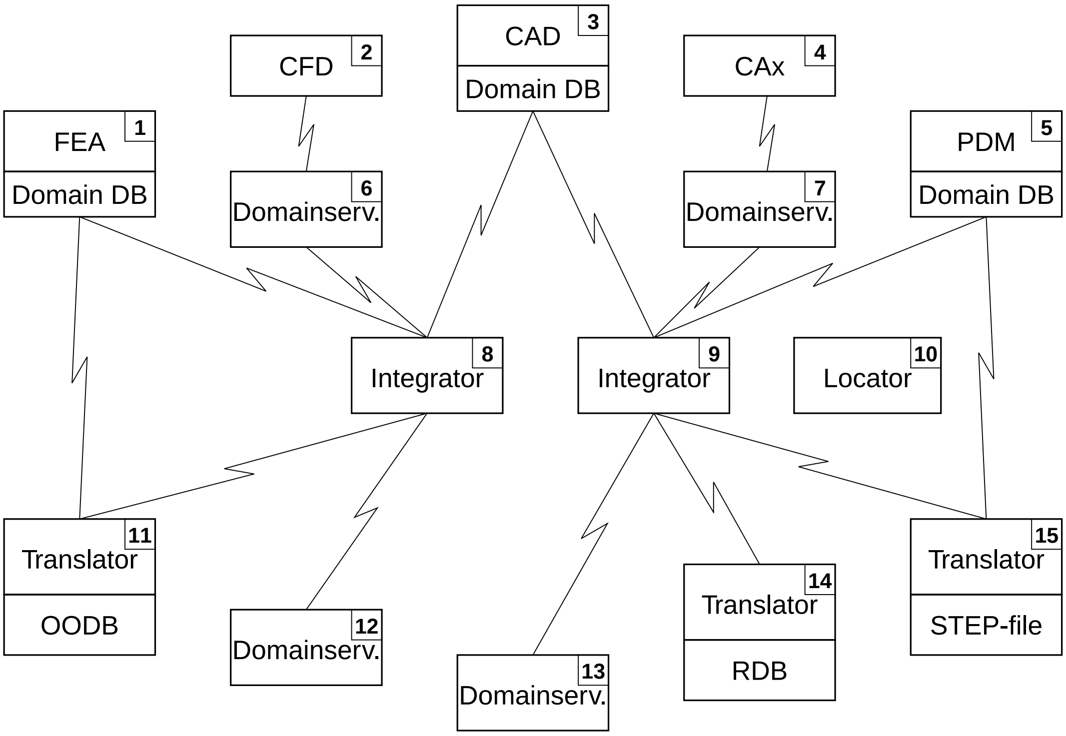

Figure 3. A scenario for a distributed IEIS environment illustrating different classes of mediators used for transforming, combining, and exchanging data between applications and data sources.

4.2 Data storage

A typical IEIS environment requires the storage of large amounts of data with varying durability. For example, a relational database RDB[14] can store business data such as components, parts, prices and suppliers. The RDB can also store measurement data from design tests, load data for load cases, and data about construction materials. An object-oriented database OODB[11] is used for long term storage of analysis models for FEA[1] that is to be reused several times. In the scenario STEP files managed by a PDM system [5] are used for communication and exchange of data with other organizations (customers, suppliers, etc.)

4.3 Domain databases

Pre-existing applications, exemplified by CFD[2] and CAx[4], access data by calling a domain database server[6][7]. A domain database server can be specialized for only a single application type[6][7] or it can be shared by several applications[12][13]. Domain database severs are useful when several databases and mediators need access to domain oriented data and operators. For example, a domain server can support geometrical models for some design details[12] or it can contain a company private materials database[13].

4.4 Embedded databases

Large applications often require high performing data management support by embedded domain databases, as exemplified by FEA[1], CAD[3], and PDM[5]. The domain databases supply database functionality embedded in the application. Embedded applications permits very high performance (in large main memories) which is necessary for many computational applications. For example, the embedded database in FEA[1] can provide specialized data storage structures for numerical matrixes that can be directly utilized by the analysis.

4.5 Translators and Integrators

An assumption in the scenario is that the mediators provide a uniform way of representing data, i.e. they provide a *common data model*. If a data source has a different data representation a *translator* is needed from the data format of the data source into the common data model, as exemplified by OODB[11], RDB[14], and STEP file[15]. When data from two or more data sources need to be combined, *integrators*[8][9] combine and transform data from the different data sources. For example, in order to compare materials it is needed to integrate[9] data about materials in a company private database[13] with materials data in an RDB[14] and from a STEP data exchange file[16].

4.6 Locators

In order to keep track of where specific types of data are located the mediator architecture uses one or several *locators*[10]. For example, a locator can tell in which mediator or data source there are analysis models for gear boxes developed by the company Ultima Design AB during the first half year of 2002. In another case one might need to know what translators and integrators are suitable for transforming and integrating data from that data source. Locators provide *meta-data* about mediators.

4.7 Use scenario

A possible use scenario could be that a design detail is designed in a CAD system[3]. As long as the design of the detail is going on, data about it is stored in a domain database embedded in the CAD system[3]. During the design materials data are retrieved from the domain server[13] and from the materials database RDB[14]. When the design of a version of the detail is completed data are transferred into the design database in the domain server[12]. The mechanical properties of the detail are analyzed using FEA[1]. Geometry data for the analysis is retrieved from the design database in the domain server[12]. During the analysis data is stored in the embedded domain database[1]. When the analysis is done the analysis model can be permanently stored in OODB[11]. Some computational results can also be fed back to the design database in the domain server[12] for further evaluation. It may further be needed to perform fluid mechanical analysis for the detail during which one loads data from a preceding termical analysis in FEA[1] whose results are in the design database in the domain server[12] and the discretized model in

OODB[11]. In this case the CFD application[2] communicates with the domain server[12] using the integrator[8] and a translator from the OODB[11].

4.8 Utilization of database technology

The scenario shows examples of how a future IEIS environment based on mediation technology could work. Using database technology for building the different classes of mediators opens up possibilities to utilize the general and powerful mechanisms of DBMS to represent and process data in the entire IEIS environment. Through the mediator architecture the IEIS environment is provided with database facilities, e.g., for optimizing data flows and data processing. In order to realize these possibilities research is needed to study how database and mediation technology should evolve in order to satisfy the demands of the IEIS environment and its applications.

5 Research Program

IEIS has many research challenges, as can be seen from the descriptions above. Below some examples are given of areas where new research is required. By providing qualified research activities in these areas infrastructures and knowledge can be built for complementing and supporting applied research and development in IEIS application areas. Sec. 11 describes some more detailed project scenarios within these areas.

5.1 Data integration

- The same and similar information (e.g. measurements, prices, design structures, etc.) can be represented in different ways in different systems. Techniques need to be developed for building *integrators* that combine and convert different data representations.

- Knowledge is needed about how related information is represented in different systems, companies, and countries. This information is needed to be able to combine data from different data sources with different structure, *semantically heterogeneous* data. Tools are needed to *describe* the properties and the structure of heterogeneous, overlapping, and complementary information.

- It should be possible to *search* for relevant data from different information sources and to *define abstractions*, views, using queries over these semantically heterogeneous data sources.

- Methods and systems should be developed to *adapt existing systems* (legacy systems) for the IEIS environment. It should be possible to access data from existing systems without a too large work effort by defining translators. The translators contain descriptions of some of the properties of the legacy systems and define interfaces to them.

- Many applications depend on consistency among data values, such as that data from different data sources are not contradictory. Therefore methods are needed to *specify and monitor the consistency* between related data. *Active databases* provide rules to maintain consistency and to alert users and adapt data when inconsistencies occur. Such active database technology should be developed to handle heterogeneous data sources.

5.2 Domain specialized database technology

- Technology should be developed to *efficiently represent and manipulate* the domain specific data and operators used in IEIS applications for design and computation. For example, engineering applications often need to manage numerical *vectors*, *matrixes*, *mathematical formulae*, *spatial data* (e.g. geometrical data), and *temporal data*. For efficient management of such non-traditional data new types of extensible DBMS should be developed where application specific data types, storage representations, indexing, and query optimization methods can be included.

- The requirements for exchanging data between applications in an IEIS environment have implications for the *architecture* of these systems. This includes the use of high performance and extensible database functionality as a *tool box* for developing IEIS applications making database functionality available to the IEIS applications. This includes, e.g., query languages, storage management, indexing, and transactions. Important enabling technologies for this are DBMS *embedded* in IEIS applications, *main memory databases*, and use of a *query language* as an alternative to traditional APIs and file exchange formats.

5.3 Meta-data management

A mediator system includes meta-data that describe properties of both the mediators and the data sources.

- *Meta-data systems* should be developed for describing properties of sub-systems and data sources, e.g. information about quality, price, location, format, and availability of various types of information. This simplifies maintenance and comprehension of the information systems. The meta-data is furthermore useful for many tools.

- The meta-database can also contain knowledge about *applications* that use the information. This makes it possible to develop, e.g., source code generators that automatically generate both database schemas and interfaces with various types of tools.

5.4 Standardization

- The information exchange can be simplified by *standardization* of the different data formats used in the IEIS environment. For example, new STEP models need to be developed for new application domains, such as CAD, CAM, CFD, and FEA. A number of such standardization efforts are going on in Sweden and internationally (e.g. STEP/EXPRESS, IGES, SGML, VRML, EDI, and CALS). These standardization activities should be surveilled and made available in the IEIS environment.

- New standards are contiguously evolving. Methods and tools therefore need to be developed for *exchange* and *retrieval* of data based on these new standards. For example, support for data transformation is being introduced into the EXPRESS-X standard. Furthermore, CORBA based standards are being developed for new application areas, and SQL, ODBC, JDBC and OQL have become popular interfaces to databases.

5.5 Data presentation

- *Method*, *tools*, and *interfaces* need to be developed for effectively use modern presentation tools to *present* and *interact* with data extracted from different data sources. Data use in the ISIS environment include both geometry and time. Engineering applications often have very high performance requirements that require efficient techniques and hardware for the user interfaces that *visualize* the data effectively. Often multi-dimensional models are used. A lot of software and hardware have been developed for presentation of such models, using multi-dimensional models presented with modern visualization tools based on, e.g., virtual reality techniques.

- Visual presentation tools often use standardized description languages (e.g. VRML). Methods should be developed to automatically *generate presentation models* in these languages from the result of a database search. Furthermore, the presentation models should be made available to the rest of the IEIS environment as a data source, e.g. for retrieving interesting substructures in a design.

5.6 Immaterial rights

It is important that the goals of the IEIS program are made available for the Swedish industry. As is common policy in Swedish universities the researcher has full rights to the inventions made within the program. However, s/he should also get assistance in making patents of the inventions, e.g. by using the companies specializing in this. When a project is financed by an industrial company an agreement will be signed for the intellectual property.

6 Scientific Collaboration

From the start the main research groups involved in IEIS Linköping University (LiU), Luleå University of Technology (LTU), and Chalmers Institute of Technology (CTH). The research is done in collaboration with a number of Swedish industries. Furthermore, there is expected to be international collaborations with non-Swedish universities. In this section the research groups and industries are described that are associated with the IEIS program.

6.1 EDSLAB, Linköping University

The Laboratory for Engineering Databases and Systems (EDSLAB) in the Department of Computer Science (IDA) at Linköping University is a research group with about 10 people supervised by Professor Tore Risch. The research is focused on modern database technology primarily for engineering applications. Mediator technology is central for the research and several members work on technology relevant for IEIS. The group was founded in the fall of 1992. So far 4 PhDs and 6 Licentiates have been produced in the group.

6.2 Computer Aided Design, Luleå University of Technology

The Division of Computed Aided Design at Luleå University of Technology has 5 persons working on IEIS related subjects. The department is managed by Professor Lennart Karlsson. The research includes work on database design and systems integration for design systems. This implies a close coordination of different parts of the design originating in the functionality of the product and where the properties of the product are defined in a product database.

Examples of existing Swedish research efforts within IEIS related areas are Ph.D. Theses by Dr Peter Jeppsson (Luleå), Dr Olof Johansson (Linköping), Dr. Kjell Orsborn (Linköping) and Dr Martin Sköld (Linköping).

6.3 Other scientific partners

The following other national groups are interested in the program:

Prof. HANS JOHANNESSON	Professor Sören Andersson
Inst. för Maskin och Fordonskonstruktion	Inst. för Maskinkonstruktion
Chalmers Tekniska Högskola	KTH

6.4 Cooperation with industry

Each of the research project in IEIS will have some industrial partner in order to give an industrial perspective for the research. Industrial contact groups will also be set up where researchers and the participating industries have a forum to discuss technical problems. Furthermore, IEIS will offer specialized courses for the industrial partners when in demand.

The following Swedish industrial organizations have expressed interest in IEIS:

ABB Atom AB, Västerås	ABB Stal AB, Finspång
ABB Corporate Research, Västerås	ABB Generation, Västerås

Hägglund Drives AB, Mellansel	Hägglunds Vehicle, Örnsköldsvik
SCANIA AB, Södertälje	SAAB Military Aircraft, Linköping
SAAB Automobile AB, Trollhättan	CelciusTech Electronics, Järfälla
Indexator AB, Vindeln	Inexa Profil AB, Luleå
Sandvik Coromant AB, Sandviken	Vattenfall AB Vattenkraft, Luleå
Volvo Aero Corporation, Trollhättan	Volvo Lastvagnar AB, Göteborg
Volvo Personvagnar AB, Göteborg	

6.5 Collaboration with other national programs

The IEIS program is intended to be a supporting program for other programs for manufacturing systems funded by the Foundation for Strategic Research such as ENDREA, PROPER and IVS. The IEIS research provides basic technology to be applied by these programs. Furthermore the applicative Complex Systems program funded by NUTEK is also serviced by IEIS.

It is expected that collaborative projects are going to be set up where some researcher funded by IEIS collaborates with one or several researcher funded by the other programs. The IEIS financed researcher is thereby mainly concentrating on studying how the IEIS technology can support the application areas studied by the other involved researchers. In Sec. 11 we show some examples of such collaborative projects.

6.6 International cooperation

EDSLAB has developed a number of international research contacts that are relevant for IEIS. Within the area of domain databases for computational applications the group has developed a close cooperation with Professor John R Williams at MIT. Dr Kjell Orsborn has received a Post Doctoral fellowship for two years of research by Professor Williams. The research of EDSLAB is well known within the international database research community and the group has many contacts within the area. Of particular interest for IEIS is cooperation with Professor Peter Gray in Aberdeen (Scotland), Professor Peter Lockemann in Karlsruhe (Germany) and Professor Witold Litwin in Paris (France). In North America the group has contacts with Professor Gio Wiederhold (Stranford), Professor Tamer Özsu (U. Alberta), Professor Marek Rusinkiewicz (MCC and Huston Univ.), Professor Sham Navathe (Georgia Tech), and Professor Sharma Chakravarthy (Gainesville, Florida). There are furthermore industrial contacts with HP, IBM and MicroSoft.

The division of Computer Aided Design in Luleå has cooperation with Professor F-L Krause at IPK (Frauenhofer Institut, Berlin) and with Professor Martin Hardwick at the Design and Manufacturing Institute at Rensselaer Polytechnic Institute (RPI), Troy, New York.

7 Graduate Education

7.1 Purpose and goals

The purpose of the Graduate Education program within IEIS is to create a new foundation for research education that increases the national competence within the area of Engineering Information Systems. The intention is to provide a means for developing a number of new courses that are given at the participating universities. The courses will broaden the national knowledge within the areas of modern technology for databases, distributed systems, design systems, analysis systems etc. It is expected that such knowledge is of utmost importance for the Swedish manufacturing industry in the future. The program furthermore provides a foundation for academic research which is important for the strategic development of the industry and for providing up-to-date education. Since knowledge in the area is very important for the industry there should be very easy for the students to find employment after finished degrees. It can furthermore be expected that such advanced education is required in order to create internationally competitive products in the IT area in the future.

7.2 Admission

Admission to the IEIS graduate school is decided by the IEIS program director. The student is interviewed by at least two active researchers within the program and assessed whether s/he is expected to be able to finish a degree within the program.

7.3 Equal opportunity

The field of computer and engineering science is heavily dominated by men. Therefore the IEIS program will encourage the participation of female students and researchers. The board will actively promote recruitment of female students in the advertisement for new positions and will assure that women participate in the selection process.

7.4 Courses

The students within the IEIS program are required to take a number of graduate courses tailored for the IEIS research. These courses are primarily given at the participating universities, Linköping University and Luleå University of Technology. Because of the distributed nature of the program, it is expected that the courses are given intensively during short periods to provide the opportunity for students from different parts of Sweden to actively participate in classes and group work. It is also envisioned that modern technology is used for distributed education and group collaborationse.

These tailored IEIS courses should be complemented with some regular graduate courses at the participating universities, after consulting the Thesis supervisor on the relevance of the courses for the program. Approximately 40% of the student time spent will be devoted to course work. To support IEIS students from different disciplines some courses may overlap with the undergraduate in other disciplines. A student need not take such courses if it can be shown that s/he has the equivalent knowledge of their contents.

Every IEIS graduate student should have the knowledge covered by the following courses:

Fundamentals of Database Systems (3p): This is database course only for those IEIS students without any previous database knowledge. Includes basic database design, relational databases, and the SQL query language. It is given at Linköping University.

Advances in Engineering Database Systems (4p): This graduate level course provides the knowledge of the state-of-the-art within Engineering Databases. Includes concepts such as Object-Oriented Databases, Active Databases, Advanced Query Processing, Extensible Databases, Spatial Indexing, and representation of Engineering Data. This course must be taken by every IEIS student. It is given at Linköping University.

Advanced Data Integration Techniques (4p): This graduate level course teaches how to build systems for data integration. Includes multi-database systems, multi-database query languages, meta-database systems, data exchange formats, and interoperability of objects. This course is given at Linköping University.

Geometric Representation and Visualization (4p): This is a graduate level course that teaches the principles of how to represent geometrical objects in computers and how to visualize these objects. This includes solid modeling techniques and virtual reality techniques. This is an obligatory course given at Luleå University.

Computer Aided Engineering (4p): This graduate level course teaches the functionality of modern CAD and CAE systems, how design parts, how to combine parts into assemblies, how to simulate various properties of the designs, how to simulate the manufacturing process, etc. This course is given by Luleå University.

7.5 Degrees

IEIS will offer the conventional advance degrees given by Swedish universities, i.e. the Licentiate degree (40p) and the PhD (80p).

7.6 Thesis supervision

Each IEIS student has an advisory committee consisting of a main advisor from one university, an assistant advisor from another university, and an industrial advisor. Since the intention is that the work on the IEIS technology is to be made in cooperation with applicative work in other programs, the main advisor supervises the generic IEIS work while the assistant advisor typically is the main advisor for the corresponding applicative project. Each student must have a personal study plan which must be approved by his/her advisory committee and revised yearly. A major revision of the study plan is made after two years to determine whether the subject is likely to be strong enough for a PhD.

8 Organization

8.1 Participants

The IEIS program is a national program with cooperation between several Swedish universities. From the start the participants are Linköping University (LiU), Luleå Technical University (LTU) and Chalmers University of Technology. The program is *generic* which means that it aims to promote research on technology for information management for manufacturing and design applications. The intention is to develop cooperative projects where students developing the core technology within the IEIS program cooperate with students working on engineering applications within the ENDREA, PROPER or IVS programs.

8.2 Board

The board for IEIS has the overall responsibility for the activities within the program. The board is appointed for a period of 3 years in cooperation with the Foundation for Strategic Research.

The following persons are proposed to be members of the IEIS board.

1.
2.
3.
4.
5.

8.3 Administration

The program has appointed a part time Program Director to manage the program. The PD is employed by LiTH, which assumes the role of employer and provides office facilities.

Program Director (acting) for IEIS is Professor Tore Risch, LiU.

The PD is responsible for the budget. The research projects as well as the budget must be approved by the IEIS board.

8.4 Advisory Committee

There is an advisory committee belonging to the program. Its purpose is to give advise concerning the academic level of the program. The following members are possible candidates as they are doing IEIS related research.

1.
2.
3.
4.
5.

9 Long term plan

There will be a continuous strive for complementing the SSF financing with funding from other sources. High priority is here given to funds from the participating companies that are assumed to complement the SSF financing, as also indicated in the budget below. The funding from other sources is expected to increase as the program evolves. Every effort will be made in increasing the external funding after 2002 to allow for a continued strong program. It is, however, highly desirable that the funding remains on the 2002 level for another six years for retaining the strength of the program.

In case the program is unsuccessful if attaining its goals there should be a phasing out of the program over a period of 3 years to allow for admitted students to finish their education.

10 Budget

The budget below assumes that the program starts in July 1998 and will get an initial funding of 5M SEK per year. This is a very tight budget that will attain only some of the proposed research goals. For a fully functional a budget of about1 10M SEK is required. That would then include more projects on distributed engineering using broad-band technology.

Incomes	***1998***	***1999***	***2000***	***2001***	***2002***
SSF	2500	5000	5250	5512	5788
University	500	800	840	882	926
Industry	0	800	840	882	926
Sum	**3000**	**6600**	**6930**	**7276**	**7640**

Expenses	***1998***	***1999***	***2000***	***2001***	***2002***
8% VAT	200	400	420	441	463
Graduate courses	100	150	158	165	174
Courses for industry	0	50	52	55	58
PhD Students	1900	4400	4620	4851	5094
Advisors	400	800	840	882	926
Administration	250	500	525	551	578
Board	150	300	315	331	347
Sum	**3000**	**6600**	**6930**	**7276**	**7640**

11 Project Outlines

In this section a number of projects are outlined that are suitable for the IEIS approach. The projects have been developed with assistance from researchers at LiU, LTU and CTH.

11.1 The Data Integrator

The purpose of this project is to develop and demonstrate a mediator architecture for the IEIS environment, according to the requirements outlined in Sec. 5.1 and Sec. 5.2. This mediator database architecture is intended to be used in the applications outlined after this subsection. Figure 4 illustrates the proposed architecture.

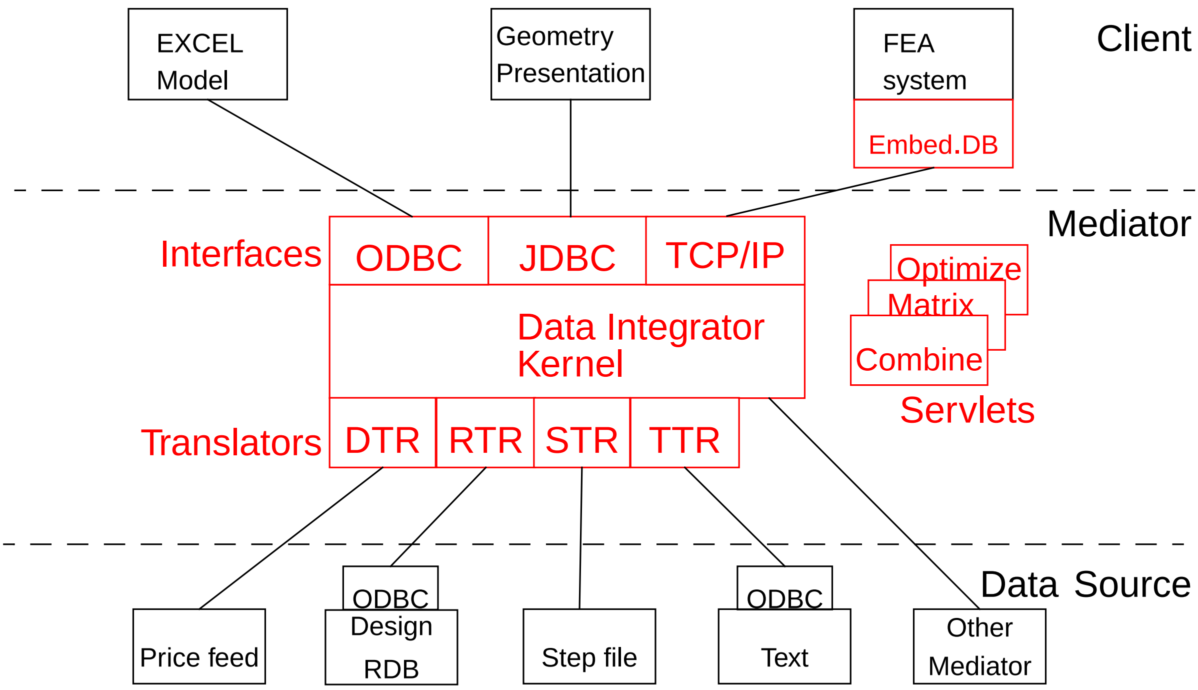

Figure 4. The Data Integrator with examples of applications, Data Sources, and Servlets

The architecture is a 3-layered architecture with applications at the top and data sources at the bottom. In the figure the applications are an analysis spreadsheet in EXCEL, a geometry presentation program in Java, and a FEA system with an embedded database. The data sources are a materials pricing data feed, a materials relational database in Oracle, a design STEP file, a text file with design documentation, and information from some other mediator. The Data Integrator runs as a middle layer server. It provides the services required for data integration, as outlined in Sec. 5.1.

Standard interface protocols on different levels (e.g. ODBC, STEP, JDBC, TCP) are used for communication with the applications and data sources which makes it easy to embed the Data Integrator in complex industrial systems. The system has *translators* for converting data extracted from the various data sources. In the figure the translators are DTR (Data Feed Translator), RTR (Relational Database Translator), STR (STEP Translator), and TTR (Text

Translator). The Data Integrator needs to be callable from other systems through several *interfaces*, e.g. ODBC, JDBC, or CORBA through which queries to views of the integrated data can be specified. Some applications (e.g. visualization systems) may furthermore require that the emitted data is converted by an *output translator* into a format (e.g. VRML) suitable for the application.

The Data Integrator is designed using a very light weight DBMS which is extensible and permit high-level object-oriented views and queries over the data sources. Object-oriented database query technology is used to customize it for the IEIS application areas and to make the system open and extensible. The extensibility is provided through servlets that are program modules that can be downloaded and connected to the mediator kernel. In the example there are servlets for customized reconciliation of data from some data sources (*Combine*), for providing customized matrix packages (*Matrix*) and for providing customized optimization rules (*Optimize*). The servlets can be written in various languages. For example the *Combine* servlet can be written in JAVA, while *Matrix* needs to be in C for high performance, and the *Optimize* servlet needs much symbolic computation and is written in Lisp.

To integrate and coordinate data from many data sources that are distributed over the network, many Data Integrators can run in different locations and communicate with each other in a client-server or peer-to-peer fashion.

11.2 Cooperative product development

(with Professor Lennart Karlsson, LTU)

The purpose of this project is to develop a system for integrated engineering of product components where several engineers participate, Figure 5. For example sheet metal profiles in a car body, such as the A-pillar, should be designed so that all the functions of the complete body are optimal with minimal weight and low price. During the design phases three different design teams are working on different parts of the design. For example, one team uses ADAMS to simulate forces on some details and their displacements to optimize the design. ADAMS is used in combination with some system for Finite Element analysis. Another team does manufacturing simulation in I-DEAS of other parts of the body, and a third team designs other details. The complete current state of the body is then visualized through a Virtual Reality system, e.g. Dive or Division, where several engineers concurrently can inspect and analyze how the body being designed currently appears. The visualization uses combinations of data from all the design teams, and data integration techniques are therefore needed to integrate the design data. The integrator contains high-level descriptions of how the design data is combined. The data integrator furthermore needs an output translator that transforms integrated data to a format suitable for the visualization program, e.g. VRML code.

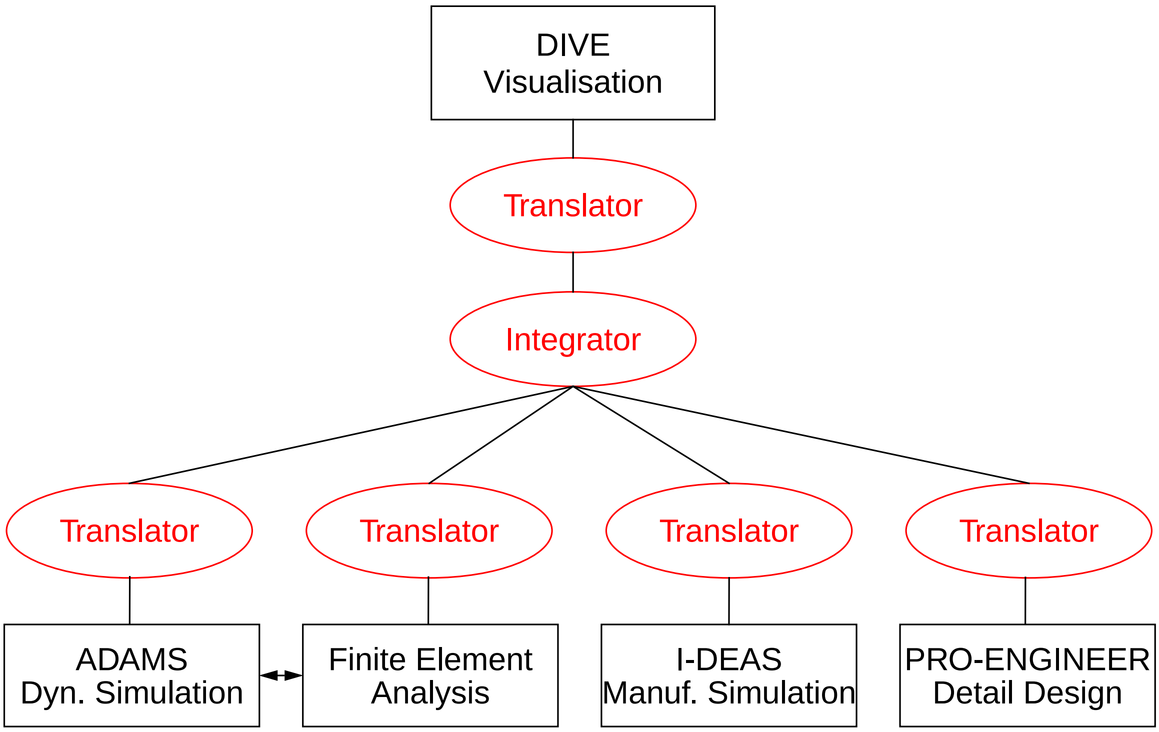

Figure 5. An example of how design data could be combined for visualization

11.3 Integrated design optimization

(with Professor Anders Klarbring, LiU)

The purpose of this project is to develop an architecture for an integrated computing environment for optimal structural design. Figure 6 illustrates two kinds of optimizations needed for this, *topology optimization* and *shape optimization*. The topology optimization needs to combine the output from the topology optimization with *geometric data*, *load cases*, and *materials data*. The shape optimization in addition needs access to the results from the topology optimization. The applications access the integrated data by querying mediators. The data sources can be of different kinds. For example the geometry data might be stored in a regular file, while the load cases and the materials database are stored in relational databases. The mediators handle the integration of these data and makes it possible to add more data sources without changing application interfaces. Also the availability of the data used in the optimizations are easily sharable with other application, since it can be accessed through high-level queries rather than hidden inside internal data representations and file formats.

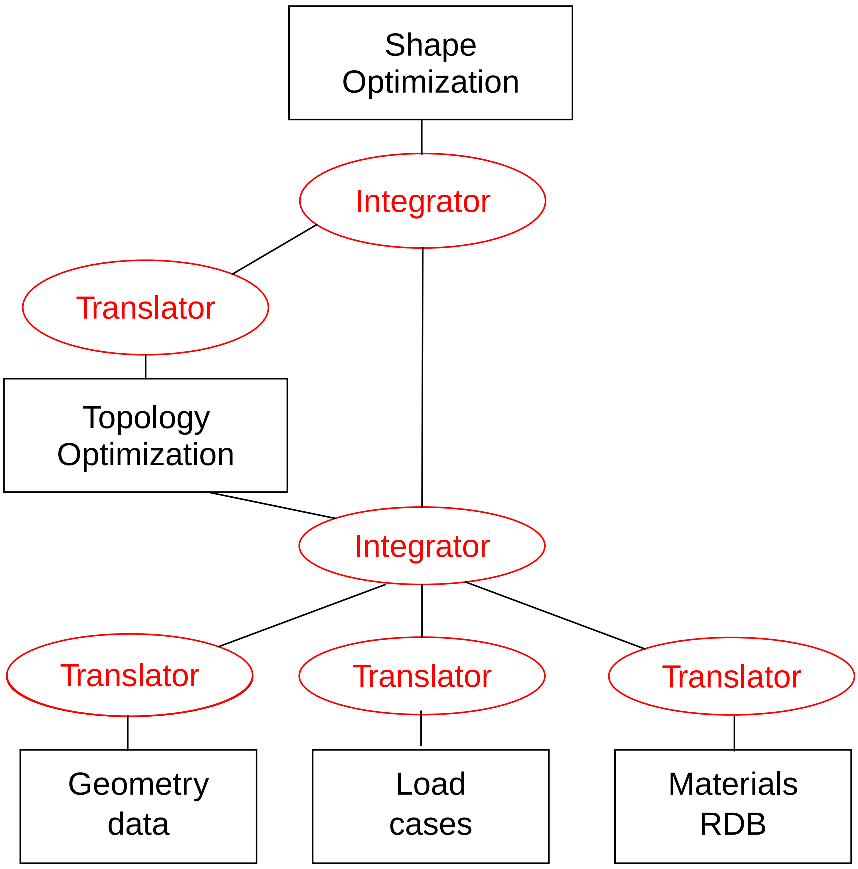

Figure 6. Data Integration for Design Optimization

11.4 Integrated dimensional management

(with Assistant Professor Lars Wennström, LiU)

The shortening of product development times requires monitoring of the behaviour of the products' individual components regarding form, fit, function, collision, etc. In particular the tolerances of individual parts need to be verified early in the production process to ensure no conflicts with the functional requirements of the final product. The manufactured products furthermore contain tens of thousands of components and hundreds of thousands of fasteners. In order to find and avoid already in the design phase such tolerance related problems, there is need for retrieving relevant data from large data sets in many different data sources within the production process. Figure 7 illustrates a proposed architecture to improve the information supply for better tolerance design. Statistical Process Control (SPC) is introduced for the different manufacturing steps to measure and collect data about how well each step in the production process fulfills various requirements. In particular measurements from Critical Geometric Characteristics are collected which are important feedback for the tolerance design as well as for R&D in general and for work on improving the manufacturing processes themselves. The design is also influenced by feed-back from the users of the final product quality, price changes, etc. We propose to use data integration techniques to provide relevant and up-to-date information to the design teams and systems. With this approach queries to select the relevant information can be submitted to the *Integrator* which knows how to efficiently extract convert the relevant data from the heterogeneous data sources though the *Translators*.

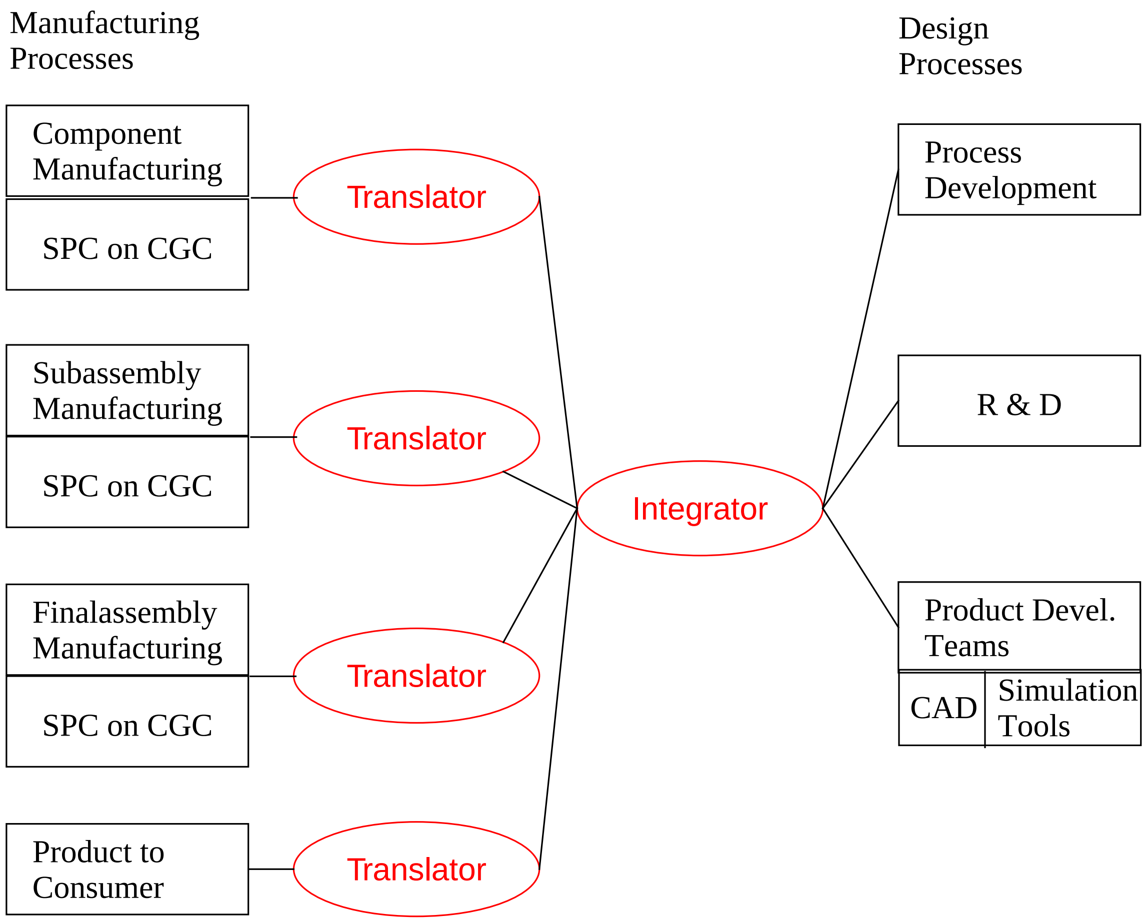

Figure 7. Data Integration for Dimensional Management

11.5 Integrated Product and Design Process Modeling

(with Associate Professor Johan Malmqvist, Chalmers UT)

The CAD/CAM/CAE systems of today features increasingly sophisticated support for geometric modeling, process planning and various kind of analyses. However, these functions are often supported by different systems leading the problems with data sharing and translation. Moreover, these models typically only describe the **end result** of the design process. For the system to support the entire design process we also need models that state the **goals** for the design, i.e. requirements and function models, and be able to link requirements objects to property objects for monitoring the state and progress of the design. The potential benefits of such a system includes simplified verification that a design meets its specification, prediction of the consequences of a design change, simplified re-use of earlier designs, support for high-level reasoning concerning the system design and support for multi-functional design teams.

The realization of such a system requires developments in the area of requirements modeling: currently requirements, functional descriptions and design decision documentation are typically collected in text documents making them difficult to extract for some kind of systematic evaluation. Systems in which requirements are managed as separate objects in a database would make this much easier. There is also a need for development of methods for tracing the consequences of requirements changes. Examples of queries are: what components need to be changed if we change requirement x?, what requirements must this component meet?

Figure 8 illustrates a product model architecture that supports the above requirements. In this scenario, the design process would start with a **requirements analysis** activity, resulting in a requirements model, that can be managed in a PDM system or a conventional database. The design team would then **synthesize** the new design using the CAD system and link the CAD model to the requirements model. The team would then **analyze** the design in order to verify that it meets the requirements. Some kinds of analysis data, e.g., weight and inertia can be derived directly from the CAD model and compared to the corresponding requirements. Other analysis data, e.g. FEM or CFD, requires the use of specialized software for computation. These software often cannot use the CAD model as it is, but require some kind of data translation. After these more advanced analyses, data may need to be translated and integrated for comparison with the requirements. This can be the basis for design changes, followed by new analyses until the all requirements are fulfilled.

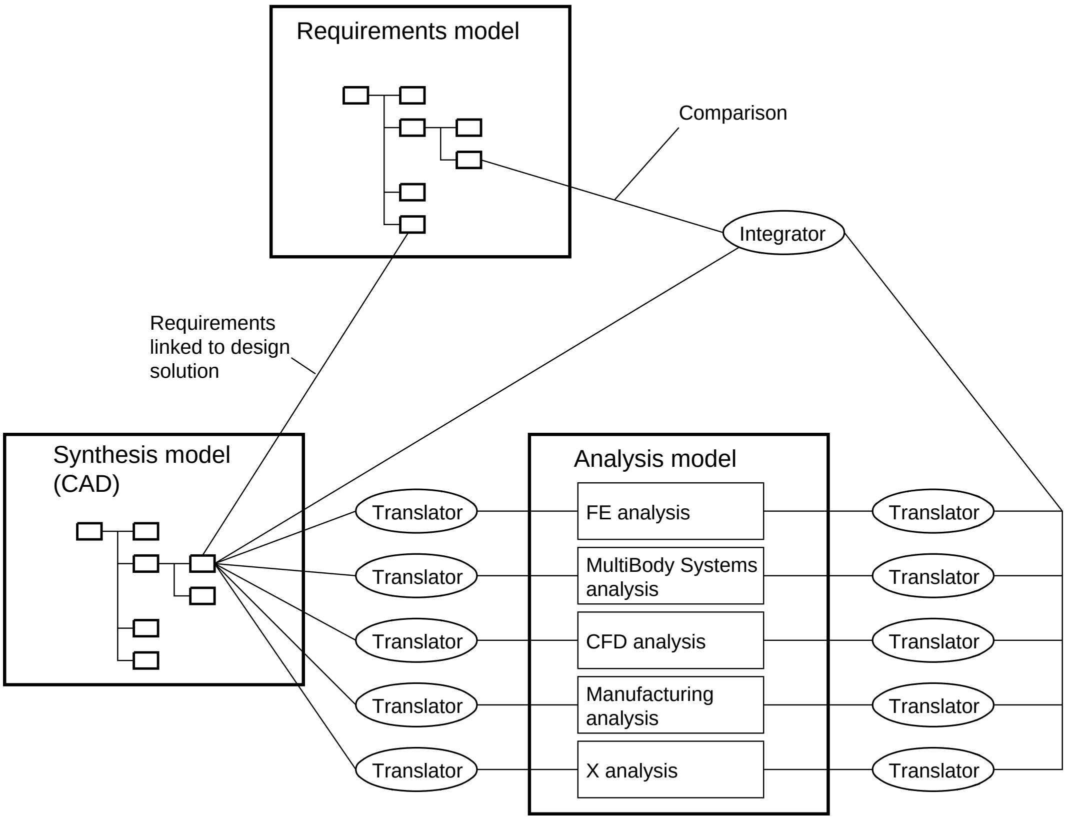

Figure 8. An integrated product model architecture

11.6 Integrated manufacturing and function simulation for the design of light structures

(with Professor Lennart Karlsson, LTU)

The goal for the project is to develop a system for integrated manufacturing and function simulation for the design of light structures, in particular for automobiles. The early concept development should be supported by verifying analyses of the manufacturing of components and of system functions. For example, a sheet metal profile, such as the structure supporting 'A-pillar', should be designed so that all the functions of the body are optimized. The component should have minimal weight and should be possible to manufacture inexpensively.
The following steps are performed during the design process:

1. Design the body to satisfy the functional specifications, but where the parts have simplified geometries, in order to simplify the subsequent steps.
2. The manufacturing process is simulated and analyzed. For example, sheet metal is formed to various shapes that are heated, bent, and hardened.
3. The steel pillar in its final stage is brought to function simulation, to be developed in this project.
4. The part and its properties are entered into a FE model of the body structure which then passes through a number of tests, e.g. crash analyses, and stiffness computations.
5. The results from the function analyses are evaluated and gives designers and manufacturers input about geometry modifications and changes to the production process.

This process is iterated a number of times to converge towards a design that both designer and manufacturers accept. The design is then refined successively w.r.t. tool geometry and process steps, etc. The methodology is about the same during the entire concept development but the tools may differ somewhat depending on
i) how detailed the geometry is,
ii) whether manufacturing simulation is necessary and how fast it should be, and
iii) how well other requirements are know.

For such a system to be realistic there is need for integration of a number of functions and for the translation and integration of various data sets.

The following kinds of data producing systems (data sources) are involved in the process:

- Geometry generators (CAD programs) are used for the design of the geometries and assemblies, e.g using I-DEAS.
- Finite element generators are used for creating FE models from the geometry models and for composing FE models into FE-structures. They are also used for creating analysis models of simulated physical tests, such as crash, torsion and vibration tests.
- From the parts geometries FE models are developed of the process steps that give the desired final geometry.
- Finite element evaluators, e.g. IDEAS, TAURUS, and VISLAB.

The following kinds of data integration facilities and data stores are needed:

- Finite element *transformers* that transfer and transform FE models and results between manufacturing and function analyses.
- Data stores for FE models and results, e.g. to store the latest state of a process.
- Data stores and translators for geometry models and histories for maintaining the product design intent, such as logging design decisions and cross-referencing analyses used to support the design decisions.
- Data stores for materials data which always need to be available during the process. The materials data base may be updated during the design process.
- Storing functional requirements, such as FE test models, requirement specifications etc. to be used as input the function simulation.

Figure 9 gives an outline of the components of this kind of system and the kinds of data integration, storage, and transformation needed. In the research project it will be investigated exactly what data transformation and integration facilities are required for the proposed scenario.

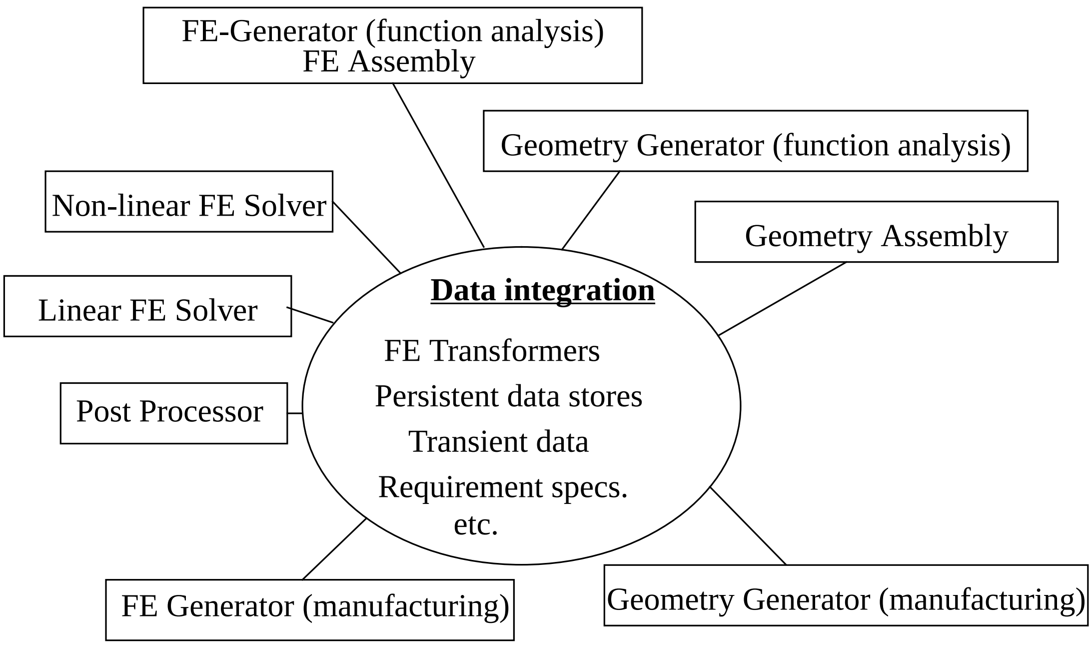

Figure 9. Data integration needs for the design of light structures