# Misconceptions of an IPDB

R. Amor and I. Faraj
*BRE, Bucknalls Lane, Garston, Watford WD2 7JR, UK*
*{trebor, faraji}@bre.co.uk, Phone +44-1923-664549, Fax +44-1923-664689*

ABSTRACT: The notion of an integrated project database (IPDB) has existed for decades. Over that time many projects have been undertaken to develop the technologies and frameworks required to implement an IPDB. Also over that time, there has been promotion of the benefits and impacts that IPDB systems will have on the industry. As there are still no industrially stable IPDB systems in existence, the industry's perception of what they are and what they can do has diverged from many of the original presentations. It is also clear that researchers and developers involved in IPDB development have many different ideas about what constitutes an IPDB and what is, or is not, possible to create. This paper aims to describe misconceptions which are growing up around IPDB systems, and presents the authors' view of reality. Consensus in this area is currently being sought through the majority opinion of the UK network of experts in objects and integration (URL-1 1999) which is run by the DETR.

## 1 INTRODUCTION

This paper aims to promote discussion on what is, and is not, possible with an Integrated Project Database (IPDB) with an attempt, at least within the UK, to draw out a consensus and common understanding for those who work in the area. With the concept of an IPDB promoted for the industry at top levels (Egan 1998) there are many new people coming into the area. The authors' perception is that this is also leading to a plethora of views and standpoints, not all of which reflect the reality of IPDB research and development.

In this paper the authors raise a number of ideas and issues where they believe there are misconceptions. A short case is argued for each of these. The authors are aware that they may have their own misconceptions about the area, and there may be important points missed from this list. This paper is being used as the starting point for a discussion within the UK network of experts in objects and integration (URL-1 1999), in order to reach consensus and generate a document on these issues for those who are new to this field, as well as government. The views of the audience and other readers are most welcome, either during the conference or through later correspondence, for inclusion in our analysis and summary.

### *1.1 Definition of an IPDB*

This initial description and definition of an IPDB is taken from Anumba and Amor (1999).

There are several views within the construction industry and the research community on what constitutes an integrated project database (which is also sometimes referred to as the shared construction project model). Some see it simply as an amorphous collection of all the information relating to a project, irrespective of the storage medium (people's heads, paper drawings and specifications, CAD files, etc.) or the method of dissemination of the project information. Others see it in terms of a single database which holds all the information on a project and which is accessible to all members of the project team. Yet others view the integrated project database as an integration of product models (which hold information relating to the building product) and process models (which hold information regarding the construction and business processes required to translate the product information into a constructed facility). These different perspectives are reflected in some of the following definitions:

- Gann et al. (1996) 'a single project database is an electronic data model to which all participants refer throughout the processes of design, construction, operation and maintenance'.
- Bjork and Penttila (1989) 'project models are conceptual structures specifying what kind of information is used to describe buildings and how such information is structured'.
- Fisher et al. (1997) 'project modelling is object modelling applied to a project and including more information than just geometry'.

Although the concept of an integrated project database may be difficult to define precisely, the above definitions focus too much on the data representation aspects and thus, are neither wholly accurate nor comprehensive. Greater insight into what constitutes an integrated project database can be gleaned from its requirements and characteristics.

### *1.1.1 Attributes, Requirements and Characteristics*

Several attributes, requirements and characteristics have been associated with the integrated project database. Many of these are reflective of the individual perspectives and biases of the authors whilst others are more robust and generic. There are also those that constitute no more than a wish list.

Anumba et al. (1997a and 1997b) see the shared construction project model or integrated project database as central to concurrent engineering in construction and vital for facilitating effective communications between project team members and between stages in the project lifecycle. They suggest that, as a minimum, it should support the following:

- individual discipline interactions with the central model;
- heterogeneous intra-discipline tools;
- configuration management;
- perpetuation of design intent and rationale across stages in the project lifecycle;
- emerging standards for information representation, interchange and interoperability;
- integration with a robust and multi-faceted project communications infrastructure;
- enhanced visualisation of design and construction processes based on multimedia, virtual and mixed reality, simulations, video, etc.
- an open architecture to facilitate extensions and customisation to suit individual project and team requirements.

Similar views on attributes, requirements, and characteristics are promoted by Construct IT (1996), Fischer and Froese (1992), Froese et al. (1996), Arnold and Teicholz (1996), Law and Krishnamurthy (1996), and Gadient et al. (1996).

These attributes, requirements and characteristics of a shared construction project model or integrated project database extend the definitions provided earlier well beyond the scope of just data modelling. They are reflective of the huge potential that many in the construction industry (researchers and practitioners alike) think is embodied within the concept of the integrated project database. Some of the general approaches being employed in the development of the integrated project database are summarised below.

### *1.1.2 Approaches to Development*

Although there is a consensus that an integrated construction project database is highly desirable for computer-integrated construction, there is far less agreement on what form it should take. This was alluded to in the discussion of definitions of the term, 'project model' or 'integrated project database'. It is also reflected in the approaches that have been proposed or adopted so far in the development of the model. Some of these approaches are briefly summarised here with references, where appropriate, to research prototypes.

1. Project Model as Reference Model - This is the approach that many practitioners seem to favour. This is based on having a 3D CAD or Virtual Reality (VR) model, which simply acts as a common reference model for the project team. In this case, the model does not necessarily hold all project information but acts as a gateway to it.
2. Centralised Project Database - This approach involves the use of a single centralised database to which all members of the project team have controlled access. The main difficulty with this approach is that the database can become very large and unwieldy with consequent maintenance and information retrieval difficulties, particularly in a multi-user environment. This approach also raises issues of ownership and control of the centralised data. This approach includes current systems that contain only project documents (Document Management System).
3. Distributed Project Database - In this approach, there is no single repository. Rather, aspects of the project database (such as those produced by each discipline) are held at various locations and accessed via a common, standard interface (such as CORBA or DCOM). This approach requires that the different applications support the standard interface, but is potentially very effective.
4. Neutral Format Project Database - A neutral format database is the core of this approach which requires that individual applications transfer information to a central project database in a neutral (STEP-based) format which can be read by other applications. This has potential for facilitating multi-lateral information interchange but requires that all applications have pre- and post- processors for effecting the bi-directional transfer of information. There is also potential for the loss of data integrity and semantics in this approach. Currently, the neutral standards required for this approach are not developed to an extent that makes it commercially feasible (Eastman and Augenbroe 1998).
5. Proprietary Approaches - In addition to the above generic approaches, there have been a number of proprietary developments which embody some features of the above. A couple of these will be given a brief mention. Fischer and Froese (1992) propose an object-oriented system called OPIS that provides for integration of a product model, a process model, a resource model and an organisation model. It also allows for objects to be classified as either project-specific or project-independent. Tah et al. (1997) describe a concurrent engineering environment for integrated design and construction (CEE-IDAC) which links CAD and project management applications using a central object-oriented database management system (OODBMS) and Microsoft's OLE/COM distributed computing standards. Knowledge-based system (KBS) techniques are also used to generate construction project programmes.

## 2 MISCONCEPTIONS

### *2.1 OO provides the solution*

The fad of the moment is objects. Object technology has been embraced in almost every area of IT and construction. This ranges from object-oriented modelling (e.g., UML, EXPRESS-G), to object-oriented programming languages (e.g., C++, Java), through to object-based CAD systems (e.g., ArchiCAD, AutoCAD Architectural Desktop). The appeal is easy to see, analogous to Minsky's frames (Minsky 1975), objects are intuitive to specialists and non-specialists alike. For example, in contrast to relational representations, where there is a formal underpinning which requires specialist knowledge to utilise (e.g., 3rd normal form), object-oriented representations require no rigorous analysis to apply. This means that object-based systems can more easily model a user's view of the world (a major criticism of relational systems whose requirements often render a user's view incomprehensible), however, it also means that object-based systems are more likely to contain inconsistencies and redundancies.

In the computer-science arena it is recognised that objects are just one of a set of approaches which form the toolkit required to solve problems. In the USA a huge set of research projects have been initiated to find the successor paradigm to objects. This is in recognition of some of the problems associated with objects, and these are worth considering in the construction IT area. Major problems include the following:

- Object-oriented modelling and programming is not well suited to large systems. This should be of serious concern to the construction industry's modellers as construction models are arguably some of the largest and most complex models which have ever been developed. The main problem here is that object-based systems are well suited to micro-level specification, but have few constructs which enable a macro-level specification to be managed. Those who have worked with large object-based models will recognise this problem (e.g., try understanding ISO-STEP or IAI-IFC models with two to three hundred object definitions), even where higher level graphical representations are employed (e.g., EXPRESS-G).
- Object-oriented systems are not easy to validate. It is not easy to show that an object-oriented model is consistent and non-redundant. This is due to the paradigm having no underlying formalism, unlike relational systems where, when the steps are followed, it is possible to guarantee that there is no duplicate or inconsistent information in the model, and that changes will never cause inconsistencies. It is also not easy to test (or prove) that an object-based system works correctly. This is mostly due to the method paradigm for describing the functionality of an object. As public methods can be invoked by any object, it is very difficult to show that a large object-based system will work correctly under all circumstances.

The main point here is that object-oriented modelling and systems are not a panacea for the construction industry. While they offer a range of tools and systems which provide great benefits for construction, they bring as many problems to the table as they solve. The industry needs to recognise that object-based approaches are not inherently better than non-object-based approaches, and that object-based systems and not guaranteed to be better that non-object-based systems.

### *2.2 The single data model will appear*

This idea, which is intoxicating in its naivety, seems to be raised by everyone new to this field. Indeed, august organisations such as ISO initially promulgated this idea when they started working on the STEP data models in the 1970's. History has consistently shown this to be impossible for complex domains and it is frustrating to have to reiterate the problems with, and arguments against, a single data model with every new generation of modellers in this area. There is a wealth of publications and books in this area and it would behoove new researchers to study papers more than two to three years old in this area to gain a proper understanding of the problems. Summarised, there are several issues which make a single data model improbable:

- A major issue in developing a single data model for the built environment is the scope this embraces. When one starts mapping out the various axes of this area it becomes clear that a single data model would have to encompass an enormous range of information. For example, requirements of clients, architects, engineers, constructors, facility managers, renovators, and demolition specialists, along with sub-domains such as landscape architecture or civil engineering. There is even major overlap with other industries, for example, ship building and aeronautical for structures, HVAC, wiring, etc where there are existing data models.
- Following on from the point above is the number of objects that would need to be modelled for such a data model. Current views are that two to three hundred objects in a data model are not easily manageable. A single data model would have tens of thousands of objects to describe, with complex inter-relationships between one another.
- Another major issue related to the domains involved is how the differing views would be reconciled. For example, the world view of an architect (and hence the model they would require) is very different from that of a structural engineer, or a quantity surveyor. Even if a single model could be created for each domain (and even this is not thought to be possible) it would not be possible to merge all views into a single coherent whole (see Section 2.4).

### *2.3 We represent reality*

Some people believe, in a similar way to the notion outlined in Section 2.2, that it is possible to completely model reality. It is clear that a data model is an approximation of reality, or some conceptual notion, for example, space and zone. The effort that would be required to model the full range of attributes of any single object utilised in construction is beyond the effort available for simple objects (e.g., a nail) and representationally impossible for more complex objects (e.g., a door or chair). Simply put, it is impossible to model all the designed combinations of the majority of objects. This means that every model that is generated is only capable of representing a portion of the possible structures that could be created. For those interested in how complex this can be there is an excellent data model developed in the EC funded ATLAS project which, despite its prodigious size, represents a small percentage of the door types which exist in this world.

There are few rigorous approaches to tackling this problem. For example, in the IAI every model is developed for a closely defined set of processes in the industry. This helps draw the bounds around what needs to be modelled, but does not help in the final decision of what range of a particular object should be modelled (e.g., 80% of what is used in the world, at least all of the products of the manufacturers involved in defining the standard, etc), and who validates this, or how it is validated. When creating a model there are three aspects of modelling which are used to ensure a larger set of possible representations; these are:

- Structural specifications which define a particular type of object fairly abstractly and then have more detailed specialisations based upon them (e.g., door with sliding, revolving, etc specialisations). This also allows the major components of an object to be identified and modelled independently, for example with a door having its frame, hinges, handle, lock, and inserts. The structural specification provides the most information about the object in terms of explicitly representing the major aspects, but provides the greatest difficulty in providing enough structure to cover all possibilities (e.g., all structures which can be doors).
- Functional specifications allow the modeller to define what an object does rather than the structures and components which makes it possible. This is a less detailed, but more general

method of providing a model with information about the utility of an object. For example, that a door allows egress and ingress for a space and the parameters of the possible movement.

- Graphical specifications which, if general enough, allow any permutation of a structure to be represented graphically. This provides no machine understandable structuring for an object (e.g., not possible to infer that the fire resistance is 2.5 hours), but allows every possible type to be represented in a way that can be comprehended by a human user. Herein lies the dilemma: we can represent everything if we have dumb graphics, but can not do anything intelligent without the structural and functional specifications which can never be comprehensive.

### *2.4 User views are reconcilable*

This is a view which is proffered in many modelling areas; the assumption seems to be that because professionals are able to communicate and understand each other, then their views are obviously reconcilable. This view is closely related to issues in Section 2.2 and 2.5 where a parallel is drawn from human comprehension to computer-based models and systems. However, the basis for this association is unclear.

The authors' contention is that different professionals' perspectives of a system are often irreconcilable. The only reason that the utterances of different professionals are understood and lead to a single view of a construct is through the application of considerable human intellect and experience to understanding them. To reconcile formalised models of two different professionals' world view would require a greater application of artificial intelligence than is currently possible; let alone to meld two models into one and still represent both of those world views. An analogous problem of similar complexity is language translation. It is clear that two humans who know part of the other's language can communicate and reach an understanding. However, despite many years of intense research, it is still not possible to completely represent different languages in computerised form, let alone translate between them consistently, let alone have a single common representation which covers the complexities of both languages.

Two concrete examples of the types of views which cause problems are described below. One is the difference between a space-based view of a structure versus a component-based view (e.g., walls, floors, etc to enclose a volume). The models to represent these two views are structurally very different and hard to reconcile without loss of detail from one of the views. Another, an example of aggregate versus component views, is where one professional would consider a series of walls which are vertically aligned as individual components, the second would view this as a single element (e.g., a structural wall).

### *2.5 Mapping is easy*

There are many who proffer a view that once two data models exist it is then a relatively straight-forward process to provide a mapping between them. This is unfortunately not the case. Because construction encompasses a number of inter-disciplinary domains and work efforts a major study must be undertaken to know what has been done and what can't be done. There are many examples of very simple mappings which are not possible to perform; the majority in the category of aggregate to component views. For example, a U-value is a concept well understood by certain professionals and used consistently in their domains. However, in other domains there is a requirement for greater detail than offered by a U-value, and the equivalent representation of thickness, capacitance, and resistance is utilised. It is impossible to map automatically from a U-value to thickness, resistance, and capacitance.

Even if such simple mappings did not trip up the mapping process, there is a difficulty in providing bi-directional mappings between models. In relational systems there is a concept of a view which is utilised to provide different representations of an underlying model (analogous to a one-way mapping). However, the view mechanism is only bi-directional under very tightly constrained conditions. This is due to the great difficulty that exists in being able to describe a mapping which can work equivalently in both directions. The best solution would be if the same mapping code could be run in both directions. Anyone conversant with procedural programming will understand that this is not feasible. So, to create a bi-directional mapping, it is necessary to write two sets of mappings, one for each direction. This approach makes it impossible to prove that the two mappings are equivalent, and hence whether moving information from one view to another, and then back again, will preserve the information that existed in the first place. This must be a serious concern to all attempts to create a collaborative work environment around an IPDB.

### *2.6 The Internet solves the communication problem*

The Internet is assumed to be able to solve all communication problems existing between dispersed project partners. While it is indeed becoming a ubiquitous transfer mechanism, the control offered by the Internet is still quite simple in comparison to the requirements of collaborative construction projects. As the Internet does not have a session-based protocol, or centralised management, many collaborative tasks are harder to manage. For example, the Internet does not allow you to identify who is working on a particular resource (e.g., a document) at any one time, or to work out when the resource is available for another project participant to work on. Another example, is the openness of the Internet which allows anyone to communicate with whoever they wish; whilst this makes it very easy to interact with others, it does not enforce the centralised management and registering protocols which are liability management requirements for many firms. It is recognised that these are mostly process management issues. However, it must be realised that the Internet does not currently offer solutions to enforcing good practice for collaborative working.

### *2.7 XML solves the representation problem*

If you listen to the marketing, XML (eXtensible Markup Language) is the hottest new technology, with everyone jumping on the bandwagon. However, all it really provides is another (albeit Internet-based) method of coding data. The STEP physical file format (ISO-STEP Part 21) is a comparable (albeit not Internet-based) notation. In order to be usable for construction it is still necessary to have an agreed standard data model (e.g., ISO-STEP or IAI-IFC). The initial attempt at this for construction is aecXML which was submitted by Bentley and is currently closely aligned with the IAI. Their aim is to develop and adopt aecXML as a means of representing construction information to be used in electronic communications.

A widely adopted and comprehensive XML schema for the A/E/C domain could greatly benefit the construction industry. It could play a major role in the future development of software applications, in particular how information is coded, represented, accessed, queried, shared, and exchanged on the Internet. aecXML is not intended to be a format for transferring and exchanging project information, but rather a technique for expressing what is in them. For example, this would enable users (human and software) to interrogate the aecXML product representation to find out its cost information. For aecXML to be used as a means of exchanging project data, a standard schema or set of standard schemas capable of supporting the project lifecycle need to be developed. This process of defining standard schemas is resource-intensive and requires the consensus of the industry. This task has been undertaken by the IAI (International Alliance for Interoperability) and STEP (Standard for the Exchange of Product Model Data) in their development of the Industry Foundation Classes (IFC) and STEP Application Protocols (APs) respectively.

### *2.8 Documents will disappear*

IPDB systems will not cover the whole project cycle, from inception to demolition, in the foreseeable future. Further, construction projects involve a large number of participants with varying capabilities which range from a one person company to multinational organisations, which makes it difficult to assume that everyone has the same capability to work and share information electronically.

Many information systems have been developed; the intended benefit is to cut down the time and cost associated with exchanging information between the project teams and members through a paperless link between the remote offices. Our observation shows that although a large amount of the information is generated electronically, they are printed on paper to be processed by other members of the team, or re-keyed into another system. Although this process adds no value to a project, is time consuming, and makes the management of information difficult as the same data exists in more than one form, many find it necessary. For example, site engineers carry a paper copy of the design around the site with their notes upon it. These notes are considered important to the individuals. Subsequent electronic updates to the design necessitate the need for the updated copy of the design to be reprinted, which many cause loss of information.

Although data models have the potential to support a large subset of the project lifecycle, not all the documents are, or can be, derived from a data model e.g., contracts for a project or site instructions.

### *2.9 CAD is the centre of an IPDB*

Today, powerful geometric modellers exist which are capable of representing 3D models. However, geometry alone is limited in its usefulness for driving construction applications where other non-geometrical information is required. The current support provided by these modellers is not sufficient to support construction applications. For example, it is not a straightforward task to transfer design information from a 3D modeller into an estimating or planning application. These geometric modellers still suffer from the following (Faraj 1999):

- Incomplete database. Attributes which are required by construction applications are not represented e.g. specifications, quantities, resources, etc.
- The geometry is stored in a very low level of detail which is not related to the construction processes. There are no higher levels of abstraction of a project's description, such as columns or walls. However, a number of software packages are now available to address this particular issue, by providing a library of predefined objects.
- There are no query mechanisms in these systems, e.g. how many columns are there on the first floor?
- Similarly, in other domains of construction, where software is used, existing data that has been generated elsewhere has to be re-keyed by the user, resulting in data duplication, multiple representations, and possible transcription errors.

This limitation to the development and use of 3D modellers has created the need for an environment to facilitate the flow of project information between the construction applications. Geometry forms a very important part of the developed standards (e.g., STEP and IFC) and many CAD vendors are implementing them. One might argue that CAD systems could be the centre for project information as geometric data is required by many construction applications. However, the non-geometric information also needs to be represented, an area that may be outside the interest, scope, and expertise of CAD vendors.

Whilst data sharing and exchange in an integrated computer environment can be achieved through a project model inside any CAD system this is not a necessity. This data model can be distributed across a range of databases, each supporting a portion of the project data and, perhaps, each owned and controlled by a different player in the process.

### *2.10 IPDB solves information ownership problems*

Many of the legal implications of exchanging and sharing project data between partners need to be clarified in order for an IPDB to be assimilated into current processes. In the construction industry, the ownership of and liability for data, as well as the way projects are managed, impose restrictions on how, what, and when data should be exchanged. It is within this context that other components must be added to the integrated environment.

The issues of private and public data should be considered in order to ease and control the exchange of data between the different disciplines. Data which are required by other disciplines to effectively fulfil their roles must be made available as and when required. These are referred to as `public data', while data which are confidential and not necessarily required by other disciplines are known as `private data'. This distinction between data is important in addressing the problem of liability and ownership within organisations. Data become public only when they are formally released by an organisation. This situation is similar to the practice of posting out documents to other disciplines. In such a case the data will be marked according to their released version, status, and date.

Current data models contain no mechanism for distinguishing between ownership and use of data. While these are similar concepts; both convey a variety of privileges, from read and query access to creation and modification rights, ownership refers to the right to determine these privileges for others.

If the liability issues are to be resolved, mechanisms must be implemented that impose control over the data, and ensure that these issues are managed.

### *2.11 IPDB enables collaborative working with coordinated and consistent information*

Computer integrated environments are still an active research area, and many issues of collaboration have not been investigated fully. If computer integrated environments are to support collaborative work, then the following must be considered:

- Data access: Who should access certain data and at what stage of the project lifecycle; who should be responsible to assign such privileges, and how do they specify and manage these.
- Data change: Who can change the data and who should be notified. Should notification be sent out even if irrelevant parts of the data are changed; who should decide what are the relevant parts of the data.
- Database locking: What happens when more than one person works on the same set of data. Some activities require a team member to work continuously for a long period; as a consequence the database will be locked for the duration, resulting in the other team members being unable to access the data.
- Data partitioning and integrity: Can users retrieve and save only part of the instances of the model without the need to retrieve or save the whole model every time an alteration is performed. For example, if two designers are working on the same project can the first designer work on his/her part of the design without affecting the other designer's data. What

happens when the first designer wants to save the data, would it automatically overwrite the original data in the model.

- Co-operation: Is the industry ready to share its data and co-operate to solve problems. Project information needs to be carefully monitored and controlled by the project management organisation where it may need to apply its own experience and rules to run the project effectively. It is not clear how members of a team (from different organisations) would react to such an approach. Orlikowski (1992) has observed that an industry standard groupware product, company-wide introduction, and senior management approval were insufficient reasons for employees in a major consulting firm to share information. Culture and incentives opposed the knowledge transfers which the technology was designed to support.

To date, the authors are not aware of any IPDB system that addresses all these issues. However, the COMMIT project has addressed some issues associated with data access and data change. WISPER (Faraj et al. 1998) and ToCEE (Amor et al. 1997) implemented a distributed environment based on the IFC Standard.

The area of collaborative environments is gaining a lot of support from both the research community and software vendors. Currently, systems developed by industry are based on more mature technologies with very limited capabilities e.g. BT Construct , CADWEB, etc. However, research projects such as at CIFE (Fruchter 1998) and Phase(X) (Schmitt 1998) are more focused on issues associated with how information, construction processes, and culture can affect the work performed through collaborative environments.

### *2.12 The industry is ready for IPDBs*

Effective implementation of large IT systems, such as integrated environments, require a substantial process and culture change. This is likely to produce a new set of business processes, which could have a significant impact on the organisational structure and current practices. It is therefore important that construction organisations first of all ensure that any significant departures from traditional practices which are essential for achieving the business objectives are implemented in well-measured stages.

Many prototypes have been developed as a proof-of-concept and tested on small projects. The real challenge begins when the technology is applied to more complex projects and when the organisation as a whole begins experimenting with or using such tools. Organisations need to develop well thought-out strategies for moving to integrated systems. The transition to such systems is not only a transition in tools and techniques. It is an evolution into a new way of performing today's tasks and a new approach to project problem-solving.

## 3 CONCLUSIONS

This paper presents twelve views that the authors believe are misconceptions about IPDB systems. The aim of this exposition is to gain critical feedback and suggestions which will form a consensus on what IPDB systems will deliver. Some of the arguments are show-stoppers for the type of IPDB system which are promoted for the future (e.g., if it is impossible to map between views). If these views are indeed incorrect then the IPDB researchers need to ensure that what is sold to the industry and government funders does not over state the case and benefit. The risk of over-hyping the possibilities is that IPDB development will be viewed with the same suspicions as AI work is currently.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support of the UK Department of the Environment, Transport, and the Regions (DETR) in this work.

## REFERENCES

Amor, R.W., Clift, M., Scherer, R., Katranuschkov, P., Turk, Z. and Hannus, M. (1997) A Framework for Concurrent Engineering - ToCEE, European Conference on Product Data Technology, PDT Days'97, Sophia Antipolis, France, 15-16 April, pp. 15-22.

Anumba, C.J. and Amor, R. (1999) A Survey and Analysis of Integrated Project Databases, Proceedings of Concurrent Engineering in Construction'99, Espoo, Finland, 25-27 August, pp. 217-228.

Anumba, C.J., Baron, G. and Evbuomwan, N.F.O. (1997a) Communications Issues in Concurrent Life-cycle Design and Construction, BT Technology Journal, 15(1), January, pp. 209-216.

Anumba, C.J., Baron, G. and Duke, A. (1997b) Information and Communications Technologies to Facilitate Concurrent Engineering in Construction, BT Technology Journal, 15(3), July, pp. 199-207.

Arnold, J.A. and Teicholz, P. (1996) Data Exchange: File Transfer, Transaction Processing and Application Interoperability, Proceedings 3rd ASCE Congress on Computing in Civil Engineering, Vanegas, J. and Chinowsky, P. (Eds.), ASCE, June, pp. 438-444.

Bentley Systems (1999) aecXML Preliminary Specification: Working draft 0.87, September, http://www.aecxml.org/.

Bjork, B-C. and Penttila, H. (1989) A Scenario for the Development and Implementation of a Building Product Model Standard, Advances in Engineering Software, 11(4), pp. 176-186.

Construct IT (1996) Feasibility of the Integrated Project Database, Department of the Environment.

Eastman, C. and Augenbroe, G. (1998) Product Modeling Strategies for Today and the Future, Proceedings CIB W78 Workshop on the Life-Cycle of Construction IT Innovations, Stockholm, Sweden, 3-5 June, pp. 191-208.

Egan, J. (1998) Rethinking Construction, The Sir John Egan Task Force Report, Department of the Environment, Transport and the Regions (DETR), London, July.

Faraj, I. and Alshawi, M. (1999) A Modularised Integrated Computer Environment for the Construction Industry: SPACE, Electronic Journal of Information Technology in Construction, Vol. 4. ISSN 1400-6529, http://itcon.org/1999/3/.

Faraj, I., Alshawi, M., Aouad, G., Child, T. and Underwood, J. (1998) The Implementation of the IFC in a Distributed Computer Integrated Environment, Second European Conference on Product and Process Modelling in the Building Industry, London, UK, 19-21 October, pp. 187-198.

Fischer, M. and Froese, T. (1992) Integration Through Standard Project Models, Proceedings CIB W78 Workshop on Computer-Integrated Construction, Montreal, 12-14 May, pp. 189-205.

Fisher, N., Barlow, R., Garnett, N., Finch, E. and Newcombe, R. (1997) Project Modelling in Construction, Thomas Telford Ltd, London.

Froese, T., Yu, K. and Shahid, S. (1996) Project Modeling in Construction Applications, Proceedings 3rd ASCE Congress on Computing in Civil Engineering, Vanegas, J. and Chinowsky, P. (Eds.), ASCE, June, pp. 572-578.

Fruchter, R. (1998) Internet-Based Web-Mediated Collaborative Design and Learning Environment, Lecture Notes in Artificial Intelligence in Structural Engineering (1454): Information Technology for Design, Collaboration, Maintenance, and Monitoring. Springer, ISBN 3-540-64806-2.

Gadient, A.J., Hines, L.E., Welsh, J. and Shwalb, A.P. (1996) Agility Through Information Sharing: Results Achieved in a Production Environment, Advances in Concurrent Engineering, CE'96, Sobolewski, M. and Fox, M. (Eds.), Technomic Publishing, Lancaster, pp. 211-218.

Gann, D., Hansen, K.L., Bloomfield, D., Blundell, D., Crotty, R., Groak, S. and Jarrett, N. (1996) Information Technology Decision Support in the Construction Industry: Current Developments and Use in the United States, Science Policy Research Unit, University of Sussex, Brighton, UK, September.

Law, K.H. and Krishnamurthy, K. (1996) A Configuration and Version Management Model for Collaborative Design, Information Representation and Delivery in Civil and Structural Engineering Design, Kumar, B. and Retik, A. (Eds.), Civil-Comp Press, Edinburgh, pp. 7-14.

Minsky, A. (1975) A framework for representing knowledge, The psychology of computer vision, McGraw-Hill, pp. 211-277.

Orlikowski, W. (1992) Learning from Notes: Organizational Issues in Groupware Implementation. ACM Conference on Computer Supported Cooperative Work, Toronto, Canada, 31 Oct-4 Nov, pp. 362-369.

Schmitt, G. (1998) A New Collaborative Design Environment for Engineers and Architects, Lecture Notes in Artificial Intelligence in Structural Engineering (1454): Information Technology for Design, Collaboration, Maintenance, and Monitoring. Springer, ISBN 3-540-64806-2.

Tah, J.H.M., Howes, R. and Wong, H.W. (1997) Towards a Concurrent Engineering Environment for Integration of Design and Construction (CEE-IDAC), Concurrent Engineering in Construction, Anumba, C.J. and Evbuomwan, N.F.O. (Eds.), Institution of Structural Engineers, London, July, pp. 206-215.

URL-1 (1999) http://www.bre.co.uk/oonet/ UK Network of Experts in Objects and Integration, for Architecture, Engineering, and Construction