LOCAL VALUES
in a
NETWORKED
DESIGN WORLD

ADDED VALUE OF COMPUTER AIDED
ARCHITECTURAL DESIGN

DUP Science

scientific committee

DUP Science is an imprint of
Delft University Press
P.O. Box 98
2600MG Delft
The Netherlands
Phone +31.15.2785678
Fax +31.15.2785706
E-mail info@library.tudelft.nl

Editors
Stellingwerff, Martijn — Hogeschool voor Wetenschap & Kunst, Sint-Lucas Architecture and TU-Delft, Faculty of Architecture, Form & Media Studies
Verbeke, Johan — Hogeschool voor Wetenschap & Kunst, Sint-Lucas Architecture

Keywords — Architecture, Local values, Globalisation, Computer Aided Architectural Design.

ISBN 90-407-2507-1
DUP Science

Printed in the Netherlands.

The promise of peer-to-peer computing versus the utility of centralised data models in collaborative design

AUTHOR

Michael Cumming

Faculty of Architecture
Delft University of Technology
The Netherlands

Abstract

Peer-to-peer (P2P), or distributed computing, involves having computers on a network - peers- acting as both suppliers, as well as consumers of information. With recent developments, most notably the JXTA initiative by Sun Microsystems, such P2P technology will soon become quite easy to implement, in a standardised and secure fashion. P2P technology holds promise in the domain of collaborative design in that it allows design collaborators to exchange information in a manner that appears to have certain advantages over centralised systems, such as greater spontaneity, the ability to self-organize, better scalability, and the ability to handle transient resources in a more robust manner. However, it is not clear how this new technology can be applied to the information needs of collaborative design, in which centralised data models are usually seen as useful. This paper examines some of the positive and negative implications of this new technology in the context of collaborative design.

What is P2P?

Introduction to P2P

Peer-to-peer (P2P) involves having computers on a network -- *peers* -- acting as both suppliers, as well as consumers of information. P2P does not constitute a new idea -- it has been around as long as computing itself but several recent developments have increased its visibility.

These developments are: 1) the use of P2P software to facilitate the trading of digital music files between individual computer users, or ‘peers’. These users are often college students who have access to fast Internet connections. 3) The availability of a P2P development framework called JXTA by Sun Microsystems, which is the latest and the most advanced framework for the development of P2P applications [13].

The JXTA framework, due to the comprehensiveness of its design, and its general excellence of design, from a software engineering viewpoint, appears to be set to become a common platform that could experience widespread popularity. This could much in the same way that another important software development effort from Sun -- the Java language, has also changed the face of software development substantially in recent years.

Examples

Sun: JXTA

JXTA is a standardised, open-source initiative that provides a protocol, with language bindings for several languages, that allows for the design and implementation of secure P2P applications. JXTA is based on open-source, standards-based protocol specification, and can be implemented in Java or any other languages [11]. JXTA also provides a generic infrastructure to deploy P2P services and applications [6].

JXTA is built out of five key abstractions [11]:

- Uniform peer ID addressing,
- Peer-groups,
- Advertisements,
- Resolvers, and
- Pipes.

The JXTA protocols work by stipulating layers of several services, which peers may or may not chose to implement. These include, among others, those for discovery of peers and peer groups, ones for membership to manage who gets to join peer groups, pipe services to create network connections between peers, and end points to describe how messages should be routed between peers. JXTA is protocol-based, and is not language dependent. It is also open source and network independent, since peers located on different networks can communicate easily using the standard JXTA protocols.
What JXTA offers that sets it apart from previous efforts at P2P is that it is a community and standards-based effort that aims to provide a standardised set of tools and services that are likely to be of benefit to the whole P2P community. It appears that within the P2P world, JXTA may have the same beneficial effect that the design of the Java language itself has had on standards-based software engineering.

Others

There are other P2P frameworks, however these have tended to decrease in attractiveness with the appearance of JXTA. These include:

- Groove networks: corporate decentralised file sharing, secure IM [8]
- Gnutella: music file sharing [5]
- Freenet: anonymous file sharing [2]

What does P2P mean for computing?

The 'buzz word' advantages of P2P

The supporters of P2P list its many apparent advantages:

- *Scalability*: the ability of P2P applications to increase their performance as more users are added, rather than to decrease it.
- *Robustness and fault tolerance*: the ability of P2P to degrade gracefully when network connections, or computing resources in general, become unavailable or corrupted.
- *Dynamic behaviour*: the ability to handle and dynamically adjust to the presence or absence of specific computer resources.
- *Spontaneity*: the ability of applications to respond to changes brought to computer systems by inputs from new peers and new computing resources, without having to pre-conceive these changes or to do special work to handle them when they do occur.
- *Self-organisation*: the ability of people working on a P2P network to quickly organise into specific peer groups of their own design, without the requirement of any centralised or pre-conceived interventions.

The added capabilities that P2P allows users

It is clear that P2P, in theory, allows ordinary computer users to do things that may be difficult to do otherwise.

Ordinary users as information and application servers

P2P allows all peers, simply be being on a P2P network, to become information and application servers. This is one of the most basic ideas, and one of the most compelling advantages of P2P technology.

Without P2P, peers can share information, in a two-way, spontaneous, reciprocated fashion by such means as e-mail. However, e-mail depends on such centralised resources such as POP mail servers, and on registries of e-mail users' login names and addresses. In addition, the flexibility of e-mail to handle a variety of information types is often primitive, compared to what is possible using more advanced P2P technologies.

Peer groups initiators

P2P technology allows peers (ordinary users of P2P applications) to form 'peer groups'. Finding suitable peer groups to join, or in other words, finding groups of people in which certain types of ideas might find a receptive audience, is an important aspect of collaborative activity. This is also one capability for which P2P is very attractive.

Within the boundaries of customised and specialised peer groups, sharing of specific information can occur easily. These peer groups can arise very quickly and spontaneously, with very little effort. Peer groups formed in this way can also quickly attain a group identity based on a build-up of common ground, which tends to encourage the sharing of information, or to provide a reason to share information. The build-up of common ground is viewed as an essential aspect towards encouraging collaborative activity [1].

With P2P, any user can send out resources necessary to join a specific peer group. In the JXTA system, for example, a peer uses this resource, called an advertisement whose message might be: 'join my discussion group on the sharing of XML data in building construction' called 'XML in Building Construction'. The remote peer could use this advertisement to instantly join this peer group. If the demand was great for discussion of this sort, and if enough interested parties happened to see the advertisement, then this peer group could grow very quickly in size.

Access to transient resources

One of the major ideas behind P2P computing is the idea that resources available from the Internet should be treated as transient resources, rather than permanent ones. P2P applications work by discovering resources on a real-time basis. This means that one day certain resources may be present, whereas the next day these resources may have disappeared, without a trace. This is a normal occurrence in computing. With any network configuration or architecture, on-line resources may in fact may disappear at any time: the network, or a portion of the network could go down or suddenly be clogged, a web server could be attacked by viruses, a company could suddenly go out of business, etc.

Software that treats such inconveniences as regular features of computing life, rather than as a crippling event, tend to be more reliable. The use of P2P might be expected to lower the expectation of its users that resources are a permanent fixtures in networked computing environments This reduction of that expectation is probably a good thing, because even when resources are centralised, and seemingly permanent, they in fact are not.

Is P2P a more general computing architecture?

In building a P2P system it is possible to specify that certain resources *must* come from only one location. This would allow all peers to access this resource much in the same way as with a client/server system. However, with P2P, the normal practice is to design the system such that multiple resources are accessed in a similar way to the singular one.

With centralised architectures such as client/server, there is a difference of type between the two classes of users: the client and the server. They entail two quite different pieces of software. The client tends to be a lightweight piece of software, while the server tends to be quite heavyweight, with demanding requirements for security. If a server is popular with its community of users, perhaps hundred of thousands of requests for information, from around the world may need to be serviced daily.

In P2P systems, all peers are able to function as both clients and servers, all of the time. Therefore, it appears that P2P applications can easily duplicate that which centralised information architectures allow, whereas the reverse does not appear to be true.

Overall significance of P2P

P2P could either be a fad promoted by selfish, bandwidth-hogging college students engaged in illegal or semi-legal activities, or it could be a fundamentally new paradigm of computing that recognizes the transient nature and fragility of computing resources available over the internet. The idea of P2P is new enough that its true nature is not yet apparent to either a majority of researchers or computing professionals, involved in the design of software or hardware, network architectures. Therefore, it is too early to tell what its true significance at this point.

Centralised vs. decentralised design systems

Introduction

Information systems that support collaborative design, can be designed either as centralised or decentralised systems. Up to now the tendency has been to build centralised systems, such as those that employ client-server architectures.

Two ends of a spectrum

P2P are seen as one end of a spectrum of available network topologies, in which completely centralised systems are at the other end. This essay attempts to analyse the implications of the two ends of the spectrum, rather than spend time discussing the myriad shades of grey in between. In the end, it seems likely that the most profitable approach in design systems will be to make hybrid systems that take advantage of the inherent advantages of both the distributed and centralised approaches.

Why the centralised approach has been popular

The main reason why the centralised approach has been more popular is:

- *The suitability of centralised architectures in the development of centralised, integrated product models*: It is usually a single unified artefact that is the intended result of a collaborative design process, it seems to make sense to attempt to make unified design representations from the beginning stages of design. Since the goal is to produce a single unified model, it makes sense to keep this model in one, centralised location. Integrated product models have the potential advantage of having a high level of internal consistency and rationality in their design. Models in which all design description information resides in one location, can be, for instance, very convenient when checking for completeness and consistency of design product models [4].
- *An approach towards collaborative design that favours design processes that in theory can be rationally planned*: The dominant design paradigm within the computer-aided design community has been one that is inspired by the promise of rational and scientific reasoning and planning in solving complex problems. One goal of technical reasoning is to provide a supportable rationale for design decisions, such that the overt subjectivity and biases of individuals can be avoided. When design problems are seen primarily as ones that can be solved by application of technical or scientific reason, then it becomes important that the actors involved in a design process have technical or scientific reasoning, and problem-solving skills. A rational, defensible design path then should be clear to most of the participants engaged in the design process, provided they are competent thinkers and professionals. Therefore, with a rational approach, specific actors, and their attendant biases, are seen as less important, and it becomes more acceptable that a reasoning engine than orders a complex design process, be located in a single, centralised location.
- *The lack of credible alternatives to centralised systems, such as P2P:* Popularity

of P2P depends both a conceptual shift, such that they can be seen to useful in theory, as well as a technological shift, such that P2P computer systems become practical to develop. P2P in its modern embodiment is a relatively new idea that has failed to achieve a ‘critical mass’ of popularity among users, researchers and developers -- except in domains such as on-line file sharing or instant messaging (IM). Until recently, there has been a lack of reliable P2P technology and of suitable P2P application development frameworks. These frameworks allow a standards-based, non-proprietary approach for P2P application development, which is seen as an important factor in popularising P2P theory and applications. It is the author’s opinion that the appearance of JXTA by Sun has changed this situation, and that there are now few impediments to discourage the growth of P2P

Disadvantages of centralised systems

Covert conceptual prescriptions

The process of conceptual design usually involves coming to a consensus with your design collaborators as to what an appropriate conceptual organisation for the project should be. In centralised design systems, this type of consensual pre-design work is often contained implicitly within the design of the computer system itself. In some cases this pre-definition of the semantics of design objects of interest, could conceivably have an unwelcome and constraining effect on the types of solutions that could result from the use of such a system. The same could be true of P2P systems, although it is expected that the type of covert prescriptions might be of a different type.

Location of proprietary data

Centralised systems usually assume that participating designers in the collaborative design process are willing, or able, to submit their design contributions to a party, or a computer-based system, that maintains a central data store or representation. In order to conform to a central representation, data translation and formatting work may be involved on the part of individual contributors. Some designers and consultants may have a proprietary interest in not allowing their specialised design representations to reside in any location other than their own private and secure databases. They may only share a subset of their data such that design collaboration is possible, without offering the full richness of the data, they may use internally within their own organizations [12].

Necessity of ‘up-front’ work

Centralised architectures tend to depend on substantial quantities of ‘up-front’ work to build suitable information infrastructures. Centralised systems, by definition, require the people they might affect -- their ‘stake-holders’, get together and work out what would be an appropriate, supportive system. Such consensus-building work takes much effort, and ideas what constitutes an appropriate system may vary widely, even among skilled professionals acting in good faith. Work on computer-based information infrastructure is usually work of a technical nature that designers in many domains may be unqualified to perform, without support from specialised information professionals. Such work, especially with centralised systems, tends to require making prescriptive and predictive assumptions about the nature of the information to be exchanged, as well as the composition and organizational hierarchy of the design team. Such aspects of collaborative practice may become clear to the design team only once a design process is well established.

This is commonly recognised problem with design, especially ‘early design’ support systems: how to support a design process without unduly shaping it to conform to a computer system designer’s preconceptions. In order for a computer system to be useful in supporting design, the system must exist. The same is true for both centralised or distributed systems. What is important is the effort required to get useful systems working, and whether these systems support, or unduly shape the nature of a design process. It appears that centralised systems tend to be weak in both these respects. However, with the absence of complex P2P design systems to compare them to, it is difficult to determine at this time whether P2P systems will be any better.

Advantages of centralised systems

Accessibility of unified design representations

Having a centralised representation means that this representation is available without any additional effort on the part of the administrators of this data. Therefore, the documentation process does not require the burden of a process of assembly of documents, from their variety of authors, such as from the various consultants involved in a collaborative design process. Such an assembly process can sometimes be prohibitively expensive. This means, for instance, that historical records of building projects can be maintained much more easily when there are integrated and centralised design representations.

Rational design of information infrastructures

Despite the fact that various design agents may have different conceptualisations of design data and of the design process, there remains the fact that information infrastructure design, such as database design, can be helped enormously when it is designed by people skilled in this domain. Distributed logic may allow design participants in theory to express anything they wish to express. However, this kind of freedom is may not be necessary in many cases, and simple logical structures may satisfy most, if not all of the design participants.

Usefulness in routine design processes

In the context of design systems, the intended degree of innovation in the design process is an important factor. In routine design processes -- ones in which the participants may have long experience, working within conceptual frameworks that are unlikely to change dramatically, centralised systems can obviously provide useful support for designers. In routine design, the issue of design freedom is not normally relevant. Preconceived goals in such design situations are not really unwelcome constraints, but rather an essential feature of this type of design.

Disadvantages of distributed systems

Lack of a central representation

The most salient feature of distributed systems is that their control and data are distributed. To maintain data integrity and consistency in distributed environments is usually much more difficult, than in centralised situations [3]. Since construction of centralised data models is often seen to be an important aspect of collaborative design practice, it appears that P2P is best suited for tasks other than the development of consistent and logical product representations.

Lack of central control

The distribution control and data found in distributed systems can have disadvantages according to Jennings, in that 1) each agent only has a partial and imprecise perspective, 2) there is increased uncertainty about each agent's actions, 3) it is more difficult to attain global behaviour, and 4) the dynamics of such systems become extremely complex [10].

Distributed control when placed in a design context is not a concept that may not have much intuitive appeal to designers. Designers are usually trained to view their primary job description as 'controllers of design processes'. The traditional expectation is that in order for a designed product to have some kind of aesthetic or functional coherence and integrity, a single cognitive entity such as a designer, must conceive and coordinate the design in its entirety. For smaller design problems, this is quite possible. For more complex problems, or for those that take place over an extended period, it becomes more difficult.

Behavioural chaos

Distributed systems since they lack central control, often exhibit non-linear interactions, as noted by Hogg. He notes that such systems can display a wide range of behaviours including stable equilibria, continual oscillations, and chaos [9]. Chaos is considered a destructive aspect of distributed systems in that it introduces global unpredictability into the system. Hogg proposes that simple reward mechanisms, based on the assessed performance of software-based agents, can help eliminate such chaos.

Advantages of distributed systems

A better model of data sharing?

In collaborative design, similar to what happens in P2P systems, individuals agents are often placed in the role of both being information providers as well as information consumers. The fact that P2P systems allow this process to occur transparently is seen as a major advantage of this technology.

Distributed control

Decentralised systems do not require that one party assumes a position of control over the work of others. This may or may not be the organizational approach that is appropriate for a specific design project. However, recent managerial trends that emphasize the advantages of flatter, leaner management hierarchies in developing more 'agile' and productive organizations, suggest it is a trend growing in popularity.

In complex, collaborative design projects, where the input of specialised design experts may be crucial to finding acceptable solutions, hierarchical control of such experts may not be, for instance, politically appropriate. Instead, complex systems rely on independent or autonomous agents interacting with each other in plausible ways, generally without access to global knowledge. Some would argue that this approach better simulates the behaviour of real designers as they perform their jobs - especially in complex, non-routine design situations.

Within the Distributed Artificial Intelligence (DAI) community, the strategy of distributing control, data, as well as knowledge sources, is now widely supported [14]. Such an approach has been shown to have several advantages, including the reduction of performance bottlenecks, the increase in reliability, and the soft, rather than steep or complete degradation of performance, when systems are under stress.

Chance of creative emergence

Decentralised systems work by allowing independent agents to interact in a manner that does not rely on pre-articulated or pre-conceived goals. In complex systems research, of which design of decentralised systems is a part, mechanisms of self-organisation have been used to explain behaviours and constructions that appear to have resulted from hierarchically controlled, top-down processes, but in fact were not. In nature, ant colonies are prime examples of such a 'design without designers' phenomenon [7].

No requirement for 'global' knowledge

Decentralised systems do not require a top-level party who is responsible for acquiring and maintaining 'global knowledge' within the context of what may be a dynamic, distributed, and highly interactive process. In a complex systems literature, the idea of global knowledge is questioned on practical as well as on theoretical grounds.

In practice, it is often difficult for any one party to actually have sufficient insight, and objectivity to acquire such knowledge. What individual design agents 'know' tends to be influenced by their specific educational and professional backgrounds. Usually this diversity of conceptual outlooks is considered a positive feature of multi-disciplinary design teams.

In theory, the basic idea that there exists global knowledge that is qualitatively more reliable or objective than the subjective knowledge that any single agent might acquire is also questioned.

Multiple resources rather than singular ones

Designers of P2P applications tend to view on-line resources as something that increase in quality with the increased *diversity* of these resources. For instance, if a user is on the hunt for specific music files by a particular artist, it is probably preferable to him if there is a variety of these types of files available for him to download. In this situation, a little bit of data redundancy is also not a bad thing. With diversity of data resources, of course there is the possibility that the quality of some of these resources may not be up to requirements.

Conclusions

Situations that favour the centralised approach

- When one party assumes a central, authoritative role -- which of course happens frequently in collaborative design.
- When design processes and the conceptual organization of product models are well understood, and are unlikely to evolve significantly.
- When design collaborators understand the appropriate role they should undertake, rather than having this role be defined dynamically within a design process.
- When there is a requirement for complete data reliability and coherence -- which usually is easiest to achieve using a centralised data model.

Situations that favour the decentralised approach

- When the intention of the design team is to design in a highly innovative fashion.
- When the design problem presents great conceptual or technical challenges.
- When suitable approaches to a design problem are poorly understood.
- When the team members have little experience with the current type of design problem, or with the process of designing with each other.
- When design team members have dynamically-defined roles rather than fixed ones.

Is P2P relevant for design education?

It appears that P2P is best suited to situations in which the level of design uncertainty is high, while centralised approaches are best when the level of design uncertainty is low. In routine design, uncertainty levels in design are much reduced, compared to when design innovation is desired. This perhaps provides a clue to whether P2P is a suitable technology for design education.

One intent of design education to impart design information and theory, which from the students' viewpoint may appear to be a static body of knowledge. In this case, the level of design uncertainty is low: the student both knows and understands this new material, or she does not. However, there is also a *constructive* aspect to design education, in which students are expected to build a knowledge base and a progression of design proposals that have personal relevance and meaning for themselves.

To design students, due to their inexperience, most design problems appear to be new. They have not yet had the opportunity to develop habituated responses to such problems, as have designers in professional practice. Therefore, design problems all tend to appear to be non-routine for them. Design education based on a constructive theory of education, depends on a student's active participation, and is one in which the levels of design uncertainty is expected to be high. Therefore, in these types of educational processes, P2P technology is expected to be able to play a valuable role.

However, both type of approaches are important in both design practice and design education: one in which design information is seen as a fairly static body of knowledge, and one in which design knowledge is seen as something that requires active construction by interested and involved agents.

References

1. Clark, H.H. Using Language. Cambridge, UK: Cambridge University Press, 1996.
2. Clarke, I. A Distributed Decentralized Information Storage and Retrieval System. Division of Informatics, University of Edinburgh, Edinburgh, UK 1999.
3. Ferber, J. Multi-agent Systems: An introduction to distributed artificial intelligence. Harlow: Addison-Wesley, 1999.
4. Flemming, U., R. Woodbury. "Software Environment to Support Early Phases in Building Design (SEED): Overview." Journal of Architectural Engineering. 1 (1995): 147-152.
5. Gnutella Gnutella News: offering the latest news and information about Gnutella and other file-sharing networks. 2003. Electronic source: http://www.gnutellanews.com/. [2003].
6. Gong, L. "JXTA: A Network Programming Environment." IEEE Internet Computing. 5 (2001): 88-95.
7. Gordon, D. Ants at Work: How an Insect Society is Organized. New York, NY: The Free Press, 1999.
8. Groove_Networks_Inc. Groove Networks. 2003. Electronic source: http://www.groove.net/. [2003].
9. Hogg, T. "Controlling Chaos in Distributed Computational Systems." IEEE Transactions on Systems, Man, and Cybernetics. 1 (1998): 632-637.
10. Jennings, N.R. "Coordination techniques for distributed artificial intelligence." Foundations of Distributed Artificial Intelligence. Eds. G. O'Hare, N.R. Jennings. New York, NY: John Wiley & Sons, 1996. 187-210.
11. Oaks, S., B. Traversat, L. Gong. JXTA in a Nutshell. Sebastopol, CA: O'Reilly, 2002.
12. Snyder, J. Conceptual Modeling and Application Integration in CAD: The Essential Elements. Ph.D. Dissertation, School of Architecture, Carnegie Mellon University 1998.
13. Sun Microsystems, I. Project JXTA: Java Programmers' Guide. 2002. Electronic source: http://www.jxta.org/docs/jxtaprogguide_final.pdf. [2002].
14. Whitfield, R.I., G. Coates, A. Duffy, B. Hills. "Coordination Approaches and Systems - Part I: A Strategic Perspective." Research in Engineering Design. 12 (2000): 48-60.