<Position paper for NSF Science of Design: Software-Intensive Systems Workshop, Nov. 2-4 2003.>

Assumptions and Design Rationale

David C. Brown
AI in Design Group, CS Dept., WPI, MA 01609, USA.

Software tends to need to be changed. It's seen as malleable, as "soft"ware, and hence something that can be changed easily.

This is an illusion, as software tends to be complex, both in terms of the structure, and the mapping of parts to requirements, but also in terms of dependencies between parts of the system. Any change to a software system can be dangerous, producing unexpected results, unless all that complexity is well understood, or well controlled.

Even in those situations, there are still pitfalls. Design decisions are usually informed by a lot of information. Some of it is given in the problem. Some of it comes from the best practices of the field, often imposed as design standards. But some information comes from the experiences of the software's designers and implementors.

That experience suggests architectures to prefer in different situations, reliable/familiar algorithms and methods to use, and ways to best represent and store data. With those preferences come assumptions. Designers tend to assume "normal" situations. They tend to make assumptions about the match between the current design situation and one where their chosen technique worked well before. They tend to make incorrect abstractions across all the situations where a particular techniques worked well before, by assuming that some key detail isn't relevant. These assumptions are not deliberate, but a form of tacit knowledge underlying expert skill, or are rarely stated tacit beliefs about things.

Hence, in order to support changes to software designs, there's a need to make assumptions visible, as changes can violate assumptions. If one could record assumptions, then they could be attached to design decisions. Easier said than done, as most of the assumptions are made without knowing that they've been made.

We need to find ways to infer and collect what assumptions are being made. The user needs to be warned about their (possible) assumptions at design time, and records of the acknowledged assumptions need to be attached firmly to the system.

Assumptions are just one part of the, usually hidden, reason for a design decision. A software engineer could, in theory, provide an extensive and rich description about "why" different parts of the system look the way they do: why a variable has that name, why it is type integer and not real, why a condition was tested at that point, why some code was made into a module, and what tradeoffs were made.

This Design Rationale is hard to collect, mainly because that process is very intrusive. In addition, its value is not immediately obvious, as up until now, retrieval of rationale at the time of change has been its main use. Only altruistic users want to spend lots of effort with no obvious reward. However, if it could be reasoned with, to provide immediate feedback to the system developer about the consistency of the reasons for design decisions, for example, then there would be much more motivation to supply the rationale, and its collection would be easier.

Note that this all requires programmed systems to be tightly tied to the assumptions and other design rationale, so that "touching" some portion of the system automagically activates the associated meta-knowledge. Software changes must be tightly associated with rationale changes.

Short Bio: David C. Brown

Dave Brown is Professor of Computer Science and has a collaborative appointment as Professor of Mechanical Engineering at Worcester Polytechnic Institute. He has B.Sc., M.Sc., M.S. and Ph.D. degrees in Computer Science, and is a member of the ACM, IEEE Computer Society, AAAI, and an elected member of IFIP WG 5.2. He was a consultant on the manufacturing applications of AI at Digital Equipment Corporation.

He is the Editor in Chief of the Cambridge UP journal AIEDAM: AI in Engineering, Design, Analysis and Manufacturing; and is on the Editorial Boards of several Journals, including: Concurrent Engineering: Research and Application; Research in Engineering Design and the International Journal of Design Computing. He is a Vice Chair of the AI in Design conferences, was Local Chair for AID'00, and is a member of the Advisory Committees for the AI in Engineering and IFIP WG 5.2 Conferences.

Dr. Brown's research interests include computational models of engineering design, and the applications of Artificial Intelligence to Engineering and Manufacturing.

He is the author, with B. Chandrasekaran, of the book Design Problem Solving: Knowledge Structures and Control Strategies, Morgan Kaufmann Publishers, Inc., and a co-editor of Intelligent Computer Aided Design, Elsevier Science Publishers B.V. (North-Holland). He is the founder of the AI in Design Webliography.

Recent publications include:

M. E. Balazs & D. C. Brown (2001) "Design Simplification by Analogical Reasoning", In: *Knowledge Intensive Computer Aided Design,* (eds.) Rizzi, Cugini & Wozny, Kluwer Academic Publishers.

J. E. Burge & D. C. Brown (Nov. 2001) "Design Rationale for Software Maintenance", *Proc. Automated Software Engineering Conf.,* San Diego.

J. E. Burge & D. C. Brown (July 2002) "Integrating Design Rationale with a Process Model", *Proc. Workshop on Process Modelling, AID'02: International AI in Design Conference,* Cambridge, UK.

Z. F. Chen & D. C. Brown (Aug. 2002) "Explorations of a two layered A-design system", *Proc. International Workshop on Agents in Design, WAID'02,* MIT, Cambridge, USA

D. C. Brown (Sept. 2003) "Functional, Behavioral and Structural Features", *Proc. Design Theory and Methodology Conference, ASME Design Technical Conferences,* September 2-6, 2003 Chicago, Illinois.

C. Shakeri & D. C. Brown (Sept. 2003) "Constructing Design Methodologies Using Multi-agent Systems", to appear in special issue on *Intelligent Agents in Design, AI EDAM journal,* J.S. Gero & F.M.T. Brazier (Eds.), CUP.

Contact Information:

Dr. David C. Brown
AI in Design Group, Computer Science Dept.
Worcester Polytechnic Institute,
100 Institute Rd., Worcester, MA 01609, USA.

Email: dcb@cs.wpi.edu
Phone: (508) 831-5618 ---- FAX: (508) 831-5776
Web: www.wpi.edu/~dcb/