Participatory Design versus Joint Application Design: Trans-Atlantic Differences in Systems Development

Erran Carmel[1], Randall Whitaker[2], and Joey F. George[3]

[1]The American University
Department of Management, Kogod College of Business Administration
Washington D.C. 20016 USA; 202-885-1928; ECARMEL@AUVM

[2]Umea University
Sweden

[3]University of Arizona
Tucson

JAD (Joint Application Design) and PD (Participatory Design) are two established user involvement methodologies in (respectively) North America and Scandinavia. JAD is a practitioner-derived methodology focusing on structured, facilitated meetings through which user involvement is elicited in systems development. PD stresses the social context of the workplace in workshops in which designers and workers collaborate in design and development activities. JAD and PD are introduced noting their significant ideological differences, but the focus is a comparison of the underlying techniques: participant selection, involvement points, team composition, team interaction, facilitation, structure, and development speed. Suggestions are made for areas in which the two approaches can benefit from each other.

Keywords: JAD, Joint Application Design, user involvement, North American practices, international comparisons, facilitation, creativity

In *PDC'92: Proceedings of the Participatory Design Conference.* M.J. Muller, S. Kuhn, and J.A. Meskill (Eds.). Cambridge MA US, 6-7 November 1992. Computer Professionals for Social Responsibility, P.O. Box 717, Palo Alto CA 94302-0717 US, cpsr@csli.stanford.edu.

Europeans and Americans have often talked past each other about aspects of organizational life. One such aspect is user involvement in systems development. Participatory Design (PD) and Joint Application Design (JAD) are methodologies that have established themselves in Scandinavia and North America (respectively) as influential thrusts in software development, yet there is virtually no cross-fertilization. PD and JAD are simultaneously similar, complementary, and contradictory. Consequently, a careful analysis and comparison of them would benefit those who teach and work in information systems development.

Klein and Hirschheim (1987) refer to such differences as *Information Systems methodological pluralism.* Pluralism offers the double edged sword of offering choices, but leaving the *practicing* designer/systems analyst "in the trenches" in a state of confusion. Like Klein and Hirschheim, we do not believe that there is one correct methodology. We present this discussion and comparison of PD and JAD (which some have classified, erroneously, as "polarized views") in order to help researchers with cross-fertilization and to help practitioners understand the choices when they need to make selections.

JAD and PD are well-known methodologies for operationalizing *user involvement* and *user participation.* Both focus on structured, facilitated interactions between users and designers wherein dynamic group techniques are employed for eliciting and refining ideas. They differ in structure, the degree of facilitators' control, the type and style of user involvement, and point(s) of user involvement. They also differ in their goals -- JAD is intended to accelerate the design of information systems and promote comprehensive, high-quality results, while PD seeks to accentuate the social context of the workplace and promote workers' control over their work and their lives.

This paper begins with a review of the rationale for JAD and PD: the need to involve the users, followed by a review of both approaches. Sections 4 and 5 present the heart of the paper: the comparison of PD and JAD and our discussion of areas in which each methodology can benefit from the other.

1 User Involvement

Since the first multi-million dollar system was rejected because users were not using it, the Information Systems (IS) community has been promoting the truism: the more users are involved in systems development the more successful the resulting system will be. The inverse holds true as well: fewer users involved in the process implies the system will be less successful. There is little empirical proof of this truism (Ives and Olson, 1984), but it is viewed by many in the community as axiomatic and it has become one of the six myths that systems developers use as a guide to design (Hirschheim and Newman, 1991). The concept of user involvement intrigues MIS researchers since it addresses a confluence of complex social factors (Henderson, 1987; King and Rodriguez, 1981; Robey and Farrow, 1982), but research has left open more questions than it has answered.

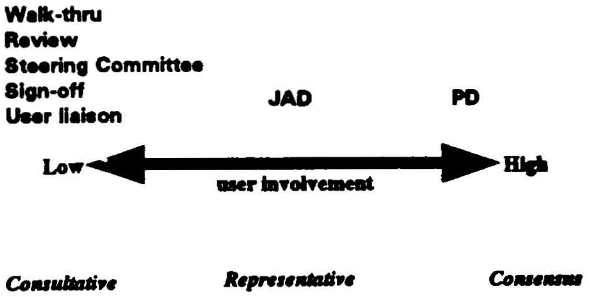

Figure 1: The user involvement continuum

An important framework of user involvement in systems development is that of Mumford (1981). (Mumford's framework addresses the traditional IS custom development context and *not* product development (Grudin, 1991) which is not discussed in her paper, nor in this paper). Mumford delineates three degrees of user involvement forming a continuum (Figure 1):

- *Consultative design.* The IS staff make the decisions. Users are simply sources of information with little to no influence or control. This is descriptive of the one-on-one interview based approaches of user involvement that are still in common use today. In industry, users are usually involved at discrete points in the SDLC (Software Development Life Cycle) using a variety of involvement techniques: sign-off meetings, managerial reviews, steering committees, and user liaisons.
- *Representative design.* Some user representatives, who are either elected or appointed, have influence and affect decisions. JAD falls into this category.
- *Consensus design.* Users have responsibility for the system. The users are involved continually throughout the design process. We place PD in this category (with some qualification since for PD "compromise" is much more descriptive than "consensus," but the thrust of this category is correct). (Other user involvement methodologies that fall into this category are Socio-Technical Systems design (Mumford, 1981), Soft Systems Methodology (Checkland, 1981)).

Prototyping, a common user involvement method, is one that, depending on its use, may span the spectrum of Figure 1.

This paper focuses on the practical implementation of the methodologies in question and hence our interest is in how to operationalize the abstract notion of user involvement, i.e., which methods and techniques to use to get the users involved. Hence our focus is on the methodological implementation factors. Since JAD is not as well known, we devote a larger share of our background review to its introduction.

2 Joint Application Design -- JAD

JAD has become, perhaps, the most common user involvement methodology in North America for two reasons: first, the IS organizations realized that a methodology with a high degree of user involvement would lead to better systems and they found that solution in JAD; second-- by and large-- it worked. The essence of getting the users involved in the JAD methodology is the structured meeting (the session). The JAD user meeting becomes the event around which the rest of the system development activities revolve. The methodology is *participatory* in that the users are queried more (and hence involved more) than users typically were before the advent of JAD. The innovation in JAD, as it has developed today, is that the user meeting is structured, disciplined, and is a foundation of the SDLC. JAD is said to lead to increased quality, reduced costs, and life cycle time reduction.

JAD originated at IBM in the late 1970s (see End Note 1) and began receiving industry attention several years later (Rush, 1985; EDP Analyzer, 1986; Gill, 1987). The interest in JAD has remained exclusively in industry where, by our estimates, there have by now been well over ten thousand meetings labeled JAD (or one of its close cousins that have appeared in the marketplace; See End Note 2). JAD is diffused in the community through manuals (Guide, 1986), books (Wood and Silver, 1989; August, 1991), and continued exposure in the trade press (Martin, 1990a,b,c; Andrews, 1991; Crawford, 1991). We have found that most IS practitioners in large North American organizations have had some direct or indirect exposure to JAD.

As JAD has matured it has become part of industry's "new thinking" about systems development methodologies, or: JAD is a component of *Best Current Practice* (McDonnell Douglas, 1991). The "new thinking" is an amalgamation of the most successful concepts in systems development today: JAD, small teams, rapid prototyping, CASE (Computer Assisted Software Engineering), and rigid time limits (Martin, 1991).

We suspect that the reason for absence of academic interest is that JAD developed and flourished completely outside the academic world. The theoretical basis of JAD is minimal. The JAD meeting methodology has been influenced by the group dynamics discipline and the study of group work and meetings. This makes JAD's contribution one of behavioral underpinnings supporting a technical goal. Indeed, the focus of reported gains, as seen in Table 1, are those of technical progress. The methodologies used for most of the findings in the data are not available and cannot be verified. Some seem to be post-hoc estimates by method advocates.

Table 1: JAD statistics

Time Savings:
- Repair effort p/defect is only 10% in JAD phase as compared with system test phase (Jones, 1991)
- 30-40% in design and 20-30% in implementation (FASC, 1990)
- 15% cycle reduction (Guide, 1986)
- 80% time savings (Boeing Computer Services, 1990)
- 8 hrs/Function Point for traditional method vs. 2.5 hrs/Function Point for JAD (EDP Analyzer, 1986)
- A project at Western-Southern Life: 4 to 6 weeks (Wood and Silver, 1989)

Cost:
- 50% cost reduction (Boeing Computer Services, 1990)
- A project at Texas Instruments: cost avoidance of $0.5 million (Wood and Silver, 1989)

Completeness:
- JAD removes 50% of the defects of the requirements phase and 25% in design phase (percents are not cumulative) (Jones, 1991)
- A project at CNA: 25% increase in num. Function Points (Guide, 1986)

Subjective Evaluation:
- 99% of users would do it again (Guide, 1986)
- 94% of users said they had a better understanding of the system (Guide, 1986)
- 100% of the users said the system would be at least "good" (Guide, 1986)

The Techniques

There is no one structure or definition for JAD. Over the years JAD has evolved to become a framework for "how to run a meeting" (Note the "typical JAD room" shown in Figure 2). Users attend the meeting to define or design an information system. JAD can be viewed as both a technique and a methodology. It is a technique because it is a structure for conducting a design meeting with user participants. It is a methodology because when introduced into the SDLC, JAD sessions/workshops/meetings form the core around which all the activities revolve.

The JAD methodology emphasizes structure and agenda. This is evident in the JAD literature that reads somewhat like cookbooks (JAD, 1986; Guide, 1986; Wood and Silver, 1989; August, 1991). Everything is explained in great detail: "to do" lists are included, as are copies of useful forms.

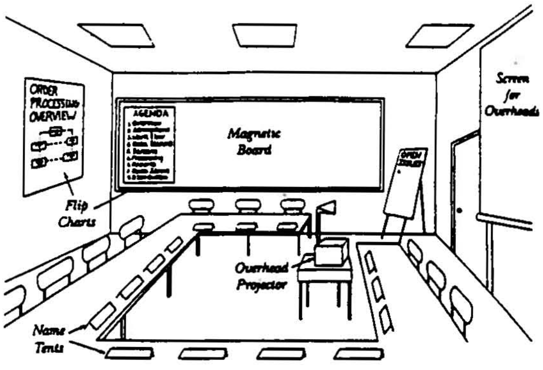

Figure 2: The typical JAD room (from Wood and Silver, 1989)

There are four necessary building blocks for a "good" JAD session:

1. *Facilitation.* A designated leader (or leaders) manages the meeting. Most JAD practitioners consider the meeting leader to be key to process success, more so than the act of gathering the users in one place, the raison d'etre of JAD.

2. *Agenda setting/structure.* The meeting must have a plan of action.

3. *Documentation.* One or more designated scribes carefully document everything in the meeting. Lists are rigorously maintained.

4. *Group Dynamics.* Group dynamics techniques such as those described in Doyle and Straus (1976) are used for inspiring creativity (e.g., brainstorming), resolving disagreements (e.g., airing facts, documenting them as "issues," taking notes), and handling speaking protocols (e.g., enforcing "one conversation at a time").

The conduct of the JAD session changes at different points in the SDLC. JAD sessions early in the SDLC deal with higher level issues: defining objectives, decomposing the domain into smaller functions, defining boundaries and scope, deciding what should and should not be included. In these sessions, participants begin to compile a list of assumptions, constraints and open issues; to target specific people and organizations for tasks; and construct timelines. Lists and other text are often maintained on wall charts, such that, by the end of the session, the walls are covered with flip-chart paper. Some facilitators encourage the users to roam around the room and fill in the wall charts (DEC, 1990), while the more traditional techniques allow only the facilitator writing privileges. Once JAD sessions get into the latter phase -- the design phase-- the users are asked to provide ever increasing detail. At this stage sessions are often longer in duration, perhaps 3-5 days, compared to 1-2 days in earlier stages.

JAD techniques are fairly strict about assigning roles to the various participants in the sessions. For example, the roles in the IBM JAD methodology defined here are fairly typical:

- *Users.* The people who will use the system or are affected by it. The users most knowledgeable about the use of the system should be present at the session(s).
- *Executive Sponsor.* The (user) sponsor defines the overall project purpose and direction, but is usually not present for the entire session, if at all.
- *Facilitator/Session Leader.* A neutral facilitator leads the session. The facilitator (a member of neither the IS team nor the user group) is specifically trained to lead such meetings (many firms provide training specifically for JAD facilitators). The facilitator should have training in group dynamics (or an instinctive flair) and in systems development methodologies. She or he is responsible for all activities: the agenda, the discussion, and documentation of the session results. She or he carefully controls all discussions, guiding and interrupting where necessary.
- *Scribe.* The scribe captures the proceedings of the session: charts, flows, lists and definitions. The "group memory" of the meeting is the scribe's responsibility.
- *IS Project Team.* The IS staff includes analysts, project managers, database personnel, and technical experts. Some, though not all, professionals in the field suggest that they not be involved in the session per se, for they might intimidate the users.

The use of creative visual aids is broadly recognized as helpful for users, many of whom are computer novices, in visualizing the software. For example, M.G. Rush, as a part of the company's week-long JAD facilitator training program, offers a $400 suitcase of custom-designed magnetic color-coded symbols that the facilitator can use during a session, on a whiteboard, for presentation purposes.

The JAD methodology has matured over the years (Table 2) with perhaps the greatest controversy amongst JAD practitioners being computer support in JAD sessions. Today, some parts of some JAD sessions are conducted using CASE tools: graphic tools for depicting data flow diagrams, Entity-Relationship diagrams, state transitions and other diagramming techniques, and screen painters (Kerr, 1989; Semich 1990). Another technology gaining adherents in the JAD community is groupware and electronic meeting systems (Carmel et al, 1991; Carmel, 1992).

In contrast, some practitioners stress the behavioral, managerial and organizational aspects (e.g., Crawford, 1991; Hill, 1991; Kettlehut, 1991). These and many of the JAD practitioners that we have encountered try to minimize the technology that is brought into the JAD meeting room in order to keep the sessions simple and non-threatening.

	1st Generation JAD	Next Generation JAD
Focus	Process	Data and process
System type	Transaction	Transaction and MIS/Decision Support Systems/ Executive Information Systems
Participants	Users only	Users and designers
Meeting Memory	Scribe/ word processing	Design Analyst / CASE
Orientation	Applications level only	Applications level, Enterprise modeling, Functional testing, Engineering approach

Table 2: The generations of JAD (adapted from ATLIS/PRI 1990a).

3 Participatory Design-- PD

Participatory Design (PD) -- widely termed the "Scandinavian model" of systems development -- advocates a much stronger form of user involvement than that of JAD, in which workers participate in designing computer systems they will employ. Czyzewski, et al (1990) outline some key PD tenets: (1) Workers should be given better tools instead of having their work or their skills automated. (2) Users are best qualified to "...determine how to improve their work and their work life." (p. ii). (3) Users' perceptions and feelings about technology are as important as technical specifications or performance indices. (4) Information technology can only be appropriately addressed within the context of the workplace.

Given the wealth of literature on PD, we refer the reader to these for more background (Bjerknes, et al., 1987; Ehn, 1988; Floyd, et al., 1989; Greenbaum & Kyng, 1991; Whitaker, et al, 1991; Wynn, 1983).

PD is still in its infancy in North America, receiving attention primarily in academic circles (e.g., a now bi-yearly PD conference; the CSCW conference-- computer-supported cooperative work;). One must wonder about the degree to which principles engendered in Scandinavia can be employed in North America. Wynn (1983) identifies two common stereotypes about users and science as key problems of general orientation, while Greenbaum (1990b) lists some American perceptions as obstacles to PD acceptance: PD is too idealistic; PD is biased toward workers; PD lacks method or model; and PD designers need to rely strictly on experience. These last two issues may be addressed without recourse to ideological differences and they will be our focus in the rest of the paper.

PD Techniques

There seems to be a disdain for specifying and enforcing "techniques" in the PD community, presumably because this is too closely associated with the engineering scientism PD proponents oppose (Nygaard, 1990). Two principles govern practical implementation of PD principles (Floyd, et al., 1989). The first is *mutual reciprocal learning* by users and designers working together, often through creating "joint experiences" (Kyng, 1991). Training has been an integral part of PD all along -- e.g., the early PD efforts to familiarize workers with computer technology so as to improve their qualification levels. This was extended in the more recent PD practices to include designers' familiarization with users' work settings and activities.

The second principle is *design by doing*, where experimentation, testing and prototyping prevail and there is an emphasis on "hands-on design" and "learning by doing." The PD methodology is very innovative in getting users involved in creative design through various hands-on techniques. Most such practices employ "low-tech" tools. Blackboards, index cards, and Post-It Notes affixed to the wall are common documentation tools during the modeling phase. Later, prototyping is commonly done with (e.g.) cardboard props and HyperCard prototypes. The flexibility required for support of PD practices is problematic for current CASE tools and work is underway toward developing design tools with the capacity for adaptation in response to changes generated from the mutual learning process (Kyng, 1991).

Some sample techniques are described below, divided (arbitrarily, we admit) according to their applicability to either modeling and specification formulation or iterative evaluation of prototypes for the envisioned system.

Modeling I: Visualizing the current workplace

Historical Aspects. (Kensing and Madsen, 1991). This technique involves focus on historical aspects of shared practice to facilitate people in discussing their individual skills, knowledge, and judgment.

Immersion. Kensing and Madsen (1991) suggest designers/facilitators immerse themselves in the workplace, for example, working as clerks at a library for which they are designing a system (Bodker, 1990).

Games (Ehn and Sjogren, 1991). The authors use games (structured actions and interactions) as a method of learning and articulating the practices in the workplace. They describe an example that they developed called Carpentryopoly. Game-like activities are common in PD modeling practice (e.g., Norder et al., 1991).

Consciousness raising sessions. These sessions are derived from the women's movement in the 1970s in which women were encouraged to speak in "their own voice" (Greenbaum and Kyng, 1991).

Modeling II: Visualizing the possible workplace

Future workshops (Kensing and Madsen, 1991). This methodology, derived from Jungk and Mullert (1987), focuses on generating visions of the future workplace. There are three phases: critique (to draw out specific problems with work practice); fantasy (what-if scenarios about the workplace); and implementation (which determines what resources are needed to make realistic changes using user action and to-do lists). The STAR methodology (Ehn, et al, 1990; Ehn and Sjogren, 1991) also follows this general course.

Metaphor-based design (Kensing and Madsen, 1991). Metaphors for current work situations and future scenarios are developed and extended as a conceptual prototyping process.

Site visits. This is a simple and powerful way of getting users to understand the broad spectrum of possibilities (one might consider this immersion in the opposite direction).

Some other techniques mentioned include: storyboarding (best known from advertising), video and multimedia (Fischer, et al., 1990; Harrison and Minneman, 1990; Allen and Pea, 1990), brainstorming (Kensing and Madsen), theatre and role playing, and drawing (Crane, 1990).

Prototyping: Presentation and evaluation of concrete options

Cooperative prototyping (Ehn, 1989; Floyd, et al., 1989; Bodker and Gronbak, 1989; 1991; Thoresen, 1990). Cooperative prototyping involves the user more than the traditional modes of prototyping, in that they actually work with a prototype and experience it. When a breakdown occurs, users and designers actively discuss the reason for the breakdown. Prototyping also supports mutual learning by promoting cooperative communication.

Props and mock-ups. Cardboard system mock-ups (e.g., Kensing and Madsen, 1991; Ehn and Kyng, 1991) are frequently used, inspired by the industrial design notion that the artifact is much more tangible than the idea. Ehn and Kyng argue that many aspects of computer systems can be made equally as tangible.

4 Comparison of JAD and PD

It is first worthwhile noting the similarities between PD and JAD. Both methodologies stress a high degree of user involvement as imperative to good design of information systems. Both represent new thinking on the traditional forms of user involvement that were described in Section 1. Both involve the users in workshops that, to various degrees, encourage creativity and new thinking. Practitioners in both JAD and PD often employ simple, low-tech documentation and visualization methods in their workshops. Both acknowledge the central goals of the other -- JAD proponents speaking of worker empowerment, PD proponents citing benefits of higher quality systems.

The similarities are not just in the approach but in the contexts. Both PD and JAD face considerable obstacles to implementation. There is a reluctance on the part of both IS professionals and executives to increase user involvement or to experiment with new methods and techniques. Once either PD or JAD is accepted, there are numerous local problems in successful implementation: managerial resistance, user conservatism, lackluster workshops, and poor facilitators. Getting user participation is always a test of perseverance; the managers are too busy, the low-level workers are not given approval to spend much time away from their jobs. Lastly, as well-intentioned as both methodologies may be, the users themselves can be uncooperative and unmotivated.

Now to some contrasts (summarized in Table 3). The software engineering approach that effectively serves as the basis for development in North America is based on fixed requirements, communication through documentation, and rules of work enforced through methods -- functional foci which are de-emphasized or dismissed in the PD literature. Conversely, the PD thrusts of mutual learning, joint experiences, and workplace democratization -- what might be termed "social" foci -- do not receive explicit emphasis in JAD. With reference to the methodologies' histories, we might say that JAD represents a movement toward more collaborative practices to enhance the viability of given technical goals. In contrast, PD represents a movement toward more technical practices to enhance the viability of given social goals.

The discussion of several points of departure in techniques follows.

At what point(s) do the users participate?

In theory, both PD and JAD support the entire SDLC. Meetings, sessions, and workshops with the users can be conducted at all points with great frequency. In practice, however, involvement points are not clear cut.

We have found that JAD sessions are most often used only in the requirements stage of the SDLC. This is the stage in which the benefits are considered the greatest. However JAD practitioners stress that multiple JAD sessions need to be run throughout the SDLC-- at many points along the timeline. Two other stages where JAD sessions are often used are for IS Planning (in which we include Enterprise and Business modeling), and the design stage. JAD sessions are also commonly used to help select software packages and sometimes used for other SDLC stages such as system test planning. In an iterative approach (e.g., prototyping), the JAD sessions approach is used at multiple times as newer versions are reviewed.

PD, like JAD, stresses continuous involvement of the users. In one of its forms, cooperative prototyping, this would indeed be a continuous form of development. However, the PD literature does not position itself vis-a-vis the SDLC and hence identifying involvement points is not feasible. PD does not lend itself to the "IS Planning" stage, which as typically defined, has a strong managerial/ executive flavor.

Who are the users and how are they selected?

Generally, the JAD approach has two rules of thumb for selecting user representatives (the term "user" does not indicate rank or position, but simply organizational affiliation). First, all areas of relevant expertise should be represented, minimizing potential for an issue being irresolvable owing to insufficient authority or expertise on hand. Second, JAD user participants should be those the organization can least spare from day-to-day operations; as the saying goes: "If you can't afford to lose her for three days, then that's the person we want." In short, selected users

Point of comparison	JAD	PD
Criteria for validation	Quantitative: economic optima, performance indices, time savings	Qualitative: democracy, mutual learning, mutual education, conflict resolution
Background/theory	Group dynamics, software engineering	Labor relations, group learning
Goal	Improved system	Improved workplace
Roots	- Industry - USA, Canada	- Government, unions, academe - Scandinavia
Current practice	Consultancy for profit	Consultancy on principle
Themes	Teamwork, accelerated design, completeness	Democracy of the workplace, social context, industrial democracy, empowerment, humanization
Focal activity	The meeting: - delimited by time - set agenda	Group processes: - satisfaction delimited - agenda negotiable
Techniques	Structure	Creativity
Perspective on users	- User selection based on competence criteria - users are viewed as one source of knowledge	- User participation *is* mandatory - users are viewed as primary source of knowledge

Table 3: Comparison of JAD and PD

participate to the extent they contribute breadth and depth of expertise to the team -- an instance of Mumford's (1981) "representative" design. In practice, we have found that JAD sessions involve low- and middle-level managers -- the population presumably empowered with decision making authority over a project in a North American context. The managers and supervisors who participate in the JAD meetings are sometimes augmented by non-managerial, operational user representatives -- if the JAD facilitator has enough influence to do this.

PD focuses on low-level, operational users (often excluding management from the process). In the Scandinavian context, empowerment (for project decisions) extends to the operational staff due to co-determination agreements; as such, presence of decision making authority does not distinguish the two methodologies. PD practitioners assume a priori that operational users are the most qualified authorities on improving their workplaces (Czyzewski et al, 1990). Bodker, et al. (1991) suggest workshops be made up of people from similar levels to limit any imbalance in power, but they further admit that sometimes workshops of mixed levels (i.e., with management involved) are unavoidable.

Are the IS technical staff involved?

The traditional IBM JAD approach suggests the technical people not directly participate in the session, so as not to intimidate users or shoot down good ideas. At most, some IS personnel can be allowed to sit in on JAD sessions as silent observers. Many, if not most, JAD practitioners now emphasize cooperation between IS staff and users as members of an ongoing *team*, involved through JAD sessions in a continuous dialogue (Martin, 1990a).

As for PD, the technical presence is limited to designers acting as both facilitators and technical advisors -- which leads to the next point of difference.

Facilitators and their roles

The place of the facilitator in both PD and JAD is pivotal, however the roles are subtly but significantly different. The JAD facilitator tightly controls the meetings and dictates their pace. PD does not use the term "facilitator" but, rather, the term "designer." The dual role of designers in PD as both facilitators and technical advisors contrasts with JAD, where these functions generally remain distinct and specialized. PD designers typically try to (1) collaborate as peers rather than controllers (e.g., Ehn, et al, 1990) and (2) promote maximum independent activity by user-participants. Yet, Bodker, et al. (1991) suggest (in a paradox that they acknowledge) that PD workshop rules be strictly enforced, claiming that strict usage of novel communications breaks traditional patterns and allows time for more people to speak and interact.

The team and its interaction

The PD workshop and the JAD session both foster a sense of cohesion among the group of workers, users, facilitators, designers, and technical staff; yet the goals of collaboration are differently defined.

JAD practitioners emphasize cooperation in the form of a "team" (exemplified by writings of Drucker, 1988; Johansen, et al., 1991). From the PD (socio-political) perspective, Ehn (1990) argues that the concept of the American team is a poor compromise which takes from workers without giving them anything in return.

PD strongly promotes a mutual learning process between members of the group: designers and workers. In the commonly unstructured PD atmosphere, there is little way of describing how the feedback loops operate between designers and the users. As the design progresses, both workers and designers are transformed by learning.

Structure

While JAD is a very structured approach, in which manuals and guides are reminiscent of cookbooks, PD does not insist on invariant structure. Greenbaum and Kyng (1991) criticize the rationalistic approach of systems design with its roots in scientific objectivism and the central notion of analysis through decomposition. They specifically demur from presenting any "step-by-step" PD approach, urging designers to improvise and focus on the process aspects of designing.

More specifically, PD does not structure the entire time span commonly covered by JAD. The PD techniques cited above are practices defining sessions, not entire project phases. The longest developed activity plans in PD are probably those of the futures workshops and STAR methodology (Kensing and Madsen, 1991; Ehn, et al, 1990), but these deal with generating blueprints for the future (organizational requirements specification) -- only a fraction of the total SDLC.

Speed of Development

JAD proponents typically claim that the design and implementation phases are shortened and that maintenance is reduced (Table 1), although we have spoken to some JAD practitioners who concede that JAD increases overall design time. The PD community has not consistently discussed timeframes for their practices, which are typically defined with regard to achieving stepwise goals irrespective of deadlines. The STAR methodology (Ehn, et al, 1990; Ehn & Sjogren 1991) allocates 60 hours for generating organizational desiderata (stopping short of any technical specifications), while Bodker, et al. (1991) concede the PD approach probably lengthens the design phase.

5 Conclusions

This paper does not attempt to bridge the ideological differences between PD and JAD which are plentiful; instead we conclude by highlighting, in this section, areas in which each of the two methodologies can usefully learn from the other (mutual learning in PD parlance). We begin with two areas in the JAD approach that can benefit from PD principles (participant selection and creativity), then discuss one area in which the PD approach can benefit from JAD (structure).

The User Participants. Whether or not one adopts PD's workplace democratization ideal, we have observed numerous JAD sessions in which low-level employees are overlooked as attendees. This results in a meeting room filled with middle managers and supervisors unable to specify details of day-to-day operations (e.g., what 17 fields are needed to fill out form A345). This organizational failure stems in part from an often unjustified lack of confidence that "front-line" workers can meaningfully contribute to the design process. If the victories claimed by PD practitioners could be more clearly demonstrated, this would provide a basis for opening up JAD to worker participation.

Creativity. JAD practitioners utilize many creative techniques and paraphernalia in the design process, from magnetic displays that can be moved around a whiteboard, to prototypes of various kinds. However, all too often, these are minimized, and in practice there are many JAD practitioners that utilize the old methods of long documentation, tedious text, and excessive reliance on flow charting techniques. It is perhaps difficult for many practitioners to be creative in a JAD workshop, just as many teachers lack the flair to be creative in the classroom. PD practitioners tend to have a flair for creativity that many people in the systems development field simply do not have. The creativity of PD techniques is not unique to PD, but can be found in many sources that emphasize "good design" (an excellent one is Gause and Weinberg, 1989). This suggests JAD's creative potential can be enhanced through facilitation training.

Structure The JAD approach emphasizes structure, while the PD approach devotes almost no guidelines to structure. This partially stems from the different set of underlying values that drives the two methodologies. Nevertheless, structure has merits; as Greenbaum and Kyng (1991) note, structure can actually enhance creativity when introduced properly. Introduction of a PD structure summarized in a cookbook format which, to continue the analogy, suggests a dozen ways to cook chicken, would present an important step forward. A PD cookbook would preserve the contextual flexibility that PD practitioners deem important, while at the same time serving to democratize the PD movement by pushing it further into the hands of the average designer/systems analyst in industry.

In closing, we have attempted a comparative examination of two leading user involvement methodologies: PD and JAD. Although there exist contextual differences in their origins and implementation, strong correspondences exist between them. The similarities we have noted suggest a basis for future mutual development, while contrasts suggest points of mutual learning. We hope that our comparisons and conclusions will prove informative in motivating further discussion among practitioners of diverse approaches to user involvement in IS design.

● ● ●

Acknowledgements: We thank Jonathan Grudin, Diane Lockwood, and our reviewers for their thoughtful comments.

End Note 1: More on the history of JAD: JAD was conceived by Chuck Morris and Tony Crawford of IBM in 1977. The JAD approach was loosely derived from another IBM methodology--BSP (Business Systems Planning). The first JAD sessions were held at IBM's Raleigh, North Carolina offices in design of a distribution system called Distribution Center Operations Workshop. This project used the same basic JAD concepts still used today: user participant sessions, magnetic visual displays, and careful documentation of the meeting (Wood and Silver, 1989; private conversation with Andy Algava of IBM; JAD, 1986; FASC, 1990). JAD was adapted by IBM Canada and further refined, later migrating back across the border to the United States in the early 1980s.

End Note 2: A partial list of JAD consulting firms: Andersen Consulting in Chicago; APLAN's Odyssey in Newport Beach, Calif; ATLIS/Performance Resource Inc.'s "The Method" in Rockville, MD.; Boeing Computer Services' Consensus, in Seattle, WA.; Computer and Engineering Consultants' Rapid Analysis, in

Southfield MI.; D. Appleton Co's Requirements Analysis Planning, in Manhattan Beach, Calif.; Digital Equipment Corporation (Europe)'s TOPS and RAMS; JAtec Design Systems' 4RAM; McDonnell Douglas Information Services, in St Louis; M.G. Rush's FAST; The Strategic Advantage in Nile IL.; and WISDM, in Issaquah, WA. Generic names for JAD include: Joint Application Development (IBM), Joint Application Design (IBM) Joint Application Requirements (IBM), Joint Requirements Planning (Martin, 1990a), Interactive JAD (Martin, 1990b), Interactive design (FASC, 1990), Group design (Gill, 1987; EDP, 1986), Accelerated design (Leventhal, 1986), Team analysis (Yourdon, 1989b), Facilitated Team Techniques (Lockwood, 1989).

References

- Allen, C. and Pea, R. (1990). Reciprocal evolution of research, work practices and technology. *Proceedings of Conference on Participatory Design*, Seattle. Palo Alto, California: Computer Professionals for Social Responsibility.
- Andrews, D.C. (1991). JAD: A crucial dimension for rapid applications development. *Journal of Systems Management*, March 1991.
- ATLIS/PRI (1990a). *Performance Resources Inc*, Marketing literature, Rockville, MD, USA.
- ATLIS/PRI (1990b). Conference brochure, International JAD/CASE Conference. *Performance Resources Inc*, Rockville, MD, USA.
- August, J.H. (1991). *Joint Application Design: the group session approach to system design*, Prentice Hall.
- Bjerknes, G., Ehn, P. and Kyng, M. (1987). *Computers and Democracy – A Scandinavian Challenge*, Avebury (UK): Aveshot.
- Bodker, S. (1990). Workshop, *PDC'90*, Seattle, authors' notes.
- Bodker, S., and Groenbaek, K. (1989). Cooperative Prototyping Experiments -- Users and Designers Envision a Dental Case Record System, in *Proceedings of the First European Conference on Computer Supported Çooperative Work (EC-CSCW'89)*, London: Computer Sciences Co., pp. 343-357.
- Bodker, S., and Groenbaek, K. (1991). Design in action: from prototyping in demonstration to cooperative prototyping. in *Design at work: cooperative design of computer systems*, Greenbaum, J. and Kyng, M. (eds.), Hillsdale, NJ: Lawrence Erlbaum Assoc, pp. 197-218.
- Bodker, S., Greenbaum, J., and Kyng M. (1991). setting the stage for design as action. in *Design at work: cooperative design of computer systems*, Greenbaum, J. and Kyng, M. (eds.), Hillsdale, NJ: Lawrence Erlbaum Assoc., pp. 139-155.
- Boeing Computer Services (1990). *CONSENSUS* marketing literature, Seattle, WA.
- Carmel, E. (1992). EMS speeds JAD sessions. *Computerworld*, Volume 26, Num 19, May 11, p. 92.
- Carmel, E., George, J.F., and Nunamaker, J.F. (1992). Supporting Joint Application Development (JAD) with Electronic Meeting Systems: Moving the CASE concept into new areas of software development. *Proceedings of the 25th Hawaii International Conference on Systems Sciences.*
- Checkland, P.B. (1981). *Systems thinking, systems practice.* Chichester: John Wiley and Sons.
- Crawford, T. (1991). People considerations for a more successful JAD. *American Programmer*, 4:1, January 1991.
- Czyzewski, P., Johnson, J., and Roberts, E. (1990). Introduction: Purpose of PDC-90. *Proceedings of Conference on Participatory Design*, Seattle. Palo Alto, California: Computer Professionals for Social Responsibility.
- DEC (1990) RAMS (Requirements Analysis for Manufacturing Systems), *Digital Equipment Corp.*, Europe, marketing literature.
- Doyle M. and Straus D. (1976). *How to make meetings work: the new interaction method*. Dove Books.
- Drucker, P. F. (1988). The coming of the new organization. *Harvard Business Review*, January-February.
- *EDP Analyzer* (1986). Developing high quality systems faster, Volume 24, Number 6, June.
- Ehn, P. (1988). *Work-Oriented Design of Computer Artifacts*, Stockholm: Arbetslivcentrum (Swedish Center for Working Life).
- Ehn, P. (1989). The art and science of designing computer artifacts. *Scandinavian Journal of information systems*, August, pp. 21-42.
- Ehn, P. (1990). *Keynote address, Conference on Participatory Design*, Seattle. Author's notes.
- Ehn, P. and Kyng, M. (1991). Cardboard computers: mocking it up or hands-on the future. in *Design at work: cooperative design of computer systems*, Greenbaum, J. and Kyng, M. (eds.), Hillsdale, NJ: Lawrence Erlbaum Assoc., pp.169-196
- Ehn, P., Molleryd, B. and Sjogren, D. (1990). Playing in reality: A paradigm case, *Scandinavian Journal of Information Systems*, Vol. 2, pp. 101-120.
- Ehn, P. and Sjogren, D. (1991). From system descriptions to scripts for action. in *Design at work: cooperative design of computer systems*, Greenbaum, J. and Kyng, M. (eds.), Hillsdale, NJ: Lawrence Erlbaum Assoc., pp. 241-268.
- Federal Acquisition Support Center (FASC)(1990). *A guide to new requirements and analysis methodologies*, Internal document of Federal System Integration and Management center, Office of Technical Assistance, February.
- Floyd, C., Mehl, W. Reisen, F., Schmidt, G., and Wolf, G. (1989). Out of Scandinavia: alternative approaches to software design and system development. *Human-computer interaction*, vol 4: pp 253-350.
- Fischer G, Grigensohn A., Lemke, A., McCall, R., and Morch, A. (1990) Conceptual frameworks and innovative systems designs for Participatory Design. *Proceedings of Conference on Participatory Design*, pp. 63-81.
- Gause, D.C. and Weinberg G.M. (1989). *Exploring requirements: quality before design*, Dorset House Publishing.
- Gill, A., (1987). Setting up your own design session. *Datamation*, November 15, pp. 88-92.
- Greenbaum, J. (1990). Keynote address, *Conference on Participatory Design*, Seattle. Author's notes.
- Greenbaum, J. and Kyng, M. (1991). *Design at work: cooperative design of computer systems.* (eds), Hillsdale, NJ: Lawrence Erlbaum Assoc.
- Grudin, J. (1991). Interactive systems design. *IEEE Computer*, April 1991.
- Guide (1986). *Joint Application Design*, Chicago: Guide International Inc., Publication no. GPP-147.
- Harrison, S. and Minneman, S. (1990). The media space: a research project into the use of video as a design medium. *Proceedings of Conference on Participatory Design*, Seattle. Palo Alto, California: Computer Professionals for Social Responsibility.
- Henderson, J.C. (1987). Cooperative behavior in information systems planning and design. Working Paper, *Sloan School of Management*, MIT, Cambridge, MA.
- Hill, I.M. (1991). Not all JADs are created equal. *American Programmer*, 4:1, January.

- Hirschheim, R.A. and Newman, M. (1991) Symbolism and information systems development: Myth, metaphor and magic. *Information Systems Research*, 2(1), March 1991, 29-62.
- Ives, B. and Olson, M. (1984). User involvement and MIS success: a review of research. *Management Science*, 30, pp. 586-603
- JAD (1986). *IBM*, Training manual.
- Jones, C. *Applied software measurement.* McGraw Hill, 1991. Johansen, R, Sibbet, D. Benson, S. Martin, A. Mittman, R. and Saffo, P. (1991). *Leading business teams*. New York; Addison Wesley.
- Jungk, R. and Mullert, N. (1987). *Future workshops: how to create desirable futures*. London: Institute for social inventions.
- Kerr, J.D. (1989). Systems design: users in the hot seat. *Computerworld*, Volume 23, Number 8; February 27, 1989, pp. 87-96.
- Kettlehut, M.C. (1991). Avoiding group-induced errors in systems development. *Journal of Systems Management*, March 1991.
- Kensing, F. and Madsen, K.M. (1991). Generating visions; future workshops and metaphorical design. in *Design at work: cooperative design of computer systems*, Greenbaum, J. and Kyng, M. (eds.), Hillsdale, NJ: Lawrence Erlbaum Assoc., pp. 155-168.
- King, W.R. and Rodriguez, J.I., (1981). Participative design of strategic decision support systems: an empirical assessment. *Management Science*, 27, pp. 717-726.
- Klein, H.K. and Hirschheim, R.A. (1987). Social change and the future of information systems development. in *Critical issues in information systems research*, in Boland R.J. and Hirschheim, R.A. (eds). New York: John Wiley and Sons.
- Kyng, M. Designing for cooperation: cooperating in design. *Communications of the ACM*, 34:12, pp. 65-73. December, 1991.
- Leventhal, N.S. (1986). Accelerated design speeds the application development process. *Data Management*, July 1986.
- Lockwood, D. (1989). Using facilitated team techniques to define information systems requirements. *CIS Educator Forum*, 2:1, Fall.
- Martin, J. (1990a). User/analyst teams make the best systems developers. *PC Week*, January 29, 1990.
- Martin, J. (1990b). JAD workshops can help capture design specs. *PC Week*, February 19, 1990.
- Martin, J. (1990c). Success of JAD workshops depend largely on leaders. *PC Week*, February 26, 1990.
- Martin, J. (1991). *Rapid Application Development*. MacMillan.
- McDonnell Douglas (1991). STRADIS methodology, marketing literature, St. Louis Mo.
- Mumford, E. (1981). Participative systems design: a structure and method. *Systems Objectives Solutions*, 1:1, pp 5-19.
- Norder, L., S. Feierbach, B. Nelson, and S. Ogren (1991). Bottom-Up Business Modelling, Kyoto: *Proceedings of the 7th Symposium on Human Interface*, pp. 627-632.
- Nygaard, K. (1990). Keynote address, *Conference on Participatory Design*, Seattle. Author's notes.
- Robey, D. and Farrow, D. (1982). User involvement in information system development: a conflict model and empirical test. *Management Science*, 28:1, pp. 73-85.
- Rush, G. (1985). A fast way to define system requirements. *Computerworld*, October 7, pp. 11-12.
- Semich, W.J. (1990). CASE Power for OS/2. *Datamation*, November 15.
- Thoresen, K. (1990). Prototyping organizational change. *Proceedings of Conference on Participatory Design*, Seattle. Palo Alto, California: Computer Professionals for Social Responsibility, pp. 22-33.
- Whitaker, R., U. Essler and O. Oestberg, (1991). Participatory Business Modeling, Lulea Sweden: *Lulea University of Technology TULEA Report*, 1991:31.
- Wood, J. and Silver, D. (1989). *Joint Application Design*. John Wiley and Sons.
- Wynn, E. (1983). The User as a representation issue in the U.S., in *Briefs, U.C. Ciborra, and Schneider, L.* (eds.), Proceedings of IFIP WG 9.1, Amsterdam: North Holland, pp. 349-358.
- Yourdon E. (1989b). *Modern Structured Analysis*, Yourdon Press.