Partner Technologies: an alternative to technology masters & servants

Kevin McGee and Johan Hedborg
Department of Computer and Information Science
Linköping University
581 83 Linköping, Sweden
kevmc@ida.liu.se
johhe386@student.liu.se

ABSTRACT

There are activities where a dynamic, creative *partnership* among equals seems like an appropriate model of empowerment. Not only do good partnerships seem to help people attain or sustain powerful engagement in their current activities, in some cases they seem to enable people to successfully *enter* new activities.

In this paper we describe our initial work to develop *partner technologies*. Using examples from four partner prototypes being developed, we discuss some design patterns based on the work to date, some insights into aspects of partnership, and conclude with a discussion of some future prospects and potential of partner technologies.

1. INTRODUCTION

When it comes to the design of intelligent technologies intended to empower people, much of it is guided by two central metaphors: technologies as servants or as masters. Servant technologies can be empowering because they reduce or remove work that people find difficult, dirty, or dangerous; master technologies can be empowering because they instruct, inform, remind, cajole, nag, or otherwise force people to do things which are important – but which, for whatever reasons, people do not (or cannot) do without this assistance.

For many kinds of activities and contexts these guiding metaphors do indeed seem useful. But there are activities where a dynamic, creative *partnership* among equals seems like the more appropriate model of empowerment – as in the case of musical co-improvisation by jazz groups, where particular collections of individuals mutually inspire and support each other. Not only do good partnerships seem to help people attain or sustain powerful engagement in their current activities, in some cases they seem to enable people to successfully *enter* new activities.

One consequence of shifting the focus to partnerships and partner activities is that it suggests some serious limitations of objectivist accounts of meaning-making. Much of the work in applied AI is based on a traditional semiotic perspective – one in which goals, constraints, evaluation criteria, indeed *meaning* itself are all (objectively) "pre-given" ("in the world" or "in the mind"). For example, "intelligent tutoring systems" are often implemented with an assumption about the objective (and pre-existing) status of information/knowledge that is to be transferred to the student; likewise, search-engines are typically designed with strong assumptions about the ontological status of the "information to be found." These assumptions may (or may not) be appropriate for those systems – but a number of difficult questions seem to arise as a result of trying to apply the techniques used by such systems to other, more improvisational, domains and activities.

As an example, consider the musician who wishes to improvise with other musicians, and the goal (to the extent it is explicit) is for the activity to be as spontaneous, creative, and enjoyable as some of their more successful sessions. What is "similarity" in such cases – and how does one facilitate it? In what sense does the person explicitly represent (if at all) the qualities of "good play" in advance of new sessions – and what happens to such models as the person has more and more experience? Could it be that the process of experiencing more examples, rather than being a process of "refining" what is meant, is more like a process of *elaborating* and *enacting* what is meant? That is, could it be that rather than "narrowing in on a definition", it involves some combination of undefined praxis and experience that emerges as a co-evolution of activity, participants, and individual experience?

This kind of activity seems different in kind from many traditional computer-science tasks that have (or can be formulated in terms of) clearly defined goals, constraints, and evaluation criteria. In the case of improvisational activities, goals (if that is the appropriate word) may be ill-defined, if at all, and changeable: participants may wish to "improve performance" or "take on as much of the partnership as possible." Similarly, the desire to do "something like" an example activity may not be well-defined, it may not even be *possible* to make it well-defined in any way that it useful for the participants, and it may actually change as a result of participating in the activity.

Some of these challenges are beautifully expressed by Belinda Thom, describing her work to develop musical systems that are intended to co-improvise with human musicians, "Altering the interactive task to one of musical companionship makes it paramount that the agent automatically configure itself so as to reflect its user's particular and momentary style. Simply put, composition – the notion of setting down ideas in advance – makes less sense" [27].

First published at COSIGN-2004
14–16 September 2004, University of Split (Croatia)
0

In this regard, a *non-objectivist* [30, 14] approach to the study of partnership seems promising – one based on the assumption that meaning is not *pre-given* (either in "the world" or in "the individual"), but rather that it emerges out of *action* [32], in this case, *coordinated action*. And a deeper understanding of partnership can, in turn, inform the development of *partner technologies* – technologies that have the potential to empower people by actively *adapting to* and *evolving with* them. Such technologies would not attempt to lead people to pre-determined goals (by applying predetermined constraints) – nor would they necessarily just execute commands that presume the user has predetermined needs, desires, goals, or constraints to be satisfied.

Although the term *partnership* can be used to describe such things as "partnerships with materials" [24, 25], for the purposes of this paper the focus is upon partnerships that have *pro-active* participants, such as when individuals in a jazz group co-improvise. This kind of improvisation differs in significant ways from from so-called "solo improvisation," but (curiously) has no distinct English term, so we have coined the term *symprovisation* to highlight it.

Thus, there are three main aspects of symprovisation – and their relevance to the development of partner technologies – that we wish to explore here. Specifically, issues that arise when participants wish for an activity to have an *unpredictable* outcome (invention, creativity); to be *sustained* in an appealing way (rather than "attain some goal-state"); or to be *engaging* in ways that are difficult to make precise.

In this paper we describe our initial attempts to develop partner technologies, some design patterns based on the work to date, some insights into aspects of partnership, and conclude with a discussion of some future prospects and potential of partner technologies. Before describing our work, we provide a brief overview of non-objectivist AI and a review of some of the work that can be considered broadly relevant to the development of partner technologies.

2. BACKGROUND

There are a number of different non-objectivist concerns, claims, models, and methods. To highlight our interest in *action* – and to avoid the awkwardness of the term *non-objectivist* – in the remained of this paper we will use the term *enactive* [30] to indicate our particular focus.

The work on enactive AI tends to be motivated by insights from either enactive philosophy of mind (constructivism, phenomenology, and the like) – or by concerns about "biologically plausible" models of cognition. One consequence is the exploration of alternative mechanisms, such as neural networks or dynamical systems models (see [30] for examples and review) – or even a reformulation of classical "symbolic" approaches in innovative ways, such as the *deictic* representations of Agre and Chapman [1]. Perhaps the only attempt to combine implementation, enactive models, and some proposal for supporting and empowering people is the controversial work by Winograd and Flores [31] on a model of *decision support systems* influenced by phenomenology, theoretical biology, and speech-act theory.

The issues are many, subtle, and largely unresolved. For present purposes, it is enough to say that much of the debate turns on what can be formalized, the extent to which it can be formalized, and how this should occur. Although it is not possible to give a brief explanation of the key issues, it is important for readers to at least know that most enactive theory is *not* proposing a "subjective" alternative to objective models, where "everything is in the mind" (which is "unconstrained by reality"). Rather, the work is to articulate, and in some cases embody in working systems, an *alternative* to the traditional objective/subjective distinction. Thus, one of the main assertions is that enactive systems are still principled, viable, and robust – but this is not because they have access to objective or universal laws or phenomena.

Turning away from philosophical issues, there is some work in the field of applied AI (artificial intelligence), as well as work on the use of computation to assist in design, that falls broadly within the category of developing partner technologies. Indeed, the organizing taxonomy of cooperative human-machine models used below comes from this latter effort [13]. These categories cut across many domains – there have been partner systems developed for musical improvisation, design support, "just in time" multi-agent decision-support systems, programming partners, and team-sport agents. In the taxonomy that follows, the technology takes a progressively more pro-active role – from "maintaining constraints" to actively attempting to fix situations that it identifies as problematic.

Constraint-Based Paradigms. These systems are designed such that most of the work is done by the person – while the system participates by maintaining various constraints. In some sense, a spreadsheet is one of the simplest examples of this: various relationships are maintained even as a user makes changes to specific cells. More sophisticated versions of this approach include *programming by example* [5]. An interesting example in this regard is the Logo Turtle [20] which imposes constraints on how it can be moved, but which is also an enabling control metaphor. Constraint-based systems include systems to support human-to-human decision-support [31], collaborative learning [6], and creativity [18].

Critic-Based Paradigms. These systems have models of good (and bad) praxis and are able to provide different kinds of feedback as the design progresses (for an overview, see [7]). In many ways, such systems help people by "asking questions", raising concerns, and noting contradictions.

Improver-Based Paradigms. These systems are similar to critic-based systems, but they also include mechanisms for automatically fixing the problems they identify, as in automatic spelling-correction functions in word-processing applications or some of the *programming by example* systems referenced above.

Cooperative Paradigms. Finally, cooperative systems interact dynamically with humans, transforming their work, and making proposals of their own. One of the earliest proposals for such a model was Negroponte's vision of *The Architecture Machine* [19] (although, in recent years Negroponte has moved away from the metaphor of partnership and speaks instead of computational servants, such as butlers). More recently, there is work on systems that co-create works of art [22], music [27], the user's interface [28], or the "browsing experience" [12]. Additionally, there is work on implementing teams of cooperative player-agents [23] and

on the development of joint-control systems; one approach involves extending the subsumption architecture of Rodney Brooks [3], as in the development of semi-autonomous wheel chairs [26] and the simulation of joint-steering system for drivers of automobiles [33].

3. RESEARCH PROBLEM

There are two categories of partner-related activity that are not typically well-served by current research interests nor by conventional AI techniques and mechanisms. First, activities where the goal is to help people discover or invent something unexpected, new, surprising, or interesting – whether it is new artifacts or ideas. Second, activities where the emphasis is on increasing and sustaining ongoing *engagement*, where the *experience* is more important than the attainment of some particular goal. Some of the work on computer-assisted design mentioned earlier is starting to address the first, and some of the work on computational entertainment (games, narrative, and the like) is beginning to take seriously the second (see, for example, [16]).

Researchers working on the development of systems for musical symprovisation and collaborative browsing are among the few attempting to combine the two, though, with its emphasis on implementing explicit models of users and domains, much of that work is broadly objectivist. In this regard, the work of Agre and Chapman requires two brief comments. First, although they use the term "improvisation" (almost apologetically) to describe their model, they do so in order to stress that they are proposing an alternative to *planning* – indeed, they emphasize the *routineness* of many activities and phenomena and are very clear to say that their focus is not on issues of creativity or innovation. And second, although they make explicit reference to enactive theorists and ideas, in their writing there is a frequent implication that they believe their model of improvisation is successful precisely because "the (objective) world" is reliable.

The focus of our research is to contribute to the application of enactive insights to the development of partner technologies symprovisational activities. Our approach is to study examples of good human partnership, implement prototypes that embody possible partner techniques, and, based on the performance of the resulting systems, develop design principles for partner technologies which are then used to guide further study and elaboration of working systems.

For readers not familiar with the discipline of structured symprovisation, it is important to emphasize that it is *not* an activity in which the participants "do whatever they want." In casual usage, *improvisation* may sometimes describe unconstrained invention, but within music, theater, and dance, for example, symprovisation – and improvisation – is very disciplined, structured, and constrained. Indeed, one of the marks of excellent symprovisation is the extent to which real-time invention and creativity manages to resolve constraints. This is widely accepted in the performing arts, but is largely absent in discussions about the development of empowering technology.

To see how different is this perspective from typical AI models of activity, we need only consider the radically different status of *planning*. It is not much of an exaggerations to say that in applied AI, heuristics and other non-algorithmic methods that do not guarantee a definite result are typically considered degenerate; by contrast, in artistic symprovisation, planning is the first and foremost sign of *failure*.

The different attitudes about planning highlight another important distinction: the *enhancement of experience* versus *the attainment of goals*. Crudely put, symprovisation represents an attitude that "we do not choose destinations but headings – with the hope that the travel itself, the destination (if one should materialize), or both are interesting in unexpected and unpredictable ways." Contrast this with a dominant criterion in much of applied computer science: "it is better to arrive reliably and in the most efficient way possible at a clearly-defined destination – or to be informed at once, 'you can't get there from here.' "

For our purposes, these differences have some important consequences for both open-ended ("creative") and goal-oriented partner activities. For open-ended activities, partners need to make valuable contributions to the collective effort of both "what to do next" *and* to the ongoing enactment of "what are we trying to do"? Additionally, good partnerships allow for and support the possibility that even when the activities have more explicit goals, these may not be "shared" by participants in any usual sense of the word.

To explore these differences, the initial architecture we are developing differs slightly from most *cooperative expert systems* or *intelligent tutoring systems*. These architectures typically include components that model *domain knowledge*, various mechanisms for making inferences or drawing conclusions based on the domain knowledge, and facilities for explaining to users how they arrived at their conclusions. In the case of expert systems, the cooperative human-machine interaction is typically predicated on mechanisms for reliably and efficiently attaining pre-determined and well-defined results (or, for identifying new results that meet well-defined constraints). In the case of intelligent tutoring systems, the emphasis tends to shift to mechanisms for eliciting particular classes of pre-determined results by the person being "taught."

Our systems embody various amounts of domain knowledge or explicit constraints, but they are most different in their use of what might be called "difference reduction" mechanisms. In traditional systems, these are used as part of the overall concern to arrive at some pre-determined goal-state(s). In our systems, they are often used as part of *making hypotheses* in the form of *actions* that may satisfy evolving, mutual constraints – as the system has enacted them to that point.

On the other hand, our current approach is similar to work on expert systems in another important way. We are currently trying to identify aspects of *good* partnership – by analogy to the work done in knowledge engineering to identify "domain expertise." Thus, the results we report in this paper are mainly in the form of initial design insights – presented loosely in the form of *patterns* [2] – to potentially important dimensions of partnership. This issue, of making explicit certain aspects of design – indeed the notion of design *at all* – is vexed in the context of non-objective approaches to engineering. We will return to this point in the discussion at the end of this paper.

Before turning to the actual implementations it is important for the reader to have the appropriate expectations. The philosophical issues – and overall ambitions – discussed in the paper so far really *are* our concerns. Nonetheless, applied research on this particular combination of interests is only in its infancy – and the initial implementations are

so primitive it would be easy to dismiss them entirely. However, research needs to start somewhere; and as is the case in many symprovisations, the initial work may be most interesting in terms of what it suggests to do next.

4. PARTNER IMPLEMENTATIONS

In this section we briefly describe four Partner implementations developed to further our understanding of partnership: a Typing Partner, developed to explore some aspects of co-adaptation and support for such goals as "getting better"; a Chess Partner, that attempts to address *dynamic reconfiguration* of action in a different way; a Painting Partner, that is intended to support co-creation of art works in a particular style; and a Flying Partner, that explores a technique for the emergence and co-evolution of skilled performance in a flight simulator.

The work on the Typing Partner suggests that even goal-orient-ed activities can benefit from partner techniques that are relevant to sustaining experience. Similar insights come from the work on the Chess Partner, although the skill here is quite different. One of the insights from work on the Painting Partner is how how mutual meaning-making may arise out of shared activities with no shared frame of reference. Finally, the work on the Flying Partner raises a number of issues about "intrinsically motivating" activities, but that also have elements of danger (for both the participants and others).

It may seem counterintuitive that we explore competitive and challenging activities in the context of partnership. However, many kinds of activities – and many kinds of good partnership – involve a combination of support and provocation.

4.1 Typing Partner

We begin with a description of the Typing Partner, which in many ways is the most conventional. It is intended as a partner to help people improve their skill at touch-typing.

The basic model is as follows: the Partner challenges the person by presenting different characters – and combinations of characters – at different speeds; and the person's typed responses *challenge the partner* to present something appropriate to the person's (changing) skill-level.

Visually, the person sees "blocks" of alpha-numeric characters drop from the top of the screen – and the challenge is to type the characters before the blocks hit the bottom. Initially, characters are randomly generated in equal proportions; based on user-inputs, the presentation of characters become more "adapted" to the particular user. Adaptation takes the form of adjusting the a) actual characters presented, and b) the speed of the presentation. The current Typing Partner is basing its activity on two main parameters: typing *accuracy* and *speed* of response. If a user types the wrong key for a letter – or takes "too long" to type the correct key for a letter – the system presents that letter more often (until the system determines that the letter is "too easy").

The Typing Partner is an initial attempt to give computational form to a proposal by Csikszentmihalyi [4] that an important characteristic of *flow experiences* is that they occur "between boredom and anxiety." By this view, one of the contributing factors to flow experiences is that the activities have the appropriate ratio of *difficulty* relative to a person's *skills*: they are not too easy (boredom) nor too difficult (anxiety). This model raises some challenges for those who would like to facilitate flow: the ratio of skill to difficulty is highly individual – and it is a moving target (that is, as individuals become more skilled, what was once difficult becomes easier). Thus, we are experimenting with adaptive models where partners are constantly adapting themselves or the activity to the individual.

In this regard, there are two aspects of Typing Partner that we wish to highlight here. First, the notion that in partnership, *all* the participants are challenged "equally" – that is, appropriate to their skills and interests. Second, the issue of determining appropriate challenge is more subtle than may at first appear, even in activities based on limited parameters of characters, character-combinations, and speed. To give one example, how should a Typing Partner react to the fact that a person keeps hitting the key that is *next to* the key for the falling character? Initially, it seems appropriate for the Partner to use this information to drop more of the same characters, based on the model that "errors are a way of identifying areas of difficulty." On the other hand, at what point do repeated errors become a signal that the task is too difficult?

Although the partner challenge dimension here is slightly more sophisticated than games such as *Tetris*, it suffers from a similar limitation. Namely, that the model of challenge involves "moving up" a simple hierarchy of challenges. As with *Tetris*, there is a certain level at which people will not be able to improve – and above which they will not be able to perform. At that point, neither *Tetris*, the current Typing Partner, nor other games like this, have any mechanism for introducing different *kinds* of difficulties.

Thus, in addition to Csikszentmihalyi's parameters of skill and difficulty, there seems to be an additional parameter related to *stasis* or repetition. If a person achieves some level of proficiency in an activity where the person stalls at some upper level of skill and difficulty, and if the activity at that level does not change in any other way, boredom will also be likely.

One approach is to add additional parameters. We now turn to an example of a Partner that attempts to address this limitation in a different way.

4.2 Chess Partner

One of the barriers to entry with chess is the complexity of pieces, rules, and situations. Whereas the Typing Partner adapted by changing the speed and frequency of discrete elements (letters), the Chess Partner adapts by creating different kinds of *scenarios*. In particular, the Chess Partner creates different variations on chess – variations with simplified rules as well as piece-movement requirements or restrictions. The central metaphor of the Chess Partner is that of an experienced player who is exploring the kinds of chess scenarios a player can play – and using the results of playing those scenarios to create new scenarios.

Initially, the Chess Partner creates a scenario in which the player must take an opposing piece in three moves, starting from a standard opening position and using the conventional rules of chess. A player is free to move any pieces; if an attempt is made to make an illegal move, the system simply restore the piece to its earlier position (but does not print anything like "you cannot do that"). The Partner keeps track of which pieces are used by the player, and as play continues it makes moves (or reconfigures the board) so that the

player is challenged to use different combinations of pieces (both knights, bishops, queen, etc.) When the Partner determines that variations on this scenario are no longer challenging, it adds an additional requirement: the player *must* take an opponent piece if possible during a turn. As the player becomes comfortable with this scenario, the Partner then starts requiring that the player make moves that will lead the most quickly to a captured piece. Variations on this type of scenario are presented in increasingly more complex ways: particular challenges about taking pieces, choosing the more valuable of pieces, positioning pieces to prepare for capturing pieces, and the like. In addition, the Partner may begin to ask the player to, for example, identify all pieces that can be captured in the current move.

It is important to make clear an essential caveat about the Chess Partner: it is quite, well, un-partner-like. The current implementation has a number of specific scenarios hard-coded and is extremely limited in its ability to move between them, following a fairly rigid progression. We are in the process of developing mechanisms that will allow the Partner to invent and present its own scenarios based on the history of play, but given the limitations of the current implementation, we seriously considered omitting a description of it for this paper. In the end, we decided that, even in its current form, work on the Chess Partner raises a number of questions about what it means to adaptively create appropriate *scenarios* – or *patterns of activity* – rather than the more simplistic model of the Typing Partner which has some fairly crisp parameters for adaptively challenging the typist. Some of these issues are discussed later in the paper.

4.3 Painting Partner

When someone wishes to paint "in the style" of another painter, one of the issues that arises is the fact that different people mean different things when they say they want to do something "like that."

The central image of symprovising with the Painting Partner is as follows. Imagine two friends who have decided to create a painting together without speaking to each other. One person makes an initial sketch, then the other makes some additions to the sketch. This process continues in silence, each person elaborating upon the work of the other. In this particular case, the Partner simulates certain aspects of the painter Mondrian's work. One can imagine it in the following way: a child says to a painter, "I would like to make paintings the way you do." It is not clear what the child means by "the way you do" – so rather than asking questions, the painter decides to learn by painting together with the child. In the process, they shape paintings co-adaptively, with the child refining its request (in the form of new paintings that respond to the history of their collaboration) – and with the painter's understanding simultaneously being elaborated.

The current implementation of the Painting Partner has three different "personalities" for working with someone who wants to paint in the style of Mondrian. The first personality is one that co-creates in the style for which Mondrian is most well-known: bright colors and straight lines. The second personality is more representational, helping a person draw specific images (such as trees) in a style that is recognizably Mondrian. Finally, the third personality is a synchronous Mondrian. Whereas interaction with the two other personalities involves turn-taking, synchronous Mondrian paints at the same time as the person, while also attending to the history of the painting to that point; in this sense, painting with this personality is like real-time jazz improvisation.

When symprovising with Abstract Mondrian, a person can draw lines freely, but everything is converted into straight horizontal or vertical lines. It is also possible to draw polygons and ellipses, although these are "normalized" to the grid. And, by selecting different tools from a palette, the person can specify lines to remove, to resize, to change the color or the texture – as well as specifying various fill properties (color, shading, texture) for the rectangles between the lines. (The inspiration for this Partner was Mondrian's painting, *Broadway Boogie Woogie*.) The Partner begins by making random additions and modifications to the drawing made by the person. The Partner notes various aspects of the history of transformations made by each of them, building different hypotheses about what the person wants and likes, occasionally testing the hypotheses by undoing what the person did last, and the like.

Representational Mondrian works much the same, except that it does not transform everything into vertical and horizontal lines. Rather, it "geometrizes" the person's naturalistic drawings to a certain extent – and in the current implementation, the palette is gray-scale. (The inspiration for this Partner was Mondrian's painting, *Grey Tree*.)

Finally, although Synchronous Mondrian is otherwise similar to the other personalities, it does not "wait for a turn." It interacts synchronously with the human painter, painting at the same time.

Much like the Chess Partner, but to an even greater degree, the Art Partner is not trying to "improve" the person's ability nor to "correct" what the person does. Rather, it is trying to *actively contribute* to the joint-painting in a way that is acceptable to the person (and which meets its own constraints). To the extent that these contributions are also hypotheses about "what the person will accept," these are not used to try and change what the person is (or should be) doing, but rather to elaborate and refine an evolving model of "what is mutually acceptable." Of course, the Painting Partner has some internal model of a particular style (in this case, Mondrian's). However, in a very real sense the Partner is not even trying to develop its own explicit model of what the person *means* by "in the style of Mondrian" – and certainly not in any sense that is intended to be later confirmed or correlated with the person's model (if, indeed, such a mental model even exists in any explicitly identifiable form).

4.4 Flying Partner

The Flying Partner is an implementation of a flight simulator in which the airplane has some built-in intelligence.

One of the major concerns in the design of airplanes is the trade-off between maneuverability and control. In order for a plane to turn, it essentially has to be destabilized. Modern fighter jets are designed to be extremely responsive – and in order to attain this responsiveness, they are remarkably unstable. So much so that some of them are designed such that it is not possible for a human to fly them without computer-assistance. The airplane cannot simply maintain control because it is the pilot who knows where to fly – but, the plane cannot simply yield control entirely since the pilot will be unable to maneuver the plane (except within a

limited range of control) without crashing it. Add to this the different capabilities and skill-levels of different pilots, various weather conditions, and other factors and a number of interesting partnership challenges arise.

The Flying Partner simulation is designed to address this problem in the following way. The Flying Partner initially has complete control of the plane, keeping it stable and on-course. If there are buffeting winds or turbulence, it is the Flying Partner that maneuvers the plane to regain stability. As time goes on, the Partner progressively releases control, bit by bit. Thus, the human pilot is given control over a few degrees of freedom. The Partner "observes" how the pilot handles different situations and, as the pilot demonstrates competence, the Partner yields more and more control.

Consider two of the major difference between the Flying Partner and the other Partner implementations.

First, it embodies a model of "instant participation." In other words, the pilot can begin flying the plane at once. Compare this to, say, the Chess Partner where the model is one of inventing and presenting simplified versions of the activity. The Flying Partner is designed as a joint-control (cybernetic) system. It has a set of internal correlations that allow it to maintain balance, and the challenge is how to mediate over time between the changing skill of the pilot, the intentions of the pilot (flying a loop), the various forces (weather), conditions (angle and altitude of the plane), and the like.

And this leads naturally to a second important difference: the assumption behind this model is that the Partner will always *remain* a partner in this activity. That is, in the other examples it is possible (and perhaps desirable) to participate in the activity without the Partner. In the case of the Flying Partner, the attentiveness to the pilot "never ends"; each pilot/partner pair will enact a different history – and may continuously enact a different balance of competences in the shared activity.

5. PARTNER PATTERNS

In this section we describe some initial Partner design *patterns*, motivated by and illustrated with examples of the Partner implementations, that are intended to highlight some aspects of partnership that should be present.

Support Instant Participation

One of the things that hinders entry into a new activity or domain is that there are not always good ways to *experience* "what it would be like" to participate in them. This makes it difficult to know whether to commit to the (sometimes) years of preparation and pre-requisites necessary to gain such experience. One approach to this problem is seen in the Chess Partner: create simplified versions of the activity that a novice *can* experience. However, such approaches are not always satisfactory – people want to experience more of the "actual" activity.

An alternative approach can be seen in the design of certain equipment for physical training. As an example, there are machines in which one does "chin-ups" by kneeling on a pneumatic pad that carries most of the person's full weight. Thus, the person has the experience of doing the full-motion range of a chin-up – even though the person does not have the physical strength to do so unassisted. Over time, the person can adjust the amount of assistance provided until the strength is enough to do the activity unassisted.

The Flying Partner is one implementation of this approach, where the technique can be characterized as "progressively and adaptively relinquishing control." Thus, rather than "constructing" progressively more challenging scenarios and situations (as the Chess Partner does), the Flying Partner supports the experience of immediately participating in something like the full activity, while progressively *yielding* more and more of the challenges of that activity to the participant.

This model of partnership can be generalized to such activities as riding a bicycle – or even to the development of partners that can empower swimmers or dancers (in this regard, the work in robotics on swing-dance partners [10] is promising). To flesh out this idea in a bit more detail, consider the case of learning to wind-surf, where one of the dominant aspects is *falling over* – in fact, it dominates to such an extent that people are often motivated to quit. Imagine developing a wind-surfboard that is designed to automatically maintain its balance out on the water, indeed, to maintain its balance in the presence of disturbances by a human rider. Thus, when novices first starts wind-surfing, they actually have the experience of immediate participation. And, as with the Flying Partner, they slowly handle more and more of the challenges of the waves, the wind, and the like. Not only that, but the surfboard can be designed to *initiate* additional challenges for riders who gain enough proficiency with wind-surfing unassisted (and who wish for additional challenge).

Of course, there are a number of technical challenges to be addressed in realizing such a vision. However, the current state of the art – Dean Kamen's remarkable "balancing wheelchair," the *iBot* [17], for example – suggests that it is attainable.

Therefore, for activities where a system can be implemented to perform unassisted, consider elaborating the implementation as a Partner that co-adaptively relinquishes control to the other participant(s).

Elaborate Without Prejudice

In good partnerships, partners know how to work with the ideas, proposals, and contributions of their partners. In other words, good dynamic partnership is often characterized by responses along the lines of "that's interesting – and we could do *this* with that." An important concept emphasized in theater symprovisation is "*yes, and*", as in, "yes, and here's something interesting/useful we can do with the results of what you just did." The idea is that when one participant in a symprovisation does something, the others cannot complain, or stop the activity, or otherwise reject what has been done. They must somehow incorporate it, use it, and build on it. Thus, symprovisation at its best is not a struggle by participants to impose pre-conceived ideas about how the events should unfold or what the final result should be; rather it is a dynamic, real-time creative activity that tries to satisfy certain mutually developed and evolving constraints. Among the most important is the constraint that a symproviser's contribution must "work well" in the context of the current activity (and the history of activity leading up to the current moment); and, crucially, it must contribute in a satisfying way to the context of the immediate future in which the next symprovisational activity will take place.

We see this most clearly in the Painting Partner. Consider

how different is its behavior to, say, that of a critic whose role is typically to *pass judgment upon* some performance or result; this can often be characterized as *yes, but* – as in, "yes, but here's what you are doing wrong." The Painting Partner does not reject, deny, or attempt to invalidate the activity of its partner. In this sense, good partners often take the contributions of their partners seriously. And seeing that partners work with their contributions, that those contributions are taken seriously and become useful in the ongoing joint-effort, helps participants develop a powerful sense of the consequences of their actions and their relevance to the ongoing activity. This also differs from what is often meant by "constructive criticism" which implies an outsider giving advice, rather than a participant constructively *contributing* to the elaboration of the joint-project.

Therefore, for activities in which participants *create* things – whether artifacts or ideas – develop partners that elaborate on, rather than critique or reject, the actions of their partners. Said another way, take the actions of partners seriously – and make meaning out of their meaning-making actions.

(Note: it is interesting to consider that even standard servant technologies can use this technique to make them more partner-like. Consider the behavior of some search engines that implement mechanisms one could liken to "assuming the user is asking something meaningful." If the returned results are few, the system assumes this might be a problem. It then looks up more popular results that are similar – by some definition of similar – and when it presents the original results, also presents the option of running the alternative query that will return more results, asking, in effect, "would you like to try this?" Compare this to other systems employed by online merchants that attempt to "suggestive sell" by taking a query that would returns few or no results and simply generating results that contain any of the search-terms. Such systems tend not to feel partner-like, but rather stupid or irritating.)

Allow Partners to Be Challenged

In order to develop symprovisational partners that are capable of rich and expressive interaction is important to implement them in such a way that they can be challenged by the actions of the person. In other words, it is difficult to imagine that people will have fulfilling experiences interacting with systems that are designed with, for example, lookup tables for every possible user-action.

One of the important dimensions of good partnerships is that each participant grows, learns, and changes as a result of the partnership. And this is true even in situations that do not seem at first glance to be partnerships, such as the relationship between a coach and an athlete. We do not mean to labor the point in too much detail, merely to say that the more adaptive the situation between a coach and an athlete, the closer it comes to a partnership in our meaning. In such cases, we would say that all the partners are equal in their relation to the challenges proposed by each other.

Similarly, the work on the Chess Partner is motivated by such concerns. The goal is not to identify and implement some set of appropriate scenarios. Rather, it is to develop a Partner that uses the performance of the player as the creative source for *inventing* appropriate scenarios. That is, to develop a Partner capable of treating the response to the challenges it proposes as challenges *it* must act upon.

Therefore, when activities involve creating something, good partners allow themselves to be challenged by the actions of other participants, rising to these challenge by treating them as sources of inspiration or constraints within which they must contribute.

Respond in Kind

In much of the research on empowerment there is a strong emphasis on making things explicit. Thus, pedagogical research typically looks at ways of explicitly representing, presenting, and helping people to remember and apply explicit rules for different kinds of tasks, whether it is spelling rules, riding a bicycle, or learning a new language. Similarly, much of the work on assistive technologies involves building explicit representations of expert knowledge – and of helping the user by presenting advice and feedback in explicit, often linguistic, form. One obvious alternative is simply to tell people to "just keep trying." Thus, if we take the example of helping someone to learn to ride a bicycle, this usually consists of a combination of "think about this" heuristics and admonitions to "keep trying."

In many activities we find the *actions* of our partners more important and useful than what they *say*. Of course, being explicit is sometimes appropriate, but, on the one hand, there is a risk that the activity drifts away from the activity of concern (to an activity of discussion, analysis, and the like) – and, on the other hand, that the activity of the partner is no help at all.

This is not a proposal for "mindless praxis" – nor even a proposal to always avoid explicit feedback. There are, of course, activities and situations where explicit linguistic understandings and explanations are appropriate, indeed, crucial. However, it is also important to realize that these are not always appropriate – and that, in fact, an overvaluation of such explications can actually hinder praxis for certain activities.

For the most part, the Partner implementations described here represent one alternative: *respond in kind*. The basic idea is that in many cases partners find most useful suggestions and counter-proposals that are similar in kind to the actions they have themselves performed. One aspect of this is to look for and identify, where appropriate, alternatives to strictly verbal or linguistic responses. Another aspect is that these responses may not be entirely symmetrical, as when the Chess Partner plays to create challenging scenarios and the participant plays to answer those challenges. Both these aspects are visible in techniques used by masters in the so-called martial arts [11]. Part of the support consists not in verbal explanations, but rather the creation of new situations in which the person is required to *experience* – and deal with – certain kinds of challenges. Highly skilled teachers are able to adapt the creation of these scenarios to the level and particular challenges of the individual. One could even say that the "cryptic" koans of Buddhist teachers embodies this in its most extreme form, as when it is used to "test the balance" of practitioners [21] with a complex use of language that requires an active demonstration of linguistic competence that is difficult to put into words.

Whatever the form, one of the main advantages of such methods is that practitioners actively work to encourage other participants to "answer back" in the same mode, as when a sports-trainer requires an athlete to respond in the form of *athletic activity*. This is perhaps clearest in the case

of the Painting Partner. Similarly, the Chess Partner is not designed to explicitly *instruct* the person. Rather, the Partner is mostly silent as it presents the different challenges, allowing players to invent their own models of "good play."

Obviously, none of the Partner implementations described here are very sophisticated. Nonetheless, they do embody these principles in simplified form. The Painting Partner responds to drawing activity with drawing activity of its own – and the Chess Partner responds to chess-playing with chess moves that create contexts for more chess-playing. And none of the Partner implementations allow people to "explain what they mean or want" in any other way than through the target activity.

Therefore, even when the activity is in large part non-linguistic, implement Partners that respond in kind – and require other participants to do the same.

Strive For Mutual Benefit

Good partners act to maximize *mutual* benefit [9]. Obviously, there are deep philosophical questions about the extent to which it is reasonable to describe computational systems in these terms. In what sense, for example, is it meaningful to say that these software programs "satisfy their own needs"? One view is that human beings are essentially computational, and thus, there is no difference in kind between the kind of concern expressed by humans and that expressed by computer programs. An alternative view is that there is such a difference in kind that it strains credulity to speak of computer programs expressing concerns, having interests, or caring about mutual benefit.

We are mindful of this debate and do not wish to participate in it here. Rather we simply wish to note that it seems intuitively plausible that the notion of *mutual benefit* is important to partnership – and that future research on partner technologies should address it in some satisfying way.

The key issue is to embody a particular kind of action: "how can I do something in the current context that takes into account my interests and constraints *as well as* those of my partner?" To give this idea more context, consider the difference between asking colleagues for feedback on a single-author paper – and the kinds of feedback and discussions that often occur between two people who are co-authoring a paper. Of course, the feedback from colleagues is important and valuable, but it is also different in kind than the dynamic of co-authors. One difference, we suggest, is that good feedback between co-authors tends to satisfy mutual interests. And, although we are acutely aware of the limits of this claim, it seems reasonable to say that the Painting Partner embodies this to a certain extent.

Therefore, partners should strive to ensure that their actions are of mutual benefit. In the cases where the activity involves enacting an interesting experience, it should be interesting for all; in the case of inventing or creating something, it should be something satisfying to all.

Coordination rather than Reconciliation

To what extent does good partnership involve *unified frames of reference* – "shared" beliefs, goals, and models? When discussing examples of good partnership, there is a common assumption that "what made the partnership work" has something to do with shared values or habits. It is beyond the scope of this paper to enter this topic in any detail, but only to note that "shared models" are not necessary for good symprovisation. Indeed, the results of research on group-cooperation and conflict-resolution by the Harvard Negotiation Project [8, 9] suggest that one of the contributing factors to conflict is an assumption that participants share the same values and goals – and one way to resolve certain conflicts is to develop solutions that do not attempt to *remove* differences, but rather *address them*.

Coming after a pattern on *mutual benefit* this pattern may seem to be its opposite. However, there is no contradiction. None of the Partner implementations described in this paper attempt to create or propose shared frames of reference. Rather, they act based on certain constraints – and to the extent that they develop hypotheses about aspects of their partners, these are provisional and used as the basis for constructive actions (rather than attempts to "get on the same page").

Therefore, good partners do not entirely take over an activity; and if they must (as the Flying Partner may occasionally need to), they do so in ways that all participants agree are necessary.

Leave No One Behind

Good partners are equal in a significant sense; they make contributions (which may be quantitatively or qualitatively different) that result in a process and product where none of the participants feel as if "they are carrying too much" nor that "the other people have taken over." A Flying Partner that simply takes control and flies the plane on behalf of the pilot is no partner. Nor is a Painting or Chess Partner that does all the work (no matter how successful the results).

Of course, partners need not share the same expertise, skills, or background – but there is some significant sense in which they are working together as equals. Although there may be brief instances when one partner "knows more" than another (or teaches), the result of a good symprovisation is something that all the contributors "own." Who, for example, creates the drawings made with the Painting Partner?

Therefore, the actions of good partners are such that they encourage and support further contributions by other partners.

Maintain History

This final point is less of a design pattern and more of a reminder.

If good symprovisation cannot rely on extensive planning, normative rules, or models of an objective world, what *can* it rely upon? One part of the answer is the joint *history* of participant's activity. This may seem so obvious that it does not require comment, but experience shows that when we emphasize that symprovisation involves "acting in the Now," people frequently assume that this involves "forgetting (or ignoring) the past." And this is true even for readers with a background who may be familiar with work on "reactive systems" [3]. It is important to note that the emphasis on symprovisational techniques does not – indeed, *should* not – preclude memory. Although symprovisation is very much about "acting in the current context" (rather than extensive planning), it makes use of the *history* of the symprovisation in various ways.

We need only look at the infamous "Paperclip" in Microsoft Word to see some of the consequences of having a short memory. For all the complaints about the Paperclip, it is a serious attempt to implement something partner-like.

The original model for the Paperclip involved maintaining and using an extensive history of the person's activity; many of the annoying characteristics of the final product can be traced to the fact that, for various reasons, contrary to the researcher's original design, the shipped product maintains an extremely short history of actions.

Similarly, many of the limitations of our current Partner implementations stem from the limited degree to which they maintain and make use of the history of the partnership. Of course, *knowing what (and how) to do* with that history is not trivial. For now, we merely wish to signal our belief that good symprovisational systems will need to make use of their histories in ways that go beyond simply "reacting."

6. CONCLUSION

The work reported here is part of our ongoing research to develop working implementations of partner technologies – technologies that help people enter new domains or activities, or empower them in various ways within their current domains and activities. Although the work to date on partner technologies is still in its earliest stages, and the implementations are so far quite primitive, we are encouraged.

There are a number of obvious ways to continue the work reported here: implementing further the existing prototypes, improving the pattern descriptions, and re-incorporating the insights they represent back into the implementations. This work also involves the exploration of additional classes of domains (music, literature, computer programming, etc.), activities (composing versus performing), and participant epistemologies (blind programmers, dyslexic poets, deaf musicians, etc.) – and, especially, to investigate activities than can or should take place away from the (desktop) computer. Similarly, it will be important to extend the scope of investigations to understand better which kinds of support are applicable in which cases. In other words, in addition to studying different dimensions of partnership (domains, activities, epistemologies) it will also be important to consider the design perspective. For which kinds of activity are the different Partner patterns appropriate – and how would we characterize those activities? The examples in this paper try to indicate this to a certain extent, but much work remains to be done.

This work, in which enactive media studies, design theory, and cognitive science co-inform each other, also raises some larger questions.

It seems that one important aspect of partnership is the ability to elaborate upon the work of one's partners. Contrast this notion of *use* with the notion of *re-use* common in manufacturing and object-oriented programming. By that view, one identifies common components that can be reused in different contexts. But, as famously demonstrated by Kuleshov [15], semiotics and media studies raise a number of issues about what is meant by the transformation of "things" in different "contexts." What are some of the implications for the implementation of partners with the ability to *elaborate upon*, to *repurpose*, to "decontextualize the familiar"? These questions challenge us to follow Kuleshov in an effort to extend media studies, not only as an *analytical* discipline, but as one that is also *generative*.

On a related note, what happens to the notion of *design* as we develop systems that are intended to creatively co-enact? This tension is already present when considering the traditional role of theater-directors with their possible role(s) in improvisational theater groups. Is the role of a director for such groups one of design? If so, in what sense? In literary theory, the "death of the author" is meant to suggest the importance of the *reader* in constructing what is read. It may be that for partner systems we are similarly confronting the potential "death of the developer" – but it is not clear whether (or how) this can give rise to participant creation in a *computational* sense (as opposed to an "interpretive" or "cognitive" sense).

And finally, to what extent is it possible for partner technologies to embody mechanisms for the kinds of phenomena that seem important in symprovisation, such as *shared significance*, *creativity*, and *mutual interest*? It seems very likely that partnerships that are both effective and experientially satisfying are those in which the activity and/or the product of the partnership is important in some non-trivial way to all participants. It is really not clear what this can mean in the case of computational partners. Indeed, for many it is not clear that this is a meaningful concept even in biological-based cognition. Nonetheless, it will be interesting to explore possible cognitive models and mechanisms with the potential to extend or revise traditional formalist representations of creativity and cognition.

It seems appropriate to conclude this paper, which has been concerned with different ways we can empower symprovisation, by noting an intriguing proposal that cognition itself – broadly construed – may be essentially improvisational. In the words of Francisco Varela, "...the nature of the environment for a cognitive self acquires a curious status: it is that which *lends itself* (*es lehnt sich an ...*) to a surplus of significance. Like jazz improvisation, environment provides the "excuse" for the neural "music" from the perspective of the cognitive system involved" [29].

This statement expresses a trend among theoretical biologists to view life and cognition as co-evolutionary processes, and the organism and its environment as a coupled, co-adapting system. Similarly, it may be more useful to frame the study of partnership in terms of mutual adaptation rather than more conventional cognitive models of "adaption as selection for fitness by an independent environment." In future research it will be interesting to explore in more detail some of the ways that co-adaptive models of cognition and symprovisational models of partner technologies can inform each other.

7. REFERENCES

[1] P. E. Agre and D. Chapman. What are plans for? Technical report, MIT, 1988.

[2] C. Alexander. *The Timeless Way of Building*. Oxford University Press, New York, 1979.

[3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.

[4] M. Csikszentmihalyi. *Beyond Boredom and Anxiety: the experience of play in work and games*. Jossey-Bass, San Francisco, 1977.

[5] A. Cypher, editor. *Watch What I Do: programming by demonstration*. MIT Press, Cambridge, MA, 1993.

[6] P. Dillenbourg, editor. *Collaborative Learning: cognitive and computational approaches*. Elsevier Science LTD, Oxford, 1999.

[7] G. Fischer, A. C. Lemke, T. Mastaglio, and A. I. Morch. Using critics to empower users. In *Proceedings*

of the SIGCHI conference on Human factors in computing systems, pages 337–347. ACM Press, 1990.

[8] R. Fisher and S. Brown. *Getting Together: building relationships as we negotiate*. Penguin Books, New York, 1989.

[9] R. Fisher, A. Sharp, and J. Richardson. *Getting it Done: how to lead when you're not in charge*. HarperBusiness, New York, 1998.

[10] S. Gentry and R. Murray-Smith. Haptic dancing: human performance at haptic decoding with a vocabulary. In *IEEE International conference on Systems Man and Cybernetics*. IEEE Press, October 5–8 2003.

[11] E. Herrigel. *Zen in the Art of Archery*. Pantheon Books, New York, 1953.

[12] A. Kerne. Collage machine: Interest-driven browsing through streaming collage. In *Proceedings of Cast01: Living in Mixed Realities*, pages 241–244, September 2001.

[13] S. Kochhar, M. Friedell, J. Marks, S. Sistare, and L. Weitzman. Interaction paradigms for human-computer cooperation in design. In *Conference companion on Human factors in computing systems*, pages 187–188. ACM Press, 1994.

[14] G. Lakoff and M. Johnson. *Philosophy in the Flesh: the embodied mind and its challenge to western thought*. Basic Books, New York, 1999.

[15] R. Levaco, editor. *Kuleshov on Film: writings by Lev Kuleshov*. University of California Press, Berkeley, CA, 1974.

[16] B. Magerko, J. E. Laird, M. Assanie, A. Kerfoot, and D. Stokes. Ai characters and directors for interactive computer games. In D. L. McGuinness and G. Ferguson, editors, *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence*, pages 877–883. AAAI Press / The MIT Press, July 25-29 2004.

[17] B. Metcalfe. More than a wheelchair, the ibot is on the move. *InfoWorld*, November 22 1999.

[18] K. Nakakoji, Y. Yamamoto, and M. Ohira. A framework that supports collective creativity in design using visual images. In *Proceedings of the third conference on Creativity & Cognition*, pages 166–173. ACM Press, 1999.

[19] N. Negroponte. *The Architecture Machine*. MIT Press, Cambridge, MA, 1970.

[20] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York, 1980.

[21] P. Reps, editor. *Zen Flesh, Zen Bones: a collection of Zen and pre-Zen writings*. Charles E. Tuttle Co., Rutland, VT, 1957.

[22] M. Riedl, C. J. Saretto, and R. M. Young. Managing interaction between users and agents in a multi-agent storytelling environment. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 741–748. ACM Press, 2003.

[23] Robocup: The robot world cup soccer games and conferences, http://www.robocup.org.

[24] D. A. Schon. *The Reflective Practitioner*. Basic Books, New York, 1983.

[25] D. A. Sudnow. *Ways of the Hand: the Organization of Improvised Conduct*. Harvard University Press, 1978.

[26] K. A. Tahboub. A semi-autonomous reactive control architecture. *Journal of Intelligent and Robotic Systems*, 32(4), 2001.

[27] B. Thom. Bob: an interactive improvisational music companion. In *Proceedings of the fourth international conference on Autonomous agents*, pages 309–316. ACM Press, 2000.

[28] J. Vanderdonckt, N. J. Nunes, and C. Rich, editors. *Proceedings of the 9th international conference on Intelligent User Interfaces*. ACM Press, January 13-16 2004.

[29] F. Varela. Autopoiesis and the biology of intentionality. In B. McMullin and N. Murphy, editors, *Proceedings of the Conference on Autopoiesis and Perception*, pages 4–13, Dublin City University, 1992.

[30] F. Varela, E. Thompson, and E. Rosch. *The Embodied Mind: cognitive science and human experience*. MIT Press, Cambridge, MA, 1991.

[31] T. Winograd and F. Flores. *Understanding Computers and Cognition*. Ablex Publishing, Norwood, NJ, 1987.

[32] L. Wittgenstein. *Philosophical Investigations*. Macmillan, New York, 1953.

[33] R. Woltjer and K. McGee. A joint subsumption architecture as a framework to address human machine function allocation. In *Proceedings of SIMSAFE 2004: Improving Public Safety through Modelling and Simulation*, June 15–17 2004.