Modeling Human Activity Systems for Collaborative Project Work: An IS Development Perspective

Kam Hou VAT
University of Macau, Macau

fstkhv@umac.mo

Abstract

This paper investigates the idea of human activity systems (HAS) appropriate to the characterization of the purposeful human activities behind the design of suitable information systems (IS) support, especially in the context of group-based project work. Specifically, we are interested in the knowledge context of a group of people collaborating in the peculiar scenario of project development. Our discussion describes a traceable framework of information systems development (ISD), which should accommodate the application of soft systems methodology (SSM) that acknowledges the importance of people in organization. The paper situates our discussion in the action research experience of the author conducting a junior core course of *Software Psychology*, delivered through the pedagogy of problem-based learning (PBL), in our four-year undergraduate program of Software Engineering. We intend to clarify the contextualization of designing IS support in relation to teamwork design. This is done by elaborating the IS design issues through the exposition of the human processes in which, in a specific organizational scenario, a particular group of people can conceptualize their world and hence the purposeful action they wish to undertake. That provides the basis for ascertaining what IS support is needed to undertake the necessary action and how modern information technology (IT) can help to provide that support. We conclude by reiterating the challenge of designing truly relevant HAS systems in which people selectively perceive parts of their world, attribute meaning to what they perceive, make judgment about their perceptions, form intentions to take particular actions, and carry out the action themselves. All these activities carry tremendous connotations in the HAS-based process of IS development for group project work, especially in collaborative project participation.

Keywords: Collaboration, design scenarios, human activity systems, soft systems methodology

Introduction

In the emerging knowledge economy (OECD, 1996), as the possibilities of the information revolution challenge our ongoing efforts in information systems (IS) design, as witnessed by the continuous shift of IS support from a principle of automation (Venkatraman, 1994) to the practice of informating (Zuboff, 1988) and knowledging (Savage, 1990), it is very easy to be blinded to the essential uselessness of our computerized support by the sense of achievement we get from getting them to work. Indeed, many of today's information systems are difficult to learn and awkward to use; they often change our activities in ways that we do not need or want. The problem lies in the IS development (ISD) process (Lederer & Sethi, 1988; Vat, 2004c). Oftentimes, IS designers have to face convoluted networks of trade-off and inter-dependence, the need to coordinate and integrate the contributions of many kinds of experts, and the potential of unintended impacts on people and their social institutions. If project development for IS support is concerned with understanding, designing, evaluating and implementing interactive computer systems to match the needs of people, it is convinced (Vat, 2004b) that through maintaining a continuous focus on situations of and consequences for human work and activities, IS designers could become more informed of the problem domains, seeing usage situations from different perspectives, and managing trade-offs to reach usable and effective design outcomes. However, getting users' work right involves capturing and accommodating users' emergent (or subject-to-change) analytical activities, which are open-ended yet integrated and opportunistic yet coherent. IS developers must understand the intertwined regularities and idiosyncrasies of human activities and create software that support the right moves and degrees of agility at the right times and places and for specific purposes.

In this regard, the problem of designing IS support for knowledge work should never be thought of as something to be defined once and for all, and then implemented. Instead, it must be based on the observation that all real-world organizational problem situations contain people interested in trying to take purposeful action (Checkland, 1981, 1999). And the idea of a set of activities linked together so that the whole, as an entity called the human activity systems (HAS) from the viewpoint of Soft Systems Methodology (SSM) (Checkland & Holwell, 1998; Checkland & Scholes, 1999) could prove useful. In fact, as far as pursuing a purpose is concerned, a HAS model could be considered as a representative organizational scenario for exploring any situation-specific IS support, which is never fixed once and for all. In this paper, we are particularly interested in the IS design scenario for a group of people engaged in collaborative project work. Our discussion is elaborated in terms of a curriculum action research report relating our students' PBL (problem-based learning) course experience (Vat, 2000, 2001) to the IS design issues behind creating the necessary HAS models (Vat, 2005) to represent the various organizational dynamics of teamwork involvement (Vat, 2004f). Of particular interest here is a proposed actionable ISD framework, in which SSM is integrated as the essential strategy of requirements elicitation and analysis.

Research Background

The teaching of *Software Psychology*, or more properly renamed as human-computer interactions (HCI) (Vat, 2000, 2001) in the undergraduate curriculum has always been a challenge as it is composed of such a mix of elements as human factors, user expectations, man-machine interfaces construction, cognitive psychology, computer science, and those latest developments on contextual design in interactive systems. In the case of the author's teaching experience, since 1998, the pedagogy adopted to deliver such a course has been shifted from a conventional instructivist approach to the constructivist method of problem-based learning (PBL) (Greening, 2000; Ryan, 1993). Besides, with the increasingly accumulated course material to cover in a single semester, the idea of scenario-based design (Carroll, 2000) has also been incorporated in 2000 with an attempt to help undergraduate Software Engineering students deepen the idea that HCI is concerned with understanding, designing, evaluating and implementing interactive computer systems to match the needs of people. It is our experience that the constructivist's ideas of problem-based learning (PBL) (Barrows, 1986) revolving around a focal problem, group work, feedback, skill development and iterative reporting, with the instructor playing the coach by the side, guiding, probing, and supporting student-groups' initiatives along the way, could help students develop a unified team-based approach to better manage the underlying software requirements. Methodically, we need some working scenarios to try out some iterative process involving researchers (instructor) and practitioners (students) acting together on a particular cycle of development activities, including problem diagnosis, action intervention, and reflective learning. Particularly, our action research approach should involve evaluating how well the students playing the role of practitioners, could function as self-directed work teams (SDWTs) of software professionals, following the constructivist's tenets of PBL, in performing group-based software development for a specific user scenario. Against this backdrop, the use of soft systems methodology (SSM) (Checkland & Holwell, 1998; Checkland & Scholes, 1999) has demonstrated quite a promise in enhancing the student-practitioners' learning to deal with the design difficulties typified in the complex domain of ill-defined problem situations.

The Issue of Team-Based Collaboration for IS Work

Those who are engaged in the attempt to build IS support for specific user situation, are involved in the delicate business of creating a conglomeration of various human activity systems. This endeavor requires the effort and commitment on the part of everyone involved, as well as a good imagination in the mind of those charged with directing its implementation (Fisher, 2000; Ginac, 2000; Gregory, 2000). In the instance of a project team charged with the mission of creating IS support for group-based project work, what makes the team work is people's mutual understanding of their own and others' interests and purposes, and the recognition that their interests are somehow bound up in doing something to which they all contribute. In a strict sense, it is in the course of interaction that people's sense of purpose and even their contributions, come to be defined. As collaborators in an IS team, we face the tremendous challenge of how team members move from being individual spokespeople to a unified, collaborative body. In his book on group decision-making, Sam Kaner (1996) calls the transition from the divergent zone of the individual to the convergent zone of the team member the "groan zone." In a team, even though every member wants to contribute to success and to get the project going, each has a different perspective, a different experience, or a different context to bring to the project. Each person's thinking is divergent, bringing diversity to the process, but not much agreement. Convergence occurs as the group's individual ideas are integrated into a whole solution. This process of integration does not entail compromise (Fisher, Ury & Patton, 1991), in which every one gives up something and no one is happy with the result, nor does it mean that everyone is in complete agreement. What convergence means is that everyone has participated and will support the final decision. Kaner calls this period between divergence and convergence the groan zone because it is the time during which team members groan and complain. In the divergent zone, most group members voice their opinions to make sure their ideas being heard by the group. In the groan zone, however, an individual digs behind other people's ideas to try to uncover their reasons, assumptions and mental models. Difficult problems and wrenching decisions cause teams to spend time in the groan zone because of the required interchange, sharing, and resolution of ideas, and viewpoints. Likewise, the groan zone is also used to describe the transition zone in which innovative, emergent (or unexpectedly desirable) results are generated. Indeed, collaborative groups, especially those in fast-paced environments, groan a lot. They struggle to create the services that converge on the mission profile. They struggle to integrate their own, and others' diverse perspectives.

The PBL Paradigm for Collaboration

To allow our students to experience the groan zone themselves in the process of collaborating with their project work, the constructivist idea of problem-based learning (PBL) (Barrows, 1986; Greening, 2000; Ryan, 1993) has been adopted in the enactment of our junior core course of *Software Psychology* since 1998. Problem-based learning, briefly stated, is designed to actively engage our students, divided in groups, in opportunities for knowledge seeking, for problem solving, and for the collaborating necessary for effective practice. At the heart of PBL is a set of group-based activities, including climate setting, starting a problem, following up the problem, and reflecting on the problem. A brief description of the PBL model of collaboration could be presented as follows:

- *The Climate Setting Phase.* At the outset, before the PBL group work begins, students must get to know one another, establish ground rules, and help create a comfortable climate for collaborative learning. Meeting in a small group for the first time, students typically introduce themselves, stressing their academic backgrounds to allow the facilitator (instructor) and each other to understand what expertise might potentially be distributed in the group. The most important task is to establish a non-judgmental climate in which students recognize and articulate what they know and what they do not know.

- *The Problem Initiation Phase.* The actual PBL episode begins by presenting a group of students with minimal information about a particular problem. Students then query the given materials to determine what information is available and what they still need to know and to learn to solve the problem. During this phase, students typically take on specific roles. An example is the scribe, who records the group's problem solving, including listing the facts known about the problem, students' ideas, additional questions about the problem, and the learning issues generated throughout ensuing discussion. Such written record helps the students keep track of their problem solving and provides a focus for negotiation and reflection. Throughout the problem-solving process, students are encouraged to pause to reflect on the data collected, generating additional questions about that data, and hypothesizing about the problem and about possible solutions. Early in the PBL process, the facilitator may question students to help them realize what they do not understand. As students become more experienced with the PBL method and take on more of the responsibility for identifying learning issues, the facilitator is able to fade this type of support, or scaffolding. After the group has developed its initial understanding of the problem, the students divide up and independently research the learning issues they have identified. The learning issues define the group's learning goals and help group-members work toward a set of shared objectives. These objectives can also help the facilitator to monitor the group's progress and to remind members when they are getting off course, or alternately, to ask if they need to revise their goals.

- *The Problem Follow-up Phase.* In the problem follow-up phase, students re-convene to share what they have learned, to re-consider their hypotheses, or to generate new hypotheses in light of their new learning. These further analyses, and accompanying ideas about solutions, allow students to apply their newly acquired knowledge to the problem. Students share what they have learned with the group as they interpret the problem through the lens of their newly accessed information. At this point, it is important for the students to evaluate their own information and that of the others in their group. In the PBL group, information is not often accepted at face value. Students must discuss how they acquired their information and critique their resources. This process is an important means of helping the students become self-directed learners.

- *The Problem Reflection Phase.* During post-problem reflection, students deliberately reflect on the problem to abstract the lessons learned. They consider the connections between the current problem and previous problems, considering how this problem is similar to and different from other problems. This reflection allows them to make generalizations and to understand when this knowledge can be applied. Finally, as the students evaluate their own performance and that of their peers, they reflect on the effectiveness of their self-directed learning and their collaborative problem solving.

The Scenario of IS Design in *Software Psychology*

At each semester when *Software Psychology* is offered, our course scenario begins when the instructor helps the class evolve into its team-based organization. Typically, students embark on the PBL cycle of learning through organized groups of 4-6 members (one being the team leader). Each PBL group will be given a dual role to explore as client and as developer within a specified period of time. Namely, each team, acting as the developer, is to complete an interactive system design and prototype for another team acting as the client. Yet, the same team is the client of another group, responsible for clarifying the project, and resolving ambiguities as they arise, but in any pair of PBL teams (say, A and B), they cannot be the client and developer of each other at the same time. It should be noted that an even number of teams is important to facilitate pair-wise client-developer interaction. Meanwhile, the instructor, more appropriately called the facilitator, acts as project sponsor for each client team, and as project supervisor for each developer team. Each client team is handed a design project by the sponsor. It is then given some inception time to elaborate on the specifics of the project. At the end of the inception period, each client team is assigned a developer team from among the remaining client teams. After a developer team has been identified, the working and performance of the developer team is guided and monitored by the project supervisor played by the instructor. In a typical semester, there might easily be six to ten PBL teams of students, with each team composed of four to six members each.

Essentially, each design project invites our PBL student-groups to embark on a journey to develop some interactive system that meets customers' real needs in Web-based development. The general requirement is for each PBL team to create and maintain a review Web-site to keep all team members up-to-date on all possible aspects of the project. It is also where the PBL team will work (report) collaboratively on the project. Through the review Web-site, our PBL teams can conduct reviews with their clients, who can view their project in progress, give feedback on a design, get in touch with the developer PBL team, and check the project schedule. The review Web-site contains numerous information such as: the roles and responsibilities of the project team, contact information for all team members, the project mission, the vision document, the project schedule, and all design reviews. It is designed that the first thing our PBL teams have to learn is a systematic approach to eliciting, organizing, and documenting the requirements of the system to be built for the client team. Also important is a process that establishes and maintains continuous agreement between the client and the developer teams (Curtis, Krasner, & Iscoe, 1988) on the changing requirements of the system. Individual PBL teams have to understand users' problems in their culture and their language and to build systems that meet their needs. Practically, the HCI context for the course is designed around four core development processes to be experienced by our PBL student-groups within the semester's duration constraint.

- *Analyzing the Problem*. This involves a set of skills to understand the problem to be solved before application development begins. It is the process of understanding real-world problems and user needs and proposing solutions to meet those needs. We consider a problem as the difference between things as perceived and things as derived (Gause & Weinberg, 1989). Accordingly, if the user perceives something as a problem, it is a real problem, and it is worthy of addressing.

- *Understanding User Needs.* Software teams are rarely given effective requirements specifications for the systems they are going to build. Often they have to go out and get the information they need to be successful. Typical methods include interviewing and questionnaires, requirements workshop, brainstorming and idea reduction, storyboarding, role playing, and prototyping. Each represents a proactive means of pushing knowledge of user needs forward and thereby converting fuzzy requirements to those that are better recognized.

- *Defining the System.* This describes the process by which the team converts an understanding of the problem and the users' needs to the initial definition of a system or application that will address those needs. Our PBL teams should learn that complex systems require adaptive strategies to organize information for requirements. This information could be expressed in terms of a hierarchy, starting with user needs, transitioning through feature sets, then into the more detailed software requirements.

- *Managing the Project Scope.* Project scope is presented as a combination of the functionality to be delivered to meet users' needs, the resources available for the project, and the time allowed in which to achieve the implementation. The purpose of scope management is to establish a high-level requirements baseline for the project. The team has to establish the rough level of effort required for each feature of the baseline, including risk estimation on whether implementing it will cause an adverse impact on the schedule.

Throughout the course delivery, each PBL team is required to present their work in progress, and lead class forums to elicit students' discussions. The team leader, equivalent to project manager, has to coordinate the team activities, and ensure effective team communications. And team members have to help set the project goals, accomplish tasks assigned, meet deadlines, attend team meetings and participate in editing project documents and integrating work-products to be combined as the final project report. At the end of each project milestones, each member of the respective PBL teams is required to make a presentation of his or her project involvement, with a question and answer session for the client team and the whole class. The instructor, acting as the project sponsor for each client team, and as the project supervisor for each developer team, designs the necessary scenario details (Whitten, 1995) to guide, motivate and provide feedback to the PBL groups. Also, the instructor has to evaluate how well students perform in the PBL groups (Doyle & Straus, 1982), and how well such groups behave as SDWTs (self-directed work team) in managing software requirements (Conklin & Burgess-Yakemovic, 1991), and provide the necessary adjustments following the ideas of soft systems methodology (SSM) (Checkland & Scholes, 1999; Wilson, 2001).

Enacting SSM as a Scenario-Based Learning Process

A very important activity for each developer PBL team is to actively engage its client PBL team in helping solve each of the four core processes described in the previous section to ensure the quality and timeliness of the software outcomes. Undeniably, setting up interactive IS support is a social act in itself, requiring some kind of concerted action by many different people; and the operation of an IS entails such human phenomena as attributing meaning to manipulated data and making judgments about what constitutes a relevant category. In this regard, the use of scenarios (Vat, 2004b, 2004d) in the creation of IS support, can be seen as a process which learns its way to the meanings which characterize an organizational context. John Carroll (2000) characterizes scenarios as concrete stories about use through which IS architects could envision and facilitate new ways of doing things and new things to do. Specifically, scenarios provide a vocabulary for coordinating such development tasks as: understanding people's needs, envisioning new activities and technologies, designing effective systems and software, and drawing general lessons from systems as they are developed and used. Namely, scenarios help IS designers analyze the various possibilities by focusing first on the human activities that need to be supported and allowing descriptions of those activities to drive the quest for correct problem requirements.

Still, the important point is that we must be conscious of the fact that any scenario of HAS (human activity systems) models created are merely abstract logical machines for pursuing a purpose (Checkland, 1979; 1984), defined in terms of declared worldviews, which can generate insightful debate (or rational discussion) when set against actual purposeful action in the real-life situation.

The implicit belief behind constructing the HAS models is that social reality – what counts as facts about the social world inside an organization, say the client PBL teams of students – is the ever changing outcome of a social process in which human beings (individual client team members) continually negotiate and re-negotiate, and so construct with others (members of the developer team) their perceptions and interpretations of the world outside themselves (expected IS support for specific activities), and the dynamic rules for coping with it (cooperating or manipulating). In the process, we do not expect to discover unchanging social laws to set alongside the laws of natural sciences. Rather, researching social reality in the context of IS development then becomes an organized discovery of how human agents (both the client and developer team members) make sense of their perceived worlds, and how those perceptions change over time and differ from one person or group to another.

Nevertheless, the basic shape of the scenario-based learning approach for the developer PBL team could be described as follows: Find out about the problem situation that has provoked concern; Select relevant concepts that may be integrated into different human activity systems; Create HAS models from the relevant accounts of purposeful activity; Use the models to question the real-world situation in a comparison phase. The debate initiated by the comparison normally entails the findings of accommodations between conflicting interests, that is to say, situations that may not satisfy everyone, but could still be lived with, enabling action to be taken. Oftentimes, the purpose of the debate is to collectively learn a way to possible changes (improvements) to the problem situations, by activating in the people involved (client PBL team), a learning cycle, which counts on their ability to articulate problems, to engage in collaboration, to appreciate multiple perspectives, to evaluate and to actively use their knowledge. It is worthwhile to notice that taking the purposeful action would itself change the situation, so that the whole cycle could begin again, and is in principle never ending. Likewise, through scenarios of HAS models, IS architects could provide help in articulating the requirements of specific technical support through operating the learning cycle from meanings to intentions to purposeful action among the specific group of organizational members.

Some Knowledge Models of Human Activity Systems

Essentially, the use of scenarios in IS work always assumes that the purpose of creating an organized IS support, is to serve some real-world action; namely, organized provision of information is always linkable to action (Checkland, 1983). Thereby, in scenario-based design (Vat, 2004d), when a real-life problem situation arises, our typical approach of enquiry is to formulate some HAS models of purposeful activities, which it is hoped will be relevant to the real-world situation, and use them by setting them against perceptions of the real world in a process of comparison. That comparison could then initiate debate leading to a decision to take purposeful action to improve the part of real life, which is under scrutiny. Thus, designing an IS will require attention to the purposeful action which the IS serves, and hence to the meanings which make those particular actions meaningful and relevant to particular groups of actors in a particular situation (Vat, 2004a). This idea of learning the meanings, by which people sharing a human situation seek to make sense of it, is a significant feature of SSM. What follows is our appreciation of three important knowledge models considered as indispensable in the daily operations of collaborative project development among our PBL student-groups: the personal model, the social model, and the organizational model. Of particular interest here is the idea of appreciative settings, which according to (Vickers, 1972, p.98), could refer to the body of linked connotations of personal interest, discrimination and valuation which we bring to the exercise of judgment and which tacitly determine what we shall notice, how we shall discriminate situations from the general confusion of ongoing events, and how we shall regard them. The word "settings" is used because such categories and criteria are usually mutually related; a change in one is likely to affect others.

The Personal Model

Consider us as individual conscious of the world outside our physical boundaries. This consciousness means that we can think about the world in different ways, relate these concepts to our experience of the world and so form judgments (Vickers, 1965) which can affect our intentions and, ultimately, our actions. This line of thought suggests a basic model for the active human agent in the world. In this model we are able to perceive parts of the world, attribute meanings to what we perceive, make judgments about our perceptions, form intentions to take particular actions, and carry out those actions. These change the perceived world, however slightly, so that the process begins again, becoming a cycle. In fact, this simple model requires some elaborations. First, we always selectively perceive parts of the world, as a result of our interests and previous history. Secondly, the act of attributing meaning and making judgments implies the existence of standards against which comparisons can be made. Thirdly, the source of standards, for which there is normally no ultimate authority, can only be the previous history of the very process we are describing, and the standards will themselves often change over time as new experience accumulates. This is the process model for the active human agents in the world of individual learning, through their individual appreciative settings. This model has to allow for the visions and actions, which ultimately belong to an autonomous individual, even though there may be great pressure to conform to the perceptions, meaning attributions and judgments that belong to the social environment, which in the case of project development, is the project team itself.

The Social Model

Although each human being retains at least the potential selectively to perceive and interpret the world in their own unique way, the norm for a social being is that our perceptions of the world, our meaning attributions and our judgments of it will all be strongly conditioned by our exchanges with others. The most obvious characteristic of group life is the never-ending dialogue, discussion, debate and discourse in which we all try to affect one another's perceptions, judgments, intentions and actions. This means that we can assume that while the personal process model continues to apply to the individual, the social situation will be that much of the process will be carried out inter-subjectively in discourse among individuals, the purpose of which is to affect the thinking and actions of at least one other party. As a result of the discourse that ensues, accommodations may be reached which lead to action being taken. Consequently, this model of the social process which leads to purposeful or intentional action, then, is one in which appreciative settings lead to particular features of situations as well as the situations themselves, being interpreted in specific ways by standards built up from previous experience. Meanwhile, the standards by which judgments are made may well be changed through time as our personal and social history unfolds. There is no permanent social reality except at the broadest possible level, immune from the events and ideas, which, in the normal social process as evidenced by a role taken up by two or more team members, continually change it.

The Organizational Model

Our personal appreciative settings may well be unique since we all have a unique experience of the world, but oftentimes these settings will overlap with those of people with whom we are closely associated or who have had similar experiences. Tellingly, appreciative settings may be attributed to a group of people, including members of a team, or the larger organization as a whole, even though we must remember that there will hardly be complete congruence between the individual and the group settings. It would also be naïve to assume that all members of an organization share the same settings, those that lead them unambiguously to collaborate together in pursuit of collective goals. The reality is that though the idea of the attributed appreciative settings of an organization as a whole is a usable concept, the content of those settings, whatever attributions are made, will never be completely static. Changes both internal and external to the organization will change individual and group perceptions and judgments, leading to new accommodations related to evolving intentions and purposes. Subsequently, the organizational process will be one in which the data-rich world outside is perceived selectively by individuals and by groups of individuals. The selectivity will be the result of our predispositions to "select, amplify, reject, attenuate or distort" (Land, 1985, p. 212) because of previous experience, and individuals will interact with the world not only as individuals but also through their simultaneous membership of multiple groups, some formally organized, some informal. Perceptions will be exchanged, shared, challenged, and argued over, in a discourse, which will consist of the inter-subjective creation of selected data and meanings. Those meanings will create information and knowledge which will lead to accommodations being made, intentions being formed and purposeful action undertaken. Both the thinking and the action will change the perceived world, and may change the appreciative settings that filter our perceptions. This organizational process is a cyclic one and it is a process of continuous learning, and should be appropriate to the scenario of group-based project work, especially when there are more than one PBL teams involved.

Conceiving the Collaborative Context for IS Support

It has been our experience in conducting *Software Psychology* that a Web portal is needed for managing collaborative project development. This portal should lead to a Web-based organizational space for each project, $OS_{Project}$, which renders a number of peculiar services to client and developers (MacLean, Young, Bellotti & Moran, 1991), in the form of distributed applications customizable to their project-related activities. In a specific project context, there must also be a number of Web-based collaborative spaces, CS_{Group}, to enable group-based project work to be performed. For example, there have to be a CS_{Group} for each of the client PBL team and the developer PBL team. Besides, to support the interactions among project members, and between the project manager (PBL team leader) and specific team members, the provision of a personal electronic space for each of the project members such as client, project manager and team members, $PS_{Participant}$ (PS_{Client}, or $PS_{Developer}$) is essential to facilitate individual work performance. The linkages from the project space, to the respective collaborative spaces, to the individual personal spaces, must be closely updated to facilitate the Web-based auxiliary processes of collaborating in project development. The challenge is to ensure that the sites should complement the project work by enabling both client and developer teams to interact asynchronously or synchronously through the different customizable services offered. An expression for this project-based IS support (Vat, 2004e) could be written as follows: $\langle\textbf{IS-Support}\rangle_{\textbf{Project}} ::= \boldsymbol{OS}_{\textbf{Project}} + \{ \boldsymbol{CS}_{\textbf{Group}} \} + \{ \boldsymbol{PS}_{\textbf{Participant}} \}$, where the braces {} represents the repetition of the element embedded. It is intended that the provision of the project spaces, the collaborative spaces, and the personal spaces in the Web portal for collaborative project work could facilitate the formation of specific sets of IS support for different human activity systems originated from different project participants. To elaborate on the design of specific IS support we consider the following HAS-based project scenarios of planned and unplanned communication events.

The HAS Scenario of Planned Communication

Planned communication events in project development are scheduled points in time during which participants exchange information on a specific topic or review a work product. Such events are often formalized and structured to maximize the amount of information communicated and to minimize the time participants spend on communication. Typical planned communication events (Bruegge & Dutoit, 2004; Royce, 1998) include problem presentation, client reviews, project reviews, peer reviews, status reviews and brainstorming.

. *Problem Presentation.* The focus here is the presentation of the problem statement that describes the problem, the application domain, and the desired functionality of the system. It should also include some non-functional requirements such as usability and platform specification. The problem statement does not include a complete specification of the system. It is meant to be a preliminary requirements activity that establishes common ground between the client and the developer team.

. *Client Review.* The goal of client reviews is for the client to access the progress of the development and for the developers to confirm or change the requirements of the system. The client review is used to manage expectations on both client and developer sides and to increase the shared understanding among participants. A client review is conducted as a formal presentation during which developers focus on specific functionality with the client. The review is preceded by the release of a work product, such as a specification document, an interface mock-up, or an evaluation prototype. At the outcome of the review, the client provides feedback to the developers. This feedback may consist of a general approval or a request for detailed changes in definition, functionality or schedule.

. *Project Review.* The goals of a project review are for the project manager to assess status and for team members to review subsystem interfaces. Project reviews can also encourage the exchange of operational knowledge across teams, such as common problems encountered with tools or the system. A project review is typically conducted as a formal presentation of several developer teams during which each team presents its subsystem to the management (project sponsor and supervisor) or to other teams that depend on the subsystem. The review is usually preceded by the release of a document describing the aspects of the system under review. At the close of the review, the specific developer team may negotiate changes in the interfaces and changes in schedule.

. *Peer Review.* The goal of peer review is to increase the quality of a work product produced by any designated team member. It is composed of two steps: the walkthrough and the inspection. During the walkthrough, a member of the developer team presents to the other members of the same team, his or her artifact, say, the line-by-line code, or a sequence of user-interface mock-ups. During inspection, the other members challenge any suspicious area and attempt to discover as many issues as possible based on a predefined list of criteria. Communication among participants is artifact-based. The peer review is similar in nature to the project review (typically involving more than one teams), except that they differ in their formality, their limited audience, and their extended duration within a single team of project members.

. *Status Review.* The focus of status review is the tasks distributed among team members. Status reviews are primarily conducted within a team (say, weekly) and aimed to detect deviations from the task plan and to correct them. Status reviews encourage developers to complete pending tasks. The review of task status encourages the discussion of open issues and unanticipated problems, and thus encourages informal communication among team members. Often, solutions to common issues can be shared and operational knowledge disseminated more effectively when discussed within the scope of a team. Increasing the effectiveness of status reviews normally has a global impact on the team performance.

. *Brainstorming.* The goal of brainstorming is to generate a large number of solutions to a problem, regardless of their merit, then evaluate them. The functional idea behind brainstorming is that ideas, however invalid, proposed by any participant, can trigger other ideas and proposals from other participants. Brainstorming encourages thinking “outside the box.” It has two beneficial side effects: evaluating proposals within the group will lead to more explicit evaluation criteria, and the brainstorming process itself has the effect of building consensus for the chosen solution.

The HAS Scenario of Unplanned Communication

In an ideal project, all communication takes place during planned communication events. In practice, it is difficult to anticipate all information needs and plan all communications. In general, issues resulting from a combination of seemingly isolated facts from different areas of the project are difficult to anticipate because no participants could have a global overview of all the facts. Consequently, a project should be prepared to deal with unexpected situations, often under pressure. We call the communication resulting from such crises unplanned communication events, including requests for clarification, requests for changes, and issue resolution.

. *Request for Clarification.* This request represents the bulk of the communication among developers, clients, and users. Such requests are largely unplanned. A participant may request clarification about any aspect of the system that seems ambiguous. These requests may occur during informal meetings, e-mails, or any other communication mechanism available to the project. It is worthy to note that if most information needs are handled through requests for clarification, such situations represent symptoms of a defective communication infrastructure. And the result could lead to serious failures downstream owing to misunderstandings and missing and misplaced information.

. *Request for Change.* This request represents a channel to report any problems with the system itself, including its documentation, the development process, or the project organization. Typically, a change request contains such details as the classification (say, severe, moderate, or annoying), a description of the problem (say, rationale, or communication), a description of the desired change (say, a proposed solution). Requests for change are often formalized when the number of participants and the system size is substantial.

. *Issue Resolution.* An issue represents a concrete problem, such as a requirement, a design, or a management problem. To each decision in project work represents an issue to be solved so that development can proceed. Alternatives are possible solutions that could address the issue under consideration. Criteria are desirable qualities that the selected solution should satisfy. During requirements analysis, criteria are nonfunctional requirements and constraints such as usability. During system design, criteria are design goals such as reliability. During project management, criteria are management goals and tradeoffs such as timely delivery versus quality. A decision is the resolution of an issue representing the selected alternative according to the criteria that were used for evaluation and the justification of the selection. Typically, it is only after much discussion (or argumentation) that a consensus is reached or a decision imposed, covering all aspects of the decision process, which includes criteria, justification, explored alternatives, and trade-offs. Using these issue-modeling concepts in the context of capturing project rationales (Dutoit & Paech, 2001; Moran & Carroll, 1996) in review meetings, we often write an agenda in terms of issues that we need to discuss and resolve. We state the objective of the meeting (formal or informal) to be a resolution on these issues and any related sub-issues that are raised in the discussion. We structure the meeting minutes in terms of proposals that we explore during the meeting, criteria that we agree on, and arguments we use to support or oppose proposals. We capture decisions as resolutions and action items that implement resolutions. During subsequent meetings, we review status in terms of the action items that we produced in the previous meetings.

Crafting an ISD Framework for Collaborative Project Work

From our discussion built up so far, while conceiving the necessary IS support to serve the specific organizational project requirements (Checkland, Forbes, & Martin, 1990; Checkland & Holwell, 1995), it is our experience that the fundamental ideas behind an actionable ISD framework could be presented as follows: Always start from a careful account of the purposeful activity to be served by the system. From that, work out what IS support is required (by people) to carry out the activity. Treat the creation of that support as a collaborative effort between technical experts and those who truly understand the purposeful action served. Meanwhile, ensure that both system creation and system development and use are treated as opportunities for continuous learning. In this way, models of purposeful human activities can be used as scenarios to initiate and structure sensible discussion about information support for the people undertaking the real-world problem situations. Put more concisely, the emphasis in information systems development, it is argued, must be based on several stages of analysis as well as an important stage of technical design and construction:

Analysis of Human Activity

In this first stage of work, the distinction between issue and task (Kunz & Rittel, 1970) is critical because it is too easy to concentrate on tasks (say, what job is the computerized IS going to have to do?) when computerizing and to overlook important issues (say, what do we hope to achieve for the project as a result of installing the IS support?) which need to be resolved. Too often, issues are ignored in the rush to computerize. But we cannot solve a problem until we know what the problem is. Thereby, the use of SSM at this early stage helps resolve the issue-related aspects (Galliers, 1992) which are concerned with debating on the definition of what real-world problems the system is to solve. After important issues have been resolved, task-related aspects work toward forming the system that has been defined with appropriate emphasis on various technical and human views. It can be visualized as a learning cycle including discovery, invention, production, and generalization, as well as double-loop learning (Argyris & Schon, 1978) to bring about the surfacing and challenging of deep-rooted assumptions which were previously unknown or un-discussable. Consequently, the system, once created, is not just a computer system; it is also composed of people performing meaningful jobs.

Analysis of Information

If information is interpreted as what we get when human being attribute meaning to data in a particular context, then an information system (IS), in the full sense, will be a meaning attribution system in which people select certain data out of the mass potentially available and get them processed to make them meaningful in a particular context in order to support those engaged in purposeful action (Checkland & Haynes, 1994; Checkland & Holwell 1995). Thus, if we wish to create an appropriate IS in the exact sense of the phrase, we must first understand how people in the specific situation conceptualize their world. We must find out the meanings they attribute to their perceptions of the world and hence understand which action in the world they regard as sensible purposeful action, and why. Having obtained that understanding we shall be in a position to build some of the purposeful models, and use them to stimulate debate aimed at defining some human activity systems (HAS) widely regarded by people within the situation as truly relevant to what they see as the required real-world action. Once an agreed truly relevant system has emerged, the use of scenario-based development requires us to ask of each activity in the model the following questions: What information would have to be available to enable someone to do this activity? From what source would it be obtained, in what form, with what frequency? Besides, we need to be aware of what information would be generated by doing this activity. To whom should it go, in what form, with what frequency? In this way, an activity model may be converted into an information-flow model. Given the information-flow model, which is agreed to be a necessary feature of the situation studied, we may then ask: What data structures could embody the information categories that characterize such information flows? It is only then that we could start the design of a suitable information system. This should yield the information categories and information flows required by the structured set of activities regarded as truly relevant to the real-world action. Such action is itself relevant according to the meanings which people in the situation attribute to their world as a result of their worldviews.

Analysis of Socio-Technical Aspects

According to (Checkland & Holwell, 1995), the main role of an information system is that of a support function in an organizational setting. More specifically, the IS function is to support people taking purposeful action by indicating that the purposeful action can itself be expressed via some activity models, which are called the HAS models from the perspective of SSM (Checkland, 1979; 1984; Checkland & Scholes, 1999). As an account of the socio-technical context of IS work, we can consider an actionable model in which organization meanings are created. Briefly, there are seven elements in this model, worthy of our attention. Element 1 consists of people as individuals and as group members in the project organization. Element 2 is the data-rich world people perceive selectively through their various taken-as-given assumptions. Element 3 is the organizational discourse in which meaning is created inter-subjectively. Element 4 denotes the attributions of meanings which yield the necessary information and knowledge through a very complex social process involving perhaps, persuasion and coercion. Element 5 represents the assemblies of related meanings, intentions and accommodations among conflicting interests. Element 6 represents the purposeful action, best thought of and expressed as a managing of relationships. Element 7 covers the formally organized information systems based on various information technologies (IT) which support organization members in conceptualizing their world, finding accommodations, forming intentions, and taking actions (elements 5 and 6). In fact, this model is conceived not as a descriptive account of the specific organization process, but a defensible device with a structure to make sense of life in real organizations and their provision of IS support (Weick, 1995). In a particular situation, the initial focus might, for example, be on action (element 6). It might be found to be inadequately supported by the IS in element 7, or it might be found that some boring action previously taken by people could now be automated. In another situation, a new development in IT (element 7) might cause a re-think of possible knowledge (element 4), intentions (element 5), and action (element 6). Meanwhile, from an IS architect’s viewpoint, elements 1-5 describe the organizational context in which people create meanings and intentions; this leads to purposeful action being taken (element 6). Element 7 provides what would usually be described as information support. Thus, we have a process (elements 1-5) and a form of support (element 7) for a main outcome of that process, namely, the purposeful action (element 6), which people take as a result of the process. In general, the model should have pathways, which link all elements with one another; namely, there is no clear starting point for use of the model. However, the cycle might be dominated, in particular circumstances, by changes in (or changed perceptions of) any of the elements in the model.

Design and Construction of Technical Artifacts

This design stage renders a rational, engineering-based approach to systems development in which the aim is to produce technical artifacts that will support purposeful human and organizational activities. After working through the previous stages of analysis, the technical requirements for the necessary IS support, must have been formulated with both social and technical objectives in mind. At this stage, we concentrate on efficient design and the production of a full systems specification. The major approach followed is the traditional systems engineering method whose taken-as-given assumption is that "what-to-do" has been well defined, and there is a gap to be bridged between the desired future state and the present state; how to bridge it is the problem. The issue is goal-oriented: How can we provide an efficient means to meet the specific objective which declares that the system when engineered accordingly, should meet the need, and the mission to be accomplished.

Remarks for Continuing Challenge

Group-based project work often requires the collaboration of team members possessing a variety of knowledge and skill. Yet, organizing a group of individuals to work together is a difficult problem in any type of undertaking. It takes time for team members to pool their knowledge in discussions, negotiate decisions, coordinate their efforts, and incorporate the work of others in their own efforts. As an account of the context of IS work for collaborative project development, we consider the idea of human activity systems in characterizing the various teamwork dynamics of a group of people working cooperatively to accomplish the project mission. Our discussion is based on our PBL course experience of *Software Psychology*, whose curriculum action research is concerned with understanding the organizational behaviors associated with individual PBL team members during their project involvements, including the description of the personal, social and organizational knowledge models that are indispensable in the process of collaboration. In fact, any particular project in IS support must describe how and with what the organization performs its work. Here the alignment context could be expressed in terms of the dynamics of the people-process-system issue. Namely, we need to design suitable information systems to help people with knowledge to perform the processes involved to produce results of value to the organization. In fact, Zuboff (1988) has written extensively on the interaction of people and information technology (IT), and the all-important shift in management thinking from automating to informating, where *automating* typically refers to the use of IT during process change to lower uncertainty and increase management control. *Informating*, in contrast, refers to the effect IT may have on the understanding and transparency of a process. Informating makes people more productive through their use of, and process integration with IT. It serves to increase the capacity of people to understand the entire value-adding organizational process. Thus, informating concerns itself with the connection people have with their specific tasks as well as the whole flow of work. Certainly, the notion of knowledge cannot be neglected. While informating concerns IT and task integration, the idea of *knowledging* (Savage, 1990) refers to the context of individual and organizational learning, and is characterized by the process of knowledge creation and the active involvement of the individual with his or her work. Knowledging includes a dynamic interaction between the known (explicit) and the vision (tacit) forms of knowledge. In fact, each context from automating to informating to knowledging, may be thought of as an important stage to characterize the correspondent purposeful human activities. In particular, the trade-off between individualism and community may impact the movement from informating to knowledging. Individualism drives individual knowledge and rewards, and thus encourages informating, while a community emphasizes sharing and is more closely associated with knowledging, including the interaction of computers, people, lateral relations, work processes, and organizational learning (including knowledge creation). Thereby, in order to create a pragmatic model of IS support for collaborative project work, each successive organizational transformation, from automating to informating to knowledging, must be seriously examined in terms of creating the specific HAS models specifying higher levels of process abstraction and a broad range of process integration and alignment threads with the underlying project-based activities. Thereby, the introduction of our ISD framework in this article represents another empirical step of faith in this direction.

References

Argyris, C. & Schon, D.A. (1978). *Organizational learning: A theory of action perspective.* Reading, MA: Addison-Wesley.

Barrows, H. S. (1986). A taxonomy of problem-based learning methods. *Medical Education*, *20*, 481-486.

Bruegge, B. & Dutoit, A. H. (2004). *Object-oriented software engineering using UML, patterns, and Java* (2nd ed.). Prentice Hall (Pearson Education International).

Carroll, J. M. (2000). *Making use: Scenario-based design of human-computer interactions*. MIT Press.

Checkland, P. (1979). Techniques in soft systems practice, part 2: Building conceptual models. *Journal of Applied Systems Analysis*, *6*, 41-49.

Checkland, P. (1981). *Systems thinking, systems practice.* Chichester: Wiley.

Checkland, P. (1983). Information systems and systems thinking: Time to unite? *International Journal of Information Management, 8*, 239-248.

Checkland, P. (1984). Systems theory and information systems. In Th. M.A. Bemelmens (Ed.), *Beyond productivity: Information systems development for organizational effectiveness.* North- Holland, Amsterdam.

Checkland, P. (1999). Systems thinking. In W.L. Currie & B. Galliers (Eds.), *Rethinking management information systems.* Oxford University Press.

Checkland, P., Forbes, P. & Martin, S. (1990). Techniques in soft systems practice, part 3: Monitoring and control in conceptual models and in evaluation studies. *Journal of Applied Systems Analysis*, *17*, 29-37.

Checkland, P. & Haynes, M. (1994). Varieties of systems thinking: The case of soft systems methodology. *System Dynamics Review, 10* (2-3), 189-197.

Checkland, P. & Holwell, S. (1995). Information systems: What's the big idea? *Systemist*, *17* (1), 7-13.

Checkland, P. & Holwell, S. (1998). *Information, systems, and information systems: Making sense of the field.* Chicheser: John Wiley and Sons.

Checkland, P. & Scholes, J. (1999). *Soft systems methodology in action.* Chichester: Wiley.

Conklin, J. & Burgess-Yakemovic, K.C. (1991), A process-oriented approach to design rationale. *Human-Computer Interaction*, *6*, 357-391.

Curtis, B., Krasner, H. & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM, 31* (11), 1268-1287.

Doyle, M. & Straus, D. (1982). *How to make meetings work.* New York, NY: The Berkeley Publishing Group.

Dutoit, A.H. & Paech, B. (2001). Rationale management in software engineering. In S.K. Chang (Ed.), *Handbook of software engineering and knowledge engineering*, *Volume 1*. World Scientific Publishing.

Fisher, K. (2000). *Leading self-directed work teams: A guide to developing new team leadership skills.* McGrawHill.

Fisher, R., Ury, W. & Patton, B. (1991). *Getting to yes: Negotiating agreement without giving in* (2nd ed.). Penguin Books.

Galliers, R. (1992). Choosing information systems research approaches. In R. Galliers (Ed.), *Information systems research: Issues, methodology and practical guidelines.* Oxford: Blackwell.

Gause, D. & Weinberg, G. (1989). *Exploring requirements: Quality before design*. Dorset House.

Ginac, F.O. (2000). *Creating high performance software development teams.* Prentice Hall PTR.

Greening, T. (2000). Emerging constructivist forces in computer science education: Shapiong a new future?” In T. Greening (Ed.), *Computer Science Education in the 21st Century*, Springer, 47-80.

Gregory, V. (2000). Knowledge management and building the learning organization. In T.K. Srikantaiah, and E.D. Koenig (Eds.), *Knowledge management: For the information professional* (pp. 161-179). ASIS: Information Today.

Kaner, S. (1996). *Facilitator’s guide to participatory decision-making.* Philadelphia: New Society Publishers.

Kunz, W. & Rittel, H. (1970). Issues as elements of information systems. Working Paper No. 131. *Institut fur Grundlagen der Plannung*, Universitat Stuttgart, Germany.

Land, F. (1985). Is an information theory enough? *The Computer Journal, 28* (3), 211-215.

Lederer, A. L. & Sethi, V. (1988). The implementation of strategic information systems planning methodologies. *MIS Quarterly,* 12 (3), 445-461.

MacLean, A., Young, R. M., Bellotti, V. & Moran, T. (1991). Questions, options, and criteria: Elements of design space analysis. *Human-Computer Interaction*, *6*, 201-250.

Moran, T. P. & Carroll, J. M. (Eds.). (1996). *Design rationale: Concepts, techniques, and use.* Mahwah, N.J: Lawrence Erlbaum Associates.

OECD (1996). *The knowledge-based economy.* Organization for Economic Co-operation and Development, OCDE/GD(96)102, Paris.

Royce, W. (1998). *Software project management: A unified framework.* , Reading, MA: Addison-Wesley.

Ryan, G. (1993). Student perceptions about self-directed learning in a professional course implementing problem-based learning. *Studies in Higher Education*, *18*, 53-63.

Savage, C.M. (1990). *Fifth generation management: Integrating enterprises through human networking.* Digital Press.

Vat, K. H. (2000). Teaching software psychology: Expanding the perspective. In *Proceedings of the Thirsty-first SIGCSE Technical Symposium on Computer Science Education*, Austin, TX, Mar. 8-12, 2000, pp. 392-396.

Vat, K. H. (2001). Teaching HCI with scenario-based design: The constructivist’s synthesis. In *Proceedings of the Sixth Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE2001)*, Canterbury, U.K., Jun. 25-27, 2001, pp. 9-12.

Vat, K. H. (2004a). Conceiving a learning organization model for sustainable development: The IS manager's perspective based on soft systems methodology. In *Proceedings of the IEEE International Engineering Management Conference 2004 (IEMC2004)*, Oct. 18-21, Singapore, pp. 500-504.

Vat, K. H. (2004b). Conceiving scenario-based IS Support for knowledge synthesis: The organization architect's design challenge in systems thinking. In *Proceedings of the 10th International Conference on Information Systems Analysis and Synthesis (ISAS2004)*, Orlando, Florida, USA, July 21-25, pp.101-106.

Vat, K. H. (2004c). On the idea of soft systems methodology for IS development: A perspective based on purposeful action. In *Proceedings of the International Conference on Computing, Communications and Control Technologies (CCCT2004)*, August 14-17, Austin, Texas, USA, pp. 227-232.

Vat, K. H. (2004d). Systems architecting of IS support for learning organizations: The scenario-based design challenge in human activity systems. In *CD-Proceedings (ISSN: 1542-7382) of the 2004 Information Systems Education Conference (ISECON2004)*, Nov. 4-7, Newport, Rhode Island, USA.

Vat, K. H. (2004e). Towards a learning organization model for PBL: A virtual organizing scenario of knowledge synthesis. In *CD-Proceedings of the Seventh Annual Conference of the Southern Association for Information Systems (SAIS2004)*, Feb. 27-28, Savannah, Georgia, USA.

Vat, K. H. (2004f). Toward a learning organization model for student empowerment: A teacher-designer's experience as a coach by the side. In *Proceedings of the 2004 IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA2004),* Dec. 15-17, Lisbon, Portugal, pp. 131-140.

Vat, K. H. (2005). On the importance of human activity systems in organization modeling for IS development. In *CD-Proceedings of the Eighth Annual Conference of the Southern Association for Information Systems (SAIS2005)*, Feb. 25-26, Savannah, Georgia, USA.

Venkatraman, N. (1994). IT-enabled business transformation: From automation to business scope redefinition. *Sloan Management Review,* Winter.

Vickers, G. (1972). Communication and appreciation. In G. B Adams, J. Forester & B. L Catron (Eds.), *Policymaking, Communication and Social Learning: Essays of Sir Geoffrey Vickers.* New Brunswick, NJ: Transaction Books.

Vickers, G. (1965). *The art of judgment.* London: Chapman and Hall.

Weick, K.E. (1995). *Sense-making in organizations*. Thousand Oaks, CA: Sage.

Whitten, N. (1995). *Managing software development projects* (2nd ed.). John Wiley & Sons.

Wilson, B. (2001). *Soft systems methodology: Conceptual model building and its contribution.* New York: John Wiley & Sons.

Zuboff, S. (1988). *In the age of the smart machine: The future of work and power.* New York: Basic Books.

Biography

Kam Hou VAT is currently a lecturer in the Department of Computer and Information Science, under the Faculty of Science and Technology, at the University of Macau, Macau SAR, China. His current research interests include learner-centered design with constructivism in Software Engineering, architected applications developments for Internet software, information systems for learning organization, information technology for knowledge synthesis, and collaborative technologies in electronic organizations.