The Pennsylvania State University
The Graduate School
College of Information Sciences and Technology

AWARENESS PROMOTION FOR COORDINATING DISTRIBUTED GEO-COLLABORATIVE ACTIVITIES

A Dissertation in
Information Sciences and Technology
by
Bo Yu

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

Spring 2013

The thesis of Bo Yu was reviewed and approved* by the following:

Guoray Cai
Associate Professor of Information Sciences and Technology
Thesis Advisor, Chair of Committee

Alan M. MacEachren
Professor of Geography

Mary Beth Rosson
Professor of Information Sciences and Technology

Xiaolong Zhang
Assistant Professor of Information Sciences and Technology

*Signatures are on file in the Graduate School.

Abstract

One of the major challenges to support complex, distributed geo-collaborative activities is to integrate effective coordination mechanisms to manage different types of work dependencies. This study focuses on event-based awareness mechanisms to support the management of dependencies by giving team members an awareness of what each other are doing or have done so that participants can adjust and coordinate their work in a more flexible way. Unlike existing event-based mechanisms, this study (1) relies on a deep understanding of the variety and dynamics of dependencies in these activities to distribute awareness events, and (2) provides explicit visualization and interaction support for the interpretation of awareness events.

The central idea of the approach to modeling dependencies and awareness events is based on a formal structure of collaborative activities. Rooted in the SharedPlans theory [19, **?**], the activity model treats a collaborative activity as an evolving shared plan situated in a set of physical and mental contextual factors. Such a model of collaborative activities then can be used to (1) interpret awareness events according to how they are related to the current activities, and notify the relevant users of the events based on the identification of dependencies between the events and users' current focuses.

Then this study illustrates the visualization and interaction provided by the activity model through a design scenario, where a couple of first responders in an emergency response team can use it to help generate and interpret awareness information to manage dependencies. To investigate the impacts of complexities in geo-collaborative activities on the actors' ability to interpret awareness events, an experiment is being designed and performed. The general hypothesis is that the effectiveness of our visualization and interaction design to support awareness interpretation is correlated to the level of complexity of collaborative activities.

Table of Contents

List of Figures

List of Tables

List of Corrections

Chapter 1

Introduction

1.1 Problem Scope

Support for awareness is one of the most active research areas in computer-supported cooperative work (CSCW) [11, 39, 34]. At its essence, awareness refers to the ability of collaborators to understand each others' activities and relate them to a joint context [34]. This context is essential for collaboration as it is used to ensure that individual contributions are relevant to the groups activity as a whole, and allow groups to coordinate the process of collaborative working [11].

One general design concern for awareness systems is that awareness must be achieved with minimal attention and effort from the participants of teamwork [26]. Awareness falls into the category of 'the articulation work' that is required for coordination but not the primary goals for collaboration [**?**]. Taking extra time and effort to achieve awareness can interrupt the current line of action, and therefore damage team performance [18]. As a result, a large number of awareness mechanisms and tools have been proposed in the literature, aiming at promoting awareness in a relatively effortless way [34].

Although much progress has been made in designing awareness mechanisms to support team activity at relatively small and medium scales [2], it becomes a much more difficult task to promote awareness in complex and highly distributed activities [5]. The geo-collaborative activities of interest in this study are a subset of these complex collaborative setting, where awareness is unlikely to be achieved effortlessly, and hence it becomes even more important to design computational artifacts that actively promote awareness. In the rest of this section, we first describe the characteristics of collaborative settings that we consider as geo-collaboration in this study, and then present the challenges for supporting awareness in geo-collaboration that motivates our work.

1.1.1 Geo-Collaboration

The geo-collaborative activities we consider in this paper are a subset of complex collaborative setting, with the following characteristics:

1. Multiple team members are geographically distributed.

2. They are engaged in tightly interdependent activities that require effective coordination.

3. They work in dynamic setting that entails rapid and frequent changes in environment and activities.

Examples of geo-collaboration within these boundaries abound in practical applications such as crisis management, resource management, military exercises, and science explorations. For the ease of illustrating the work we present in this study, we will frequently refer to the following scenario in emergency response.

An Emergency Response Scenario. A chemical factory near an urban area was exploded and caused a major pollution. To respond to this critical incident, task force is formed that includes search and rescue teams, decontamination teams, medical treatment teams, and transportation teams. Teams are dispatched and configured geographically to cover the impacted area. Each team cover a functional area of the overall mission. Search and rescue teams patrol the incident area to search for victims and report their locations and status. Discovered victims are first decontaminated (by one of the Decontamination teams) before they can be moved to other facilities. If a victim is wounded, he or she will be scheduled and transported to a medical station for treatment. All transportation needs for moving victims to treatment stations and shelters are handled by the transportation team. Although teams are working autonomously on their local tasks, they must coordinate their capacity, schedule, and priority to deal with emerging and unexpected situations in order to save and protect all the victims in an efficient fashion.

Such an emergency situation usually involves multiple individuals and organizations that are distributed in different geographic locations. Figure 1.1 shows a hypothetical distribution of tasks and teams in relation to the disaster area. Because of the distribution, workers structure their tasks so that they can work relatively autonomously within their local environment and responsibilities. In the same time, due to interdependencies among sub-activities [41], they find themselves working on a multitude of different activities, interleaving them in ways that seem best suited for getting their work accomplished given the practical pressures. Furthermore, exceptions to the planned responses are a common and critical factor in these activities [**?**]. What specific information is of concern and interest to a given individual is changing rapidly.

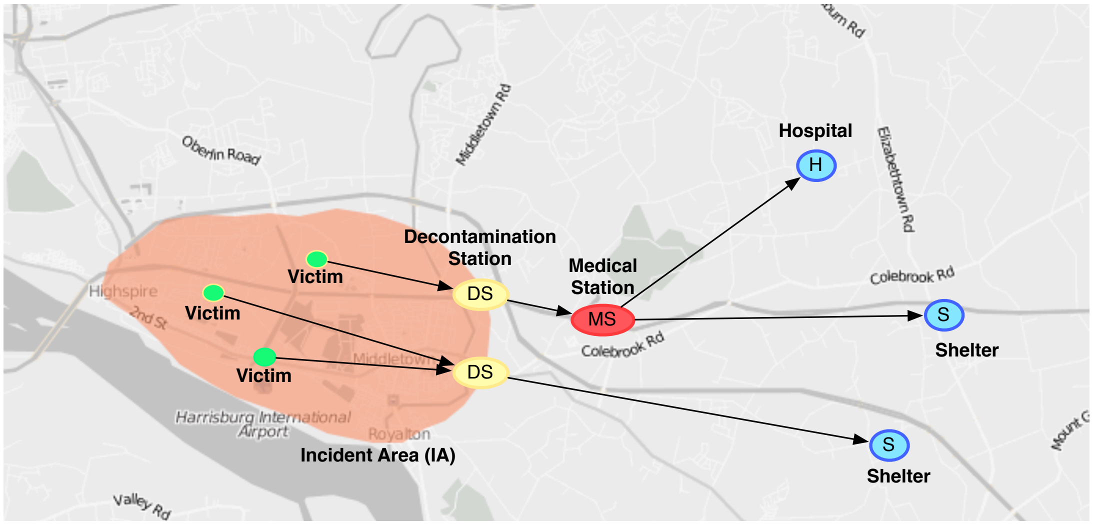


Figure 1.1. An emergency scenario

1.1.2 Challenges in coordinating geo-collaboration

This scenario clearly demonstrates the three major challenges to coordinate complex geo-collaborative activities: the complexity and contingency of collaborative activities, and the involvement of geographical space.

1.1.2.1 High level of complexity

With low degrees of complexity, the coordination of cooperative work can be achieved by means of mutual awareness and alignment [**?**]. As demonstrated by the body of rich empirical studies of cooperative work within CSCW [**?**, 21], actors tacitly monitor each other; they perform their activities in ways that support coworkers' awareness and understanding of their work; they take each others' past, present and prospective activities into account in planning and conducting their own work; they gesture, talk, write to each other, and so on, and they mesh these interactional modalities dynamically and seamlessly . This appears effortless, because, to a competent member in the flux of doing the work and thus attuned to the changing state of the field of work what the colleagues next to him are doing is immediately meaningful; it does not require interpretation, reflection, contemplation to know why they are doing what they are doing or why they are not doing something else [**?**].

However, in the complex work settings that characterize modern industrial, service, and administrative organizations where hundreds or thousands of actors engaged in myriads of complexly interdependent activities, the task of coordinating the interdependent and yet distributed activities is of an order of complexity where the mutual awareness mechanisms are far from sufficient. Carstensens study of a software development project [**?**] clearly illustrates the problem of scaling with complexity. In previous projects the systems they had been constructing had been small and the programming work had been done by a couple of programmers. In these projects they had been able to manage their interdependencies practically effortlessly. They had been working next to each other and had practically unconstrained access to consulting each other and to monitoring each others work. At the time of the study, however, a new project had been undertaken in which the engineers were building a significantly larger system comprising many hundred thousands lines of code. Their traditional coordinative practices were now quite inadequate. The interdependencies of their cooperative effort now transcended the local practices, and they were faced with situations that the effects of their local activities to other regions of the cooperative effort are not immediately and straightforwardly evident. To deal with the ensuing crisis, the ensemble had to develop a set of formal coordinative artifacts, such as the bug report forms, to regulate local practices.

One of the major challenges demonstrated in the scenario is characterized by the degree of *complexity* of dependencies that can exist in coordination work. With low degrees of complexity (e.g. when the number of dependencies that need to manage is limited, or their effect is only within the scope of local practices), the coordination of collaborative work can be achieved by means of human communicative and cognitive skills [40]. However, in the complex work settings as described in the scenario, multiple dependencies can co-exist at the same time, and they together form a web of dependencies where state changes of one of them can cause chain effects on others. Faced with a high degree of complexity of coordination work, collaborative actors often use a category of symbolic artifacts which, in the context of a set of procedures and conventions, stipulate and mediate coordination work and thereby are instrumental in reducing its complexity and in alleviating human effort [45]. These artifacts, together with the concomitant procedures and conventions are called 'coordination mechanisms' [40]. The first goal of this study aims at building an integrated coordination mechanism that would keep track of the state of work and to manage relations and dependencies among actors, tasks, and resources in order to reduce the complexity of coordination work in geo-collaborative activities.

1.1.2.2 High level of contingency

Developing appropriate coordination mechanisms must be based on a solid understanding of various collaborative dependencies, i.e. the capabilities of coordination mechanisms should match coordination requirements of the task at hand. In relatively static work domains such as process control or manufacturing, the set of work dependencies is largely known in advance and thus it becomes feasible to specify formal coordination mechanisms to manage them, such as routines, pre-planning, or standardization. Increasingly, however, collaborative work is characterized by high levels of interdependence, uncertainty, and time constraints, such as cooperative design [**?**] or emergency response [41]. In these dynamic contexts, interdependence of work is emerging and rapidly shifting over time [16]. Due to changes and evolution of collaborative plans, dependencies may emerge, sustain, or disappear as the activity advances.

The contingency of dependencies can be clearly evidenced in the motivating scenario, where patterns of dependencies among people are in fact quite volatile, either due to the changes of environment in which the work is done (environmental uncertainty), or the changes of the task that the contributors are trying to accomplish (task uncertainty) [24]. For instance, while the decontamination process at a given decontamination station is under way, a first responder reports that five new victims have been discovered and are on their way to the decontamination station. As a result, a request to deliver an additional operator is initiated in anticipation of exceeding its maximum capacity of victims the station can handle. Suddenly, the decontamination process becomes intertwined with the process of delivering the operator, and the five new victims cannot be decontaminated until the new operator arrives at the station. In this way, such unexpected events cause seemingly un-related tasks to suddenly become interdependent.

As a result, we believe that the contingency in the interdependence of work represents a major feature of complex geo-collaborative work, but existing coordination tools have not taken this into account seriously. When a collaborative activity involves an extended period of work with volatile dependencies, the potential dependencies (i.e. the collective set of all dependencies that could potentially happen during the whole activity) are a quite large number, but the actual dependencies (i.e. those dependencies that are live and need to be coordinated) are rather a relatively small number. Coordination tools could play a more effective role if they target coordination of actual dependencies, rather than potential dependencies. However, identifying the set of actual dependencies at any given moment is difficult due to the fact that such dependencies are unknown a priori and they emerge as a consequence of the evolving activity. It would be highly desirable for collaborative tools to be able to assess the characteristics of the activities, computationally identify actual dependencies, and track their changes over time.

1.1.2.3 The uniqueness of being spatial

Coordination work varies, according to the complexity of the interdependence, that is, depending on factors such the distribution of activities in time and space, the number of participants in the cooperative ensemble, the structural complexity posed by the field of work (interactions, heterogeneity), the degree and scope of specialization among participants, and so on [40]. As a result, different types of dependency relationships have been discussed in the literature. For example, Yu and Mylopoulos [51] illustrate that actors can depend on each other in various means: they may depend on each other for goals to be achieved, activities to be performed, and resources to be furnished. In the case of task dependencies, Malone and Crowston [25] identify several common types of dependency relationships between tasks, including temporal relationships, resource-related relationships, and goal-related relationships.

In the context of geo-collaborative activities, some of dependency relationships may be directly cast on the geospatial relationships between entities. For example, in the scenario, the different rescue resources, e.g. vehicles, equipments, the victims, the response actors, and their activities are spatially distributed. The locations of these entities and spatial relationships among them can raise dependencies that need to be managed explicitly in the coordination work. A traffic jam at a given location may have no direct impact on the actor who is performing the task to take care of victims. However, the fact that the location of the traffic jam is on the path of a rescue vehicle delivering vital medical equipments to the actor may poses a dependency between the traffic jam and the actor's current work. As a result, the actor may have to switch the focus to explicitly manage this dependency before the domain task can be performed.

The nature of being spatial in these geo-collaborative activities provides a unique type of dependency relationships that need to be explicitly managed. However, existing models of dependencies lack of support for dependencies that involves spatial relationships. To understand and manage the dependencies involving spatial relationships, we must develop a deep understanding of how geographic space and human activities are related to each other, and integrate these spatially-related dependencies with other types of dependencies in a coherent framework.

1.2 Research Objectives and Approach

By setting the scene in the previous section, the overall objective of this dissertation is to addresses the major challenges in coordinating complex, dynamic, and distributed geo-collaborative activities by developing an awareness-based coordination mechanism that can utilize the knowledge of activities and dependencies.

To achieve the research objective, this study follows the design-science paradigm in information systems research (Hevner et al. 2004). Knowledge and understanding of the aforementioned coordination problems in geo-collaboration and their solutions are achieved through a set of design activities, including building, evaluation, and application of the designed artifact. Figure 1.2 provides the overview of the research approach in this study.

The research starts with justifying the relevance of the research to the problem domain, i.e. geo-collaborative activities. The unique characteristics of geo-collaboration motivate this study. Then the existing knowledge about theories, models, and methods of coordination is reviewed to provide the scientific foundation of this study. Based on the grounding work in the literature, the overall design framework is developed. Such a design framework help the author to identify knowledge gaps in existing studies, and develop concrete research questions that need to be answered. Answers to these research questions are achieved through a set of design research activities. During the process of these design activities, more relevant knowledge from existing literature is applied.

Following the design-research paradigm, the expected contributions of this research are two-folded. On the one hand, the research activities in this study have the potential to extend our understanding of spatial dependencies in geo-collaboration, and provide a comprehensive approach to supporting awareness process in coordinating geo-collaborative activities. On the other hand, the proposed approach can be applied in many geo-collaborative activities, such as emergency response, transportation management, to offer solutions to important real life problems.

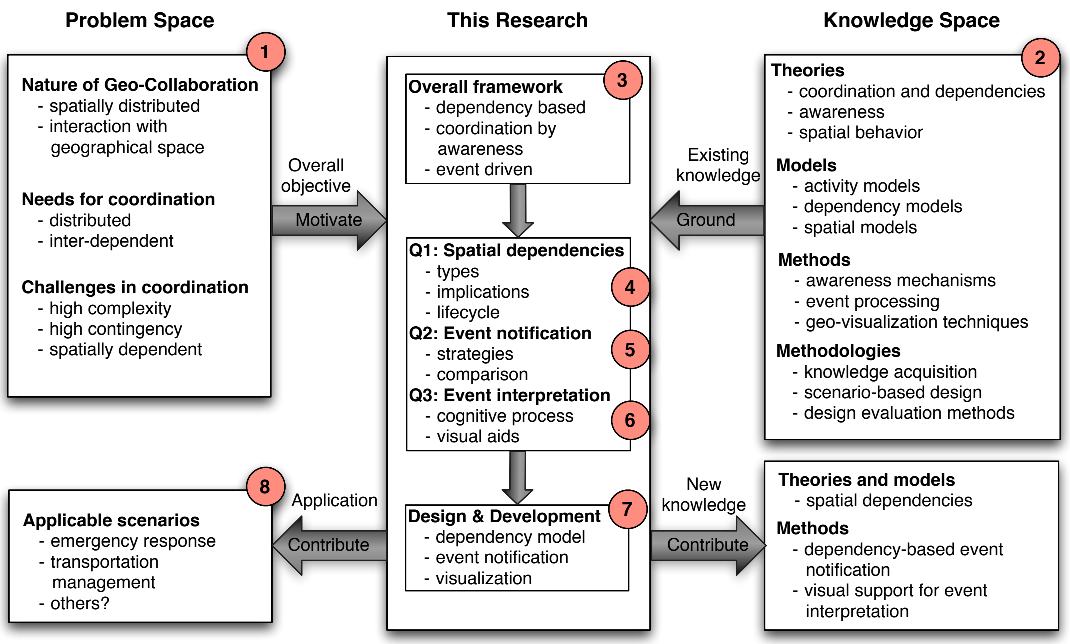


Figure 1.2. Overview of the research approach

1.3 Thesis Structure

The rest of this document is structured corresponding to how each chapter fits into the overall research framework. Chapter 2 presents the grounding work of this study, including the various studies on coordination and awareness mechanisms. Chapter 3 provides the overall design framework and relevant research questions. Chapter 4-6 provide the research activities and answers to these research questions respectively. Chapter 7 presents the design and implementation details. Chapter 8 demonstrates the use of the approach in the domain of emergency response. Chapter 9 concludes the study.

Chapter 2

Understanding Awareness

Add a general diagram to layout the framework, discuss existing work in the framework, add a table to compare and summarize the literature

The concept of awareness has come to play a central role in both social and technical research in CSCW. However, what in CSCW labeled as 'awareness' has little in common, besides the fact that it represents some aspect of human interaction that is important for successful collaboration [39]. In a broad sense, two types of awareness can be distinguished in the CSCW research area: *social awareness* and *task-oriented awareness* [32, 39].

Social awareness addresses the availability of different kinds of information about the social context of the team members, e.g. awareness about what they are doing, if they are talking to someone, if they can be disturbed etc. *Social awareness* thus is conceived of as something that engenders "informal serendipitous interactions" [22] and "a shared space for community building" [12]. Awareness of the general social context is an important aspect of collaborative work, especially in domains where the the actors are engaged in cooperative work in a loose and broad sense, or domains where socialization is crucial [39].

However, when the tasks of collaborating actors become closely interdependent on each other, more urgent concerns need to be given to the aspect of *task-oriented awareness*. The *task-oriented awareness* focuses on practices through which actors seamlessly align and integrate their distributed and yet interdependent activities, e.g. awareness of things being done or in need of being done, of developments within the joint effort that may be advantageous or detrimental for ones own work, of occurrence that makes ones work more urgent or leads to changes to the intended course of actions, etc. [39]. The major difference of *task-oriented awareness* from *social awareness* is that it focuses on activities performed to achieve a specific shared goal [6] and the actors being interdependent in their work [39], which lead to the unavoidable requirement for coordination.

In this study, we focuses on the **task-oriented aspect of awareness**, because the primary goal of supporting awareness in this study is motivated by the actors' being interdependent in their work and hence by the unavoidable requirements of coordinating and integrating their various actions in distributed geo-collaborative activities.

Within the scientific investigation into task-oriented awareness phenomena in collaboration, two lines of research can be identified. On one hand is the research aiming to establish conceptual understanding of the awareness phenomena from the cognitive and social aspects [37], and on the other hand is an increasing bearing on system design and development in particular technologies to promote awareness [34]. Although our work has an emphasis on the second aspect, i.e. the supportive technologies to promote awareness, we believe that a solid conceptualization of awareness phenomena in collaboration is extremely important for awareness promotion. System designers need to know what awareness might comprise and also how it is built and maintained in order to identify the specific awareness features they want to support [47].

As a result, before we move to the computational issues for awareness promotion, this chapter attempts to identify which of the existing theories and conceptualization of awareness in the literature is the most suitable for understanding the awareness phenomena in real world complex collaborative activities. A review and critique of what is currently known on the concept of awareness at both individual and team levels is presented, following which an integrated conceptual framework of awareness phenomena in complex collaborative activities is presented. In next chapter, we will show how such a conceptual framework helps us to evaluate existing computational awareness models and systems, and informs our design of the computational framework for awareness promotion.

2.1 Situation Awareness

Research into awareness at the individual level originated from the study of situation awareness (SA) in the human factors research community. Situation awareness is considered as knowledge created through interaction between a person and his/her environment, i.e. "knowing what is going on" in the situation [15]. A good general definition of situation awareness is as "the up-to-the minute cognizance required to operate or maintain a system" [1]. Although most of the situation awareness models in the literature are individual focused theories [37], it has also been well recognized as an important element in collaborative environments. For example, Gutwin and Greeenberg [20] view their workspace awareness as a specialization of situation awareness tied to the specific setting of the shared workspace. The concept of activity awareness proposed by Carroll et al. also subsumes situation awareness with an emphasis on aspects of the situation that have consequences for group work towards shared goals [6]. Hence, to understand the awareness phenomena in collaboration, it is important to start with understanding the practice of how individuals maintain and develop the situation awareness.

The human factors community has not settled on a common explanation of situation awareness, but we still can summarize some of the important characteristics that are well recognized in the literature, and also applicable to collaborative environments:

1. The products of awareness is the knowledge about the elements of the environment that is

hierarchically structured [15].

2. The awareness phenomena should be seen as both product and process. As product, it is the knowledge that an actor can make use of. As process, it includes the cognitive processes through which the knowledge is achieved and developed) [1].

3. The process of achieving and developing awareness revolves around internally held, mental models, which contain activated information regarding current situations [46]. The activation of awareness information into the mental models are directed by the actor's activity [4].

We elaborate these three characteristics in the following of this section.

2.1.1 Hierarchically Structured Awareness Knowledge

Among the numerous attempts at specifying the products of situation awareness, i.e. what must be known to solve a class of problems posed when interacting with a dynamic environment [37], Endsley's three-level model [15] has undoubtedly received the most attention. The three-level model describes situation awareness as the operator's internal model of the state of the environment, comprising three hierarchical levels that is separate to the process used to achieve it [46]:

1. Level 1: *perception of relevant elements in the environment.* An actor must first be able gather perceptual information in the surrounding environment, and be able to selectively attend to those elements that are most relevant for the task at hand. At this stage, the information is merely perceived and no further processing takes place.

2. Level 2: *Comprehension of task-related elements in Level 1.* Level 2 involves the interpretation of the perceptual information from Level 1 in a way that allows an actor to comprehend or understand its relevance in relation to their tasks and goals.

3. Level 3: *Projection of the states in the near future.* Using a combination of Level 1 and Level 2 awareness-related knowledge and experience in the form of mental models, actors forecast likely future states in the situation.

Endsley's three-level model presents an intuitive description of situation awareness and has been applied in a plethora of different domains [50]. Its simplicity and the division of SA into three hierarchical levels allows the construct to be measured easily and effectively [13], and also supports the abstraction of situation awareness requirements and the development of design guidelines [37]. Furthermore, it has been extended in order to describe team situation awareness [14], and the three levels of awareness information are applicable in many collaborative situations [20].

Despite its popularity, the three-level mode has some important flaws. One of the key assumptions of the three-level model is the separation between depicting situation awareness as a product and the cognitive processes used to achieve it [37], which leads to the inability to cope with the dynamic nature of situation awareness [46]. For example, Uhlarik and Comerford [49] suggest that the process of achieving situation awareness presented by the three-level model is both static and finite. Nevertheless, the model is also criticized by the ill-defined concept of mental models. Although Endsley's model emphasizes the critical roles of mental models in directing attention to critical elements in the environment (Level 1), integrating the elements to aid understanding of their meanings (Level 2), and generating possible future states (Level 3), the definition only includes the long-term knowledge that is formed by training and experiences, more important factors, such as the actor's goals, conceptual model of the current situation, are neglected [4].

2.1.2 The Development of Awareness

To address the dynamic nature of situation awareness, many researchers have used Niesser's perceptual cycle model [29] to clarify the cognitive components involved in the acquisition and development of situation awareness [46, 1, 20, 48]. According to the perceptual cycle model (Figure. 2.1), an actor's interaction with the world continues in an infinite cyclical nature. By perceiving the available information in the environment, the actor modifies its knowledge. Knowledge directs the agent's activity in the environment. That activity samples and perhaps anticipates or alters the environment, which in turn informs the agent. The informed, directed sampling and/or anticipation capture the essence of behavior characteristic of situation awareness.

Based upon Niessers perceptual cycle model, Smith and Hancock suggest that situation awareness is neither resident in the world nor in the person, but resides through the interaction of the person with the world [46]. Thus they viewing situation awareness as a generative process in 'an adaptive cycle of knowledge, action and information' [46]. In a similar fashion, Adams et al. [1] used a modified version of Niessers perceptual cycle model to describe how situation awareness works. They argue that the process of achieving and maintaining situation awareness revolves around internally held mental models, which facilitate the anticipation of situational events, directing an actor's attention to cues in the environment and directing their eventual course of action. An actor then conducts checks to confirm that the evolving situation conforms to their expectations. Any unexpected events serve to prompt further search and explanation, which in turn modifies the actor's existing model. Gutwin and Greeenberg used the perception-action cycle to explain how the awareness is maintained in a shared workspace, in which awareness knowledge both directs and is updated by perceptual exploration of the workspace environment [20].

Unlike the three-level model that depicts SA as a product separate from the processes used to achieve it, models based on the perceptual cycle view situation awareness as both process and product, offering an explanation of the cognitive activity involved in the development of situation awareness, and also a judgment as to what the product of SA comprises [37]. One of the key assumptions of these models is the interplay between the awareness information and

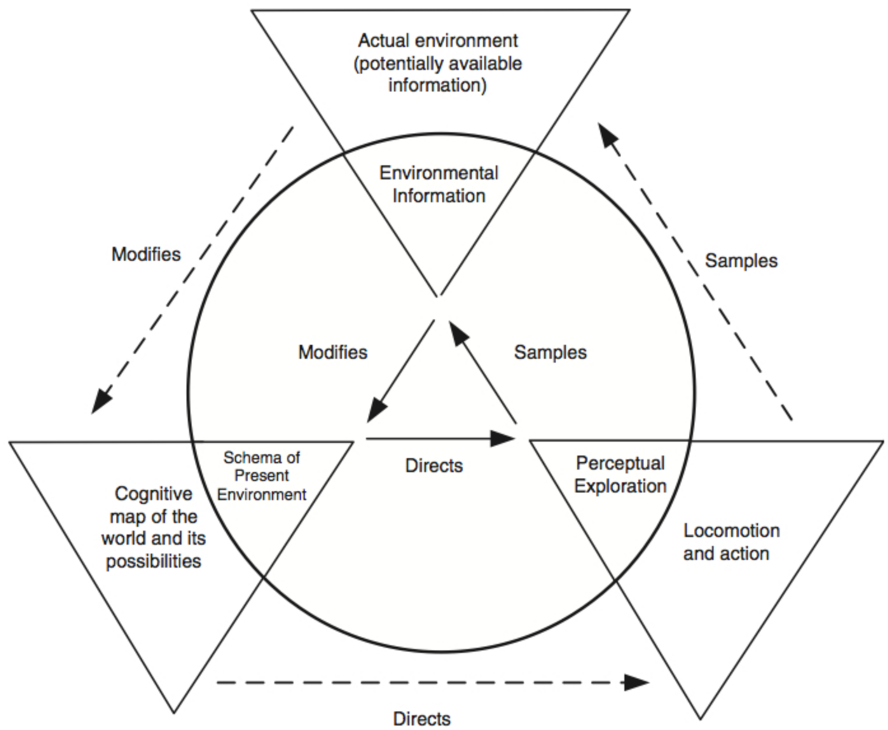


Figure 2.1. Niesser's Perceptual Cycle Model [37]

the internal mental model of current situation. In the process of awareness development, some knowledge is activated and integrated into the mental model, while some becomes inactive or removed from the mental model. Although Smith and Hancock suggested that the adaptation of awareness information into the mental model should be goal-directed, i.e. it must reside in the task environment rather than in the actor's head [46], little detail has been given about the cognitive processes that guide the selection and interpretation of awareness information into the mental models.

2.1.3 Activity Directed Awareness Process

Bedny and Meister propose a description of situation awareness based on the activity theory in an attempt to clarify the cognitive processes involved in the interpretation of awareness information [4]. Based on the activity theory, they purport that individuals possess goals that represent an ideal image or desired end state of activity, which direct them towards the end state or methods of activity (or actions) that permit the achievement of these goals. It is the difference between the goals and the current situation that motivates an individual to engage in the awareness process and take action towards achieving the goal. They conceptualize activity in three stages: the orientational stage, the executive stage, and the evaluative stage. The orientational stage involves the development of an internal representation or picture of the world or current situation.

The executive stage involves proceeding towards a desired goal via decision-making and action execution. Finally, the evaluative stage involves assessing the situation via information feedback, which in turn influences the executive and orientational components.

Instead of considering the actor's internal mental model as a whole, they suggest that the mental model is comprised of several functional blocks as presented in Figure 2.2. Each functional block has a specific role to play in the development and maintenance of situation awareness and that the blocks orientate themselves towards the achievement of situation awareness. The interpretation of incoming information (function block 1) is influenced by an individuals goals (function block 2), conceptual model of the current situation (function block 8), and past experience (function block 7). This interpretation then modifies an individuals goals and experience and conceptual model of the current situation. Critical environmental features are then identified (function block 3) based upon their significance to the task goals and the individuals motivation towards the task goal (function block 4), which directs their interaction with the world (function block 5). The extent to which the individual proceeds to engage the task goals is determined by their goals (function block 2) and their evaluation of the current situation (function block 6). The resultant experience derived from the individuals interaction with the world is stored as experience (function block 7), which in turn informs their conceptual model (function block 8).

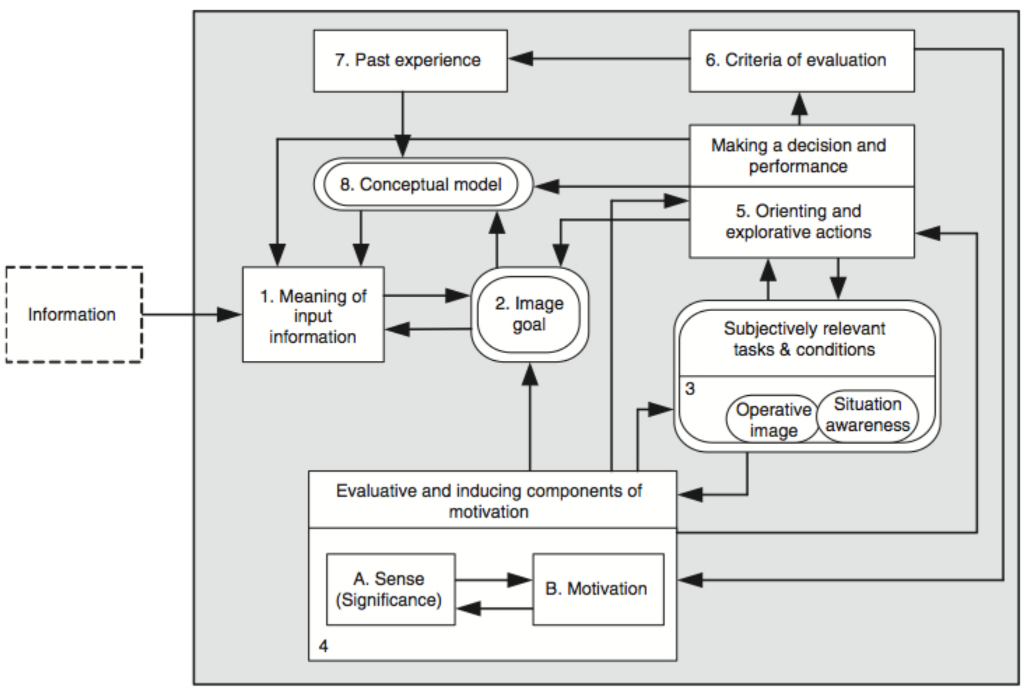


Figure 2.2. Bedny and Meister's Activity-Directed Awareness Process [4]

Based on the activity theory, Bedny and Meister clearly elucidate the functional blocks within the actor's mental model and their roles in the development of situation awareness [4]. Their model clearly shows how the actor's goals and activities play a central role to drive the process of awareness development. However, like most other situation awareness-related models, Bedny and Meister's activity theory model does not attempt to cater for, or explain the awareness phenomena in collaborative settings.

2.2 Awareness In Collaboration

The awareness phenomena in collaborative environments is indubitably more complex than situation awareness at the individual level. Salas et al. [35] point out that there is a lot more to team level awareness than merely combining individual team member's situation awareness. Beyond knowing what is going on in the environment, in their tasks, team members also need to develop an understanding of the activities of others, which provides a context of their own activities. This context is used to ensure that individual contributions are relevant to the group's shared goal as a whole [11]. This section reviews three prominent conceptualizations of awareness in collaboration that informs this study: team situation awareness, activity awareness, and distributed team awareness.

2.2.1 Team Situation Awareness

The research on team situation awareness (TSA) attempts to extend the theories and models of situation awareness to collaborative settings. Endsley et al. [14] suggest that, during team activities, situation awareness can overlap between team members, in that individuals need to perceive, comprehend and project awareness elements that are specifically related to their specific role in the team, but also elements that are required by themselves and by members of the team. Successful team performance therefore requires that individual team members have good situation awareness on their specific elements and also the same awareness for those elements that are shared. It is therefore argued that, at a simple level, team situation awareness comprises three separate but related components: individual team member's situation awareness; situation awareness of other team members; and situation awareness of the overall team (Figure 2.3).

Similar to individual situation awareness, team situation awareness should be considered as a dynamic process. Salas et al. [35] propose a framework of team situation awareness that comprises two critical processes, individual situation awareness and team processes. According to them, team situation awareness depends on communications at various levels. The perception of SA elements is influenced by the communication of mission objectives, individual tasks and roles, team capability and other team performance factors. The comprehension of awareness information (i.e. Level 2) is impacted by the interpretations made by other team members, so it is evident that SA leads to SA and also modifies SA, in that individual SA is developed and then shared with other team members, which then develops and modifies team member SA. Thus, a cyclical nature of developing individual SA, sharing SA with other team members and then modifying SA based on other team members SA is apparent.

Most attempts to understand team SA have centered on a 'shared understanding' of the

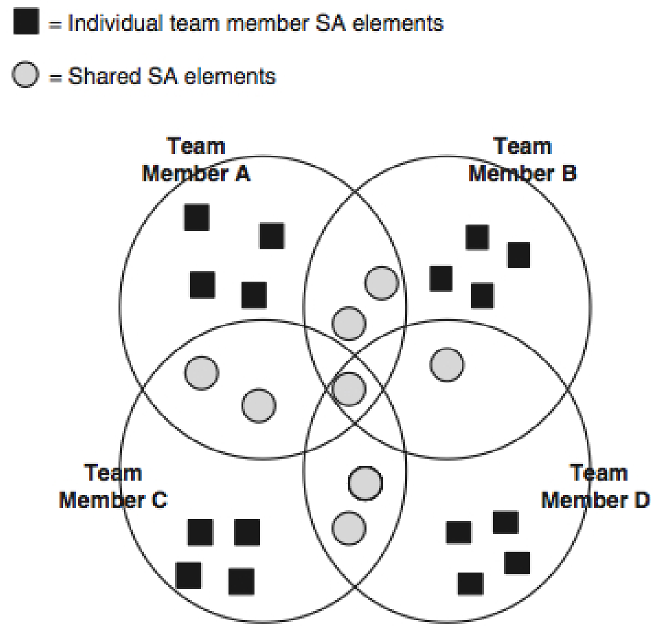


Figure 2.3. Team Situation Awareness (adapted from Endsley 1995 [15])

same situation. Nofi [30], for example, defines team SA as: 'a shared awareness of a particular situation' and Perla et al. [31] suggest that 'when used in the sense of "shared awareness of a situation", shared SA implies that we all understand a given situation in the same way'. Shu and Furuta sugggested that TSA comprises both individual SA and mutual awareness and can be defined as 'two or more individuals share the common environment, up-to-moment understanding of situation of the environment, and another person's interaction with the cooperative task' [42].

Team situation awareness has been broadly recognized as a critical factor to understand awareness in collaborative environment, because its compatibility to individual situation awareness, and the abstraction of individual awareness process from team processes [42, 37]. However, TSA also has many limitations.

1. First, a critical factor of team situation awareness is to define the configuration of shared situation awareness requirements, i.e. the degree to which team members understand which information is needed by which team member, or by the whole team. The identification of such shared awareness requirements is feasible in simple, small-scale collaborative scenarios However, for complex, real world collaborative scenarios, such a task becomes more intricate. In complex scenarios that involve numerous agents and artifacts working both collaboratively and dispersed geographically, viewing and assessing team situation awareness is actually too complex [37].

2. Second, the role of team processes, such as communication, mutual monitoring, in maintenance and development of team situation awareness, is only barely touched. It is recognized

that an increased level of team processes will lead to enhanced levels of team situation awareness. However, the specific relationships between team situation awareness and team processes remains largely unexplained [37].

3. Last, the team situation awareness adopts the knowledge-in-common view of shared mental models [27], i.e. it focuses on how the shared understanding of the same situation is developed. However, as argued by Mohammed and Dumville, the knowledge-in-common view may be appropriate for only certain task domains and types of groups [27]. For example, in teams with high level of division of work, the distribution of knowledge and skills across the team typically is not uniform, as a result, a high level of overlapping knowledge in such teams might be inefficient.

2.2.2 Activity Awareness

To address the problem of team situation awareness that posits ‘knowledge in common’ as a basis for awareness, Carroll et al. proposes a new framework for understanding awareness in collaborative environment, based on the concept of ‘activity awareness’ [6, 7]. The major distinction between team situation awareness and activity awareness is that, in realistically complex circumstances, instead of merely sharing relatively static and stable constructs such as knowledge in common, people share their activities [7]. In framing activity awareness, they appropriate the concept of *activity* from Activity Theory to emphasize that collaborators need to be aware of a whole, shared activity as complex, socially embedded endeavor, organized in dynamic hierarchies, and not merely aware of the synchronous and easily noticeable aspects of the activity [8].

Similar to the activity-directed SA model proposed by Bedny and Meister [4], activity awareness emphasizes the importance of using the concept of activity to structure the products and processes of awareness phenomena. The ultimate motivation of human actors to acquire and maintain awareness in the collaborative environment is to achieve their shared goals by performing their activities. As a result, the context surrounding a collaborative activity (e.g. the manner in which a shared activity is decomposed into smaller inter-related tasks, how these subtasks are assigned or adopted by collaborators, and when and how distributed subtasks are interdependent on each other), becomes the most important aspects of the situation that the team members need to be aware of [6].

By shifting the focus from shared knowledge to shared activity, activity awareness aligns the development of awareness with the development of collaborative activities. Most basically, activity awareness is achieved and developed through the join construction of common ground - shared knowledge and beliefs, mutually identified and agreed upon by members through a rich variety of communication protocols [7]. In long-term, open-ended activities over significant spans of time, the construction of shared practices, social capital, and human development become also important to enhance team member’s activity awareness.

Activity awareness with its basis on Activity Theory, primarily focuses on the social aspect of the awareness phenomena, i.e. how the team members’ awareness of the social context of collaborative activities is developed through common grounding, construction of shared practices, social capital, and human development. Although the authors claim that activity awareness subsumes situation awareness [6], little discussion has been given to how these higher-level social processes are connected to the cognitive processes to maintain situation awareness. Issues, such as how the state change in the external environment can lead to a team member's internal goal change, which could furthers lead to re-planning of their activities, cannot be explained.

Furthermore, the activity awareness focuses on the sharing of activities, i.e. the importance of a common picture of the shared collaborative activities. However, such a common picture is usually distributed in the whole group, instead of in any single actor's mind [48]. Each actor in the group has their own awareness, related to the goals they are working towards. However, this seldom includes the whole picture of the collaborative activity, and only when all the actors' awareness knowledge is meshed up together, the common picture emerges. Activity awareness framework provides little support to explain how the activity knowledge is distributed across multiple actors.

2.2.3 Distributed Team Awareness

A more recent theme to conceptualize awareness in collaboration is the concept of distributed or systemic team awareness [48, 3]. Distributed team awareness approaches are borne out of distributed cognition theory [23], which describes the notion of joint cognitive systems comprising the people in the system and the artifacts that they use. Within such systems, cognition is achieved through coordination between the system units [3] and is therefore viewed as an emergent property (i.e. relationship between systemic elements) of the system rather than an individual endeavor. Distributed team awareness approaches therefore view awareness in collaboration not as a shared understanding of the situation, but rather as a characteristic of the socio-technical system itself [3]. Whilst recognizing that team members possess their individual SA for a particular situation and that they may share their understanding of the situation, distributed team awareness assume that awareness is distributed across the different human and technological agents involved in collaborative systems [48].

The main difference between distributed team awareness and other TSA and activity awareness models relates to the concepts of *compatible* and *shared* awareness. *Shared* awareness accounts suggest that efficient team performance is dependent upon team members having the same awareness knowledge. Distributed team awareness, on the other hand, postulates that, within collaborative systems, different team members have unique, but *compatible* awareness, regardless of whether the information that they have access to is the same or different [48]. Team members experience a situation in different ways, as defined by their own personal experience, goals, roles, tasks, training, skills, and so on. So whilst some of the information required by two different team members may be 'shared' in the sense that they both need to attend to it as part of their job, their resultant understanding and use of it is different [38]. Ultimately, the picture developed by each team member is unique for themselves. *Compatible* awareness is therefore the phenomenon that holds distributed systems together. Each team member has their own awareness, related to the goals that they are working towards. Although different team members may have access to the same information, differences in goals, roles, the tasks being performed make them view it differently. In this way, each team members awareness is different in content, but is compatible in that it is all collectively required for the system to perform collaborative activities successfully.

While the distributed team awareness emphasizes the distribution of awareness, it does not discount the social interactions among different team members. The term ‘transactive’ awareness is used to describe the notion that distributed team awareness is acquired and maintained through *transactions* that arise from communications or other team processes [38]. A transaction in this case represents an exchange of awareness between one agent and another (where agent refers to humans and artifacts). Agents receive information, it is integrated with other information and acted on and then passed on to other agents. The interpretation on that information changes per team member. The exchange of information between team members leads to transactions in the SA being passed around. For example, an agent may perceive certain awareness element in the environment, interpret the meaning, and then pass it to another agent via a transaction. The second agent then builds its own interpretation upon the first agent’s interpretation, and may start a new transaction to pass the awareness to other agents. Hence, it is the systemic transformation of awareness elements as they cross the system boundary from one team member to another that bestows upon awareness in collaboration an emergent behavior [48].

The concept of distributed team awareness has been investigated by the authors in a number of domains, including naval warfare [47], energy distribution [36], and air traffic control [48]. The major strength of the approach is related to the systemic approach that it advocates, which is more suitable to analyze the awareness phenomena in complex, real world collaborative activities [48]. However, the main weakness, as admitted by the authors, is also related to it complexity [38]. Similar to other team situation awareness models, it uses concepts as the basic unit to analyze awareness elements, which often leads to extremely large networks in order to represent all the concepts and their relationships. A possible remedy is to integrate the distributed team awareness with the activity-directed models and switch the focus from concepts to activities to understand the awareness phenomena.

2.3 An Integrated Conceptual Model of Awareness

By reviewing the existing theories and conceptualizations of awareness, we believe that, instead of adopting one particular viewpoint, a suitable conceptual model for understanding the awareness phenomena in real world complex collaborative activities requires the integration of multiple constructs in the literature. Specifically, we identify the following requirements:

1. *The integration of individual cognitive processes and social processes.* As most theories and models of awareness in collaboration claim that individual situation awareness is still an important component in collaborative environment, the conceptual model should be able

to account for both cognitive processes at the individual level and social processes at the team level, and emphasize on how these two aspects interplay with each other.

2. *The integration of compatible and transactive aspects of awareness phenomena.* We agree with the distributed team awareness approaches [38] on that, because of the differences in goals, roles, the tasks being performed make, each team members awareness is different in content, but at the same time is compatible for the team to perform collaborative activities successfully. Hence, the conceptual model should be able to account for how the awareness is distributed across multiple team members, and meanwhile can interact with each other via transactions.

3. *The integration of awareness and activity.* We resonate with the activity-directed SA model [4] and the activity awareness framework [6] to emphasizes the importance of using the concept of activity to structure the products and processes of awareness phenomena. As the purpose of human actors to acquire and maintain awareness in the collaborative environment is to achieve their shared goals by performing their activities, it is very natural to use the activities to structure their awareness requirements.

To satisfy these requirements, we propose an integrated conceptual model of awareness in complex collaborative activities, by combining multiple constructs in the literature. In general, the model has two major components: (1) an integrative model of activities, local scopes of work, and dependencies that form **the field of collaborative work**, (2) and **a set of awareness processes** built on top of it (Figure 2.4). The goal of the former is to establish the necessary knowledge that is needed to understand the content of awareness, i.e. *aware of what*, and the later focuses on the awareness processes, i.e. *how awareness is achieved and developed.* In the following of this section, we elaborate each component in more details.

2.3.1 The Field of Work

Following the activity-directed SA model [4] and the activity awareness framework [6], we focus on the concept of activity to structure the content of awareness, i.e. the field of collaboration work, built on top of three interrelated concepts:

1. We consider *activity* as the basic unit of analysis to support coordination in geo-collaboration.

2. Due to the distributed nature of geo-collaboration, the various activities have different implications on different actors, and form their respective *local scopes of work.*

3. Activities in different local scopes of work are interdependent due to various types of *dependencies* existing among them. Dependencies serve as the bridges between different local scopes of work to evaluate the implications of remote activities transcending multiple local scopes.

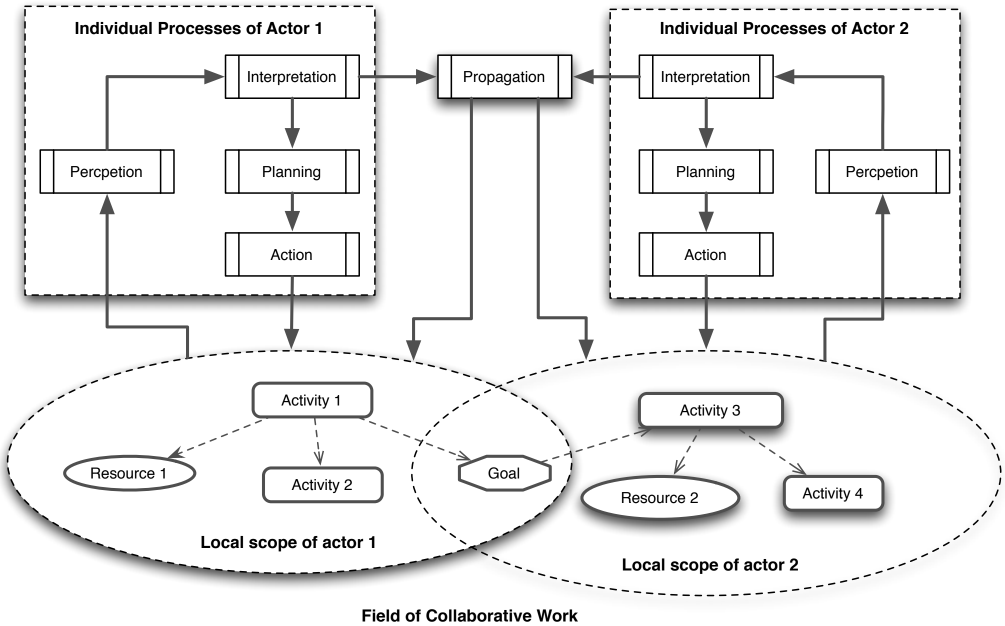


Figure 2.4. The Integrated Conceptual Model of Awareness

2.3.1.1 Activity as the basic unit

In our approach, we subscribe to Activity Theory to conceptualize human activities [28]. From this perspective, the basic structure of an activity can be defined as several basic elements and mutual relationships between them.

1. **Actions** An action specifies a particular way of doing something. An action can be either basic or complex. A basic action can be directly executed by one or multiple actors. A complex action needs to be decomposed into subsidiary actions. When an action is specified as a sub-component of a higher action, this restricts the higher action to that particular course of doing. An action is assigned to a list of actors who are responsible to executing it.

2. **Goals** A goal is a state of affairs in the world that the actors would like to achieve. How the condition is to be achieved is not specified, allowing alternatives to be considered. For example, a goal can only specify that a victim needs to be medically treated, but how it is achieved is not specified.

3. **Actors** are defined as entities capable of performing actions and capable of making decisions on the performance of their actions.

4. **Resources** are anything that can be used in the transformation process of an action, including both material resource and resources for thinking. For example, in the case

of response emergency, resources may include rescue vehicles, medical equipments, and information such as the locations of victims.

2.3.1.2 Local scope of work

Although various activities can be identified in a complex collaborative environment, each actor usually only engages in a small set of them. In fact, one of the fundamental motivations for team work is to decouple a complex problem into a set of smaller ones that are much easier to manage and tackle. One result of the decoupling is that the work of actors is distributed and each actor is only interested in the states of a small set of activities, resources, and conditions that are relevant to their roles, current goals and tasks. To characterize the distributed nature of team awareness, we define the local scope of work for each actor as the set of activities, resources, and conditions that have direct or potential impact on the actors work.

The concept of local scope of work have several characteristics of the distributed nature of emergency response activities:

1. First, the local scope of work is a relative concept that changes with the actors current goals. Whenever an actors goal is changed, the local scope of work may also be changed.

2. Second, the local scope of work may cover multiple portions of the collaborative activity as a single actor may focus on several activities at the same time.

3. Third, the local scopes of two actors can overlap with each other. It is common that the same activity, resource, or condition falls into the local scopes of different actors, even though it may have different impacts on these actors. Actually, these overlapping elements across multiple local scopes of work play an important role in supporting the transactive awareness process.

2.3.1.3 Dependencies

Although activities are largely distributed and belong to local scopes of different actors, they cannot be performed without interacting with each other. Although an actor is only responsible for or interested in the activities within her/his local scope of work, the activities outside the local scope can still potentially impact her/his work through dependencies.

Dependencies have been studied by many researchers with aspect to coordination, from organizational studies ([51, 10]), multi-agent systems (MAS) ([**?**, 44, 43]), and computer-supported cooperative work (CSCW) ([33, 9, **?**]). Dependencies within a collaborative process can be of various forms. In general, we can summarize three types of dependency relationships that have been well recognized in the literature: temporal relationships, resource-related relationships, and goal-related relationships.

1. *Temporal dependencies.* The activities of the actors might be interdependent due to the constraint of ordering them in a certain order [44]. In some case, certain activities cannot be

started until others are finished (the problem of sequencing) or a certain group of activities have to be performed at the same time (the problem of synchronization).

2. *Resource dependencies.* Resource dependencies can be analyzed in terms of common objects, i.e. resources, that are used by multiple actors or involved in multiple actions. An example is when two or more users simultaneously want to alter the same part of a document in a collaborative authoring system [?]. These common objects constrain how each activity is performed. Different patterns of use of the common objects by the activities will result in different kinds of dependency relationships [?]: A *fit dependency* occurs when multiple activities collectively produce the same resource. A *flow dependency* arises whenever one activity produces a resource that is used by another activity. A *sharing dependency* arises whenever multiple activities all use the same resource.

3. *Goal dependencies.* Goal dependencies involves the subsidiary goals that might be interdependent across various actors or their tasks [44]. A goal-related dependency reflects the fact that the depender depends on the dependee to bring about a certain state in the world. Unlike the temporal or resource-related dependencies in which the depender knows what tasks are involved in these relationships, the depender in a goal-related dependency does not care how the dependee goes about achieving the goal, i.e. the dependee is free to, and is expected to, make whatever decisions are necessary to achieve the goal [51]. A goal-related dependency is usually characterized by the structural relationship resulting from goal decomposition of the whole shared goal, i.e. task A depends on the other task B because B is a means to achieve a subsidiary goal that must be satisfied in order to perform A.

2.3.2 Awareness Processes

Built upon the field of work, a set of awareness processes can be identified to describe how the awareness is acquired and developed in a cyclical way (Figure 2.4). In general, we can identify the awareness development cycles at both the individual and team levels.

2.3.2.1 Individual Processes

At the individual level, each team member develops his/her own awareness in the similar way as described in the Neisser's perceptual cycle model [29]: the development of awareness starts with the perception of selective elements in the actor's local scope of work, which is then interpreted with the help of the actor's existing knowledge. The result of interpretation then help the actor to make certain decisions and perform actions, which then further update the actor's local scope of work. A important difference in our model from the original perceptual cycle model is that, instead of action directly stimulated by the acquired awareness knowledge, we emphasize the individual's planning process before the action performance, in which the individual needs to make decisions on what to do based on the interpretation of awareness information, such as whether he/she needs to change the goal, perform re-planning, or ask for help etc.

1. *Perception.* The process of perception is very similar to the concept in the perceptual cycle model. An actor must first be able gather perceptual information in the surrounding situation, and be able to selectively attend to those elements that are most relevant for the task at hand. The only difference is that the perception is constrained by the actor's local scope, i.e. he/she can only perceive the information about the environment, acitivities, or other objects defined in his/her local scope.

2. *Interpretation.* The process of interpretation includes generating the awareness knowledge at both the comprehension and projection levels in Endsley's three-level model [15]. The actor comprehends or understands the relevance of perceptual information in relation to their tasks and goals, and predicts likely future states in the situation.

3. *Planning.* The process of decision-making involves two major tasks. The first is to elaborate the actor's plan based on the interpretation. This involves decisions such as commitment to certain activity, focus switching, re-planning. The second task includes decisions about whether the actor wants to propagate the interpretation to other actors, which is important for initiating the awareness development cycle at the team level.

4. *Action.* Based on the result of decision-making process, the actor may perform some action to manipulate his/her local scope, which can lead to further changes that will start another round of perception.

2.3.2.2 Team Processes

Beyond the individual processes of awareness development, our conceptual model allows the transactive awareness development across multiple actors in a team. The development of awareness at team level can be performed explicitly or implicitly. Based on the interpretation of awareness information, an actor may decide to propagate the change to other actors by indicating how the other actor's activities can be impacted by his/her interpretation. The result of propagation is then perceivable by the other actor and may initiate the other actor's individual awareness development cycle. As the other actor develops his/her individual awareness, he/she may decide to propagate his/her interpretation back to the actor or other actors in the team, or he/she may perform actions that lead to changes in other actors' local scopes. In either way, new individual cycle may be initiated and the awareness development continues at the team level.

The development of awareness at the team level is guided by the dependencies among activities and the shared activities in overlapping local scopes. The dependencies allow the actors to cascade the awareness interpretation across multiple activities. The shared activities in overlapping local scopes enable the propagation process to cross the boundary of multiple local scopes.

2.3.3 An Example

We use the motivating emergency response scenario described in Chapter 1 to show an example of the conceptual framework.

Figure 2.5 illustrates how the different activities are distributed into different local scopes of work (represented as dotted circles) and how they are connected by the various dependency relationships. As we can see from the example, the different actors in the scenario, the victim manager, the decontamination manager, and the transportation manager, have very different roles to play, and therefore their local scopes vary significantly. However, due to the dependencies among their activities (For example, the decontamination manager relies on the transportation manager to deliver the victim to the decontamination station; the victim manager relies on the decontamination manager to perform decontamination on the victim), their local scopes are overlapped with each other.

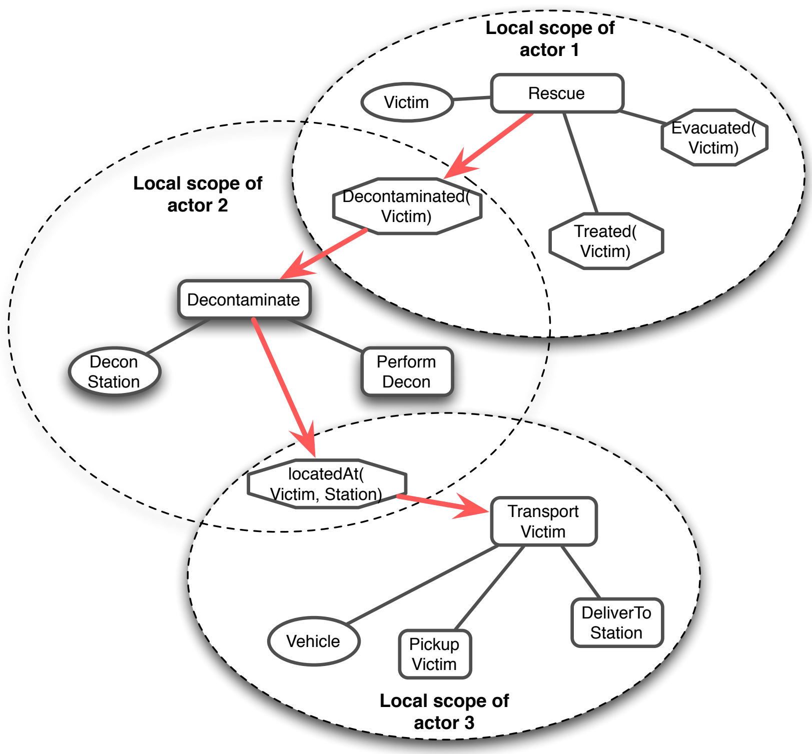


Figure 2.5. An Example: The Field of Work

Figure 2.6 demonstrates how the awareness processes work on top of the three constructs. In the beginning, Actor 1 perceives some unexpected event happening in the environment, and attempts to interpret it as whether and how it can impact the activities within her local scope of work. After understanding the event, she associates it with a state change of Activity 1. Furthermore, she predicts how the state change of this activity will impact other activities because of the dependencies among them. After Actor 1s interpretation, the event is likely to impact another activity (Activity 2), and Actor 1 decides to propagate it. The propagated event falls into Actor 2s local scope of work. Upon receiving this interpretation from Actor 1, Actor 2 first needs to understand where this event comes from by backtracking the interpretation, and evaluate how this event will impact his own line of work, which leads to a new projected state change of Activity 3. The similar process then is propagated to Actor 3s local scope of work. In this way, the teams awareness of the initial external event is collaborative developed as the relevant actors gradually attach their interpretations to it.

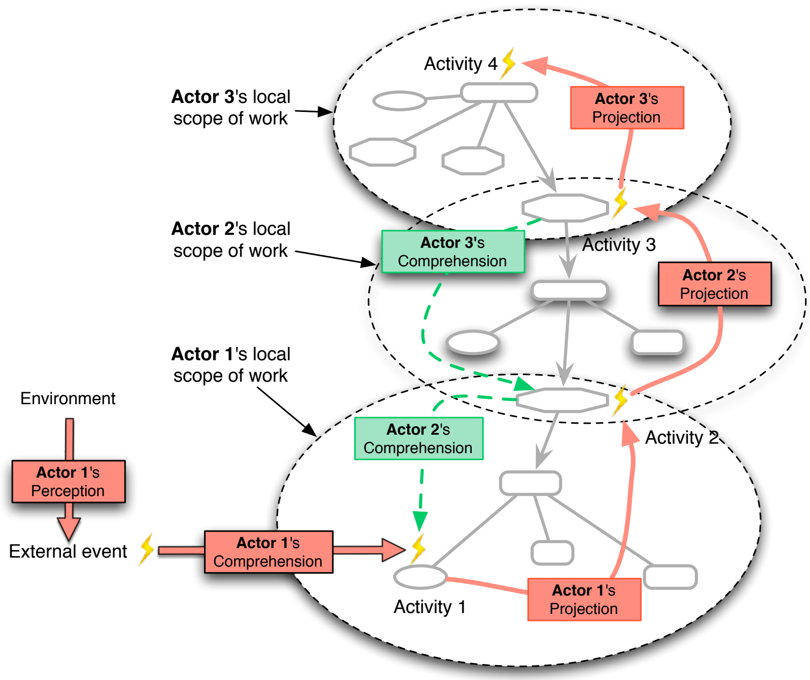


Figure 2.6. An Example: The Awareness Processes

2.3.4 Discussion

The proposed conceptual framework shows the capabilities to satisfy the three requirements we presented in the beginning of this section.

1. First, the conceptual model provides two ways to connect the individual cognitive processes and team processes. On one hand, the connection can be implicit as team members interact with the filed of work. The development of an actor's individual awareness can manipulate the field of work, which can then be perceived by other actors in their separate individual awareness processes. In addition, the actors can explicitly propagate their interpretations to other actors within their individual awareness processes.

2. Second, the concept of local scope allows us to define how the awareness knowledge is distributed across multiple actors. The different actors often attend to different sets of activities as their local scopes. At the same time, the actors' local scopes are often overlapped due to the various dependency relationships that may occur between activities, this

provides the mechanism to allow awareness transactions among multiple actors. In this way, the compatible and transactive aspects of awareness phenomena are integrated.

3. Last, our structure of the field of work is based on the activity theory, which enables the integration of awareness and activity.

Chapter 3

A Design Framework for Awareness Promotion

Following the conceptual model of awareness phenomena in complex collaborative activities, we present a design framework for awareness promotion in this chapter. The framework describes the different means to promote awareness based on our conceptual framework, and is used to review existing awareness system designs, then lead to the rationale for our computational approach.

3.1 A Review on Existing Computational Models

Event notification model (GroupDesk, PoLIAwaC et al.) Spatial model Focus/Nimbus Model Diffusion Atmosphere

Dimensions Field of Work: activity, local scope, dependencies Processes: perception, interpretation, decision-making, action, propagation Interplay: whether the filed of work can be updated by the processes whether the processes are supported by explicit representation of field of work

3.2 Formalizing the Field of Work in SharedPlan Theory

Rooted in the SharedPlans theory [19, **?**], the proposed activity model treats a collaborative activity as an evolving shared plan situated in a set of physical and mental contextual factors. In general, a shared plan is a moment-to-moment representation of an unfolding activity and includes not only a hierarchy of actions but also the set of mental states (beliefs, intentions, and commitments) that the participating actor have established towards the activity, and the knowledge preconditions and physical constraints that have to be satisfied prior to an activity being performed.

3.2.1 SharedPlan Theory

Action Decomposition

The SharedPlans theory models an activity as hierarchically structured subsidiary actions that are intended and performed in some specific context components. An **action** refers to a specific goal as well as the effort needed to achieve it. An action can be either basic or complex. A *basic* action can be directly executed by one or multiple actors. A *complex* action, however, cannot be directly executable because it needs to be decomposed into subsidiary actions. The concept of action in SharedPlan theory has a broad definition, which includes the high-level collaborative activity, the medium-level actions, and the low-level operations.

The knowledge preconditions are called **parameters** of the action that have to be identified prior to an action being performed, i.e. actors must be able to identify the parameters of the action to be performed. A parameter can represent the knowledge about an object of the action or a resource used by the action. It is shared and visible to the action it belongs to and all the sub-actions. If a parameter has not been identified, the actors need to develop a plan to identify the parameter before the action can be performed.

A **precondition** defines a certain proposition that indicates certain states of a parameter or relationships among multiple parameters must be satisfied in order to perform an action. As a result, any variables in a precondition must also appear in the action's parameter list.

Mental States

The SharedPlans theory provides a formal specification of the mental attitudes requirements of participants in a collaborative activity. The specification is given in terms of beliefs, mutual beliefs, and intentions of the participants. It deploys two intentional attitudes, intending to (do an action) and intending that (a proposition hold). Intentions-to are used to represent an agents commitments to its own actions. Such commitments instigate means-end reasoning about ways of accomplishing its intended actions that is, to planning. Intentions-that are used to represent an agents commitments toward a joint activity and the actions of its fellow participants in service of that activity. Such commitments lead an agent to engage in reasoning about other participants actions and intentions and ways the agent could contribute to their success in the context of the group activity. Besides, an indicator of the current attentional state, representing whether an action is the current focus of an actor is also included.

Partiality of Plans

A critical point made in the SharedPlans theory is that planning is interleaved with acting, which means agents can act on partial recipes, and a group of agents may not have a complete plan until after they have done some of the actions in the partial recipe. To have a partial shared plan means there are still some actions that remain to be resolved (i.e., role assignment) or decomposed further. In the beginning of the collaboration, the agents only have a partial shared plan of the activity, which merely consists of the agents initial intention of performing the activity. With the development of the activity, the agents collaboratively elaborate the activity by selecting recipe for it, breaking it down into subsidiary actions, and updating their beliefs and intentions towards these subsidiary actions. In this course, the shared plan evolves and when it reaches the status of a full shared plan, the collaboration ends up with the underlying activity successfully performed.

The SharedPlans theory has the benefits to address the aforementioned challenges of modeling an activity: (1) the internal mental states of the user are modeled as the intentions, beliefs, and mutual beliefs of the participants; (2) the activity is modeled as hierarchically structured actions that are intended and performed in some specific context components (e.g. constrains, knowledge preconditions, intentional structure, and attentional state etc.); (3) the development of an activity is modeled as the evolution from a partial shared plan to a full one.

A plan of an action is a description of the action in the specific collaborative context. The difference between a plan and a recipe of an action can be drawn from two aspects: first, a plan indicates the specific way to perform the action, which is generated in the collaborative activity. It can be considered as the internal representation of the underlying activity. But a recipe represents the external knowledge about how to perform an action. Second, a plan includes the mental states of the agents towards the action. The major components of mental states for a plan include: (1) Intentions. Based on SharedPlan theory, we distinguish between two kinds of intentions: intentions to perform an action (IntTo) and intentions that a proposition holds (IntThat). An agent intending to do an action must commit to doing that action, and must hold appropriate beliefs about its ability to perform the action. IntThat is used to represent an agents expectation that some proposition will hold or some actions will be performed (possibly by other agents). An agent intending that a proposition holds must be committed to doing what it can to help make the proposition hold. (2) Beliefs refer to what human agents believe about the action. They can be beliefs about the completion of the action, the value of a parameter, the way to perform the action, or the ability of the agents to perform the action.

3.2.2 Activities

3.2.3 Local Scopes

3.2.4 Dependencies

3.3 Event-Driven Awareness Process

3.3.1 Events: the basic unit of awareness information

The application of event-based model offers high potential for awareness support in geo-collaboration. As we described in Chapter 1, many geo-collaborative work settings are characterized by different types of dynamics, such as when cooperation breaks down, changes over time, and is perceived differently by different actors involved. Understanding the dynamics of cooperative work is extremely important as a way to understand how to design computer systems supporting coordination. If computer technology does not take into account support for the dynamic development, change, and breakdown in cooperation, the system fails [?]. As a result, we consider managing the contingencies inherent to coordination work, and assume that coordination work always become necessary when certain state of the collaborative work changes, or the normal flow of work breaks down. It is the state changes in the field of coordination work that cause the actors to identify changing dependency relations, and try to resolve possible problems or conflicts [?]. Following this argumentation, we believe that events offer a unifying and powerful abstraction to describe the necessary awareness information needed for coordination in geo-collaboration.

3.3.1.1 The concept of events

The concept of events has been used in the literature from different perspectives [?]. In general, we can distinguish two different classes: *ontological* and *cognitive* positions.

In the *ontological* position, set out by Quine [?], events and objects are not to be distinguished, as both are first-class entities that can be localized in space and time, broken into sub-parts, and arranged in a taxonomic hierarchy. Events are defined as the occurrences in the real world and *the goal of modeling the events primarily focuses on study the internal structures of the events.* For example, a traffic accident can be considered as an instance of an event type *TrafficIncident*. Event types can be arranged in a subsumption hierarchy; for example, event type *TrafficIncident* may subsume *SeriousTrafficAccident*. Events may have parts; for example, *SeriousTrafficAccident* may be composed of an event of *AccidentReport* and an event of *EmergencyResponse* [?].

On the other hand, the *cognitive* perspective defines an event as a linguistic or cognitive concept that supports the interpretation of a significant pattern of change [?]. That is, the world does not really contain events. Rather, events are the way by which agents classify certain useful and relevant patterns of change. The very same circumstance in the world might be described by any number of events. Each of these descriptions is a different way of describing the circumstances. No one description is more correct than the other, although some may be more informative for certain circumstances, in that they help predict some required information, or suggest a way of reacting to the circumstances [?].*Rather than studying the internal structure of events, modeling events from the cognitive position has mainly concentrated on what external inferences can be made from the fact that certain events are known to have occurred.* The event indicating the occurrence of a traffic accident is used to analyze possible impacts on the agent's work. For instance, the traffic accident may block the traffic flow, which make the agent's task of delivering an equipment to a destination difficult to finish on time.

In the literature of GIScience, models of geographical events have been proposed to study the dynamic geographic phenomena [?]. The concept of events in these models usually belongs to the ontological position. For example, in [?], complex geographical phenomena,such as wildfire or precipitation, have been modeled in a hierarchy of events, processes, and states. The purpose is to study the internal structure of an event. In the case of precipitation, an event marks the occurrence of precipitation in a study area. A process describes how it rains; that is the transition of precipitation states in space and time. In [?], the object-oriented approach has been adopted to model all the types of occurrences in a habor's zones. The focus here is to derive the different event types that may occur in a habor and identify the relationships between them.

In the context of awareness systems, however, events are mostly defined from the cognitive position. A common theme is that events are defined as changes that exemplify a property or relationship in some locations or objects at some time. It is an application-driven concept. A specific class of events usually defines a specific type of awareness information that is supported by an awareness system. For instance, the AREA system [17] describes events as actions performed on an artifact and the event classes are derived from the artifact class hierarchy and possible operations on them. The ENI system [?] describes events from the sensors associated with actors, shared artifacts, or any other objects that generate events related to them. The primitive events represent low-level change units obtainable from individual sensor data streams. Composite events are aggregates of primitive events from single or multiple sensor data streams. An event is represented as an attribute-value tuple that represents an instance of an event type that is supported by an awareness system.

3.3.1.2 Why Event-Driven?

1. unstable communication link
2. information push v.s. pull
3. dynamics in activities

3.3.2 Event-driven awareness process

By distinguish different types of events, the proposed awareness mechanism follows the event processing architecture (Taylor et al., 2009), and is performed in following steps:

(1) *Event generation.* Sensors recognize external events, which are represented as event objects within the system.

(2) *Event interpretation.* Event interpretation components/actors evaluate direct implications of these external events on their activities, and derive new internal events.

(3) *Event propagation.* Event propagation components/agents reason on top of these initial state changes to understand their chain effects on other activities through the web of dependency relationships.

(4) *Event notification.* All the relevant local and derived remote events are notified to the users based on their subscriptions within their local scope of interests

Figure 4.1 illustrates an example of how these steps are performed to process a single event, and how the knowledge about the field of coordination work (activities, local scopes, and dependencies) are used in the awareness process. This awareness process starts with the detection of an external event by a sensor. During the interpretation phase, the knowledge about current states of activities and their dependencies is employed as the inferential framework to understand the state changes revealed by the event, and update the current states of these components within the collaborative activity, which generates a new internal event. Due to the existence of the web of dependencies among activities, the initial internal event can (or potentially can) lead to changes outside the original actors local scope of work. The inference of effects on other remote activities is performed during the propagation step, and a list of derived internal events can be generated. If any of the events is defined within an actor's local scope of work, the actor will be notified about the change during the notification step.

Figure. 1 demonstrates how the awareness process works on top of the three constructs. In the beginning, Actor 1 perceives some unexpected event happening in the environment, and attempts to interpret it as whether and how it can impact the activities within her local scope of work. After understanding the event, she associates it with a state change of Activity 1. Furthermore, she predicts how the state change of this activity will impact other activities because of the dependencies among them. After Actor 1s projection, the event is likely to impact another activity (Activity 2), which also falls into Actor 2s local scope of work. Upon receiving this projection from Actor 1, Actor 2 first needs to understand where this projected event comes from by backtracking the interpretation, and evaluate how this event will impact his own line of work, which leads to a new projected state change of Activity 3. The similar process then is propagated to Actor 3s local scope of work. In this way, the teams awareness of the initial external event is collaborative developed as the relevant actors gradually attach their interpretations to it.

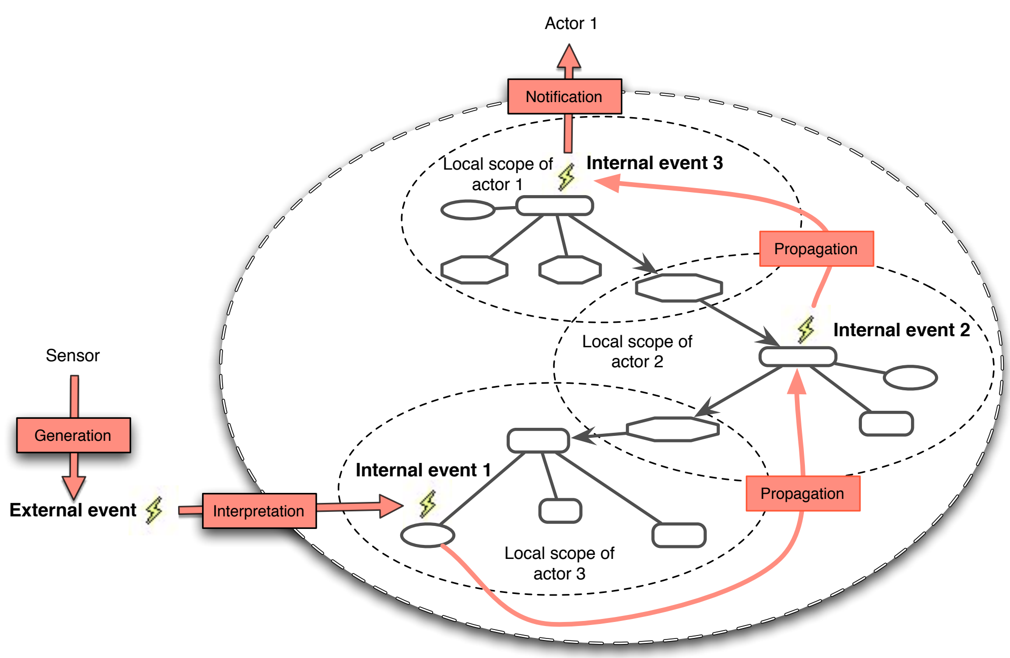


Figure 3.1. Event-based awareness process

3.4 Mediation Roles of the Computer

Our design for computer-supported awareness promotion focuses on the mediation roles that the system can play to support awareness development.

1. It provides the visual and interactive support that allow the users to interpret awareness information within their local scope of work. As the awareness information is propagated across multiple actors, the chain of reasoning can become very lengthy. By visualizing the event propagation chains with context information about activities, the users can backtrack how the awareness information is developed from the origins.

2. It supports the relay of awareness information across local scopes. Although the actors have best knowledge within their local scopes of work to interpret and develop awareness information, they often do not know who will be interested in receiving their interpretations. By maitaining a computational model of the ongoing activities and local scopes, the system can recognize whenever the interpretation transcends the boundaries of multiple local scopes and relay the awareness informaton to other actors.

3. With the situated knowledge about activities and their dependencies, the system can play a even more proactive role to automate the reasoning on how certain awareness information impact the states of activities as its effects propagate across multiple activities. In this way, the system can offload some of the interpretation effort from the human actors. Meanwhile, the human actors should have the capability to revise the systems interpretation.

In this study, we operationize these mediation roles within an event notification system to promote awareness. We consider events as basic units of awareness information, representing changes concerning the environment, resources, goals, and activities in a collaborative setting. We make distinctions between external (in the environment) and internal events (in the activities). External events are changes in the physical environment that have impact on the performance of activities. Internal events are state changes on any resources, goals, and activities involved in the collaboration.

By adopting the event-driven approach, the mediation roles of the system to support transitive awareness process can be illustrated in Figure 4.2. Whenever an external event is sensed by the system, it first matches the events with all the users predefined local scope of work and their subscriptions. If a match is found, the system notified the matched user (e.g. Actor 1) about the external event. Then upon perceiving the event, Actor 1 interprets the direct impacts of this event on his/her current activities and predicts future changes to other activities within his/her local scope of work. The inferred state changes represented as internal events are then sent back to the system. The system reasons about how these predicted changes by Actor 1 may impact other actors due to the dependencies among activities, and notify the projected state changes to other relevant users (e.g. Actor 2). Upon receiving the notification about the projected state change, Actor 2 needs to interpret the change by backtracking where this state change comes from and relating it to the activities within his/her local scope of work. Actor 2 may reject the system inference by revising the belief about the projected state change, or predict further changes due to this change, which leads to a new cycle of reasoning.

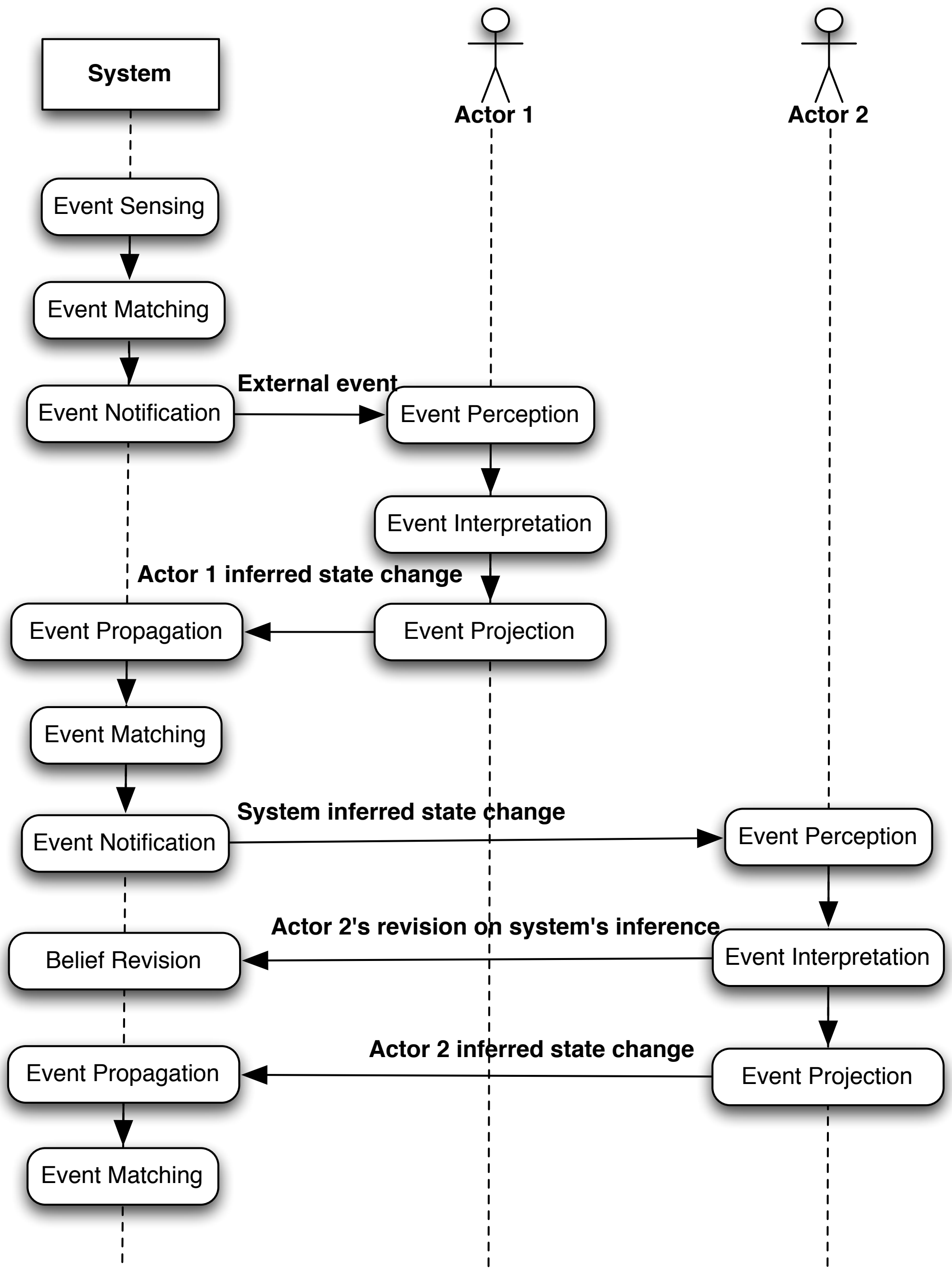


Figure 3.2. Event-based awareness process

Chapter 4

Our Approach: Overview

Following the conceptual model of awareness phenomena in complex collaborative activities, we present our computational framework of awareness promotion in this chapter. The framework includes two major components: the formalization of the field of work based on SharedPlan theory and the event-driven awareness promotion processes. Before presenting our computational framework, we use the conceptual model as a schema to compare some dominant computational models on awareness promotion in collaborative systems, which motivates us to propose our new computational framework. Then the two major components of our computational framework are described. Following that, we discuss the different types of roles the computer system can play within the computational framework.

4.1 Computational Model of the Field of Work

4.2 Event-Driven Awareness Process

4.2.1 Events: the basic unit of awareness information

The application of event-based model offers high potential for awareness support in geo-collaboration. As we described in Chapter 1, many geo-collaborative work settings are characterized by different types of dynamics, such as when cooperation breaks down, changes over time, and is perceived differently by different actors involved. Understanding the dynamics of cooperative work is extremely important as a way to understand how to design computer systems supporting coordination. If computer technology does not take into account support for the dynamic development, change, and breakdown in cooperation, the system fails [?]. As a result, we consider managing the contingencies inherent to coordination work, and assume that coordination work always become necessary when certain state of the collaborative work changes, or the normal flow of work breaks down. It is the state changes in the field of coordination work that cause the actors to identify changing dependency relations, and try to resolve possible problems or conflicts [?].

Following this argumentation, we believe that events offer a unifying and powerful abstraction to describe the necessary awareness information needed for coordination in geo-collaboration.

4.2.1.1 The concept of events

The concept of events has been used in the literature from different perspectives [?]. In general, we can distinguish two different classes: *ontological* and *cognitive* positions.

In the *ontological* position, set out by Quine [?], events and objects are not to be distinguished, as both are first-class entities that can be localized in space and time, broken into sub-parts, and arranged in a taxonomic hierarchy. Events are defined as the occurrences in the real world and *the goal of modeling the events primarily focuses on study the internal structures of the events.* For example, a traffic accident can be considered as an instance of an event type *TrafficIncident*. Event types can be arranged in a subsumption hierarchy; for example, event type *TrafficIncident* may subsume *SeriousTrafficAccident*. Events may have parts; for example, *SeriousTrafficAccident* may be composed of an event of *AccidentReport* and an event of *EmergencyResponse* [?].

On the other hand, the *cognitive* perspective defines an event as a linguistic or cognitive concept that supports the interpretation of a significant pattern of change [?]. That is, the world does not really contain events. Rather, events are the way by which agents classify certain useful and relevant patterns of change. The very same circumstance in the world might be described by any number of events. Each of these descriptions is a different way of describing the circumstances. No one description is more correct than the other, although some may be more informative for certain circumstances, in that they help predict some required information, or suggest a way of reacting to the circumstances [?].*Rather than studying the internal structure of events, modeling events from the cognitive position has mainly concentrated on what external inferences can be made from the fact that certain events are known to have occurred.* The event indicating the occurrence of a traffic accident is used to analyze possible impacts on the agent's work. For instance, the traffic accident may block the traffic flow, which make the agent's task of delivering an equipment to a destination difficult to finish on time.

In the literature of GIScience, models of geographical events have been proposed to study the dynamic geographic phenomena [?]. The concept of events in these models usually belongs to the ontological position. For example, in [?], complex geographical phenomena,such as wildfire or precipitation, have been modeled in a hierarchy of events, processes, and states. The purpose is to study the internal structure of an event. In the case of precipitation, an event marks the occurrence of precipitation in a study area. A process describes how it rains; that is the transition of precipitation states in space and time. In [?], the object-oriented approach has been adopted to model all the types of occurrences in a habor's zones. The focus here is to derive the different event types that may occur in a habor and identify the relationships between them.

In the context of awareness systems, however, events are mostly defined from the cognitive position. A common theme is that events are defined as changes that exemplify a property or relationship in some locations or objects at some time. It is an application-driven concept. A specific class of events usually defines a specific type of awareness information that is supported by an awareness system. For instance, the AREA system [17] describes events as actions performed on an artifact and the event classes are derived from the artifact class hierarchy and possible operations on them. The ENI system [?] describes events from the sensors associated with actors, shared artifacts, or any other objects that generate events related to them. The primitive events represent low-level change units obtainable from individual sensor data streams. Composite events are aggregates of primitive events from single or multiple sensor data streams. An event is represented as an attribute-value tuple that represents an instance of an event type that is supported by an awareness system.

4.2.1.2 Why Event-Driven?

1. unstable communication link
2. information push v.s. pull
3. dynamics in activities

4.2.2 Event-driven awareness process

By distinguish different types of events, the proposed awareness mechanism follows the event processing architecture (Taylor et al., 2009), and is performed in following steps:

(1) *Event generation.* Sensors recognize external events, which are represented as event objects within the system.

(2) *Event interpretation.* Event interpretation components/actors evaluate direct implications of these external events on their activities, and derive new internal events.

(3) *Event propagation.* Event propagation components/agents reason on top of these initial state changes to understand their chain effects on other activities through the web of dependency relationships.

(4) *Event notification.* All the relevant local and derived remote events are notified to the users based on their subscriptions within their local scope of interests

Figure 4.1 illustrates an example of how these steps are performed to process a single event, and how the knowledge about the field of coordination work (activities, local scopes, and dependencies) are used in the awareness process. This awareness process starts with the detection of an external event by a sensor. During the interpretation phase, the knowledge about current states of activities and their dependencies is employed as the inferential framework to understand the state changes revealed by the event, and update the current states of these components within the collaborative activity, which generates a new internal event. Due to the existence of the web of dependencies among activities, the initial internal event can (or potentially can) lead to changes outside the original actors local scope of work. The inference of effects on other remote activities is performed during the propagation step, and a list of derived internal events can be generated. If any of the events is defined within an actor's local scope of work, the actor will be notified about the change during the notification step.

Figure. 1 demonstrates how the awareness process works on top of the three constructs. In the beginning, Actor 1 perceives some unexpected event happening in the environment, and attempts to interpret it as whether and how it can impact the activities within her local scope of work. After understanding the event, she associates it with a state change of Activity 1. Furthermore, she predicts how the state change of this activity will impact other activities because of the dependencies among them. After Actor 1s projection, the event is likely to impact another activity (Activity 2), which also falls into Actor 2s local scope of work. Upon receiving this projection from Actor 1, Actor 2 first needs to understand where this projected event comes from by backtracking the interpretation, and evaluate how this event will impact his own line of work, which leads to a new projected state change of Activity 3. The similar process then is propagated to Actor 3s local scope of work. In this way, the teams awareness of the initial external event is collaborative developed as the relevant actors gradually attach their interpretations to it.

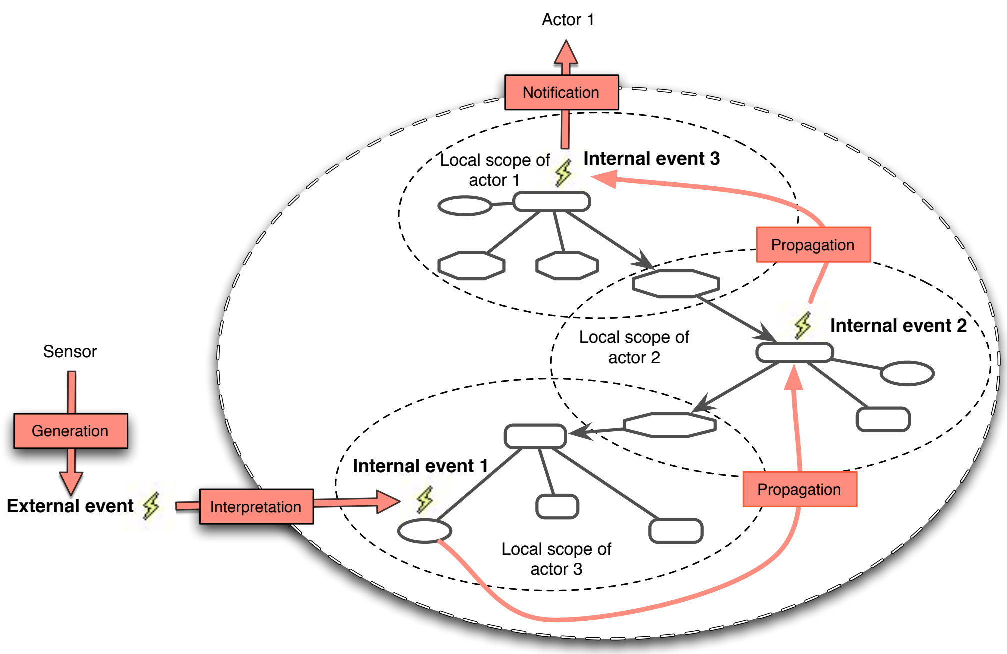


Figure 4.1. Event-based awareness process

4.3 Mediation Roles of the Computer

Our design for computer-supported awareness promotion focuses on the mediation roles that the system can play to support awareness development.

1. It provides the visual and interactive support that allow the users to interpret awareness information within their local scope of work. As the awareness information is propagated

across multiple actors, the chain of reasoning can become very lengthy. By visualizing the event propagation chains with context information about activities, the users can backtrack how the awareness information is developed from the origins.

2. It supports the relay of awareness information across local scopes. Although the actors have best knowledge within their local scopes of work to interpret and develop awareness information, they often do not know who will be interested in receiving their interpretations. By maitaining a computational model of the ongoing activities and local scopes, the system can recognize whenever the interpretation transcends the boundaries of multiple local scopes and relay the awareness informaton to other actors.

3. With the situated knowledge about activities and their dependencies, the system can play a even more proactive role to automate the reasoning on how certain awareness information impact the states of activities as its effects propagate across multiple activities. In this way, the system can offload some of the interpretation effort from the human actors. Meanwhile, the human actors should have the capability to revise the systems interpretation.

In this study, we operationize these mediation roles within an event notification system to promote awareness. We consider events as basic units of awareness information, representing changes concerning the environment, resources, goals, and activities in a collaborative setting. We make distinctions between external (in the environment) and internal events (in the activities). External events are changes in the physical environment that have impact on the performance of activities. Internal events are state changes on any resources, goals, and activities involved in the collaboration.

By adopting the event-driven approach, the mediation roles of the system to support transitive awareness process can be illustrated in Figure 4.2. Whenever an external event is sensed by the system, it first matches the events with all the users predefined local scope of work and their subscriptions. If a match is found, the system notified the matched user (e.g. Actor 1) about the external event. Then upon perceiving the event, Actor 1 interprets the direct impacts of this event on his/her current activities and predicts future changes to other activities within his/her local scope of work. The inferred state changes represented as internal events are then sent back to the system. The system reasons about how these predicted changes by Actor 1 may impact other actors due to the dependencies among activities, and notify the projected state changes to other relevant users (e.g. Actor 2). Upon receiving the notification about the projected state change, Actor 2 needs to interpret the change by backtracking where this state change comes from and relating it to the activities within his/her local scope of work. Actor 2 may reject the system inference by revising the belief about the projected state change, or predict further changes due to this change, which leads to a new cycle of reasoning.

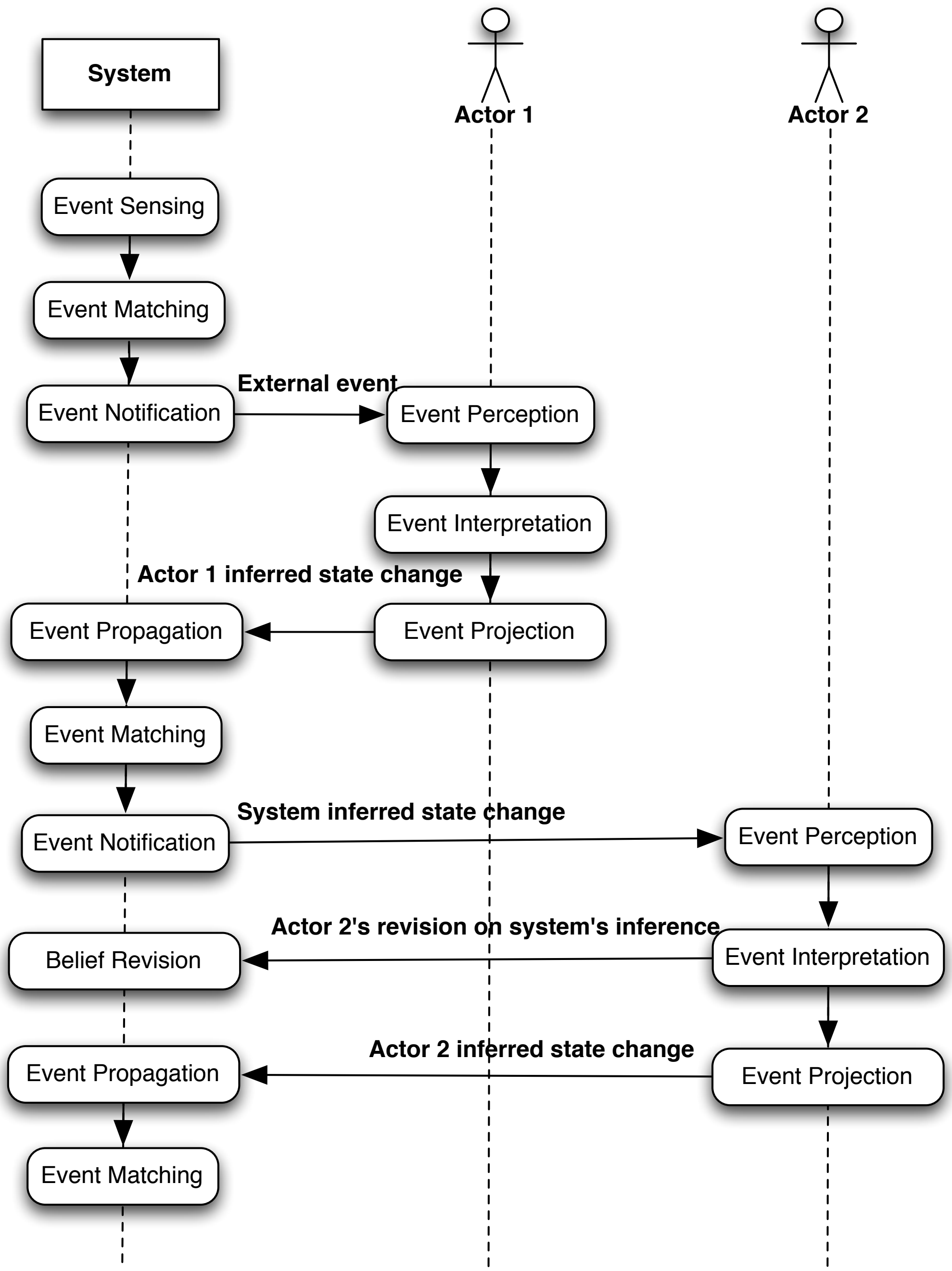


Figure 4.2. Event-based awareness process

Chapter 5

Our Approach: Modeling the Filed of Work

We employ the computational theory of SharedPlan [19, ?] to represent collaborative activities as shared plans, and then use this model of activities to derive the knowledge about local scopes of work and dependencies. In the following parts of this chapter, we first describe the basis and different components of the SharedPlan model, then describe how the different components of the filed work can be represented using the PlanGraph model.

5.1 Formalizing the Field of Work in SharedPlan Theory

Rooted in the SharedPlans theory [19, ?], the proposed activity model treats a collaborative activity as an evolving shared plan situated in a set of physical and mental contextual factors. In general, a shared plan is a moment-to-moment representation of an unfolding activity and includes not only a hierarchy of actions but also the set of mental states (beliefs, intentions, and commitments) that the participating actor have established towards the activity, and the knowledge preconditions and physical constraints that have to be satisfied prior to an activity being performed.

5.1.1 SharedPlan Theory

Action Decomposition

The SharedPlans theory models an activity as hierarchically structured subsidiary actions that are intended and performed in some specific context components. An **action** refers to a specific goal as well as the effort needed to achieve it. An action can be either basic or complex. A *basic* action can be directly executed by one or multiple actors. A *complex* action, however, cannot be directly executable because it needs to be decomposed into subsidiary actions. The concept of action in SharedPlan theory has a broad definition, which includes the high-level collaborative activity, the medium-level actions, and the low-level operations.

The knowledge preconditions are called **parameters** of the action that have to be identified prior to an action being performed, i.e. actors must be able to identify the parameters of the action to be performed. A parameter can represent the knowledge about an object of the action or a resource used by the action. It is shared and visible to the action it belongs to and all the sub-actions. If a parameter has not been identified, the actors need to develop a plan to identify the parameter before the action can be performed.

A **precondition** defines a certain proposition that indicates certain states of a parameter or relationships among multiple parameters must be satisfied in order to perform an action. As a result, any variables in a precondition must also appear in the action's parameter list.

Mental States

The SharedPlans theory provides a formal specification of the mental attitudes requirements of participants in a collaborative activity. The specification is given in terms of beliefs, mutual beliefs, and intentions of the participants. It deploys two intentional attitudes, intending to (do an action) and intending that (a proposition hold). Intentions-to are used to represent an agents commitments to its own actions. Such commitments instigate means-end reasoning about ways of accomplishing its intended actions that is, to planning. Intentions-that are used to represent an agents commitments toward a joint activity and the actions of its fellow participants in service of that activity. Such commitments lead an agent to engage in reasoning about other participants actions and intentions and ways the agent could contribute to their success in the context of the group activity. Besides, an indicator of the current attentional state, representing whether an action is the current focus of an actor is also included.

Partiality of Plans

A critical point made in the SharedPlans theory is that planning is interleaved with acting, which means agents can act on partial recipes, and a group of agents may not have a complete plan until after they have done some of the actions in the partial recipe. To have a partial shared plan means there are still some actions that remain to be resolved (i.e., role assignment) or decomposed further. In the beginning of the collaboration, the agents only have a partial shared plan of the activity, which merely consists of the agents initial intention of performing the activity. With the development of the activity, the agents collaboratively elaborate the activity by selecting recipe for it, breaking it down into subsidiary actions, and updating their beliefs and intentions towards these subsidiary actions. In this course, the shared plan evolves and when it reaches the status of a full shared plan, the collaboration ends up with the underlying activity successfully performed.

The SharedPlans theory has the benefits to address the aforementioned challenges of modeling an activity: (1) the internal mental states of the user are modeled as the intentions, beliefs, and mutual beliefs of the participants; (2) the activity is modeled as hierarchically structured actions that are intended and performed in some specific context components (e.g. constrains, knowledge preconditions, intentional structure, and attentional state etc.); (3) the development of an activity is modeled as the evolution from a partial shared plan to a full one.

A plan of an action is a description of the action in the specific collaborative context. The difference between a plan and a recipe of an action can be drawn from two aspects: first, a plan indicates the specific way to perform the action, which is generated in the collaborative activity. It can be considered as the internal representation of the underlying activity. But a recipe represents the external knowledge about how to perform an action. Second, a plan includes the mental states of the agents towards the action. The major components of mental states for a plan include: (1) Intentions. Based on SharedPlan theory, we distinguish between two kinds of intentions: intentions to perform an action (IntTo) and intentions that a proposition holds (IntThat). An agent intending to do an action must commit to doing that action, and must hold appropriate beliefs about its ability to perform the action. IntThat is used to represent an agents expectation that some proposition will hold or some actions will be performed (possibly by other agents). An agent intending that a proposition holds must be committed to doing what it can to help make the proposition hold. (2) Beliefs refer to what human agents believe about the action. They can be beliefs about the completion of the action, the value of a parameter, the way to perform the action, or the ability of the agents to perform the action.

5.1.2 Activities

5.1.3 Local Scopes

5.1.4 Dependencies

5.2 Representing the Field of Work with PlanGraph model

By modeling an collaborative activity within a PlanGraph, the knowledge about different components in the field of work can be represented as different components of the PlanGraph, and relations among them (Figure 5.1).

5.2.1 Structure of the PlanGraph

A PlanGraph is a schematic tree representation of a shared plan about a collaborative activity in a hierarchical way (Figure 5.2). The root of a PlanGraph is the overall collaborative activity, which is decomposed recursively as actions through the adoption of recipes. The whole tree therefore is the plan corresponding to the root action, while each sub-tree with a sub-action as the root represents the plan for that sub-action. In a PlanGraph, we also handle the knowledge-preconditions as a special type of node parameters. Nodes with oval shape in Figure 5.2 indicate parameters, and nodes with rectangle shape represent subplans. A plan underneath a parameter node is the plan for identifying the parameter.

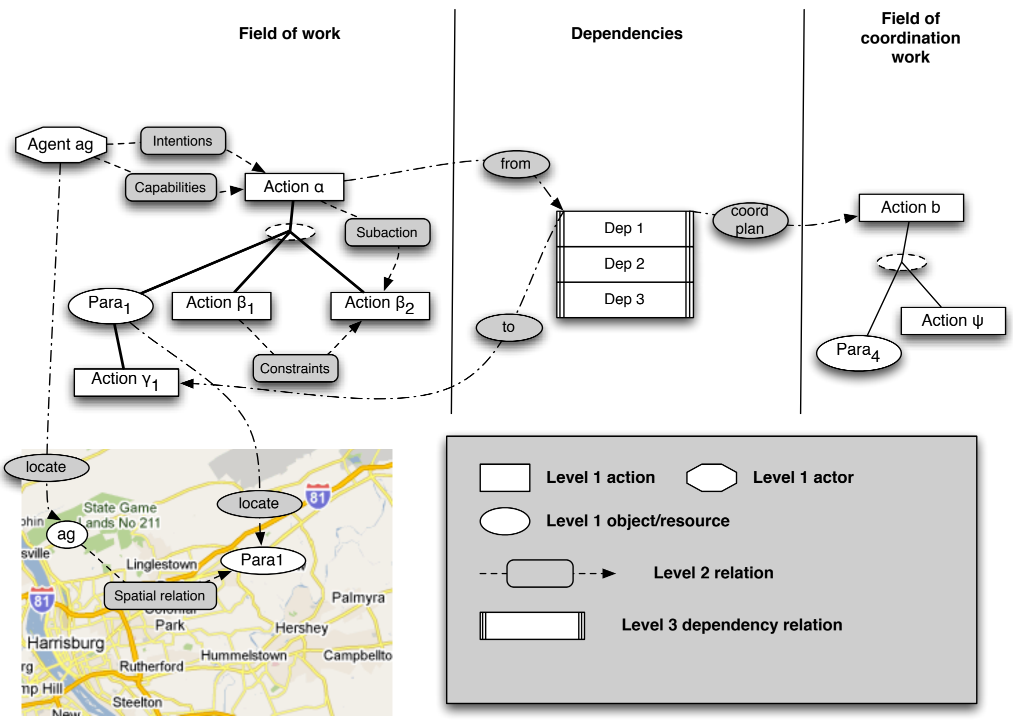


Figure 5.1. Representing knowledge with PlanGraph

Unlike the notion of recipe graph (Rgraph) developed by Lochbaum (1998), a PlanGraph not only represents how a complex action is decomposed into sub-actions, but also encodes the mental states of agents towards each action and sub-action. Therefore, each node in the PlanGraph includes several slots to store the participating agents' mental states: (a) Intentions are slots recording the system's beliefs about intentions of each agent towards the action; (b) Capability indicates the systems belief about the ability to perform the action, such as whether they can identify a recipe for the action or whether the agents can bring it about; (c) Beliefs are slots for recording the system's beliefs about what the other agents also believe about.

Therefore, the development of a collaborative activity can be represented as a dynamic process of constructing a PlanGraph. Before the collaboration begins, the PlanGraph is empty. As agents perform some actions to fulfill the shared goal, new plan nodes are introduced to the PlanGraph. If the root action in the PlanGraph is complex, agents will decompose it by selecting a recipe collaboratively. Then, agents move their attention to the parameters and sub-actions. These sub-actions may themselves be complex, and will become the subjects of further elaboration to form a sub-plan for performing. In the course of development, the PlanGraph will store the mental states of the agents toward each action node and update them in corresponding slots.

In general, a PlanGraph represents the dynamic knowledge about the underlying collaborative activity. It indicates those beliefs and intentions of the agents that have been established at that point, the actions that have been performed, and the parameters about these actions.

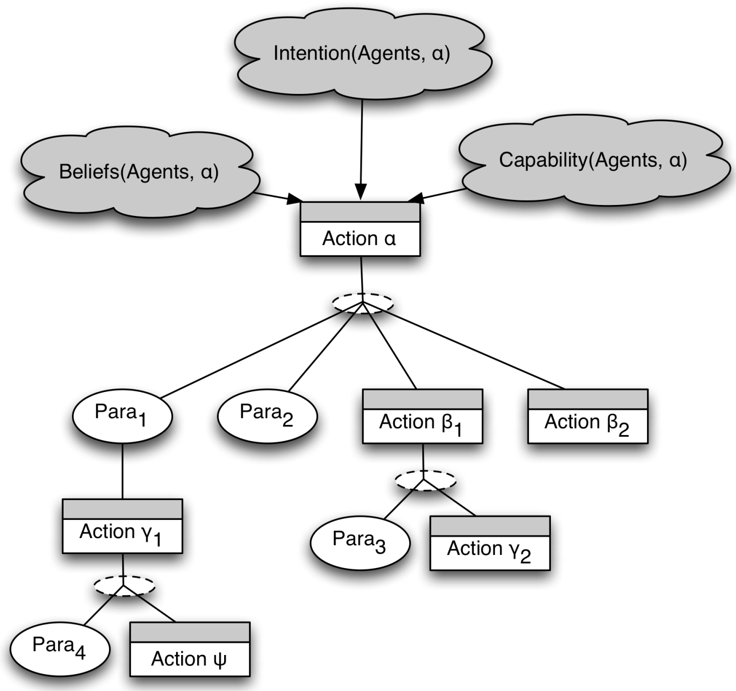


Figure 5.2. Structure of a PlanGraph

5.2.2 Representing activities

5.2.2.1 Representing basic elements

The four basic elements we define in a geo-collaborative activity are: actors, actions, resources, and objects. They can be modeled as the first-class components in a PlanGraph structure.

Actors as agents of plans.

Actors are represented as agents responsible for a plan/sub-plan and capable of performing actions in the plan. Each agent node in the PlanGraph maintain the current state of the corresponding actor, such as the location, the role of the actor, and the expertise he/she has. Each agent node also includes an cursor that is always attached to an action that is the current focus of the corresponding actor in the whole plan. In addition, each agent node has a unique ID which allows the retrieval of more long-term contextual knowledge (e.g. gender, age, familiarity with the activity involved, ethnicity, etc.) about this particular actor from the knowledge base.

Actions as plans/sub-plans

Actions are modeled within the hierarchical structure of the PlanGraph as plans/sub-plans. The goal of the action is represented as a plan node, including information about various properties, such as the type, the complexity, or the execution status of the action. Furthermore, the selected plan to perform this action can be derived from the subsidiary parameter or action nodes, which can include sub-plans by themselves.

Objects and resources as parameters

Objects and resources are modeled as parameters in a PlanGraph. They define the knowledge or physical pre-conditions that have to be satisfied prior to an action being performed. Each parameter node records information about the current states of the corresponding object or resource in the domain, such as the location of a rescue vehicle, or the capacity of the decontamination station. Parameter nodes can be collective or individual. A collective parameter indicates an object or resource with multiple values, for instance, the parameter representing all the victims in a rescue operation. An individual parameter represents an instance of an object or resource, such as the rescue vehicle used in a particular delivery task.

5.2.2.2 Representing relations

The second level of knowledge in the field of work includes the relations among the basic elements in the activity structure and in the geographic space. They can be modeled as the relations among components in a PlanGraph structure, or inferred from their properties.

Relations between actors and their actions

The relations between actors and their actions are modeled based on the formal specification of mental attitudes requirements of participants in a collaborative activity. The specification is given in terms of beliefs, mutual beliefs, and intentions of the participants. Intentions are used to represent an agents commitments to actions. Besides, several belief predicates are also defined to represent the knowledge an agent has about its ability to perform an action. Each action node in a PlanGraph includes several slots to To model these mental attitudes: (a) *Intentions* are slots recording the intentions of each participating actor towards the corresponding action; (b) *Capability* indicates the systems belief about the ability of each participant to perform this action, such as whether they can identify a recipe or they can bring it about.

Relations between actions

The two major types of relations between actions are composition and precedence. The former is directly encoded in the hierarchical structure of a PlanGraph. Each action node has a slot *subactions* recording its subsidiary actions under current plan. As a result, the fact that acton a_1 is included in the *subactions* slot of action a_2 represents the relation that action a_1 is the subsidiary action of doing a_2. The precedence relation that reflects the temporal order of doing two actions is encoded in the *constraints* slot of each action node. In a PlanGraph, each action node also includes a slot *constraints* to recording the systems beliefs about a set of propositions, each of which represents certain constraint information about the action, e.g. the deadline of doing the action, the actions that need to be performed before the action, etc. The temporal order of two actions can be represented in their respective *constraints*, indicating that the other action must be performed before/after this action.

Relations between objects/resources and actions

The relations between object/resources and actions are modeled as the composition relations between action nodes and parameter nodes in a PlanGraph. Each action node has a slot *parameters* recording all the parameters under current plan, and each parameter has a slot *subactions* recording the current plan to identify or manipulate the parameter. The distinction between objects and resources is relative, depending on their relations to the actions in the PlanGraph. A parameter is an object to the action that is used to identify or manipulate its value, but it is a resource to the action that is at the same level of the parameter and use the parameter during the performance.

Spatial relations

The spatial relations between basic elements in the field of work are not represented directly in the PlanGraph. Instead, all the basic elements with a geographic dimension in the PlanGraph are geo-referenced as properties. Each actor's current location is stored as an attribute of each agent node. Each parameter, if its values have geographic meaning, such as the location of a rescue vehicle, or the shape of an incident area, are stored with corresponding coordinates in the geographic reference framework. In this way, although the spatial relations among these elements are not directly recorded in the PlanGraph, they can be easily perceived or calculated based on their references in the geographic space.

5.2.3 Representing local scopes

To be Added.

5.2.4 Representing dependencies

According to the SharedPlans theory, the commitment to an action entails different types of obligations to the components of this action's current plan. Based on the SharedPlans theory, we can utilize the following rules to detect basic dependency relations in the activity structure:

1. **Action Performance**: if an agent or a group of agent intend to develop a FSP of collaborative action a, then they will also intend to develop a FSP for each subsidiary action:
   $Int.Th(ag, a, t) \wedge b_i \in Recipes(a) \Rightarrow Int.Th(ag, b_i, t)$
   This rule can be used to detect the subsidiary dependency relation between two action nodes. The fact that a_1 is a subsidiary action node under a_2 indicates a dependency of $subsidiary(a_1, a_2)$.

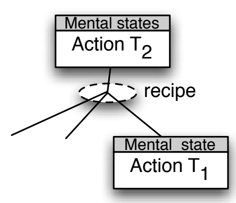
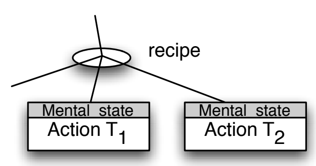
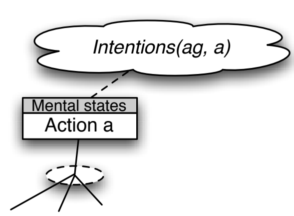
2. **Parameter Identification**: the formalization in SharedPlans specifies that all the agents must intend-that the parameters of the group action be identified. If the actors have the intention that they will develop a FSP of collaborative action a, then they will also intend that they will develop a FSP to identify each parameter of a:
   $Int.Th(ag, a, t) \wedge p_i \in Params(a) \Rightarrow Int.Th(ag, id.Params(p_i), t)$
   This rule can be used to detect the dependency relations between an action and a parameter node. $consume(a, r)$ is detected when a parameter node representing a resource r and an action node a are siblings; $producer(a, r)$ can be extracted from the fact that a is a subsidiary action node under the parameter node representing a resource r.

3. **Actor Commitment**: In order to develop a FSP of an collaborative action a, at least one of the actors need to commit to the action and has the capability to perform it:

$FSP(a, a_t) \wedge ag \in Agents(a) \Rightarrow Int.Th(ag, a, a_t) \wedge CBA(ag, a, a_t)$
This rule can be used to detect the dependency relations between an action and an agent. $commit(ag, a)$ relation between an agent and an action is encoded in the *Intentions* slot attached to each action node, and $capable(ag, a)$ is encoded in the *Capability* slot attached to each action node.

4. **Constraint Satisfaction**: it requires that all the members of the group be committed to making sure that the constraints for doing action a will hold:
$Int.Th(ag, a, a_t) \Rightarrow Int.Th(ag, constr(a), a_t)$
This rule can be used to detect the dependency relations that are encoded in the *constraints* slot of each plan node. For example, if the one of the constraints of performing action a_1 indicates that a_2 must be performed at a time before a_1, then the dependency relation of $precede(a_2, a_1)$ can be detected.

Table 5.1 indicates the basic dependency relations that can be detected based on these rules.

Dependency	Structure in PlanGraph
$subsidiary(a_1, a_2)$	a_1 is a subsidiary plan node under a_2: 
$precede(a_1, a_2)$	a_1 and a_2 as ordered siblings: 
$commit(ag, a)$	the *Intentions* slot attached to each plan node: 

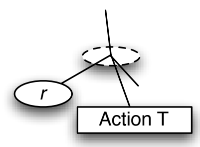
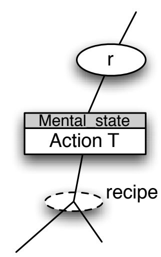
$consume(a, r)$	a parameter node representing a resource r and a plan node T as siblings: 
$produce(a, r)$	a is a subsidiary action node under the parameter node representing a resource r: 

Table 5.1: Identifying basic dependencies in PlanGraph

5.3 Construction and Development of the Model

5.3.1 The Development of PlanGraph

Considering the collaborative activity as a process from a partial SharedPlan to a full SharedPlan among the participants, the PlanGraph, which is used to represent the collaborative activity, should also be considered as dynamically developed. To facilitate the development of PlanGraph, the PlanGraph model provides the reasoning mechanisms to decide how the interaction between the system and human participants, or between the human participants, influences the state of collaborative activity and update the changes in the PlanGraph. In general, the reasoning mechanism in PlanGraph model includes two steps: the plan explanation and the plan elaboration.

Plan Explanation

Participants in a collaborative activity need to interact with each other to contribute to the activity. It is through the communication between the participants that collaboration can be enacted. Therefore, the input of an agent can be treated as the way that the agent tries to express their intentions to the other agents in the collaborative activity. Plan explanation is the step where the system attempts to explain how the meanings of the new piece of interaction relate to the current PlanGraph. In general, the new input is said to be meaningfully merged with the current collaboration context if one of the following conditions is satisfied:

1. The new input provides a piece of information that helps the agents to assign/change a value to a parameter in the current PlanGraph.

2. The new input provides more details on the performance of a complex action in the PlanGraph. For example, the user suggests performing a sub-action or identifying a parameter in order to complete the action.

3. The new input helps to establishing necessary mental states for a collaborative action. In this case, the agent's new input serves as an indicator of the speaker's mental attitudes, such as confirmation or refusal towards certain states of fairs.

If the new user input can be successfully explained in current PlanGraph, it will be merged to update the PlanGraph accordingly, i.e. the contextual information about the ongoing activity is changed.

Plan Elaboration

The main goal in elaboration stage is to advance the collaborative activity from the system side based on the change from the new input. After the interpretation process, the context of the activity is changed. Therefore, the system needs to elaborate the PlanGraph to accommodate the changes. The elaboration process begins with the root node of the PlanGraph and uses the depth-first traverse to visit the whole plan based on reasoning rules associated with PlanGraph model. The elaboration ends when no more parts of the PlanGraph can be further elaborated.

The system can elaborate the PlanGraph in several ways. First, the system can contribute to the collaborative activity by retrieving a recipe for an action from the knowledge base. Second, the system can execute the basic actions that can be performed without further inputs from other human agents. Lastly, the system can identify the value for a parameter by retrieving the default or suggested values from the knowledge base.

After the elaboration process, the PlanGraph again reflects the current state of the collaborative activity. Therefore, the information from the PlanGraph can be used by the system to represent the current state of the activity context.

This elaboration process also attempts to discover all the open nodes in the PlanGraph that might be addressed later in the collaboration, such as unidentified parameters, unperformed actions, conflicting beliefs, missing details, or ambiguous choices. These possible open nodes are stored in an agenda to include all the possible attentional states that require further development in the collaboration. During the elaboration process, when the system encounters some problem to elaborate an action node, is unable to identify the value of a parameter, or fail to execute an action, the system will put the node into the agenda.

Chapter 6

Our Approach: Supporting Event Notification

6.1 The Basic Interaction Scheme: Publish/Subscribe

6.2 Existing Notification Approaches

6.2.1 Topic-Based Approach

6.2.2 Type-Based Approach

6.2.3 Content-Based Approach

6.3 Activity-Based Approach

6.3.1 Types of Events

6.3.1.1 External and internal events

To understand events in geo-collaboration, an important distinction has to be made between *external* events (in the environment) and internal events (in the activities).

External events are changes in the physical environment that have impact on the performance of collaborative activities. For instance, the occurrence of a traffic accident may block the traffic flow, which makes an actors activity of delivering equipment to a medical station impossible to finish on time. External events in the environment can be characterized in three major categories: changes of identity (e.g. the occurrence of a traffic accident), attribute changes of an object (e.g. the contamination level of the chemical plan is increased), or spatial changes (e.g. the impacted area is enlarged due to the wind condition).

Internal events are state changes on the basic elements (i.e. resources, goals, activities) of geo-collaboration. Resource events indicate changes on the state of any resources that are used in the activities. Resource events can impact response activities in different ways: (1) the creation of a new resource can trigger new activities (for example, a new victim may lead to the activity to perform decontamination and medical treatment); (2) the state change of a resource can impact the activity that requires the use of the resource (e.g. the mechanical breakdown of a vehicle makes the delivery activity unable to complete); (3) the state change of a resource can enable/disable the activity that depends on it (e.g. a victims location change leads to the satisfaction of a precondition that the victim must be located at the medical station, which further enables the activity to perform medical treatment on the victim). Condition events indicate changes on a certain relationship between multiple objects in a collaborative activity. For example, a condition may express a spatial relationship that the victim must be located at the decontamination station. The condition will change the state from open to holding when a driver has picked up the victim and transpored him/her to the station. Activity events are described as change of activity states, such as the initiation of a new activity, the completion or delay of an ongoing activity.

The distinction between external and internal events is very important to identify the set of events that should be captured and modeled in the awareness mechanism. Not all the external events in the environment are important for the actors to perform their activities. Rather, only a subset of the external events that can lead to changes within the activities (i.e. the internal events) is meaningful. The derivation of internal events from external events is an active cognitive process that requires interpreting the meaning of external events within the context of the collaborative activities.

6.3.1.2 Local and remote events

The distinction between external and internal events is based on the boundary between the surrounding environment and the overall collaborative activities as a whole. However, in order to understand the awareness information needed for each individual, another important distinction between local and remote events has to be made.

The local and remote events are defined with aspect to the local scope of work for each individual actor. Local events reflect the state changes of basic elements (i.e. resources, goals, activities) within an actors local scope of work. For instance, the report of a new victim that needs to be decontaminated, the exceeding capacity of a decontamination station are local events within the decontamination managers local scope of work. Remote events are the changes outside an actors local scope of work. For instance, a victim that doesnt need to be decontaminated is transported to a shelter is a remote event for the decontamination manager. The distinction of local and remote events is relative to each individual actor. A local event of one actor may be a remote event for another actor due to their different local scopes of work. In addition, a remote event can be internal, reflecting the state changes of another actors activities, or it can be external, reflecting changes in the environment.

The distinction between local and remote events is important to identify the relevance of events that should be notified to each actor. Local events reflect changes within an actors local scope of work and therefore are directly relevant to the actor. On the other hand, due to the existence of the web of dependencies among activities, some of the remote events can (or potentially can) lead to changes in the actors local scope of work, i.e. the impact of a remote event may be propagated to the local scope of the actor as derived local events, and therefore they also become relevant. For instance, the traffic jam on the road is a remote event for the decontamination manager. However, because the traffic jam happens on the way of a victim to be delivered to a decontamination station, it can potentially delay the decontamination operation on the victim, which becomes a relevant local event for the decontamination manager. As a result, the goal of an awareness system is to notify an actor not only all the relevant local events, but also the subset of remote events that can be propagated to the actors local scope through the web of dependencies.

6.3.2 Event Subscription

6.3.2.1 Specifying Local Scopes

6.3.2.2 Defining Event Patterns

6.3.3 Event Matching

6.4 A Simulation Experiment

We conduct a simulation experiment to compare the content-based event notification approach and the proposed activity-based approach. The basic hypothesis is that, in dynamic environment where the user's activity changes frequently, the proposed activity-based approach provides a more accurate way for the user to express their awareness interests than the content-based approach. We use the emergency response scenario as described in the Introduction Chapter to perform the experiment, from the perspective of the decontamination manager.

6.4.1 Variables of Interests

We are interested in the capability of these two notification approaches in handling different types of activity dynamics. Particularly, we are interested in three types of dynamics that may occur in the scenario:

1. **Parameter Change**: the value of a parameter is changed. In the scenario, it can be the case when the decontamination manager re-assign a new station where a victim will be decontaminated.

2. **Plan Development**: the plan of the actor is changed. In the scenario, it can be the case when the decontamination manager starts to concern about certain equipment delivery after the stock is running low.

3. **Local Scope Change**: the actor's local scope of work is changed. In the scenario, it can be the case when another manager joins the activity, and take a subset of tasks away from the modeling manager.

To perform the experiment, we define four scenes to reflect the three types of activity dynamics:

1. Initial scene (S_0): indicates the current state of the decontamination manager's activities
2. Parameter change scene (S_1): indicates one of the parameters of the decontamination manager (station) has been assigned with a new value, comparing with S_0
3. Plan development scene (S_2): indicates one of the conditions has been elaborated into a more concrete plan, comparing with S_1
4. local scope change (S_3): indicates the another actor joins the activity, and the local scope of the modeled user has been reduced to a smaller set, comparing with S_2

6.4.2 Simulating Event Generation

A set of events are randomly generated based on the whole group activities. Given the current PlanGraph model, the events can happen on any of the activity, resource, or condition node, and indicate any types of possible changes that are defined in Chapter 5.

6.4.3 Creating Subscriptions

Two sets of content-based subscriptions will be generated.

1. The first set of subscriptions are defined merely based on the current interest of the manager in the initial scene (S_0). This set only includes the events that are relevant to the manager's current activities. Therefore, it can be considered as the minimal subscription set of events, using the content-based notification approach (CON_{min}).
2. The second set of subscriptions are defined based on considering all the three possible activities dynamics defined in scenes (S_1, S_2, S_3). It includes all the possible events that might be relevant to the manager in all the scenes. Because we attempt to include the maximum set of events that could be relevant for the manager across all the scenes, we define this content-based subscription as (CON_{max}).

In addition, the activity-based subscription will also be generated, following the steps described in previous section. The subscription includes the specification of local scope of the decontamination manager, his/her intentions and detailed event patterns on each node in the local scope. We define this activity-based subscription as ACT.

6.4.4 Procedures

The experiment includes four runs.

1. The first run R_0 is the human judgment on the relevance of the simulated event set to form the baseline result. The human analysts run through the list of simulated event set to decide on whether each event is relevant to the modeled decontamination manager, under the four scenes respectively. To reduce the subjective errors of human judgment, two analysts will perform the judgment separately and then discuss on the differences to form the final sets of events.

2. In the second run R_1, the simulated event set will be fed into the matching program following content-based notification approach, using the set of subscriptions defined as CON_{min}.

3. In the third run R_2, the simulated event set will be fed into the matching program following content-based notification approach, using the set of subscriptions defined as CON_{max}.

4. In the fourth run R_3, the the simuated event set will be fed into the matching program following activity-based notification approach, using the set of subscriptions defined as ACT.

6.4.5 Measures

Two measures will be used to compare the results of different notification approaches:

1. **Miss rate** is defined as the ratio between number of events that are considered as relevant in R_0 but missing in the result of R_i ($i \in \{1, 2, 3\}$), and the total number of events considered as relevant in R_0.

2. **False alarm rate** is defined as the ratio between number of events that are considered as relevant in R_i ($i \in \{1, 2, 3\}$), but missing in the result of R_0, and the total number of events considered as relevant in R_i ($i \in \{1, 2, 3\}$).

6.4.6 Results

Some results we expect from the experiment can be:

1. Comparing with the context-based notification using subscriptions defined as CON_{min}, the activity-based notification has lower miss rate in the scenes S_1, S_2, S_3. The difference between miss rates of the two methods increases as the scene evolves.

2. Comparing with the context-based notification using subscriptions defined as CON_{max}, the activity-based notification has lower false alarm rate in the scenes S_1, S_2, S_3. The difference between false alarm rates of the two methods decreases as the scene evolves.

6.4.7 Discussion

Chapter 7

Our Approach: Supporting Event Interpretation

7.1 The Cognitive Basic

The cognitive principles that guide the design of visual displays for event interpretation.

1. The schemata theory (Bartlett et al.): when human respond to incoming stimuli, they make use of their existing knowledge as a frame of reference to make sense of the incoming stimuli and produce behavior. The active organization of existing knowledge is defined as 'schema'. Schemata can be merely mental templates in human mind, or retrieved dynamically from the external visual artifacts.

2. The relevance principle (Kosslyn 2006): Visual displays should present no more or no less information than is needed by the user. Presenting all of the relevant information in the display relieves the user of the need to maintain a detailed representation of this information in working memory, where presenting too much information in the display leads to visual clutter or distraction by irrelevant information (Rosenholtz et al. 2007; Wickens & Carswell, 1995).

3. The task specificity principle (Hegarty et al. 2009). Visual displays are used for many different tasks, and there is no such thing as a 'best' visual display, independent of the task to be carried out with this display.

What these cognitive principles can inform us are:

1. When the user interprets an awareness event, he/she needs to activate and connect the event to his/her contextual knowledge of the situation. The visual display can enhance the interpretation by externalizing part of the knowledge in the visual representation and freeing up working memory resources.

2. However, the effectiveness of the visual display depends on how much relevant information is represented. We want to provide the users with just enough contextual information without causing visual clutter or distraction.

3. The relevance of contextual information to be displayed is dependent on the current task to be carried out, or the decision that need to be made by the user.

7.2 Activity-Aware Event Interpretation

The problem we want to address here is that: how the computational model of activities and local scopes can inform the system to decide on what contextual information should be displayed when the users interpret awareness events.

The decisions can be made based on two factors: the incoming event and the activities/local scope the user is working on. By maintaining the PlanGraph model, the system can infer how the event will impact the user's activities, which can be used to infer what decisions the user needs to make next, or what tasks the user will focus on, which can be used to decide on what should be displayed on the user interface.

Generate the set of rules based on SharedPlan Theory.

Some examples:

1. If the event initiates a new goal of the user, by elaborating on the plan, the system can infer that the user will work on identifying the set of parameters first.

2. If the event indicates the failure of a parameter, and there is no plan to fix the failure, the system can infer that the user will re-assign the values of this parameter.

3. if the event indicates the failure of a parameter, and there is plan to fix the failure, the system can infer that the user will work on the plan to fix it.

7.3 Experimental Study

7.3.1 Hypotheses

The general assumption is that maps with contextual information that is adapted to the user's current activities are more effective to support awareness event interpretation, than maps with fixed contextual information.

7.3.2 Experimental Design

7.3.2.1 Participants

Twelve undergraduate or graduate students will be recruited as participants in this study. All participants need to use computers on a daily basis. Prior experience with emergency response planning or operations is welcomed, but not required.

7.3.2.2 Tasks

The participants are asked to perform simulation tasks in an emergency response scenario, from the perspective of the decontamination manager. The tasks that the participants need to perform are driven by the events they receive.

Some example events and triggered tasks include:

1. E1: A new victim that needs to be decontaminated is reported by the victim manager. The task that the participant needs to do is to assign a decon station for the victim.

2. E2: The impacted area is enlarged and a decon station becomes inside of the impacted area. The task that the participants need to do is to re-assign the victims that are assigned to the station to other stations.

3. E3: A type of resource in a particular decon station is running low in stock. The participant need to request for a resource delivery from one of the suppliers.

4. E4: The delay on a victim's arrival at assigned station. The participant needs to check whether it will conflict with the victim's deadline on decontamination. If so, he/she needs to plan for sending the victim to other station.

7.3.2.3 Design settings

Participants are randomly assigned to one of the two design settings in one experiment session to perform their tasks. The three design settings are described as follow:

- The first design setting (D_1) shows a map showing locations of all the stations, victims, and resource suppliers as separate layers. The user can turn on/off each layer, or filter the objects based on spatial distances.

- In the second design setting (D_2), the features displayed in the map are adapted based on the event and the tasks that the user needs to perform.

7.3.2.4 Procedure

In the beginning of each experiment session, the participant is randomly assigned to one of the three design settings and given a couple of minutes to learn the functions of the assigned design setting. In addition to the training of the simulation system, the participant is also provided with the general picture of the collaborative activity. Information about other agents, related resources in the activity, and the current goals and tasks of the group are presented to the participant, so he/she has a clear understanding of the task in the context of the whole activity and provides the basis for interpreting the awareness events. After the training session, the participant is asked to perform the simulation task. During the performance of the task, the participant is notified with a list of awareness events, and asked to finish the assessment form to answer relevant questions after each event notification.

Each experiment session is composed as a series of interaction episodes between the participant and the system. Each interaction episode defines how an agent (role-played by the participant) responds to a particular event in the collaborative scenario. Each interaction episode starts with the provision of an awareness event in the interface, and then the participant is perform corresponding tasks based on their understanding of the event.

7.3.2.5 Measurement

To measure the overall performance during the experiment session, we record the completion time of each experiment session and the participant's responses to each awareness event. In the design phase of the awareness events, each event is associated with a set of optimal responses. The participant's responses are then compared with the optimal set to indicate the quality level of task completion.

7.3.3 Results

In general, we expect that

1. the participants in design setting D_2 will generate more valid responses than the participants in design setting D_1.

2. the participants in design setting D_2 should spend less time to finish the tasks than the participants in design settings D_1 in average.

7.3.4 Discussion

Chapter 8

Our Approach: Supporting Event Propagation

Activity representation as an approach to support awareness transactions between actors:

scenario-based design

a. the ability to find who should be awareness of certain events by making each other's local scopes visible

b. the ability to interpret events within the activity context

Chapter 9

Our Approach: Architecture and Implementation

Chapter 10

Case Studies

Chapter 11

Conclusion and Future Work

Bibliography

[1] Marilyn Jager Adams, Yvette J. Tenney, and Richard W. Pew. Situation Awareness and the Cognitive Management of Complex Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):85–104, March 1995.

[2] P Antunes, C Sapateiro, J Pino, V Herskovic, and S Ochoa. Awareness Checklist: Reviewing the Quality of Awareness Support in Collaborative Applications. *Collaboration and Technology*, pages 202–217, 2010.

[3] H Artman and C Garbis. Situation awareness as distributed cognition. In *Proceedings of ECCE*, volume 98, 1998.

[4] Gregory Bedny and David Meister. Theory of Activity and Situation Awareness. *International Journal of Cognitive Ergonomics*, 3(1):63–72, January 1999.

[5] F Cabitza, M P Locatelli, and C Simone. Promoting Process-Based Collaboration Awareness to Integrate Care Teams. In *Business Process Management Workshops*, pages 385–396. Springer, 2009.

[6] J M Carroll, D C Neale, P L Isenhour, M B Rosson, and D S McCrickard. Notification and awareness: synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies*, 58(5):605–632, 2003.

[7] John M Carroll, Mary Beth Rosson, Gregorio Convertino, and Craig H Ganoe. Awareness and teamwork in computer-supported collaborations. *Interacting with Computers*, 18(1):21–46, 2006.

[8] John M. Carroll, Mary Beth Rosson, Umer Farooq, and Lu Xiao. Beyond being aware. *Information and Organization*, 19(3):162–185, July 2009.

[9] Marcelo Cataldo, Patrick A Wagstrom, James D Herbsleb, and Kathleen M Carley. Identification of coordination requirements: implications for the Design of collaboration and awareness tools. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 353–362, New York, NY, USA, 2006. ACM.

[10] K Crowston. A taxonomy of organizational dependencies and coordination mechanisms. *MIT Center for Coordination Science Working Paper*, 1994.

[11] P Dourish and V Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114. ACM New York, NY, USA, 1992.

[12] Paul Dourish and Sara Bly. Portholes: supporting awareness in a distributed work group. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*, pages 541–547, New York, New York, USA, June 1992. ACM Press.

[13] M R Endsley. Measurement of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):65–84, 1995.

[14] M R Endsley, W B Jones, K Schneider, M McNeese, and M Endsley. A model of inter- and intrateam situational awareness: implications for design, training, and measurement. *New Trends in Cooperative Activities Understanding System Dynamics in Complex Environments*, pages 46–67, 2001.

[15] Mica R. Endsley. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, March 1995.

[16] A Espinosa, F J Lerch, R E Kraut, E Salas, and S M Fiore. Explicit vs. implicit coordination mechanisms and task dependencies: One size does not fit all. *Team cognition: Understanding the factors that drive process and performance*, pages 107–129, 2004.

[17] Ludwin Fuchs. AREA: A Cross-Application Notification Service for Groupware. *Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work*, pages 61–80, 1999.

[18] S R Fussell, R E Kraut, F J Lerch, W L Scherlis, M M McNally, and J J Cadiz. Coordination, overload and team performance: effects of team communication strategies. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 275–284. ACM, 1998.

[19] B J Grosz and S Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[20] Carl Gutwin and Saul Greenberg. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work (CSCW)*, 11(3):411–446, 2002.

[21] Christian Heath, Marcus Sanchez Svensson, Jon Hindmarsh, Paul Luff, and Dirk vom Lehn. Configuring Awareness. *Computer Supported Cooperative Work (CSCW)*, 11(3):317–347, 2002.

[22] Scott E Hudson and Ian Smith. Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems. In *CSCW '96: Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 248–257, New York, NY, USA, 1996. ACM.

[23] E Hutchins and G Lintern. *Cognition in the Wild*, volume 262082314. MIT press Cambridge, MA, 1995.

[24] A Kittur, B Lee, and R E Kraut. Coordination in collective intelligence: the role of team structure and task interdependence. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 1495–1504. ACM, 2009.

[25] T W Malone and K Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR)*, 26(1):87–119, 1994.

[26] P Markopoulos. A design framework for awareness systems. *Awareness Systems*, pages 49–72, 2009.

[27] Susan Mohammed and Brad C. Dumville. Team mental models in a team knowledge framework: expanding theory and measurement across disciplinary boundaries. *Journal of Organizational Behavior*, 22(2):89–106, March 2001.

[28] B A Nardi. *Context and consciousness: activity theory and human-computer interaction.* The MIT Press, 1996.

[29] U Neisser. *Cognition and reality: Principles and implications of cognitive psychology.* WH Freeman/Times Books/Henry Holt & Co, 1976.

[30] A A Nofi. Defining and measuring shared situational awareness. Technical report, DTIC Document, 2000.

[31] P P Perla, M Markowitz, A A Nofi, C Weuve, and J Loughran. Gaming and shared situation awareness. Technical report, DTIC Document, 2000.

[32] Wolfgang Prinz. NESSIE: An Awareness Environment for Cooperative Settings. *Proceedings of the sixth conference on European Conference on Computer Supported Cooperative Work*, pages 391–410, 1999.

[33] A B Raposo, L P Magalhães, I L M Ricarte, and H Fuks. Coordination of collaborative activities: A framework for the definition of tasks interdependencies. In *Proceedings of Seventh International Workshop on Groupware*, pages 170–179. IEEE, 2002.

[34] Markus Rittenbruch and Gregor McEwan. An Historical Reflection of Awareness in Collaboration. *Awareness Systems*, pages 3–48, 2009.

[35] E Salas, C Prince, D P Baker, and L Shrestha. Situation awareness in team performance: Implications for measurement and training. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):123–136, 1995.

[36] P M Salmon, N A Stanton, G H Walker, D Jenkins, C Baber, and R McMaster. Representing situation awareness in collaborative systems: a case study in the energy distribution domain. *Ergonomics*, 51(3):367–84, March 2008.

[37] Paul M. Salmon, Neville A. Stanton, Guy H. Walker, Chris Baber, Daniel P. Jenkins, Richard McMaster, and Mark S. Young. What really is going on? Review of situation awareness models for individuals and teams. *Theoretical Issues in Ergonomics Science*, 9(4):297–323, July 2008.

[38] Paul M. Salmon, Neville A. Stanton, Guy H. Walker, Daniel P. Jenkins, and Laura Rafferty. Is it really better to share? Distributed situation awareness and its implications for collaborative system design. *Theoretical Issues in Ergonomics Science*, 11(1-2):58–83, January 2010.

[39] Kjeld Schmidt. The Problem with ‘Awareness’: Introductory Remarks on ‘Awareness in CSCW’. *Computer Supported Cooperative Work (CSCW)*, 11(3):285–298, 2002.

[40] Kjeld Schmidt and Carla Simonee. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work (CSCW)*, 5(2):155–200, 1996.

[41] S Y Shen and M J Shaw. Managing coordination in emergency response systems with information technologies. *Proceedings of the Tenth Americas Conferences on Information Systems*, pages 2110–2120, 2004.

[42] Y Shu and K Furuta. An inference method of team situation awareness based on mutual awareness. *Cognition, Technology & Work*, 7(4):272–287, 2005.

[43] J S Sichman and R Conte. Multi-agent dependence by dependence graphs. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 483–490. ACM, 2002.

[44] Riyaz Sikora and Michael J Shaw. A Multi-Agent Framework for the Coordination and Integration of Information Systems. *Management Science*, 44(11):pp. S65–S78, 1998.

[45] C Simone, M Divitini, and K Schmidt. A notation for malleable and interoperable coordination mechanisms for CSCW systems. In *Proceedings of conference on Organizational computing systems*, pages 44–54. ACM, 1995.

[46] Kip Smith and P. A. Hancock. Situation Awareness Is Adaptive, Externally Directed Consciousness. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):137–148, March 1995.

[47] N A Stanton, R Stewart, D Harris, R J Houghton, C Baber, R McMaster, P Salmon, G Hoyle, G Walker, M S Young, M Linsell, R Dymott, and D Green. Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics*, 49(12-13):1288–311, January 2006.

[48] Neville A. Stanton, Paul M. Salmon, Guy H. Walker, and Daniel Jenkins. Genotype and phenotype schemata and their role in distributed situation awareness in collaborative systems. *Theoretical Issues in Ergonomics Science*, 10(1):43–68, January 2009.

[49] J Uhlarik and D A Comerford. A review of situation awareness literature relevant to pilot surveillance functions. Technical report, DTIC Document, 2002.

[50] C. D. Wickens. Situation Awareness: Review of Mica Endsley's 1995 Articles on Situation Awareness Theory and Measurement. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3):397–403, June 2008.

[51] E S K Yu and J Mylopoulos. An actor dependency model of organizational work: with application to business process reengineering. In *Proceedings of the conference on Organizational computing systems*, pages 258–268. ACM, 1993.