Design Approach for Collaborative Decision Making Games

Aida Azadegan
Bournemouth University
School of Design, Engineering and Computing
aazadegan@bournemouth.ac.uk

Casper Harteveld
Northeastern University
College of Arts, Media and Design
c.harteveld@neu.edu

Steven Sutherland
Northeastern University
College of Arts, Media and Design
ssuth5zd@siu.edu

ABSTRACT
Game design is a challenging process. Because there are multiple players and a need to design for the intricacies of these players working together to achieve goals, the complexity of this game design increases when it comes to the design of collaborative games. In this paper, we introduce a pattern-based approach for designing Collaborative Decision Making Games (CDMGs). This design approach is based on the principles of Collaboration Engineering and will provide guidance as well as proven building blocks for the design of collaborative games. To demonstrate our proposed approach, we applied the design approach to the design process of a carefully selected collaborative game that is currently being developed. This paper concludes with lessons learned, further suggestions, and potential implications for future work.

Categories and Subject Descriptors
H.5.3 [**Information Systems**]: Information Interfaces and Presentation (e.g., HCI) – *collaborative computing, computer-supported cooperative work;* K.8.0 [**Computing Milieux**]: Personal Computing – *games.*

General Terms
Design, Human Factors

Keywords
Collaboration Patterns, Collaborative Games, Game Design, Collaboration Engineering (CE), Design Science Research

1. INTRODUCTION
Game design is a challenging process. Fields of game research and development involve people from diverse creative and scholarly backgrounds. Although each party involved with the design process has its own expectations and needs, they need to be able to contribute to an overall design goal that satisfies the desired results. A systematic coherence among all game design activities will appear when conflicting constraints are resolved and each of the game's parts can meaningfully relate to each other as a whole [1]. Decomposing, clarifying, and shaping this coherence requires moving back and forth between all levels of abstraction [2].

Various practitioners and researchers have delivered methods and tools for game design to cope with its challenge and to help create a shared understanding amongst the team as well as increase the efficiency of the design process. Hunicke et al. [2] describe a formal approach to game design in general. Other general approaches concern game design patterns [3], formal abstract design tools [4], and machinations [5]. Dedicated approaches exist too such as on serious games [1], social values [6], and policy games [7]. In this paper, we present a dedicated design approach for creating collaborative games. To our knowledge a design approach specific to collaboration in games does not exist. Investigation of such an approach would be valuable because of the complexities inherent in designing for collaboration and multiplayer games in general [8], [9], which are in essence a result of a need to consider how various people play well together. With two or more decision makers, the emergent gameplay dynamics are more difficult to anticipate and, therefore, design for. Designing the interdependencies between players is more important in collaborative games than in others because in collaborative games players have shared goals and the complete game revolves around players working together and not individually [10]. An additional complication is the group dynamics between players. With this in mind, designing collaborative or multiplayer games can be seen as designing a complex system. Games are generally conceived of as systems [11], [12] and complex systems are characterized by having multiple interdependent variables [13]. Possibly because of its complexity, there is a dearth of purposefully designed games that require collaboration [14]. A systematic design approach can assist in addressing this gap.

Although no dedicated design approach for collaborative games exist, collaboration (and cooperation) has been studied in the context of games. So far, existing work includes research which has focused on developing guidelines [9], [15], [16], drawing lessons from board games [17], examining existing collaborative games [18], [19], and defining performance metrics to evaluate collaborative play [20]. In our previous work [10], we used patterns of collaboration described in Collaboration Engineering (CE) to analyze collaborative games. CE is a discipline outside of gaming that is focused on creating collaborative opportunities to reach a number of decision-making goals, where collaborative efforts are necessary to achieve those goals. Our reverse-engineering approach suggested that CE can be used to identify collaboration mechanics and help us to regulate team collaborations by either facilitating or hindering collaborative work practices to create a more challenging environment. We analyzed different collaborative games and realized that only games in which players experience collaborative decision-making and consensus building at the conscious cognitive level can be addressed and analyzed through the use of CE (see [10] for a description of the analysis and the games).

Building forth on our prior work for applying CE to analyze games, we posit that collaborative games are similar to

Proceedings of the 10th International Conference on the Foundations of Digital Games (FDG 2015), June 22-25, 2015, Pacific Grove, CA, USA.
ISBN 978-0-9913982-4-9.

collaborative work practices, able to be engineered and described using patterns of collaboration described in CE. Our research objective for this paper was to build a systematic design approach for designing collaborative games using CE. To guide us in the development, we relied on the Information Systems Design Science Research (DSR) methodology [21].

The next section describes the background for the creation of the design approach. Section 3 describes the design approach and Section 4 contains a description of how this design approach is applied to the design of *VistaLights*, a collaborative game where players have to work together to deal with disruptions in a port. This game is currently still in development and in this paper we illustrate how the design approach can help designers in exploring design alternatives. We conclude the paper in Section 5 with a discussion on this research and future work.

2. BACKGROUND

Cooperation and collaboration are two terms that are used interchangeably. A technical distinction exists when it comes to the exact definition of these two terms. According to Software Engineering and Computer Science, collaboration is defined as: "A process in which two or more agents *work together to achieve* shared *goals"* [3]. Within a cooperative mode team players have interests that are "neither completely opposed nor completely coincident." Therefore, in cooperative games, players may have different goals, whereas in collaborative games each player has the exact same goal [22]. We use the term collaboration the way it has been addressed in Software Engineering and Computer Science. Based on this, we define a collaborative game as a game in which two or more players, either co-located or distributed, work together to achieve shared goals [10]. We provide this distinction between collaboration and cooperation because the presented collaborative approach may require modifications if it is applied in the context of a cooperative game. Nevertheless, we strongly believe it has merit to both types of multiplayer games.

In the remainder of this background section we elaborate on the need and role of game design approaches, what Collaboration Engineering (CE) is, what kinds of collaborative games exist, and how design science research has informed the development of the design approach.

2.1 Game Design

Our aim is to develop a formal design approach for collaborative games. We argue that such an approach will help to systematically consider design choices on collaboration and to rely on an existing and growing body of knowledge on how to collaborate. It will make designers more mindful about their design, free up their mind for focusing on other creative aspects of the design (i.e., the distributed cognition argument for using tools in the design process), and prevent them from reinventing the wheel. These advantages are not specific to our approach; in fact, this pertains to any game design tool or method. However, our approach is unique in its focus on collaboration.

Game design is generally characterized as an iterative process where playtesting plays a crucial role in the development process [11]. Player-game interaction cannot be accurately predicted and this necessitates iteration and playtesting. Game design methods and tools can generally be distinguished in providing guidance on this process (e.g., [23]) or providing building blocks that have proven merit before (e.g., [3]). Our design approach attempts to accomplish both by integrating the use of patterns into a design process.

2.2 Collaboration Engineering

In our design approach, we prescribe the consideration of patterns of collaboration derived from Collaboration Engineering (CE). CE is an approach to create high value recurring work practices that can be executed by practitioners themselves without ongoing support from professionals. The CE approach prescribes that a collaboration engineer designs an efficient, acceptable, reusable, and predictable collaboration process [24]. According to CE, a successful collaboration should be supported by a procedural step-by-step process, which is explained according to the patterns of collaboration, and ultimately by a facilitator who intervenes and takes the role of the leader to direct the team towards a goal. In CE, the collaboration best practices are carefully selected and combined into a work process helping the participants working in teams achieve their common goal. The collaboration best practices are explained through the six main patterns of collaboration [25]:

1. *Generate*: Move from having fewer to having more concepts in the pool of concepts shared by the group

2. *Reduce*: Move from having many concepts to a focus on fewer concepts that the group deems worthy of further attention.

3. *Clarify*: Move from having less to having more shared understanding of concepts and of the words and phrases used to express them.

4. *Organize*: Move from less to more understanding of the relationships among concepts the group is considering.

5. *Evaluate*: Move from less to more understanding of the relative value of the concepts under consideration

6. *Build-consensus*: Move from having fewer to having more group members who are willing to commit to a proposal.

In CE, design patterns are used to support design processes [26]. The six patterns of collaboration help designers to have a granular perspective of team collaborations and therefore provide opportunity for the designers to create building blocks to formulate "atomic" elements of collaborations. In our design approach, these six patterns are used in considering how to design game activities. Moreover, we use understandings of Group Support Systems (GSS) design and application into team collaboration as they are described in CE. GSS are instruments used in performing an operation for moving a group toward its goals. Collaboration tools offer schemes for making sense of the range of capabilities in the collaboration space. These tools are set to make collaboration take place considering topics such as anonymity, group size, task type, and task-technology fit, etc. Examples are Group Explorer, MeetingSephere, or ThinkTank [27-29].

The CE approach was originally proposed by Briggs et al. [30], and has since been applied in a variety of case studies, e.g. [31-34]. Over the past years, efforts have been made to further define the field and conceptualized on its key design concept. In our work, we explore its usefulness to the application of games.

2.3 Collaborative Games

The creation of the design approach is based on our initial analysis of collaborative games using CE as tool. According to our previous study [10] we categorized collaborative games in three main categories:

1. *Instinctual collaboration*: Games in which collaboration happens based on players' instinct. In such games achieving agreement and consensus building among team members

happens intuitively and needs to occur swiftly. No time exists for discussion at the conscious cognitive level to collaboratively make decisions (e.g., *LittleBigPlanet*). With the conscious cognitive level we refer to a deliberate exchange of thoughts between team members about dealing with the challenges imposed by the game. Many games fit in this group.

2. *Supportive collaboration*: Games where team members are given opportunity to discuss, plan and make decisions about strategies to at the conscious cognitive level but it all happens before playing (e.g., *FIFA* soccer game series) or at predetermined intermittent points (e.g., safe rooms in *Left4Dead*).
3. *Integrative collaboration*: Games that provide opportunity for the players to experience collaborative decision making and consensus building at the conscious cognitive level. The game mechanics in such games challenge collaboration throughout the entire game (e.g., *TeamUP* - see section 3.2 for further detail*)*. Few digital games fit this group. The ones that do are not the well-known entertainment games but lesser known educational games, Alternate Reality Games (ARGs) or Serious Games.

CE takes into consideration collaboration dynamics among team members that take place at the cognitive level (e.g., brainstorming, consensus building). Therefore, applying CE to games makes most sense when integrative collaboration occurs. Therefore, our approach is most applicable to games that involve integrative collaboration. Because of our focus of collaboration at the cognitive level, we will continue to refer to the type of games we focus on as Collaborative Decision Making Games (CDMGs).

2.4 Design Science Research

In order to bring rigor to the development of our design approach, we decided to adopt Information Systems Design Science Research (DSR) methodology [21]. Design science is a methodology that is pragmatic in nature with the aim of developing innovative products (or processes) that will help understand and improve human performance. Many different outputs can be reached through the application of DSR. Design science can be used to research not just instantiations (prototypes or systems) but also constructs (symbols and vocabulary), models (abstractions and representations) and methods (algorithms and practices) [35]. We classify the design approach for CDMGs among the last, and argue that it can be designed as methods or practices.

The DSR method defines iterations within a structure that consists of five different research steps: (1) Awareness of the problem in the field of the research; (2) suggestions for solutions to the research problem; and (3) artifact development; (4) artifact evaluation; and (5) conclusion and communication of research outcomes. For applying the DSR design cycle steps (see Figure 1), we considered each step. For our problem of creating a design approach for collaborative games (1) we are proposing the use of CE principles and design patterns (2) and describe the resulting design approach in the next section (3). Our evaluation (4) of the design approach consists of applying the design approach to the development of a game called *VistaLights*. We illustrate the usefulness of applying the approach by showing the kinds of considerations designers can make with the help of this design approach. The "knowledge flow" from this evaluation exercise will be discussed at the end of this paper.

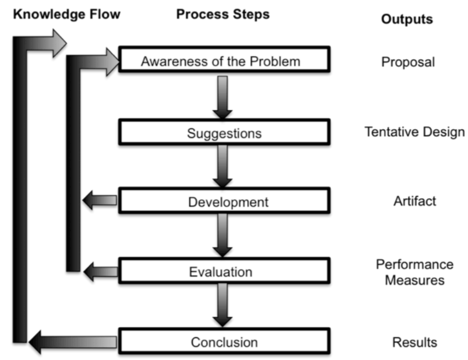

Figure 1. Design cycle steps [36]

3. Design Approach for CDMGs

The design approach we developed consists of four iterative steps: (1) identification of game activities; (2) creation of game elements; (3) allocation of game mechanics; and (4) consideration of collaboration patterns. After each cycle, validation will occur for determining the suitability of the implementation for the envisioned activity and its integration with all other envisioned activities, if any. In this validation, the dynamics and aesthetics can be considered [23]. Figure 2 visualizes our approach. The four steps are discussed in more detail below.

3.1 Identification of Game Activities

The first step that designers will need to do is to identify the possible game *activities*. The term activity with regards to game design is a nebulous concept. In our approach, an activity is defined as "a demarcated segment in a game where players face a particular challenge."

By demarcated, we mean that the activity is independent from other activities. Although previous activities may influence the state that players are in and the affordances they have (e.g., amount of ammo, types of weapons), the activity itself should be completed (or abandoned) before anything else can be done. For completing the activity, players will need to overcome the challenge offered in that activity. Quests are good examples of activities. Once a quest is accepted, players will need to complete or abandon it to continue. Levels are another good example. Each level has a goal, whether it is getting to the end of it or solving a puzzle, and players play (usually) only one level at a time. Note that it is perfectly possible that a game has just one activity. Short games or game-like exercises, such as asking a group to create a square with a rope without looking, involve one collaborative challenge. It is also perfectly possible that activities are not as neatly distinguished as in levels or quests, which are well-recognized demarcations in games, but that a change of activity is based on a significant change in what the players are doing. Our example, *VistaLights*, illustrates this. In that game players go from information gathering to prioritization and then to consensus building. Each activity has distinct game elements and a particular challenge associated with it. To help in identifying activities, we recommend the following three sequential tasks:

1. Pre-decomposition task: Designers need to identify the core objectives for the overall game and then elicit what deliverables will achieve those core objectives. To get a better understanding about the deliverables, designers need to analyze under what conditions the overall objectives will be achieved. Such conditions will be clarified through the iterative design process. In this task, designers will also need to identify the players of the

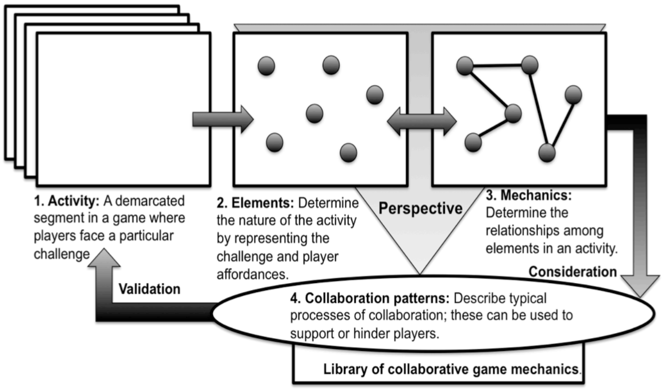

Figure 2. An overview of the design approach for CDMGs

Figure 3. The Maze in *TeamUP*.

game and their role in it. In other words, the pre-decomposition task is coming up with the higher-level concept of the game.

2. Decomposition task: Once the deliverables are identified, designers can break down the game into a number of activities that would satisfy the achievement of those deliverables. This task is analogous to level design because this is about identifying what will happen and where it will happen in the game.

3. Post-decomposition task: Once the activities are clear, designers should review them and make sure they are integrated into the game in an appropriate manner. In this task, play testing is essential because this will reveal if the design accomplishes what it is aiming for.

It is important to note that after the core game has been established, designers can go back to the decomposition task and identify more activities and then integrate them in the post-decomposition task. Designers can add more levels or quests along the lines of the core design principles that have been established for the game.

3.2 Creation of Game Elements

After activities have been identified, designers can decide what game elements are needed in each activity. Game elements determine the nature of the activity by representing the challenge and player affordances. Representation involves the objects and environment in which the challenge takes place. It can be abstract, iconic, or realistic. For example, throwing dice in a game is an abstract representation of chance. Affordances are about what players can do. With the game element inventory players can place and take out items.

Of course, the inventory is an example of a game element that is not necessarily unique to a specific activity, and many others can be thought of. There are, however, elements that are unique to an activity. In our prior work [10] we analyzed the game *TeamUp* [37], a multiplayer serious game about teamwork. In this game, a team of four players control each an avatar and has to work together to overcome a number of challenges within the game. Each challenge is designed to emphasize a specific element of effective teamwork. In one activity, which is focused on leadership, three players are put in random places in a maze. The fourth player will have an overview of the entire maze and has to guide the other three players to the exit. In this example, there are game elements where it is decided who goes into the maze and who becomes the leader (see Figure 3). Then there is the maze itself. With regards to performance, the game element of time counts how long players take to get out of the maze.

3.3 Allocation of Game Mechanics

The creation of game elements and allocation of game mechanics are tightly interrelated (hence the double arrow in Figure 2). This is because mechanics determine the relationships among elements in an activity. They provide meaning to the elements. Without mechanics, game elements are fairly static and players do not know what they can do with them. In the design process, designers may, on the one hand, connect existing game elements through a mechanic. For example, the pressure plates in *TeamUp* (see Figure 3) are part of a prior activity. Standing on top of them will open a door. However, in this activity, standing on top of the pressure plates will shoot three players down into three different locations in the maze and shoot one player upward to get an overview of the maze. Therefore, the allocated mechanic of assigning roles provides new meaning to the existing pressure plates. On the other hand, designers may realize while allocating a mechanic that a specific game element is required and in that case the mechanic drives the creation of game elements. The same example can be used here because the pressure plates had to be redesigned as elements to shoot players upwards and downwards. The original pressure plates only allowed to be stepped on. Identifying the need for a mechanic to assign roles led to redesigning existing game elements.

Similar to elements, mechanics are not always unique to a particular activity. Throughout *TeamUp*, players cannot see each other's screens and are able to move freely through the 3D environment with their avatars. Mechanics can also be re-used. The mechanic of assigning roles can be used for other activities as well.

3.4 Consideration of Collaboration Patterns

In the fourth step collaboration will be considered through the perspective of the collaboration patterns. Collaboration patterns describe typical processes of collaboration and they can be used to support or hinder players in achieving their goals. CE prescribes six main collaboration patterns (i.e., generate, reduce…build-consensus). Each pattern is considered as a functional independent module, or a building block to serve a single purpose in a collaboration process. Designers need to consider carefully what patterns are required in an activity. Then those considered patterns can serve as a perspective through which game elements and mechanics can be designed.

Although multiple patterns can be relevant, usually there is one *dominant pattern* and this pattern should be leading in the design of the activity. For example, in the *TeamUp* maze activity the pattern "clarify" is a dominant pattern because the main activity is achieving a shared understanding of how to get to the exit in the maze [10]. Realizing that the activity revolves around clarification, will lead to design suggestions and ensures that the activity is designed as intended.

In addition, the exact pattern implementations such as raising one player above all others describe a collaborative game mechanic that can be re-used. Such established specific implementations can be maintained in what we call the *library of collaborative game mechanics*, which are classified according to the patterns of collaboration. The focus in this library is on mechanics because game elements can be adjusted in accordance to the need and context of the specific game; mechanics are more generalizable and transferable. It should be noted that in CE specific implementations exist and that for each main pattern sub-patterns have been identified, which are also used to describe measurement constructs for the patterns. Measurement constructs can work as a reference point to regulate game mechanics, designing them to work for or against players. For example, Categorizing, Outlining, and Sequencing are sub-patterns of the Organize pattern. Other examples would be Gathering as a sub-pattern of the Generate pattern, which uses the number of contributions as a measurement construct (see [38] for more details).

We have argued that for creating collaborative games, game designers can learn from CE. In this fourth step, game designers can consult CE literature for how they have handled collaboration. Of course, in creating games there is a translation to be made, as the goal might be to hinder collaboration.

3.5 Validation of Game Activities

Success of collaborations has a process and a results component (deliverable) [39]. With this in mind, the validation of collaborations has three purposes:

- *Deliverables validation:* First, validation should take place to see if deliverables around which the activity is designed for are met.
- *Process internal validation:* Second, the activity process, which is designed for players to achieve the objective of the activity, needs to be validated.
- *Process external validation:* Third, the activity needs to be integrated with all other activities, if any.

The outcomes of all three validations need to be satisfactory. If unsatisfactory results are achieved, the activity needs to be redesigned and re-validated. When validating the activity, it is important to see whether the applied game mechanics direct the players to engage within the gameplay as desired by the game designer or if players interact with the elements as expected. Thus, activity validation needs to be a focus of user testing in collaborative games. Some suggested ways in CE to validate the activity are [40]:

- *Pilot testing:* Small-scale implementation of the activity that allows the designer to assess the activity effectiveness. This validation would allow the designer to understand whether the process can be achieved within the expected time line and if objectives are met.
- *Walk-through:* A way to assess the activity is by having the designer walk through the activity steps with the players. This validation will reveal unwanted pitfalls and difficulties that players face while playing the game.

4. Application of Design Approach for CDMGs

To illustrate our design approach, we will describe how it has been and continues being used in the creation of a collaborative game, titled *VistaLights*. In this application, we want to highlight how the approach can help designers to think about design alternatives. This application can be seen as "design thought experiment": thinking about what a design could be. Before illustrating our approach using this design though experiment, we must first describe the game. *VistaLights* is a game that was and continues being designed as a training tool meant to facilitate a collaborative process for decision making in response to disruptions in port operations (see Figure 4). It is in the early stages of production and thus has not been tested. Therefore, we limit ourselves to discussing the design considerations and will defer testing to future work considerations. Based on discussions with the Houston Port Bureau, the objectives for designing the game are two-folded. First, the idea is to explore the use of a game environment for testing different collaboration tools and procedures as well as other forms of decision aids (e.g., information visualization, prediction tools). Second, the goal is to create more resilient shipping supply chains by training decision makers to work collaboratively when disruptions occur. In this game, four players represent four different sections of port operations: the port authority, petrochemical, bulk, and break-bulk. Each player is tasked with representing the interests of their own stakeholders while collaborating with other players to ensure a resolution to return the port to an acceptable level of operation. *VistaLights* is a turn-based game. When a disruption occurs, players must gather information about the disruption, the recommended solutions, the requests and interests of their stakeholders, and about the current status of the port.

Each player represents four key stakeholders in the game. Stakeholders are Non-Player Characters (NPCs) or agents who have *needs*, provide *information*, and take *actions*. Stakeholders provide information to their respective players and make recommendations about how to proceed based on their needs. It is up to each player how much and which information they share with other players. When a stakeholder's needs are met, the stakeholder remains happy; however, if the player does not meet the stakeholder's needs, the stakeholder becomes unhappy. If the stakeholder becomes too unhappy with the player's performance, they will ask to no longer be represented by the player. For each stakeholder that terminates their relationship with the player, the player is imposed a large penalty to their score. Therefore, it is essential that players meet their own individual needs. The individuals' needs are challenged by the fact that the other players also have separate individual needs that may not be aligned and by the fact that there is a collaborative goal that all players must work to achieve. Whether happy or unhappy, stakeholders can take actions independently on how the disruption is dealt with collaboratively by all players, which increases the complexity of the situation and heightens the need for players to be well-informed about the intentions of their stakeholders.

The collaborative goal is to return the port to an approximation of the pre-disruption state or to the most optimal state post-disruption (i.e., if a bomb explodes and wipes out a significant portion of the port, then it will be impossible to return to the pre-disruption state) as quickly and efficiently as possible because the productivity of the port is aligned with the needs of every stakeholder. Players are responsible for selecting a response to the disruption (e.g., closing sections of the port and allowing a clean-

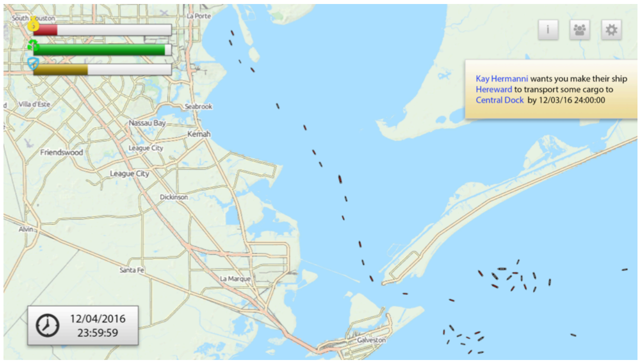

Figure 4. Screen shot of VistaLights. It depicts ships going into and out of the port. Players can provide different priorities after a disruption happens and receive messages from stakeholders.

up crew to work on an oil-spill until it is sufficiently clean). The responses will have varying sizes of effects on each player's section of the port and on each player's ability to meet the needs of their stakeholders. Therefore, players must work together to decide what course of action to take. Additionally, players are responsible for creating a schedule for ships to enter and exit the port, which is dependent on the resolution selected. Players may be largely impacted by a particular resolution, but may be able to offset the penalties by obtaining a favorable schedule.

4.1 Applying the Design Approach

Using Step 1 of our approach, we identified three activities where the design approach could be applied: Information Gathering, Prioritization, and Consensus. The game moves through these three activities in order; however, players may go between activities until consensus is reached for a particular schedule of ships and solution for the disruption. After consensus is reached, a simulation is run and the status of the port is updated based on the response the players agreed upon. If the port returns to an acceptable level of operation, the game session ends. If the port does not return to an acceptable level, the players go back through the activities. This process repeats until the players are able to return the port to an acceptable level where the overall collaborative goal of the game is met. Figure 4 summarizes this entire procedure.

Based on our prior work, we know that collaboration in each activity can either be designed to encourage or inhibit reaching both individual and group goals in this game. For the design of game elements (Step 2) and game mechanics (Step 3) that work for or against players, we can identified specific collaboration patterns (Step 4) for each activity. In the following sections, we illustrate what informed design considerations were possible using our design approach for each identified activity. We show what set of elements and mechanics were designed from the perspective of the collaboration patterns and how this impacted the overall design. We discuss the design approach that was taken for *VistaLights* and offer suggestions for alternatives that were considered or could be used to reach different goals in the game. In this paper, we limit ourselves to the design considerations to validate the usefulness of our design approach; in the future, testing of the specific design suggestions should provide further insights as to the actual impact of the game. In our discussions we further limit ourselves to discussing the dominant pattern for each activity, that is, the collaboration pattern that is most relevant for that particular activity.

4.2 Activity 1: Information Gathering

When a disruption occurs in the port, players begin the Information Gathering activity. Players must determine the nature of the disruption (e.g., oil spill, heavy fog, ship collision, etc.), determine what responses to the disruption are being suggested by the system, what responses and concerns the stakeholders have, current ships waiting to enter the port and their contents, current levels in storage facilities, and land shipping schedules. Each player receives different information from their stakeholders. The challenge in this activity is to gather as much relevant information as possible. The collaboration patterns that apply for this activity are Generate, Clarify, and Organize. The Clarify pattern is relevant because in this activity players will need to come to an understanding of what disruption happened. In addition, many factors play a role in this process and through the perspective of the Organize pattern design options can be considered for how this part of the process can be either supported or hindered.

The main outcome of this activity is for players to come up with solutions to deal with the disruption in addition to a new ship scheduling, hence the major dominant pattern in this activity is Generate. The new scheduling is necessary because under the new circumstances, priorities may need to be set differently. For example, if oil refineries do not receive new supplies within a certain amount of time, the refineries will need to be shut down, which takes an incredible amount of resources and is therefore best to be prevented. Also, certain ships carry perishables and such ships may need to be prioritized over those carrying non-perishables. However, shipping companies typically receive penalties for late deliveries and port disruptions offer no exemption, which means that ships carrying non-perishables will also be affected.

Individual vs. Collaborative Generate Process. A key decision in the generate process is whether players interact with each other and then how. The activity becomes a collective brainstorm if they are; otherwise, it will be an individual brainstorm. A design rationale for pursuing an individual brainstorm is to allow players to interact deeply with their stakeholders and take a strong position, which will more likely create conflicts among players. From the CE literature, it is known that a collective brainstorm is more efficient and effective. Using a group decision support system (groupware), for example, individuals share ideas in real-time as they are developed. They can brainstorm and build forth on each other's ideas. Game mechanics (and associated elements) should make one or the other possible. For a collective brainstorm, a variety of sub-alternatives can be imagined from simply allowing players to talk to each other (e.g., chat room), visualizing real-time what others are considering, to the integration of a groupware-like system where players share information with each other.

Elements for Generating Information. Other design decisions are how players gather information. An important design consideration other than collaboration is the allotted time to play the activity. Whereas in reality information gathering can take up a significant time, in the context of the game that would not be desirable. In conversations with subject-matter experts we were told that whatever disruption happens, representatives usually have about an hour to gather information. An hour seems relatively short, yet for playing a game that is a considerable amount of time. This may mean that the game should support this activity for reasons other than collaboration. One such possibility is including a simulation tool as element that shows the players potential outcomes for the port for different what-if scenarios.

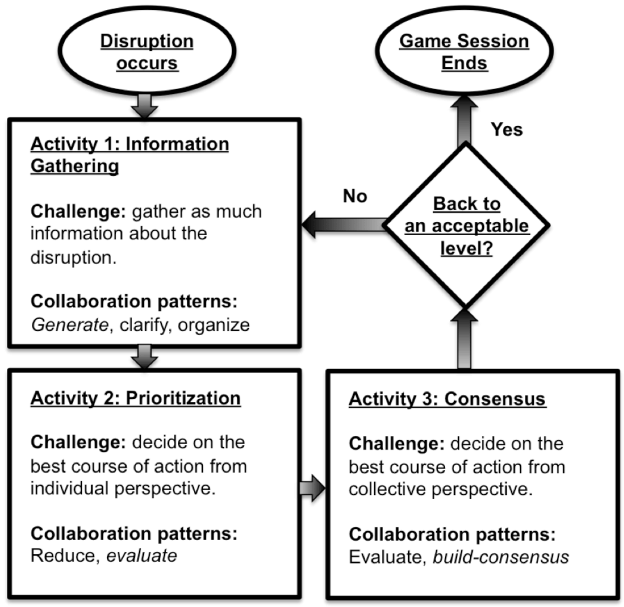

Figure 5. An overall view of the activities in *VistaLights*.

Such a tool facilitates the generate pattern by allowing players to select schedules for the ships and use that information to run miniature simulations that show them a potential outcome for the port. Alternatives can be imagined that do or do not consider the perspectives of the other players. If perspectives from other players are not considered, then such a tool may challenge collaboration further as players spent more time exploring their own perspectives and feel more invested in the solutions they came up with. For considering the perspectives of other players, a design alternative could involve approval ratings of each solution or indication of the impact of each solution for other stakeholders.

Mechanics for Sharing Information. In this activity, players are already challenged by the implementation of elements such as stakeholders in the game who need to be kept satisfied. Mechanics can be designed in the game to create situations where providing information has been forbidden by the stakeholders while on the other hand players need to share enough information with other players to achieve the overall goal of the game. Such a contrasting demand provided by this not sharing mechanic introduces an additional challenge in the game which is supported by hindering the generate collaboration pattern. In the alternative players decide themselves what or what not to share.

4.3 Activity 2: Prioritization

Once players have generated a number of possible actions, they begin the Prioritization activity by going through the Reduce and Evaluate collaboration pattern where Evaluate is dominant. In this activity, the challenge is to decide on the best course of action from the individual and overall perspectives. Here, the Reduce pattern is simply looking to reduce the total number of options to a small number that can be offered to the group and clarified. However, the nature of forecasting potential outcomes of future states, following a disruption, puts the emphasis on evaluating forecasts against the individual and collaborative goals.

Individual vs. Collaborative Evaluate Process. The Evaluate pattern is challenging because a particular resolution to the disruption and schedule for the ships may be evaluated favorably for meeting one's own individual needs for satisfying stakeholders. However, that same resolution and schedule may simultaneously conflict with returning the port to normal or with the needs of the other players. By creating conflicting evaluations for players, they are challenged to try multiple options for resolving conflicts. Individual players must evaluate a solution on how well it meets the stakeholders requirements, how well it works towards returning the port to an acceptable level of production (collaborative goal), and how well it meets the needs of the other players so that their recommended course of action is accepted by those players as well.

Elements for Evaluating Options. The game provides elements such as a mini simulation tool, stakeholders' approval rating tool, and a tool for measuring confidence in the recommended solutions. Players get a better idea of how their recommended actions may affect the state of the port by using the mini simulation tool, which would lead them to gain a better understanding of their ability to meet the collaborative goal. Although the simulation tool provides players with a forecast of a potential future outcome, it does not tell the player whether that outcome is desirable or if it is the best outcome, leaving players to determine this on their own. By providing players with a stakeholder approval ratings evaluation tool, players can learn through their actions how the stakeholders are truly measuring their performance; however, this takes time and may be difficult or costly to learn. We implemented a tool for players to submit their recommended solutions to the stakeholders to receive confidence ratings from the stakeholders regarding how well the recommended solution will meet the stakeholders' needs. Additional confidence rating measures were carefully selected to encourage the player to think about solutions as they relate to three key goals and include: How confident are you that the recommended solution will meet your stakeholder's needs?; How confident are you that the recommended solution will return the port to normal?; and, How confident are you that the recommended solution will meet the needs of the other players?. These confidence measures will be used in the next step for reaching consensus. Tying the confidence rating elements to specific goals can help facilitate Evaluation.

Mechanics for Evaluation. To challenge Evaluation, designers may implement vague measures or by using less indicative scales (e.g., a Likert-type scale of 1-3 for confidence or a simple "Yes" or "No" question). To facilitate more effective Evaluation, designers may set the mechanics to link voting to the variables that inform the underlying decision model and thus focus the evaluation on precisely what is important. The voting mechanic, using a combination of ranking potential solutions and providing confidence ratings on three levels (individual needs, others' needs, and collaborative goals), is the major mechanic used for evaluation. To facilitate the Evaluate Pattern, players may be required to request rationale for other players' recommended courses of action when there is a lot of variability (we use standard deviation as our measure of variability) for confidence ratings for how well players believe a recommended solution and schedule will meet their stakeholders' needs, the needs of other players, or whether the port will return to normal. Because information is necessary for evaluation, players will be required to provide rationale for their ratings, based on the information that they have access to and how much they can or are willing to share. If players learn to share necessary information, they can get feedback from their peers; however, if they choose to withhold necessary information, other players may be skeptical about the changes a player is making and work to prevent them from getting what they want. We have chosen to require that recommended solutions with high levels of variability (measured by standard deviations in confidence ratings) will need to reach certain thresholds of lower variability before players can move forward with implementing the solution. By requiring players to reach a particular threshold, players are encouraged to provide additional information for their ratings, giving other players necessary information on which to evaluate the solutions. As with the Generate pattern, secret information that is not permitted to be shared with other players may hinder the evaluate pattern.

The stakeholder confidence tool was introduced for getting feedback as an evaluative measure. We allow players to propose a solution and schedule to their stakeholders to determine how confident (0-100%) that the course of action will meet their needs. This can be made more challenging by limiting the number of solutions that can be submitted to the stakeholders for confidence ratings or by decreasing stakeholder approval ratings every time the player has to check with them thus introducing a cost (loss of satisfaction in one's performance).

The inhibition of collaboration occurs for the Evaluate pattern in this activity by ensuring players do not have access to the same information, which is the major resource in this game. By not having access to complete information, players may have difficulty evaluating any particular course of action without some suspicion that the actions are not meant to meet only one player's goals. This inhibition may actually work to increase collaboration through using information sharing as a technique to build trust. Players can earn trust from their collaborators by sharing information and may gain additional trust if the information seems to better serve the individual's goals than the collaborative goals. An important aspect of collaboration is ensuring that resources are equally being shared and invested by members of the collaborative team.

4.4 Activity 3: Consensus

In the final activity, players must reach a consensus on the resolution to the disruption and a schedule for the ships to enter and exit the port. Therefore, the challenge in this activity is to decide on the best course of action from both the individual and the collective perspective. The collaboration patterns present in this activity are Evaluate and Build-Consensus, with the latter being the main pattern. Evaluation is highly linked to building consensus and many of these intricacies were highlighted in the previous section. In this section we will focus on the Build-Consensus pattern.

Individual vs. Collaborative Consensus-Building Process. Because consensus, by definition, is a collaborative process, individual consensus is often not a consideration. However, in *VistaLights*, individual players must ensure that the recommended solution and schedule will meet the needs of their stakeholders, players must work towards a balance between stakeholder needs. As previously mentioned, this is done by permitting players to submit their recommendations to their stakeholders for confidence ratings from each stakeholder. Players will need to build consensus amongst their stakeholders before working with other players to reach consensus regarding which solution will be pursued in the port. The resolution for the disruption and ship schedule will not be permitted to be implemented until players reach consensus amongst each the players' confidence ratings for the each recommendation.

Elements for Building Consensus. In order to achieve collaboration, a ranking and confidence rating tool is provided for the players so that they can rank each solution and submit three confidence ratings for each resolution. For a resolution to be accepted the resolution must have the highest average rank and the variability of all confidence ratings must be lower than the chosen threshold to be considered an agreed solution. Players will be shown levels of consensus for each solution using a color-coded scheme taking into consideration different colors for either low, medium or high consensus levels (measured by standard deviation thresholds). Additional elements that designers may consider are using average ratings instead of variability, using a simple voting tool that picks the option with the most votes. Because an average does not reflect true consensus, players may initially be disappointed in the outcomes, requiring additional clarification for how other players are voting. By using a simple voting tool, players are not required to come to a true consensus or to evaluate the solutions on important outcomes or necessarily to provide rationale for their choices. Limiting the information that the game provides regarding consensus, increases the need for players to engage in additional information-seeking activities.

A scheduler tool is provided for each player in the game, as an element, that can be seen only by the player. The schedule of the ships will be handled on another single display. This dual display system allows players to make changes to their own list directly and to see how their changes affect the global schedule. The overall schedule is updated to ensure the minimum variability between each individual player's schedules, making it necessary for players to work together to get support for changes they need approved.

Mechanics for Establishing Consensus. Consensus in this *VistaLights* is determined by using the standard deviation for the confidence ratings. The smaller the value of the standard deviation, the higher the consensus rating. By using variability rather than averages, players are encouraged to discuss the solutions and work together to resolve discrepancies. This also means that players do not have to be confident in a particular solution but that player confidence is consistent across players. Therefore, players do not only decide on the best option, but can better identify the strengths and weaknesses of each recommendation as it pertains to individual needs, other players' needs, and collaborative goals. Challenge is increased through this mechanic by increasing the threshold for consensus (smaller acceptable range for standard deviation) and by increasing the number of solutions required to reach sufficient consensus. We could decrease the necessary consensus for lower ranked solutions or require no consensus for the lowest ranked options. This would decrease the number of options needing to be considered by players. Limited options would require less effort and less strain on the collaborative efforts. The top-ranked solution will only be permitted to be implemented once consensus is established. If the group does not agree, with each individual approving the actions, the system will accept whatever resolution is currently ranked highest and the schedule that minimizes variability at the time the clock runs out. Collaboration takes time; however, when supply chains are disrupted there is often a limited time to respond in order for it to be remedied efficiently. For some disruptions, such as an oil spill, an immediate response may be necessary. Although limited time may inhibit collaboration, it requires that individuals work together as quickly as possible. Collaborative games targeted at specific groups, like *VistaLights*, can help to facilitate collaboration and illustrate the importance of collaborative efforts in resolving issues that affect multiple stakeholders and organizations.

5. Conclusions and Future Work

In this paper we proposed a design approach for Collaborative Decision Making Games (CDMGs). Through our investigation of game design, as well as studies performed on collaborative decision-making processes, we suggest that the design techniques for CDMGs should be informed from the latter discipline, into the former one. The design approach we created was applied as a design thought experiment for a CDMG called *VistaLights*. Future work will need to focus on testing prototypes of this game using appropriate evaluation techniques with target players to determine how successful this design approach was for creating collaborative opportunities and challenges. This can for example be achieved by allocation of criteria for empirical evaluation of the game and by careful consideration of quantitative or qualitative deviations from expectations and initial game design objectives. Future work should also focus on using our proposed design approach for CDMGs to create a larger number of collaborative games for further validation.

We achieved an understanding about a number of interesting points as a result of the creation and application of the design approach for CDMGs. First, the design approach allows the application of a *flexible design* strategy. There are a number of possible patterns taking place within each activity. There are necessary dominant patterns, but other patterns might need to be incorporated to facilitate collaboration; however, they may have less of a role in a particular activity. It is up to the designer to select the necessary or dominant patterns and identify those that are optional to apply. We chose our dominant patterns based on the larger goal of *VistaLights* as it relates to teaching collaboration in resolving disruptions in the port, based on guidance from the Port of Houston Port Authority. In then end, we identified three patterns that are associated with a main activity in the game: (1) information gathering: generate pattern, (2) prioritization: evaluate pattern, and (3) consensus: build-consensus pattern.

Second, the design approach for CDMGs promotes *transferable design*. In the CE discipline, collaborative work practices are described through the application of patterns of collaboration. Such a design quality would facilitate knowledge transfer to other novice game designers, as practitioners in the field, so that they can re-use the documented design scripts when designing games without having a need to get help from professional designers. Over time, successful implementation could be collected through a library of collaborative game mechanics. Once fully tested, for example, the way in which *VistaLights* accomplishes consensus building using a ranking and confidence rating tool can be transferred to other games.

Third, the design approach for collaborative games encourages the use of an *adaptive design* strategy. Once the patterns within each activity are decided, applied, and validated then the overall design structure of each activity can be incorporated into the design practices of other collaborative activities, which have the same or similar objective and are embodied within the same or different CDMG platforms. It may mean that designers will be spending more time by applying the approach in a careful manner; however, long-term they will profit from this. Future work will need to evaluate player in-game performance and their application of collaboration in out-of-game situations to evaluate this assumption.

However, the approach will above all help designers to think about collaboration in a more systematic manner. Even if designers do not follow the approach step-by-step, the initial consideration or awareness of the approach will help designers in deciding how collaboration needs to occur in games and will help in coping with the inherent complexity in collaborative games.

ACKNOWLEDGMENTS

The game *VistaLights* has been developed in collaboration with Dr. David Kaeli and the students Yifan Sun, Chisheng Liang, and Nishchitha Thuljaram. In addition, members of the Engineered Cyber-Social-Physical Systems research initiative at Northeastern University, the Houston Port Bureau, and Mare Liberum have been instrumental in the development.

REFERENCES

[1] C. Harteveld, *Triadic game design: Balancing reality, meaning and play*. London, UK: Springer, 2011.

[2] R. Hunicke, M. LeBlanc, and R. Zubek, “MDA: A formal approach to game design and game research,” in *Proceedings of the AAAI Workshop on Challenges in Game AI*, San Jose, CA, 2004.

[3] S. Bjork, *Patterns in game design*, 1st ed. Hingham, Mass: Charles River Media, 2005.

[4] D. Church, “Formal abstract design tools,” in *The Game Design Reader*, Cambridge, MA: The MIT Press, 2006, pp. 366–380.

[5] E. Adams and J. Dormans, *Game mechanics: advanced game design*. New Riders, 2012.

[6] M. Flanagan, D. C. Howe, and H. Nissenbaum, “Values at play: Design tradeoffs in socially-oriented game design,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2005, pp. 751–760.

[7] R. D. Duke and J. Geurts, *Policy games for strategic management: Pathways into the unknown*. Amsterdam, the Netherlands: Dutch University Press, 2004.

[8] J. P. Zagal, M. Nussbaum, and R. Rosas, “A Model to Support the Design of Multiplayer Games,” *Presence Teleoperators Virtual Environ.*, vol. 9, no. 5, pp. 448–462, Oct. 2000.

[9] C. Harteveld and G. Bekebrede, *Learning in Single-Versus Multiplayer Games: The More the Merrier?*, vol. 42. 2011.

[10] A. Azadegan and C. Harteveld, “Work for or Against Players: On the Use of Collaboration Engineering for Collaborative Games,” in *DPG 2014 - Workshop on Design Patterns in Games*, Fort Lauderdale, FL.

[11] T. Fullerton, C. Swain, and S. S. Hoffman, *Game design workshop: a playcentric approach to creating innovative games*, 2nd ed. Burlington, MA: Morgan Kaufmann Publishers, 2008.

[12] K. Salen and E. Zimmerman, *Rules of Play: game design fundamentals*. Cambridge, MA: MIT Press, 2004.

[13] G. Bekebrede, *Experiencing complexity: A game-based approach for understanding infrastructure systems*. Delft, the Netherlands: Next Generation Infrastructures Foundation, 2010.

[14] T. Manninen and T. Korva, “Designing Puzzles for Collaborative Gaming Experience – CASE: eScape,” in *Proceedings of Digital Games Research Association’s Second International Conference*, 2005, pp. 233–247.

[15] C. Islas Sedano, M. B. Carvalho, N. Secco, and C. S. Longstreet, “Collaborative and cooperative games: Facts and assumptions,” in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 370–376.

[16] R. Hämäläinen and K. Oksanen, “Challenge of supporting vocational learning: Empowering collaboration in a scripted 3D game - How does teachers’ real-time

orchestration make a difference?," *Comput. Educ.*, vol. 59, pp. 281–293, 2012.
[17] J. P. Zagal, "Collaborative games: Lessons learned from board games," *Simul. Gaming*, vol. 37, no. 1, pp. 24–40, Mar. 2006.
[18] A. Beznosyk, P. Quax, K. Coninx, and W. Lamotte, "The influence of cooperative game design patterns for remote play on player experience," 2012, p. 11.
[19] J. B. Rocha, S. Mascarenhas, and R. Prada, "Game Mechanics for Cooperative Games," in *Actas da Conferência ZON, Digital Games 2008*, 2008, pp. 73–80.
[20] M. Seif El-Nasr, B. Aghabeigi, D. Milam, M. Erfani, B. Lameman, H. Maygoli, and S. Mah, "Understanding and evaluating cooperative games," 2010, p. 253.
[21] A. Hevner, S. March, and J. Park, "Design Science in Information Systems Research," *Manag. Infomation Syst. Q. MISQ*, vol. 28, no. 1, pp. 75–105, 2004.
[22] J. Nash, "Two-person cooperative games," *Econom. J. Econom. Soc.*, pp. 128–140, 1953.
[23] R. Hunicke, M. LeBlanc, and R. Zubek, "MDA: A formal approach to game design and game research," in *Proceedings of the AAAI Workshop on Challenges in Game AI*, San Jose, CA, 2004.
[24] G. L. Kolfschoten, G.-J. de Vreede, R. O. Briggs, and H. G. Sol, "Collaboration 'Engineerability,'" *Group Decis. Negot.*, vol. 19, no. 3, pp. 301–321, May 2010.
[25] R. O. Briggs, G. L. Kolfschoten, and G.-J. de Vreede, "Defining key concepts for collaboration engineering - Google Scholar," presented at the AMCIS, 2006.
[26] G. L. Kolfschoten, R. O. Briggs, G.-J. de Vreede, P. H. M. Jacobs, and J. H. Appelman, "A conceptual foundation of the thinkLet concept for Collaboration Engineering," *Theor. Empir. Adv. Groupw. Res. Theor. Empir. Adv. Groupw. Res.*, vol. 64, no. 7, pp. 611–621, Jul. 2006.
[27] "GroupExplorer." .
[28] "MeetingSephere." .
[29] "ThinkTank." .
[30] R. O. Briggs, G. J. de Vreede, and J. F. Nunamaker, "Collaboration engineering with thinkLets to pursue sustained success with Group Support Systems.," . *Journal of Management Information Systems*, vol. 19, no. 4, pp. 31–36, 2003.
[31] J. H. Appelman and J. van Driel, "Crisis-Response in the Port of Rotterdam: Can We do Without a Facilitator in Distributed Settings?," 2005, p. 17b–17b.
[32] J. Bragge, H. Merisalo-Rantanen, and P. Hallikainen, "Gathering Innovative End-User Feedback for ContinuousDevelopment of Information Systems: A Repeatableand Transferable E-Collaboration Process," *IEEE Trans. Prof. Commun.*, vol. 48, no. 1, pp. 55–67, Mar. 2005.
[33] A. Azadegan, K. N. Papamichail, and P. Sampaio, "Applying collaborative process design to user requirements elicitation: A case study," *Comput. Ind.*, vol. 64, pp. 798–812, 2013.
[34] G.-J. de Vreede and R. O. Briggs, "Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks," *Proc. 38th Annu. Hawaii Int. Conf. Syst. Sci.*, 2005.
[35] S. T. March and G. F. Smith, "Design and natural science research on information technology," *Decis. Support Syst.*, vol. 15, no. 4, pp. 251–266, Dec. 1995.
[36] H. Takeda, P. Veerkamp, and H. Yoshikawa, "Modeling design process," *AI Magazine*, vol. 11, no. 4, pp. 37–48, 1990.
[37] A. Bezuijen, "The further development of TeamUP, a teamwork focused serious game," 2012.
[38] J. F. Nunamaker, N. C. Romano, and R. O. Briggs, *Collaboration systems: concept, value, and use*. 2014.
[39] G. L. Kolfschoten, G. P. . Duivenvoorde, and G.-J. de Vreede, "Practitioners vs Facilitators a Comparison of Participant Perceptions on Success," presented at the 42nd Hawaii International Conference on System Sciences, Hawaii, 2009.
[40] G. L. Kolfschoten and G.-J. de Vreede, "The Collaboration Engineering Approach for Designing Collaboration Processes," in *Groupware: Design, Implementation, and Use*, vol. 4715, J. M. Haake, S. F. Ochoa, and A. Cechich, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 95–110.