Sharing design perspectives through storytelling

Ana Cristina Bicharra Garcia
Universidade Federal Fluminense
bicharra@dcc.ic.uff.br

Carlos Eduardo Carretti
Universidade Federal Fluminense
carretti@addlabs.uff.br

Inhaúma Neves Ferraz
Universidade Federal Fluminense
ferraz@dcc.ic.uff.br

Cristiana Bentes
Universidade do Estado do Rio de Janeiro
cris@eng.uerj.br

Abstract

Design consists of analyzing scenarios and proposing artifacts, obeying the initial set of requirements that lead from initial to goal state. Finding or creating alternative solutions, analyzing them and selecting the best one are expected steps on designers´ decision making process. Very often, not a sole designer, but a team of them is engaged in the design process sharing their expertise and responsibility to achieve optimum projects. In a design team, most conflicts occur due to misunderstanding of one's assessment over specifications and contexts.

Decisions explanations play a key role on teamwork success. Designers are rational agents trained to follow rational methods. Acceptable justifications lay on value function, requirements, constraints and criteria. Generally, explanations are delivered in a multimedia fashion composed of text, graphics and gestures to provide the audience the ability to perceive what was contextually imagined. The more spatial is the reasoning, the richer should be the explanation channel.

This paper presents CineADD, a design explanation generation model based on cinema techniques such as: animation, scripting, editing and camera movements. The idea is to provide designers a tool for describing the way their projects should be visually explained as in a movie. Designers develop their projects on an active design document environment. Rationale is captured as a design model, so explanations can be generated instead of retrieved. The captured design model serves as a base to visually reconstruct design giving emphasis and guidance by using movie storytelling techniques.

CineADD was implemented for the domain of oil pipeline layout showing the feasibility of this approach. We expect CineADD to become a commodity attachable to any Intelligent CAD system.

1. Introduction

Human-computer interaction (HCI) consists of a dialog between users with a set of demands and computer systems with a set of affordances, built in their codes, left by their designers. The interaction happens physically through input and output devices, such as keyboard and printer. It also happens through information exchange that let emerge the cognitive distinction between players. Depending on the complexity and the way messages are delivered, it may become a challenge for users to understand them.

The communication involves the speaker, the listener, the channel, the content to be transmitted, the code used to make the content of a message and the message itself. Texts, graphics and pictures are the common codes employed by the computer to deliver the message. However, sometimes HCI demands an "immerse" experience [Lachman 97], as in movies, for users to efficiently perceive the overall, but sometimes hidden, information. A movie has the power to: (1) connect spatial and temporal information, (2) make concrete one's perspective of the facts and processes, (3) reconstruct human memory, and (4) make the audience think.

Knowledge based systems (KBSs) have been successfully used in CAD systems to assist users in developing design projects either by offering design solutions or verifying decision alternative solutions [ten Hagen, 1987] [Garcia et al. 97]. KBSs contribute to users finding efficient solutions, given a design context. Users´ acceptance depends strongly upon the credibility of computer suggestions. An Active Design Document (ADD) [Garcia 92] is an environment for developing engineering design assisted by a computational agent trained for making decisions on projects in a specific design domain. ADD allows users to develop their project being monitored by its design agent.

Whereas the agent's knowledge base covers user decisions, explanation on those decisions can be derived without user's guidance. Whenever a user's decision on a design project conflicts with ADD's expectation, the computational agent will interact with the user to gather more knowledge to improve its knowledge base. Providing clear explanation is the key to this teamwork: user and computer agent. Furthermore, since a project is generally developed in teams, the availability of design decisions explanation allows understanding of individual perspectives on design issues.

Explanations vary from canned text (optionally multimedia message), working as pre-recorded annotations, to on-demand generated explanations. Although a textual explanation is fundamental, there are domains in which spatial and temporal reasoning are crucial to decision making. For such domains, explanations composed only by text, diagrams and pictures will not work because the spatial and temporal transformation will not emerge. For instance, planning a kitchen layout in a 2D (or 3D) space consists of optimizing space distribution obeying a set of norms such as the refrigerator should not be placed besides the oven. The designer’s task consists of moving, erasing, and reshaping objects [Fischer & Lemke et al. 91]. The decisions in this domain are well reported neither by textual notes nor by figures. They need to be reported using actions. An event in time makes difference on possible understandings of facts. When explanations reflect a set of actions or a process in a time frame, a sequence of scenes may be transformed in an animation leading to a reconstruction (full or partial) of what happened. The introduction of another visual medium (animated scenes) brings issues related to animation speed, scene selection, and user's attention guidance. Creating a scene, from a system's interaction log, is a matter of using computer graphics techniques such as rendering. The issues discussed in this paper concerns building interactive narratives as the explanations for artifact designs. In this context, designers play the role of a movie director choosing the right framing to communicate their idea when creating an artifact. In addition to allow designers play the director’s role, it is important to let end-users investigate the explanation from different perspective in order to understand it.

In this paper, we present CineADD, a design explanation model based on cinematic techniques. CineADD was planned to any CAD systems, however we develop our studies using Active Design Documents applied to engineering spatial layout domains. Our goal is to show the feasibility of using cinema and animation techniques to generate visual animated explanations that augment end-users understanding of the designer's decisions and intentions. This visual presentation represents the design story; i.e., the designers’ perspective of their project. By including special effects to the design story, designers can emphasize or hide details of the scenario. Our goal is to allow designers to create a script that describes the way they want their design to be explained to others. Designers work as filmmakers creating a script for their design movie. In addition, the other participants may change the script to further investigate the design.

In our research, we investigated the use of cinematic techniques to empower user interface systems, allowing a greater volume of knowledge to be concisely conveyed to end-users. The encoding mechanisms, that allow images and the interaction of images to carry meaning, must allow designers (filmmakers) to express their intentions and end-users (audience) to perceive them.

This paper is organized as follows. In Section 2 we discuss the design task as an activity normally conducted by a group of people. Section 3 describes the ADD approach to design development and documentation. Section 4 presents some background on cinema and animation techniques. Section 5 provides a description of our model, called CineADD, to apply cinema techniques to the design process. Section 6 presents a case study of CineADD model on the context of oil pipeline design in the ADDSub system. Finally, Section 7 provides our conclusions of this work.

2. Design Task as a Group Activity

An artifact generally emerges as a solution to a set of needs from a group of people willing to pay for it. Before the idea becomes concrete, a great deal of work must be done usually involving people from different expertise. Although it is neither the longer nor the most expensive, conceptual and preliminary design phases are crucial because the solution is conceived during this period. Mistakes can still be found and fixed at a low cost when compared with the remaining phases such as the detailing and the construction phases.

Design is a complex activity normally performed by a group of people, with different expertise, that either work together to reach a good solution, or work sequentially to carry out the project from conceptual to detailing design phases.

From the initial specifications, designers elaborate a project concept that is discussed by the design team and the end-user. During the development of the project, the set of specification grows and get modified as a result of deeper understanding of the problem being addressed by the project. Everybody in the design team shares responsibilities, however, the ones that sign the project get the fame and the blame for the success or failure of the artifact.

Design documents are produced to communicate a specification of the solution designers meant for further construction. Frequently, due to the complexity of the artifact, there is more than one writer to the artifact's story. As expected, conflicts among them appear. In this context, the documentation is also used as a communication medium to allow mutual understanding.

From the documentation, issues may be raised, conducting to a group discussion on possible better overall solutions. In summary, there are two main types of users: the documentation builder/writer and the documentation consumer/reader.

The builder/writer is the design team. They are people with the same or different background, hired to contribute with their specific expertise, but also committed to the final integrated solution. Not rarely, an designer must compromise the quality of his decision on a portion of a project to allow the best-integrated solution to happen. However, giving up a partial solution requires a great convincement effort. Consider the following scenario, plumbing and air conditioning design. There is a great deal of overlapping between the two designs, but both must be integrated in a house. Sometimes, one must worsen his design to let the other be possible. When writing the documentation, they write the final solution. Unfortunately, the discussion is let aside. Consequently, when accessing the final documentation many alternative solutions that have already been discussed raise again, and designers have to rebuild the rationale for the final solution.

The consumer/reader are people interested in reading design documentation for many different reasons such as:

- Accept/reject a solution: this scenario lets writers gain supporters to their point of views.
- Approve/reject a solution: this scenario lets writers share their responsibility with readers.
- Build the artifact: this scenario concerns making the specification concrete.
- Understand a solution: this scenario is the basis for all other scenarios (accepting, approving or building the artifact).

Even though a good documentation would save time, it has been neglected due to emphasis on generating a good quality solution. The importance of design rationale has been acknowledged and research effort has been devoted to the capture and retrieve of it [Moran & Carroll 96].

Although design rationale capture and deliver are highly connected, this paper focuses only on design rationale retrieval. When using an intelligent CAD system environment, specifically an ADD environment, designers can gradually build design and documentation for further investigation. Design rationale capture is a subproduct of developing design. The issues discussed in this paper concern delivering an effective message that reveal the designers' perspectives when building their solution.

3. Active Design Documents

Active Design Document (ADD) [Garcia 92] has been used as an intelligent CAD system, helping designers develop and document their projects [Vivacqua & Garcia 96] [Garcia et al. 97]. ADD approach uses the apprentice metaphor.

A computational agent, capable to develop a design in a specific domain, monitors a designer developing a project using the ADD environment. The computational agent creates expectations on design decisions, based on its knowledge base. Whenever an expectation fails, the agent interacts with users presenting its rationale for its expectation. Based on the presented explanation, users may have a clue on ADD knowledge representation for inputting modifications. Knowledge acquisition is restricted to scenes within a context. There is no commitment to an integrated knowledge base. The final knowledge base should cover the project develop by the team designer-ADD. Generating design decisions explanation is an important but easy task for the computational agent, considering its ability to generate a design decision expectation.

To work as an apprentice, ADD must start with an initial domain model that guides its decision-making throughout the decision process. This initial model is implemented by a knowledge engineer, and it represents all necessary abstractions on the process and parameters of a given domain. The created model, however, does not always produce the same decisions that human designers do. This happens for a number of reasons, such as: the system model does not cover all possible situations or the designer experience may be above or below the knowledge captured in the ADD domain model. In any of these situations, designers may solve conflicts and differences by either changing the system's underlying model or their own mental model. This shows that ADD is actually a learning environment for system and users.

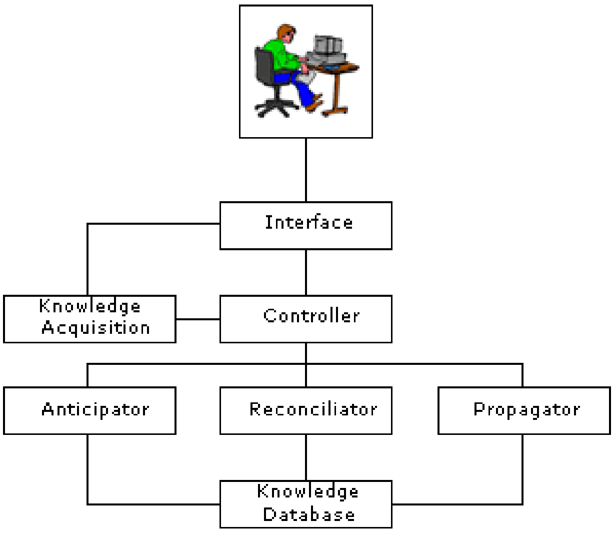

Figure 1: ADD model architecture.

Figure 1 shows the ADD architecture with its components. There are seven components:

- **Anticipator**: it is the inference engine of the apprentice agent. It monitors the design project for focus alteration. A design project is represented as a set of parameters with their values, available in a blackboard structure. Whenever a parameter gets the user's focus through the design interface, the Anticipator triggers its knowledge base to create a valid expectation for that parameter value. In addition to a parameter value, the decision may involve the existence of the parameter at all. For example, suppose the task of designing a house. Although, the owner may provide a set of specifications for a porch, later on we may conclude that there is no space for even thinking of that.
- **Reconciliator**: it identifies conflicts between the system and users' decisions. Defining the similarity function is the main issue for this module.
- **Knowledge Acquisition Elicitor**: it elicits from designer changes to the initial design model. A mismatch diagnosed by the Reconciliator triggers the action of the Knowledge Elicitor Component. Before letting users change its knowledge base, it shows an explanation of how its expectation was reached. An explanation is composed of:
  - The sequence of the decisions already made;
  - The dependency graph showing what parameters influence the current decision;
  - A tradeoff table showing the alternative values and the performance of each alternative according to the set of pertinent constraints and criteria.

  After presenting its explanation, the Elicitor lets users include or exclude: parameters, parameter dependencies, alternative parameter values, rules to produce new parameter values, rules to evaluate parameters, constraints, criteria, and evaluation functions.

After receiving the changes, the Reconciliator checks if the changes are sufficient to erase the mismatch. The elicitation process continues till the Reconciliator gets satisfied or the user wants to force a value with no explanation.

- **Propagator**: it propagates design decisions effects. Whenever a decision changes, other changes may be requested to comply the new scenario. The Propagator stops when it reaches an untouched design space area or when it reaches a design decision with a value imposed by the designer that does not comply with the knowledge base (a break on the domain knowledge consistence);
- **Controller**: it monitors the project blackboard and determines which module should be triggered;
- **Domain Knowledge Base**: it contains the heuristics ruling the decision process in the domain. A dependency parametric network represents the domain knowledge. Primitive parameters are the input data. Derived parameters have a formula, either heuristic or mathematic, for determining their values. Decided parameters require a tradeoff analysis, so alternative values must be generated. Constraints are applied to eliminate the unfeasible alternatives, while criteria are applied to order them. As a rational agent, the best alternative is preferred and selected. Sometimes, dependencies are dynamically assigned increasing the complexity of the network processing; the same happens with parameters with mutual dependencies, i.e., finding the value of one parameter influences finding the value of the other and vice-versa.
- **Design User Interface**: it consists of the CAD interface from which designers will develop their project.

This modular architecture allows ADD to capture all information needed to recover the design history, as well as the information needed to justify the decisions underlying the final product (artifact).

4. Framing Cinema Techniques to Use in Computer Interface Design

Cinema is an attractive medium for transferring thoughts. Using a set of techniques, filmmakers build a narrative to deliver a message that is communicated to an audience through a movie. Individual understanding requires a balance between the spectator’s and filmmaker’s way of seeing the world. Similarly, computer interfaces have a message from the designer that must be understood and negotiated [Persson 99], [Persson 01]. Cinema language offers an interesting approach to enrich computer interfaces to augment users´ reception. This section explains the set of techniques applied to enhance ADD interfaces dedicated to present designer’s explanations on design projects.

4.1. Cinema and Animation Techniques

Cinema language is composed by cinema techniques [Lester & Bares 97a] [Davenport et al. 91]. These techniques are heuristic rules that bring the real world to the movie screen with all its visual, temporal and sound restrictions. Animation needs further techniques, because the interface is different (there is a transition from a real visual medium to an imaginary one).

Cinema techniques are classified in five groups [Silverstein & Huss 68]:

- Camera movements, such as zoom, pan-shot;
- Camera positioning, such as close-up, wide-shot;
- Edition, such as cut, cross-cut; and
- Style, such as fiction, silent movies, documentary, and
- Narrative, such as slow motion, re-ordering and flash back.

Animation techniques can emphasize actions and physical processes that could not be perceived using other techniques [Lachman 97]. Animated movements may help users to imagine (rationale reconstruction) what might have happened during a design process, making it easier to visualize concepts, objects and thoughts. There are seven different animation rules [Thomas & Johnston 84]:

- Anticipation: the character movements are anticipated, so the audience know in advance which movement will occur, generating expectation and attention;
- Deformation: some elements are deformed during collisions. These deformations must be anticipated and exaggerated, generating expectation.
- Continued actions: two simultaneous actions that came from the same event must not begin or end together, focusing the attention to each one individually.
- Secondary movements: the effects of an action that occurred on an object must be propagated to the objects related to it.
- Movement Sequence: when an action is initiated, it cannot be drastically finished. Postponing it allows emphasis.
- Exaggerated Movement: some actions may not be perceived if they appear as they occur in real world; when some movements are exaggerated, the action is emphasized.
- Scenario Creation: the objects must be placed on the screen, so that the action of the characters can be observed clearly.

These rules are essential to emphasize, efficiently and pleasantly, the most important elements of a scene, without deviating the audience’s attention. Directing the audience attention to specific actions or scene elements is one of the most important features of the animation planning.

There are also special techniques for focusing the audience attention on some aspect of the image [Blinn 94]:

- Pointers: like arrows pointing to the object that must be highlighted;
- Blinking: objects blinking on the screen are not a very subtle way of emphasizing them, but it works.
- Saw Effect: show some interleaving scenes with the state of an object before and after an action occurred on it.

A text can also be used to emphasize, to clarify and to go deep on the information behind a movie or an animation. Including a text over the images is a common technique, even on movies, where there are captions and graphic animations. The inclusion of captions on an animation, however, must follow two main aspects: the size of the text and the screen position for this text.

The animation and cinema techniques deal with focusing the audience attention on relevant information that must be transmitted. The use of these techniques on a computational environment requires some precise definitions of its utilization and organization.

4.2. Idioms

Idioms are scripts that contain well succeeded positioning and moving camera rules. They also contain allocation time for scenes and takes, and the behaviour of the elements filmed. The rule set encompasses the filmmaker expertise for the capture of an event sequence that tends to be repeated during the film [Christianson et al. 96] [He & Cohen et al. 96].

A frequent idiom, used in movie production, is the Dialog Idiom. This Idiom defines what should be presented to the audience. This Idiom applied to the dialog of two actors consists of three steps: (1) introduce the world containing all participants, (2) present each participant individually when they are speaking and (3) show the World again. As illustrated in Figure 2, on the first shot the camera exhibits both actors in wide shot. After that the scene alternates a close shot of each actor individually. Finally, the scene ends with the same wide shot of the two actors.

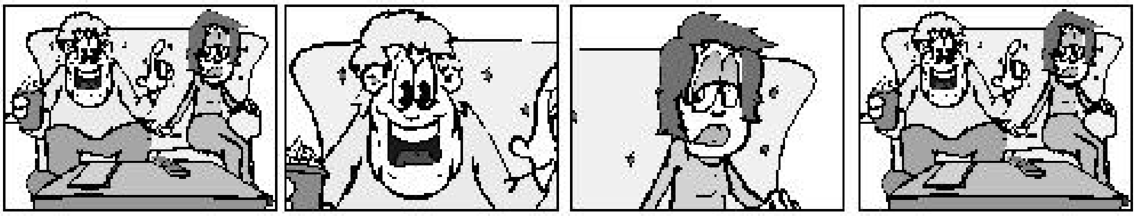

Figure 2: An example of an idiom applied to a dialog scene between two actors.

The rules for constructing scenes or takes can be decomposed and grouped, creating generic idioms, incorporating specific information on a set of cinematography techniques. Incorporating idioms to script specifications of a film allows creating independence between the used visual techniques and the domain elements that will be shot.

4.3. Storytelling Techniques and Styles

The Storytelling is where the events take place. There are different storytelling styles that have been improved over the years. New heuristic techniques have been included as new media, like TV and sound, adhere the art of making a film [Katz 91].

The cinema storytelling can be grouped in three categories [Arijon 76]:

- News Flashes: deal with unpredictable facts whose final result is a set of disconnected images that must be edited.
- Documentaries: deal with a sequence of situations with a common motivation.
- Fiction Films: deal with real events, but they can be repeated, until they capture the director’s desire. There is no single point of view.

On Lester’s work [Lester & Bares 97b], the storytelling styles are defined following the generalization of the users profiles. These profiles are defined according to the cinema techniques adopted by users to visualize animations. The defined styles are incorporated to the system and cannot be modified.

Works on interactive movies [Davenport 96] and on automatic generation of cinematography storytelling [Brooks 97] have been very helpful to turn a computer into a storytelling agent, changing the limits between the filmmaker and the audience. The movie becomes interactive in the sense the audience may change the script for producing the movie in order to get other perspectives on the facts to be revealed. There are two sides on analyzing the goodness of manipulating a story. The good part is to allow people to get deeper understanding by trying different ways to explore a story given a set of scenes. On the other hand, the audience may get confused and may miss the message filmmakers were trying to pass through the movie. This issue is not unique for Interactive movies. It covers from interactive textual narratives through Hypertexts advances.

5. CineADD: Shooting Design Decisions Scenarios

CineADD is an extension to ADD explanation model. It represents the way a movie animation is automatically generated to augment an explanation on a design built in ADD environment. Even though computer graphics techniques are used to actually produce the scenes, there is nothing new about them. The research focus is to let designers to create explanation narrative scripts for generating design explanations.

CineADD, as illustrated in Figure 3, uses three input sources to work:

- Design decision log: it contains the sequence of decisions reflecting a design project development. Playing this log remounts the entire "design movie scenes". Generally, it lacks structure and a straight line of reasoning.
- Design domain knowledge base: it contains an instance of the parametric dependency network, the assigned values and the inference rules applied in a specific design case of a domain.
- Query: it represents the user's needs entered using the Explanation Interface. It contains the type of question, such as "Why" or "How", the target decision to be explained, such as the entire design, and the focus of the explanation, such as the user wants to understand or the user needs to approve a decision.

CineADD Pre-Processing Module represents the filtering that must occur to determine the user's needs and the portion of knowledge and history that must play a role in the answer.

The Strategy Knowledge Base contains heuristics for selecting the content to respond to user's needs. For instance, a strategic rule for explaining a decision that was forced by the designer may consist of selecting the parameters that map to the decision, the immediate dependent parameters, their assigned values, what should be expected as the decision, a sign that the decision is not fully explainable and the possible annotation made by the designer on that decision.

Designers during a project development may configure the way Idioms should be selected, defining the messages format. For instance, a designer may want to make a zoom in an object under too many visual constraints to emphasize the difficulty of settlement of it in a 2D space. He may build an Idiom that shows the entire scene with all objects, followed by zooming in that specific object over constrained.

The Content Planner selects a set of strategies, from the Strategy knowledge base, and a set of Idioms, that should be applied to build the scenes to satisfy the user's needs, to produce the presentation structure.

The Presentation Planner is responsible for creating the actual movie explanation. It selects visualization techniques such as zooming or pan actions [McReynolds & Blythe 98] [Dufaux & Moscheni 96], from the Visualization Techniques Knowledge base, and applies them to the presentation structure filled with information gathered from the design domain knowledge base.

After all these processes, the created movie is apt to be played. The visual answer works as a complement of the textual answer. Further research will be including spoken language as the complement to the film.

CineADD relies upon an interaction history file, called the design log. The design log contains the user's actions while developing a design in ADD environment, and the domain knowledge base (represented as a parametric dependency network). From this raw material, CineADD applies cinema and animation techniques to organize and compose a visual presentation that works as a complement to the textual explanation generated by the system. Therefore, end-users' attention is guided to perceive the designers intent.

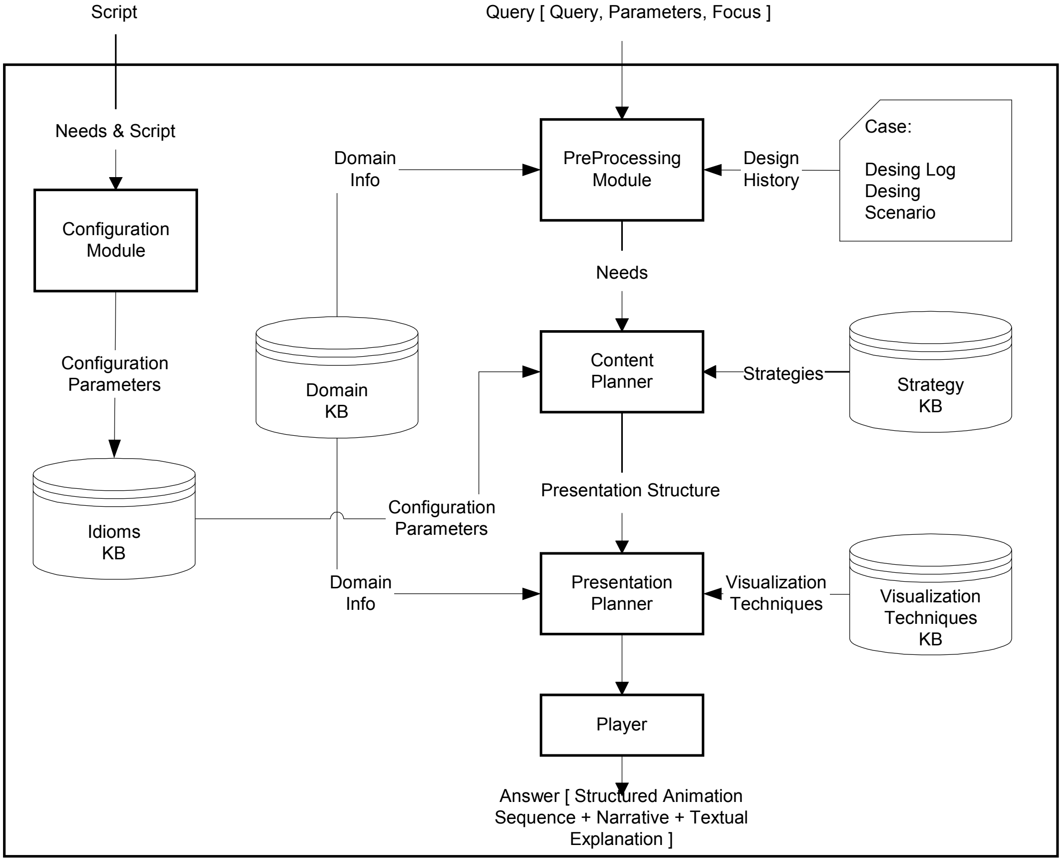

Figure 3: CineADD model as a complement of ADD textual explanation generator.

Together with the design log, the domain knowledge base is used to reconstruct the design history. Rebuilding design can be accomplished through two approaches: Replay and Rebuild.

The Replay approach consists of presenting the design history exactly as it was. The actions are shown to the audience in the same sequence they were made. In addition, even irrelevant actions are presented. It is a complete reconstruction of the facts. End-users should watch carefully the presentation observing each detail and interpreting the entire set of actions to conclude what might have happened in the design process.

The Rebuild approach consists of selecting relevant set of actions on the design log (design scenes) to create a presentation that satisfies a specific user's question. Relevance is defined in the designer's mind. When defining Idioms, designers indicate the type of scenes and the sets of actions to comply with explanation goals. The Rebuild approach follows a method consisting of:

- Interpreting the needs behind a user's question (Pre-Processing);
- Selecting relevant scenes from the design log that plays an important role building an answer (Content Planner);
- Planning the Visual Presentation (Visual Planner);
- Presenting the movie answer to the observer (Player).

The Rebuild approach offers an explicit language to designers and observers' plan design sequences to be interpreted and presented. There are three expected planner agents:

- The author: the one responsible for creating the presentation strategies;
- The designer: the one responsible for building the design, and, consequently, the design history log;
- The observer: the one responsible for studying and understanding the project; i.e., the inquirer agent.

CineADD allows designers and observers to switch roles to discuss about a project. It is expected that the presentation strategies be pre-defined and included in CineADD. The Designer should select the strategy it wants to apply to explain any specific decision. These strategies reflect the designers´ perspectives that they want to share with the audience.

CineADD also let the audience further investigate an explanation in order to allow any details that may be hidden in the designer's explanations to be brought up.

6. CineADD applied to Oil Pipeline Layout Design

In this section, we will present an example of using CineADD in a real design domain. Before explaining the use of the cinema techniques, we present the application domain in the context of an ADD system.

6.1. The Oil Pipeline Layout Design Task

Oil exploitation in deep water fields needs special processes. The oil is pumped from the bottom of the sea to offshore platforms from which it is treated to be exported to land. The oil pipeline layout problem belongs to the class of spatial layout problem. There is a set of objects on a 2D space that have to be located and connected considering a number of restrictions of the environment. Finding a solution to this problem is not a simple task, considering the complexity of the environment and the information overloaded on it. To handle the complexity of the oil pipeline layout problem, we divided the task in seven different subtasks.

Given a set of wells with their target areas and a number of oil exploration units (usually called platforms), the subtasks are:

- Finding the best wells clustering, considering the relative distance among the geometric centers of the elements;
- Assigning each platform to wells clusters, considering the maximum oil processing capacity of a platform and the maximum number of risers;
- Locating the well heads in each target region, in order to minimize the distance to the platform;
- Locating each platform in a free area as close as possible to its cluster geometric center;
- Defining the exact source and destination of a pipeline, from wells to platforms;
- Defining the pipeline route for draining the oil from wells to the assigned platforms;
- Defining intermediate draining elements to receive oil from wells and bend to platforms.

Locating the wells, the platforms and the other oil draining elements as well as designing the pipeline connecting them are the activities involved in an oil pipeline layout project. This is the most expensive part of an oil field exploitation project. Even though optimizing the project saves a great deal of money, it is rarely accomplished due to the complexity of the involved reasoning.

All above decisions are taken considering the spatial constraints of the environment. Partial decisions are also considered for the domain model. For example, designers may locate only half of the wells or they may create only one group of targets to focus their attention, leaving the remaining oil target areas to be grouped later.

6.2. ADDSub: a System to Assist Oil Pipeline Layout Design

ADDSub system [ADDSub 98] is an Intelligent CAD tool used to assist and document designer's decisions taken during oil pipeline layout project development. ADDSub helps the users to optimize their project, but is not merely a calculation tool to find the best solution for each of the subtasks described in the previous section. The problem is very complex and the order in which each decision is taken affects the overall solution of the problem. So, find one best solution for the whole problem would be "computationally intractable". The ADDSub approach takes benefit from the partnership between the designer and the system (taking advantage of the computer fast calculation with the designer expertise and visualization facility).

ADDSub offers a friendly interface presenting in a canvas active area the undersea topography and texture, as well as the existing objects. ADDSub offers a direct manipulation interface where objects are displayed and modified on this canvas. Figure 4 illustrates a small and fictitious oil pipeline layout project developed with ADDSub. As we can see from the figure, there are seven oil target areas (big circles) and a big obstacle area (irregular polygon). The wells (small circles) are connected to the corresponding platform (rectangle) by a pipeline (thick lines), and the slim lines represent the undersea topography.

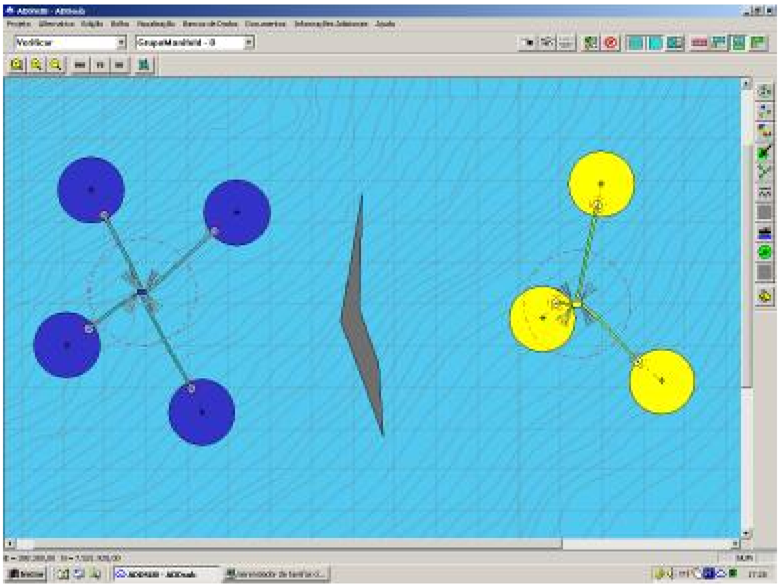

Figure 4: An example of an oil pipeline layout project developed with ADDSub.

Following the ADD model, ADDSub observes the designer actions and compares them with the rationale stored in its knowledge base. The system can disagree with the designer action, presenting another suggestion for it. The designer chooses which actions or decisions will be adopted in the final project.

ADDSub operates in six different modes: Data-Entry, Suggest, Verify, Free, Knowledge Acquisition and Explanation. In the Data-Entry mode, the designer inputs the data configuring the project to be developed. In the Suggest mode, the designer requests suggestion from the system on each of the layout design subtask. In the Verify mode, the designer proposes its own solution to a subtask and the system analyses it. In the Free mode, the designer imposes its solution without asking the system to analyse it, the system works as an usual drawing software in this mode. In the Knowledge Acquisition mode, the designer may include or modify calculation methods and design criteria. Finally, in the Explanation mode, the system provides explanations on the decisions made during the project development. In the next section we present a detailed description about the explanations given by ADDSub.

ADDSub was developed in C++ under Windows and is being successfully used by the Brazilian Oil Company (Petrobras) to develop pipeline layout projects. The use of ADDSub provides Petrobras with three important benefits: reduction on the project development time; reduction on the overall project cost (as it optimizes the project elements); and generation of automatic project documentation (with explanations).

6.3. Explaining Design Decisions in ADDSub context

When designers create a project, they do not consciously create explanations of their rationale. ADDSub automatically generates these explanations, as the system is used to make design decisions. To provide explanations, ADDSub records, in a design log, both the sequence of design decisions taken in each of its operation modes, and the agent (system or designer) that generates that decision. The design log provides a chronological view of the designer’s actions in a project.

Using the design log information, the system can provide explanations for all the decisions taken during the project development. Each decision is represented as a parameter, its assigned value, the agent who assigned this value, and information about the decision compliance (or not) with the system model.

On the Explanation mode, the designer (or the observer) interacts with the system writing a question. ADDSub provides answers for three types of questions: “How”, “Why”, and “Which”. With these questions, usually, designers/observers want to know why a system parameter has a certain value. Then, the system provides a textual explanation answering the question. Besides this textual explanation, the system provides: (1) the design history; (2) the dependence network; and (3) the heuristics. The design history contains the sequence of decisions occurred before the parameter value was calculated, which is a linear sequence of actions. The dependence network contains the part of the parametric network that refers to the parameter of the question. The heuristics contains the heuristics or formula used to calculate the value of the parameter asked.

The textual answer given by ADDSub is generated using Natural Language. The text generation uses narrative and rhetorical structure [Mann & Thompson 87] to build a textual explanation that delivers the knowledge behind the set of information pieces.

6.4. Augmenting Explanations with CineADD

Oil pipeline layout design is mainly a visual task, as it involves locating and changing the location of elements in a 2D area. The text explanations given by ADDSub may not clearly express the answer the user needs. When too many visual actions occur, a text or even pictures telling about design decisions do not suffice to let the information emerge. It keeps bouncing from wordy to concise sentences causing a cognitive overload on users to create a mental image to understand designer's intentions. Next, we will explain this argument using a very simple example (for an easy understanding) of how CineADD can improve an explanation about one decision taken by ADDSub concerning on the localization of a specific well.

Suppose a user wants to know why the well number 5 (well 5) was located on coordinates (x,y). The user formulates a question to the system as the following:

"Why well 5 was located on (x,y)?"

The system provides information about the design history, the heuristics used, the dependencies to solve that question, a view of the well location and a textual explanation as follows:

"Well 5 was located on coordinates (x,y) because:
it does not violate any spatial constraint
it is the closest location to platform 1"

This answer does not give a clue to users whether other alternative locations were even tried. Maybe the case, other solutions also leading to minimum distance were possible or even preferable. The system can deal with that flaw showing a text explaining the other considered alternative solutions and why they were discarded. The answer would be:

"Well 5 was located on coordinates (x,y) , Alternative 4, because:
it does not violate any spatial constraint; and
it is the closest location to platform 1"

Alternative 1: location (x1,y1), distance to platform 1: K1. Discarded because it violates restriction: intercept existing element.

Alternative 2: location (x2,y2) distance to platform 1: K2. Discarded because it violates restriction: it does not lead to minimum distance.

Alternative 3: location (x3,y3) distance to platform 1: K3. Discarded because it violates restriction: intercept existing element.

Alternative 5: location (x5,y5) distance to platform 1: K5. Discarded because it violates restriction: intercept existing element.

(the text continues until all alternatives are listed)

We can see from this example that the concise text explains only about location (x,y), but does not provide the answer for “why not the other positions”. The complete text answer for this question, on the other hand, becomes wordy, and it’s very difficult for users to visualize it.

CineADD model provides ADDSub with an enhanced explanation interface, as shown in Figure 5, because it gives the answer on an animated fashion. The user can see the decisions taken, without having to analyse an overloaded text. For example, for the same question on the localization of well number 5, the CineADD presents a movie showing why location (x,y) was the one chosen. Although it is very difficult to show the actual advantages of seeing a movie on a paper, we will try to show in Figure 6 some scenes of the movie that answer this question.

The whole project is shown in Figure 5. This scene is cut and edited with the next scene that is a pan on the project, to centralize the well 5 on the screen. After centralizing the well, the next scene provides a zoom in the specified well. These scenes provide the user the notion of where the target area of well 5 is on the project. After that, the movie begins to show all the possible alternatives for locating well 5 on that target area. There is a sequence (that was omitted) showing each alternative for well placing. When the alternative violates any constraint, a square marks this violation, and there is a text explaining it on the text area of the screen. The last scene shows the chosen location of the well.

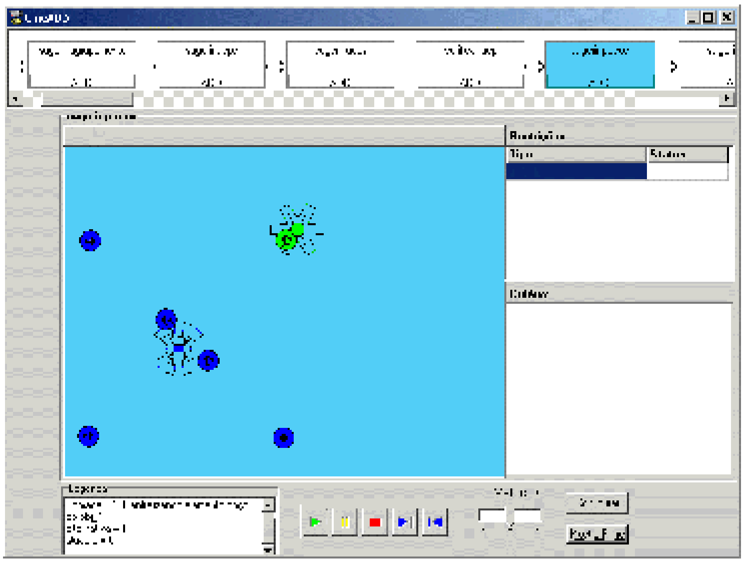

Figure 5: CineADD Interface.

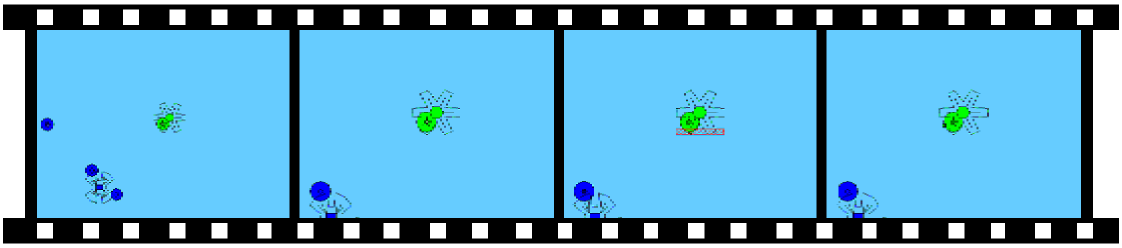

Figure 6: CineADD in progress.

6.5. CineADD in ADDSub

As we explained on section 5, CineADD is based on Replay and Rebuild approaches. In ADDSub, the Replay approach is built based on the actions and decisions stored in the design log. CineADD creates scenes for each kind of action, like data entry and creation of new elements. For the subsequent actions like moving an element or suggesting new alternatives, the system creates takes. After looking over the entire design log, the system generates all the sequence of scenes and takes that compose the Replay approach. This movie recomposes all the sequences of actions done during the project development.

The Rebuild approach is built after the interpretation of the observer question, based on a list of techniques/idioms configured by the designer (or by default if none is specified). This feature allows designers to create the Rebuild script, directing the design movie. For each decision type or even a specific decision, the designer can eliminate or create scenes, choose camera effects, include different perspectives of a scene, include texts on special boxes, include written frames and select the gluing effects between scenes.

The techniques available in CineADD to create the Rebuild script of an ADDSub project were chosen based on the characteristics of the specific domain studied. Oil pipeline projects usually involve a great number of graphically independent elements (e.g., platforms, wells, pipelines) present on a 2D canvas area, where most of the actions taken on an element affect only a small part of the project. Therefore, CineADD uses techniques that are able to highlight a single element of the project; make easy to visualize the actions taken on an element; make easy to focus on details; and show up the relevant actions that occurred on a set of elements. The set of the techniques used are divided into three groups:

- Cinema:
    - Camera movement (zoom and pan) – to emerge small details and locate in the canvas elements that must be focused;
    - Narrative (reordering, visibility, time of a frame) – to reorder the sequence of actions (e.g. show the platform positioning before the well positioning), make some takes visible or invisible, and define the time that a frame must

    appear (a frame that contains constraints violation must stay longer than others, as the observer have to realize which constraint was violated);
  - Style (silent movie) – to draw attention to transitions between scenes, explaining briefly in text what will happen next;
  - Edition (abrupt cut) – to cut some intermediaries frames with no special scenes´ merge;
- Animation:
  - Pointer and Blinking – to highlight the elements that are the focus of the attention;
- Text:
  - Insertion of text – to complement the animation, giving additional information for some specific actions (e.g., explain which constraint is violated on the well positioning);

7. Conclusions

Storytelling is a very efficient way to explain a fact to an audience [Gershon 01] [Bers & Cassell 97]. It is much more effective to give the audience an interpretation of the fact than give them all the facts and let them make sense of it. Delivering an interpretation lowers the audience cognitive load. In addition, the audience can be conducted to a specific viewpoint.

Cinema is an interesting communication medium to tell a story, and consequently send a message. This medium, when properly used, may permit long and complex messages be transformed into simple ones (as for example, The Wuthering Heights by Emily Bronte or Anna Karenina by Tolstoi).

This paper presented a model that includes design cinematic techniques to improve computer user communication for the task of design rationale delivery. Because design documentation is developed as a design task subproduct in an ADD environment, the boring, time-consuming and repetitive aspects of it can be substituted by a creative process, generating new perspectives for the problem. CineADD enhances traditional interaction by providing users an immerse experience afforded by a cinematic environment. CineADD applies cinema and animation techniques to arrange and compose a visual presentation (a movie) working as a complement to the textual explanation generated by the system.

CineADD was implemented in an intelligent CAD system, called ADDSub, used in a real design domain (oil pipeline layout). As this layout design is mainly a visual task and involves locating and changing the locations of elements in a 2D area, the main contribution of CineADD in ADDSub is to provide an enhanced explanation interface, allowing designers to configure the way a visual explanation should be created. Designers play filmmakers in CineADD. They create Idioms that define the explanation narrative, such as scene selection, scene sequencing, scene effects, and scene merge, to be created to explain each type of decision in a project.

Users may doubt an explanation and may pursue a further investigation by creating new Idioms for playing the scenes. This action may lead to reveal new perspectives in the designers' narrative. However, this functionality may lead to more misconceptions and should require a deeper HCI study. Initial results, using CineADD to deliver design rationale in the oil pipeline design domain, have shown that designers have some difficulties in learning how to build their movie; however, the increase of understanding by the design team or even the end-users is worth the learning.

CineADD was meant for designers to build design movies. However, since the design scenes are available and the movie builder allows a story to be easily constructed, end-users can also take the director's role and uncover details or interpretation that might be hidden by the designers. Therefore, CineADD allows interactive cinema to be pursued in the context of engineering design.

References

[ADDSub 98] – "Documentação do Projeto ADDSub" – Laboratório ADDLabs, Niterói, RJ, Brazil – 1998.

[Arijon 76] – "Grammar of the Film Language" – Daniel Arijon – New York: Communication Arts Books, Hasting House, Publishers – 1976.

[Bers & Cassell 97] - "Storytelling Systems: Constructing the Innerface of the Interface"- Bers, M. and Cassell, J. In proceedings of IEEE Cognitive Technologies Procedings, CT'97 (Second International Conference on Cognitive Technlogy), pp. 98-108, Los Alamitos, CA – 1997.

[Blinn 94] – "Animation Tricks" – Jim Blinn – SIGGRAPH '94 – Course Notes for Animation Tricks course – May, 22, 1994.

[Brooks 97] – "Programming Narrative" – Kevin M. Brooks – In VL'97 – IEEE Symposium on Visual Languages – Capri, Italy – September 23-26, 1997.

[Christianson et al. 96] – "Declarative Camera Control for Automatic Cinematography" – David B. Christianson, Sean E. Anderson, Li-wei He, David H. Salesin, Daniel S. Weld, and Michael F. Cohen – In Proceedings of the AAAI-96, August, 1996.

[Davenport et al. 91] – "Cinematic Primitives for Multimedia" – Davenport, G., Smith, A. and Pincever, N – In proceedings of IEEE Computer Graphics and Animation special issue on multimedia, pp. 67-74 – July 1991.

[Davenport 96] – "Smarter Tools for Storytelling Are They Just Around the Corner?" – Glorianna Davenport – In Visions and Views – IEEE Multimedia, vol. 4, no. 1, pp 10-14 – 1996.

[Dufaux & Moscheni 96] – "Segmentation-based motion estimation for second generation video coding techniques" – Dufaux, F. and Moscheni, F – In Video

Coding: The Second Generation Approach, Torres, L. and Kunt, M. (eds.), Kluwer Academic Publishers, pp. 219-263 – 1996.

[Fischer & Lemke et al. 91] – “Making Argumentation Serve Design” – G. Fischer, A. C. Lemke, R. McCall, A. I. Morch – In Human-Computer Interaction, 6(3/4) 1991.

[Garcia 92] – “Active Design Documents: A new Aproach for supporting Documentation in Preliminary Routine Design” – Ana Cristina Garcia – Ph.D. thesis, civil engineering department – Stanford, Ca. – 1992.

[Garcia et al. 97] – “ADDVAC: Applying Active Design Documents for the Capture, Retrieval and Use of Rationale During Offshore Platform VAC Design” – Ana Cristina Garcia, Joper C. de Andrade, Rogério Ferreira e Ricardo de Moura – IAAI-97 Emerging Applications – 1997.

[Gershon 01] – “What Storytelling Can Do for Information Visualization” – Nahum Gershon – In Communications of the ACM, vol. 44, no. 8, August, pp 31 – 37 - 2001

[He & Cohen et al. 96] – “The Virtual Cinematographer: A Paradigm for Automatic Real-Time Câmera Control and Directing” – L. wei He, M. Cohen and D. Salesin – In Proceedings of ACM SIGGRAPH ’96, pp 217-224 – 1996.

[Katz 91] – “Film Directing – shot by shot” – Steven D. Katz – Michael Wiese Productions, 1991.

[Lachman 97] – “Experiments in Mapping Character Animation to computer Interface” – Lachman, R. – In proceedings of IJCAI Workshop on Animated Interface Agents. Nagoya, Japan – August 1997.

[Lester & Bares 97a] – “Cinematographic User Model for Automated Realtime Câmera Control in Dynamic 3D Enviroments” – James C. Lester and William H. Bares – In UM-97 - Proceedings of the Sixth International Conference on User Modeling, pp 215-226 – 1997.

[Lester & Bares 97b] – “Realtime Generation of Customized 3D Animated Explanations for Knowledge-Based Learning Environments” – James C. Lester and William H. Bares – In Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97) – 1997.

[McReynolds & Blythe 98] – “Advanced Graphics Programming Techniques Using OpenGL” – McReynolds, T. and Blythe, D. – In SIGGRAPH `98 Course – 1998.

[Mann & Thompson 87] – “Rhetorical Structure Theory: A Theory of Text Organization” – W. C. Mann and S. A. Thompson – Report ISI/RS-87-190, University of Southern California, Marina del Rey, CA – 1987.

[Moran & Carroll 96] – “Design Rationale – Concepts, Techniques and Use” – Thomas P. Moran & John M. Carroll – LEA Publishers – New Jersey – 1996.

[Persson 99] – “Understanding Representations of Space: a Comparison of Visualisation Techniques in Mainstream Cinema and Computer Interfaces” – Persson, P. In *Social Navigation in Information Space*, Munro, A., Höök, K. & Benyon, D. (eds.), London: Springer, pp. 195-216. – 1999.

[Persson 01] – "Cinema and Computers: Spatial Practices Within Emergent Visual Technologies" – Persson, P. In *Technospaces*, Munt, S. (ed.), London: Continuum – 2001.

[Silverstein & Huss 68] – "The Film Experience" – N. Silverstein & R. Huss – Dell Publishing – 1968.

[ten Hagen 87] – "Intelligent CAD System I" – ten Hagen, P. J. W. ed. – European Association for Computer Graphics Press – 1987.

[Thomas & Johnston 84] – "Disney animation: the illusion of life" – Frank Thomas and Ollie Johnston – Abbeville Press, New York – 1984.

[Vivacqua & Garcia 96] – "The Use of Active Design Documents to Assist Conflict Mitigation in Concurrent Engineering" – A. S. Vivacqua and Ana Cristina Garcia – 3rd Conference on Concurrent Engineering and Research Applications – 1996.