ΜΟΝΤΕΛΟ ΚΑΙ ΣΥΣΤΗΜΑ ΓΙΑ ΤΗ ΚΑΤΑΓΡΑΦΗ ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΒΙΝΤΕΟ ΣΕ ΕΦΑΡΜΟΓΕΣ ΠΛΗΡΟΦΟΡΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Παντελή Αμαλία

Πολυτεχνείο Κρήτης

Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

Μια εργασία που παρουσιάστηκε στο Πολυτεχνείο Κρήτης για την εκπλήρωση των απαιτήσεων απόκτησης Διπλώματος στο τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών

Χανιά, 2010

MOBILE MULTIMEDIA EVENT CAPTURING AND VISUALIZATION

Panteli Amalia

Technical University of Crete

Department of Electronics and Computer Engineering

A thesis submitted in fulfillment of the requirements for the degree of

Diploma of Science in Electronic and Computer Engineering

Chania, 2010

ACKNOWLEDGEMENTS

ABSTRACT

The fast and immense progress of the information technology, has made the multimedia a necessary aspect of everyone everyday life.

Multimedia finds its application in various areas including, but not limited to, advertisements, art, education, entertainment, engineering, medicine, mathematics, business, scientific research and spatial temporal applications.

Capturing Multimedia Events such as economic events, accident reports, building damage reports, etc., are expensive functionalities due to the number and training of the people required, as well as the time involved in the capturing and post-processing of multimedia.

We present a model and a mobile system for multimedia event capturing by a one-man-crew. The system supports the real time capturing of complex multimedia events of different types, the recording of the capturing process and the metadata associated with the events, and the learning and preparation of the one-man-crew that will do the multimedia event capturing.

Today all the multimedia types are recorded in a digital format. With each event metadata values of different types will have to be captured and associated with the events. The one-man-crew will also be responsible for capturing and recording the metadata that are associated with the event. In some cases the beginning and the end of the event to be recorded is not known in advance. In this case the one-man-crew will have to demarcate within a longer capturing sequence the beginning and the end of the event, and associate the relevant metadata values with it in order to be able to do fast post-processing (manually or automatically). Thus the multimedia capturing environment will have to be controlled and synchronized by a mobile computer (laptop or other) which will also provide functionality for demarcating the beginning and the end of events and associate the appropriate metadata values with each event.

The first objective of this thesis is to facilitate the one-man-crew to do a complete and systematic capturing of the multimedia events of different types that compose a bigger event in real time. A second objective is to allow fast browsing and extraction of all the multimedia event information either in real time or at post-processing. A third objective is to allow the one-man-crew to be prepared through learning and reviewing for a good quality multimedia event capturing by means of event type specific learning instructions.

Multimedia events in our model belong to complex event types that may be composed of simpler multimedia event types. The metadata captured for each event depend on its type. Events are captured by multimedia shots which belong to shot types. Events and multimedia shots may be associated with maps and/or diagrams. Shots are described by the position (captured by a GPS device or otherwise), direction (compass) and camera movement of the shot placed on a diagram of a scene that describes the event, as well as with other parameters. The description of shot types can be used for learning and preparation of the one-man-crew for multimedia capturing of events of a particular type.

We describe the multimedia event capturing model, a prototype system implemented that supports the functionality of the model and two example applications that were materialized to demonstrate the functionality of the system.

CONTENTS

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1. INTRODUCTION

Many important practical applications of multimedia are associated with events or they can be described by events. For example reporting an accident, recording the events of a marriage or a baptism, recording an invited talk a debate or a lecture, recording political or economic events for presentation on TV or for video on demand consumption, recording an experiment on a lab, recording the presentation of a product, producing evaluation reports for products of various types, recording damages of a building or on disaster areas, or even systematically recording the components of a complex 3D structure such as a multi-store building may be modeled with multimedia events.

Typically a multimedia event is recorded using multiple media such as video or pictures and audio; it is associated with some spatial information described with maps or diagrams and also has some associated metadata which are used for further processing or filtering of the multimedia event. For example, in an accident event the multiple media may capture the scene of the accident (cars, people involved, etc.), spatial information such as diagrams may record the location on the road of the cars involved and the metadata may be used to record day and time as well as the names of the people involved or interviewed about the accident.

The capturing of events with multimedia is an expensive endeavor. For example TV stations utilize multiple crew members to carry, install and operate the video, audio and lighting equipment that is used for multimedia capturing. Even for smaller provincial TV stations it is not economically feasible to maintain adequately trained personnel in all the different expertise needed and it is certainly not economically feasible for small multimedia companies or for companies that only have the multimedia information management integrated within their larger business activities. Ideally multimedia event capturing should only involve one person. We call this requirement the “one-man-crew” requirement. Certainly the “one-man-crew” should posses’ expertise from several different areas of multimedia capturing like audio, video, lights, scene composition, etc. He should also be able to work fast in real time for capturing all the multimedia events and the associated metadata.

Today all the multimedia types are recorded in a digital format. With each event metadata values of different types will have to be captured and associated with the events. The one-man-crew will also be responsible for capturing and recording the metadata that are associated with the event. In some cases the beginning and the end of the event to be recorded is not known in advance. In this case the one-man-crew will have to demarcate within a longer capturing sequence the beginning and the end of the event, and associate the relevant metadata values with it in order to be able to do fast post-processing (manually or automatically). Thus the multimedia capturing environment will have to be controlled and synchronized by a mobile computer (laptop or other) which will also provide functionality for demarcating the beginning and the end of events and associate the appropriate metadata values with each event.

We present here a model and a system for multimedia event capturing by a one-man-crew. The first objective is to facilitate the one-man-crew to do a complete and systematic capturing of the multimedia events of different types that compose a bigger event in real time. A second objective is to allow fast browsing and extraction of all the multimedia event information either in real time or at post-processing. A third objective is to allow the one-man-crew to be prepared through learning and reviewing for a good quality multimedia event capturing by means of event type specific learning instructions.

Multimedia events in our model belong to complex event types that may be composed of simpler multimedia event types. The metadata captured for each event depend on its type. Events are captured by multimedia shots which belong to shot types. Events and multimedia shots may be associated with maps and/or diagrams. Shots are described by the position (captured by a GPS device or otherwise), direction (compass) and camera movement of the shot placed on a diagram of a scene that describes the event, as well as with other parameters. The description of shot types can be used for learning and preparation of the one-man-crew for multimedia capturing of events of a particular type.

We describe the multimedia event capturing model, a prototype system implemented that supports the functionality of the model and two example applications that were materialized to demonstrate the functionality of the system.

2. OVERVIEW

In the current thesis we created a conceptual model with which someone can capture multimedia information and its associated metadata in a systematic manner. Using this model a user can capture the multimedia information he desires and in the same time, capture metadata along with the multimedia, which can be used to describe the multimedia.

We are interested in mobile multimedia applications and in particular in the capturing of multimedia event information in mobile settings. Very frequently this capturing also requires the capturing of information related to the space where various events take or took place.

We present a core ontology which the user can use to create different kinds or types of multimedia application scenarios. These scenarios can then be used in the capturing of the multimedia, helping the user to capture the proper metadata needed for every different scenario.

The conceptual model we created is quite general and flexible and can represent a wide range of multimedia needed to be captured. It also facilitates the capturing of not only indoor spaces but also outdoor. In addition the information kept for the site map of every shot can be stored with many different ways.

We created an application to explore the mobile multimedia capturing model that we designed. The system functionality provided allow users that are not expert cinematographers to capture video and metadata combined with the video. The application gives the user the specific directions needed in order to capture the proper metadata. Furthermore, the application does not only give the option to capture video, but to also represent the captured video along with the relevant metadata in a database.

As mentioned above, the multimedia has applications in many areas. In every different area, the metadata needed to be captured vary. The application created, focuses on applications that are event driven as well as applications that focuses on the spatial information like architecture applications and in general spatial temporal applications.

3. THESIS STRUCTURE

In Chapter 2, we will describe the related work done in the contents of the thesis. In particular we will describe the related research in the field of cinematography as well as the technical parameters that were researched in the current thesis, and the related technology that we used.

In Chapter 3, we will describe and try to analyze the Mobile Multimedia Event Capturing Model that was created in the context of this thesis and is used to store the multimedia information along with its associated metadata.

In Chapter 4, we will describe the implementation model created in the contexts of this thesis. There, we will first analyze the core ontology created based on the Mobile Multimedia Event Capturing Model, then we will present a ontology – scenario and we will demonstrate an example of multimedia application. Finally the relational database created for the physical storage of the multimedia and its metadata will be described.

Chapter 5 will give the requirements analysis of the application which we created using use cases. Chapter 6 will describe the reference architecture along with the application model.

Chapter 7 will describe the Mobile Multimedia Event Capturing (MOME) application that was also created in the context of the current thesis and tries to give a practical application of the model. Finally, chapter 8 will suggest future work that can be done in the same field.

4. SUMMARY

In this chapter, we made an introduction to our work by providing the objectives this thesis. After that, we described in an overview what is going to follow in the thesis and finally we described the structure of the thesis.

CHAPTER 2
RELATED WORK

1. INTRODUCTION

In this chapter, we present a review of the related research that has been done. In particular we will describe concepts related to standards, camera capturing parameters, video shooting definitions and guidelines and other application environments relevant to the one implemented in the contents of the current thesis.

2. RELATED RESEARCH

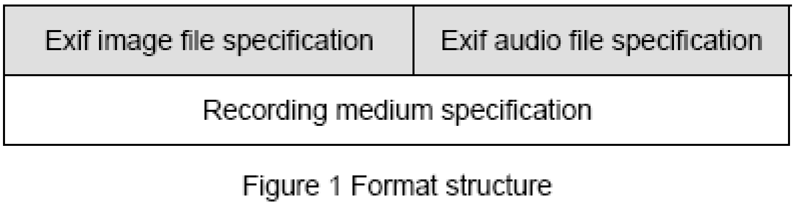
2.1. THE EXIF STANDARD

During video capturing, we need to hold information about the spatial context of the capturing location and so we need to know many of the camera's technical parameters such as the horizontal and vertical angle of view, the focal length, etc.. Thus we use the EXIF standard used in digital photography.

The EXIF (Exchangeable image file format) standard specifies the formats to be used for images, sound and tags in digital still cameras and in other systems handling the image and sound files recorded by digital still cameras (2).

- *Format Structure*

This standard consists of the Exif image file specification and the Exif audio file specification (see Figure 1). The recording medium specification shall conform to the standard for the particular medium used.



Exif image file specification	Exif audio file specification
Recording medium specification	

Figure 1 Format structure

Figure 2.1 Format Structure

- Exif Image File Specification

The Exif image file specification stipulates the method of recording image data in files, and specifies the following items:

- Structure of image data files,
- Tags used by this standard,
- Definition and management of format versions.

Features of the Exif image file specification include the following.

The file-recording format is based on existing formats. Compressed files are recorded as JPEG (ISO/IEC 10918-1) with application marker segments (APP1 and APP2) inserted. Uncompressed files are recorded in TIFF Rev. 6.0 format. The use of existing formats means files recorded using a DSC or related system can be read directly by commercial applications, and makes possible the use of functions for viewing and manipulating the images. Related attribute information for both compressed and uncompressed files is stored in the tag information format defined in TIFF Rev. 6.0. Information specific to the camera system and not defined in TIFF is stored in private tags registered for Exif. The Exif image file specification also specifies the method for recording thumbnails. The reason for using the TIFF Rev. 6.0 tag format in the compressed file APP1 segment is to facilitate exchange of attribute data between Exif compressed and uncompressed files. Compressed files can record extended data exceeding 64 Kbytes by dividing it into multiple APP2 segments. The APP2 segment is used when recording Flashpix extensions. Details of the Exif image file specification are given in Chapter 4 of this document.

- ### Exif Audio File Specification

The Exif audio file specification stipulates the method of writing audio data in files, and specifies the following items:

- Structure of audio data files,
- Chunks,
- Definition and management of format versions.

Features of the Exif audio file specification include the following.

Files are recorded using the existing RIFF WAVE Form Audio File format. As data formats, PCM and μ-Law PCM (conforming to ITU-T G.711) are used for uncompressed audio data, and IMA-ADPCM for compressed audio data. As with the image files specification, the aim is to enable direct reading, playback and manipulation of the files using the functions in commercial applications. Related attribute information is stored as an INFO list in the case of general attributes, and as Exif-specific chunks in the case of attributes specific to Exif. The Exif-specific chunks are newly provided to enable necessary information to be added by making use of chunk extensibility. The recording of data in an Exif-specific chunk is done in the same way as for an INFO list.

2.2. TECHNICAL PARAMETERS DESCRIPTION

The technical parameters that are used for the description of the camera condition during the capturing are the following:

- Camera manufacturer and model

The company that manufactured and the specific model of the camera used for the capturing.

- Horizontal and Vertical angle of view

It is the extent of the view taken in by a lens. It is determined by the focal length of a lens and film format. Angle of view may be measured horizontally (from the left to right edge of the frame), vertically (from the top to bottom of the frame), or diagonally (from one corner of the frame to its opposite corner). A “standard” 50mm lens for 35mm film has an angle of view equal to the diagonal of the film, which is 70° horizontally and 58° vertically. A 135mm "short telephoto" lens has a reduced angle of view of 29° horizontally and 23° vertically. (3)

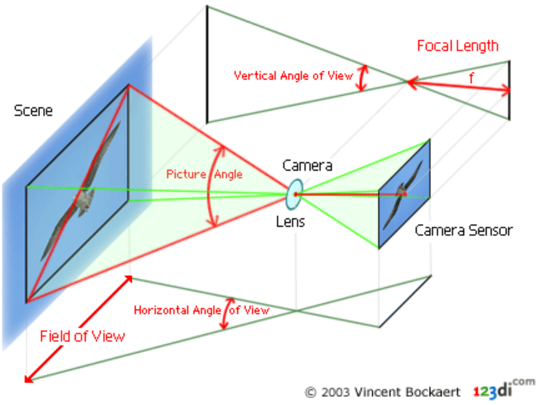


Figure 2.2 Angles of View

For a lens projecting a rectilinear image, the angle of view (α) can be calculated from the chosen dimension (d), and effective focal length (f) as follows:

$$\alpha = 2 \times arctan\left(\frac{d}{2f}\right)$$

Where d represents the size of the film (or sensor) in the direction measured. For example, for film that is 36 mm wide, d = 36 mm would be used to obtain the horizontal angle of view.

Because this is a trigonometric function, the angle of view does not vary quite linearly with the reciprocal of the focal length. However, except for wide-angle lenses, it is reasonable to approximate $\alpha \approx \frac{d}{f}$ radians or $\frac{180d}{\pi f}$ degrees.

The effective focal length is nearly equal to the stated focal length of the lens (F), except in macro photography where the lens-to-object distance is comparable to the focal length. In this case, the magnification factor (m) must be taken into account:

$$f = F \cdot (1 + m)$$

(In photography m is usually defined to be positive, despite the inverted image.) For example, with a magnification ratio of 1:2, we find $f = 1.5 \cdot F$ and thus the angle of view is reduced by 33% compared to focusing on a distant object with the same lens. (4)

Angle of view can also be determined using FOV tables or paper or software lens calculators

○ Focal length

The focal length of a lens is defined as the distance in mm from the optical center of the lens to the focal point, which is located on the sensor or film if the subject (at infinity) is "in focus". The camera lens projects part of the scene onto the film or sensor. The field of view (FOV) is determined by the angle of view from the lens out to the scene and can be measured horizontally or vertically. Larger sensors or films have wider FOVs and can capture more of the scene. The FOV associated with a focal length is usually based on the 35mm film photography, given the popularity of this format over other formats. (5)

○ Hyperfocal distance

The hyperfocal distance of a lens is the distance from the optical center of the lens to the nearest point in acceptably sharp focus when the lens, at a given f/stop, is focused at infinity. In other words, when a lens is focused at infinity, the distance from the lens beyond which all objects are rendered in acceptably sharp focus is the hyperfocal distance. For example, when a 155mm lens is set at f/2.8 and focused at infinity, objects from 572 feet to infinity are in acceptably sharp focus. The hyperfocal distance therefore is 572 feet. The following equation is used to find hyperfocal distance:

$$H = \frac{F^2}{f \times c}$$

Where H is hyperfocal distance, F is focal length of lens; f is the f/stop setting, C is the diameter of circle of confusion. F and C must be in the same units, inches, millimeters, and so forth.

Hyperfocal distance depends on the focal length of the lens, the f/stop being used, and the permissible circle of confusion. Hyperfocal distance is needed to use the maximum depth of field of a lens. To find the depth of field, you must first determine the hyperfocal distance. By focusing a lens at its hyperfocal distance, you cause the depth of field to be about one half of the hyperfocal distance to infinity. (6)

○ Circle of Confusion (coc)

A picture is basically an accumulation of many points that are exact images of points composing the subject. After light strikes a subject, it is reflected from many points on the subject. A camera lens redirects these reflected rays into corresponding points on the film. Each of these points is reproduced by the lens as a circle. When the circle is smaller than l/100 inch, it appears as a sharp point to the eye. When the circle is larger than 1/100 inch, the eye sees it as a circle, and the image is blurred or out of focus. Each out-of-focus circle on the film is called a circle of confusion and can be visualized as the cross section of a cone of a light ray. (7)

- F-number

F-numbers are numbers on the outside of the lens corresponding to the aperture opening. The larger the number is (e.g., f/22 also expressed as F 22), the smaller the opening of the lens; the smaller the number (e.g., f/2.8) the larger the opening of the lens (8). In particular, they are the numbers on the lens which is equivalent to the focal length divided by the diameter of the aperture. (9)

- GPS

Latitude (abbreviation: Lat. or (ϕ) pronounced phi) is the angle from a point on the Earth's surface and the equatorial plane, measured from the centre of the sphere. Lines joining points of the same latitude are called parallels, which trace concentric circles on the surface of the Earth, parallel to the equator. The North Pole is 90° N; the South Pole is 90° S. The 0° parallel of latitude is designated the equator. The equator is the fundamental plane of all geographic coordinate systems. The equator divides the globe into Northern and Southern Hemispheres.

Longitude (abbreviation: Long. or (λ) pronounced lambda) is the angle east or west of a reference meridian between the two geographical poles to another meridian that passes through an arbitrary point. All meridians are halves of great circles, and are not parallel. They converge at the north and south poles.

A line passing near the Royal Observatory, Greenwich (near London in the UK) has been chosen as the international zero-longitude reference line, the Prime Meridian. Places to the east are in the eastern hemisphere, and places to the west are in the western hemisphere. The antipodal meridian of Greenwich is both 180°W and 180°E. The choice of Greenwich is arbitrary, and in other cultures and times in history other locations have been used as the prime meridian.

By combining these two angles, the horizontal position of any location on Earth can be specified.

To completely specify a location of a topographical feature on, in, or above the Earth, one has to also specify the vertical distance from the centre of the sphere, or from the surface of the sphere. Because of the ambiguity of "surface" and "vertical", it is more commonly expressed relative to a more precisely defined vertical datum such as mean sea level at a named point. Each country has defined its own datum. In the United Kingdom the reference point is Newlyn. The distance to the Earth's centre can be used both for very deep positions and for positions in space. (10)

2.3. VIDEO SHOOTING DEFINITIONS

A shot is a section of a continuous, uncut footage and it's the basic of video work. The three basic shots are the close-up, the mid-shot and the long-shot and the long shot – CU, MS and LS in the abbreviations used in the television and film industry. They are defined in terms of human subjects, although of course they apply to shots of landscapes and inanimate objects too.

A close-up is a head and shoulders shot of a single individual. When only the full face occupies the screen, the shot becomes a big close-up (BCU). Any closer than this (a shot of the eyes only or mouth only, for instance) and is called extreme close-up (ECU). Each gives what the audience perceives as an increasingly intimate view of the subject.

The mid-shot extend to just below the waist (notice *not* to the waist – cut-off points that correspond to human sections look odd on the television). Where a shot contains two people it is a two-shot. The mid-shot concentrate on the subject, but include a sense of background.

The long-shot contains the full human figure, from head to toe. It is also tells the viewer where the subject is and how he or she relates to the environment. The long-shot can extend to become a very long-shot (VLS) and an extreme long-shot (ELS) (11).

- Camera Movements

Before learning to operate a camera, a user should become familiar with the most common camera movements. "Left" and "Right" always refer to the camera's point of view. The camera mounting equipment has been designed solely to help the user move the camera smoothly and efficiently in various ways. The major camera movements are pan, tilt, pedestal, tongue, crane or boom, dolly, truck or track, crab, arc and zoom.

Pan is turning the camera horizontally from left to right or from right to left. Tilt is making the camera point down or up. A "tilt up" means that the camera is made to point up gradually. A "tilt down" means that the camera is made to point down gradually. Pedestal is elevating or lowering the camera on a studio pedestal. Tongue is moving the whole camera from left to right or from right to left with the boom of a camera crane. When the user tongue left or right, the camera usually points into the same general direction, with only the boom moving left (counterclockwise) or right (clockwise). Crane or boom is moving the whole camera up or down on a camera crane. The effect is somewhat similar to pedestaling up or down, except that the camera swoops over a much greater vertical distance. Dolly is moving the camera toward or away from an object in more or less a straight line by means of a mobile camera mount. When the user "dolly in" he moves the camera closer to the object, when he "dolly out, or dolly back", he moves the camera farther away from the object. Truck or track is moving the camera laterally by means of a mobile camera mount. To "truck left" means to move the camera mount to the left with the camera pointing at a right angle to the direction of the travel. To "truck right" means to move the camera mount to the right with the camera pointing at a right angle to the direction of the travel. Crab is any sideways motion of the crane dolly base or its similar cousin, the crab dolly. A crab is similar to a truck, except that the camera mount does not have to stay lateral to the action all the time; it can move toward or away from the action at the same time. “Crabbing” is used more in filming than in television. Arc is moving the camera in a slightly curved dolly or truck movement with a mobile camera mount. Zoom is changing the focal length of the lens through the use of a zoom control while the camera remains stationary. To “zoom in” means to change the lens gradually to a narrow-angle position, thereby making the scene appear to move closer to the viewer; to “zoom out” means to change the lens gradually to a wide-angle lens position, thereby making the scene appear to move further away from the viewer. Although not a camera movement, the zoom effect looks similar to that of moving camera and is, therefore, classified as such (12).

- The RULE OF THIRDS.

This rule has two different applications

Consider a grid on the viewfinder three rows across and three rows down - you will use the tic-tac-toe grid to align your pictures.

top 1/3 L	top 1/3 C	top 1/3 R
C 1/3 L	C 1/3 C	C 1/3 R
BTM 1/3 L	BTM 1/3 C	BTM 1/3 R

When the shooting involves people, their eyes should be at the top 1/3 line. When the shooting involves landscapes with a wide shot, the horizon line should be on either the top 1/3 line or the bottom 1/3 line.

Now the up and down lines –If the shot involves a standing figure of a person or just a head and shoulder - the person may be placed to the side on either of the vertical lines. This will leave approximately 2/3 of the screen for a pretty or interesting picture. These lines will also be used to place moving figures or objects, give them the big area to run toward. This is called leading the action - for example, at a football game the runner is on one side of the screen running into the open part of the screen. This idea will hold true with all action shots.

The center area - C 1/3 C - will only be used when a detail of some object is being illustrated - most of the time a perfect centering is just not interesting. (13)

2.4. VIDEO SHOOTING GUIDELINES

For professional quality capturing of multimedia it is important that one has access to guidelines that can help him make a better video. In this section, we present generic principles and guidelines, along with an example of, a more specific situation, suggestions for video shooting.

To start, according to the Dr. Ron Whittaker (14) these are the twelve guidelines for better video:

- Use a tripod or a solid camera support. The use of a tripod or a solid camera support is the mark of a professional. This is especially important in close-ups. The exception is where you want to show a subjective camera effect, communicate a fluid or unstable situation, impart a documentary-style effect, or in news situations where you will miss the shot if you try to use a tripod.
- Rely on medium close-ups and close-ups for your basic visual material. Wide shots should only be used for establishing (and reestablishing) shots. HDTV doesn't require this same close-up emphasis, but for some time we'll have to shoot with both formats in mind.
- Eliminate shots that don't contribute to the project's goals or your basic story idea. The rule here is: If in doubt, leave it out!
- Cut away from a shot as soon as the basic information is conveyed, especially if the shot is a static one. Almost all of the student videos I see could be judiciously cut by at least 50% and be much improved in the process.
- Resist the temptation to keep the camera rolling, and pan, zoom and tilt the camera to get from one shot to another. Zooms and pans are generally just lazy and time-consuming ways of changing shots. A cut is almost always stronger and faster. Use pans and tilts when you need to reveal something or when you need to follow subject movement. True, we see a lot of zooms, pans and tilts in videos, but take a look at a good feature-length film — especially one that has won an award for cinematography. You won't often see many of these.
- Make sure your key subject matter (the talent) is not wearing white, or is against a white (or very light) background. The sky, windows, bright walls and lights in the picture are the biggest problem. The result is gray scale compression or white clipping. If you can't avoid this, you can manually open the camera's iris or engage the camera's "backlight" switch and carefully observe the effect while you make adjustments.
- Unless you are "editing in the camera," make sure you observe a five-second roll cue at the beginning of each take. Otherwise, especially considering the pre-roll requirements for many videotape editors, you may find it impossible to use the segment during editing.
- Cue up your piece to the very beginning of a ten-second countdown leader before submitting your work.
- Use an auxiliary mic for interviews, never the built-in camera mic. Use the mic as close to the subject as possible. If you don't want the mic to be conspicuous, use a

clip-on or personal mic, hide the handheld mic close to the subject, or use an off-camera directional mic.

- Select instrumental music as background for narration, not vocal, rap, or hip-hop music. You can't have two voice tracks going at the same time and expect the audience to follow both.
- Use B-roll footage with interviews whenever possible. Don't just hold a shot of a "talking head" unless the person is very dramatic or animated. Whenever possible supplement the interview footage with shots that help explain or illustrate what's being said.
- Completely and thoroughly think through and plan your piece before you start. Remember: The most important phase of production is preproduction. Plan for visual and audio variety and only include shots that are essential to getting your point across. Keep in mind the emotional element in production content.

For more unofficial home videos, these are ten essential guidelines for shooting quality videos (32). These guidelines do not refer to complex Cinematography or director skill, but they give fundamental knowledge about what goes into a good shot and what does not.

- Avoid Zooming In and Out Too Much. Many digital camcorders come with a super duper 1000X zoom in and zoom out feature. This is a good thing, but the problem is, many people get carried away. I've seen countless videos which keep zooming in and out during the shoot - they generally succeed in giving me a bad headache. Use the zoom in and out feature sparingly. Even if you must use it, do it slowly. A slow, well-controlled zoom is much more professional looking than a quick zoom. Another tip is to restrict the usage of the zoom in between scenes.
- Steady Does It. The professionals always say "Keep It Steady". There is no doubt that when we're shooting videos, our hands tend to vibrate a little. If the vibration is too much, it will badly affect the quality of the video footage. There are two ways to overcome this. One way is to do it like the professionals - get a good tripod stand. These can be obtained rather cheaply. Another way is to brace yourself against something like a wall, or perhaps stoop down on your knee while filming.
- Where Are You? One tip that I've picked up while shooting family vacations is to always, always look for a landmark, a sign or natural monument that tells the audience where you are. Do you notice the professionals do this as well? For example, in the movies, you may see the camera zooming in on the Statue of Liberty first, before cutting to a scene that happens within. Or there is a shot of the White House before the director cuts into a scene within the Oval Room. Another tip is to make the people you're filming talk or smile. This makes the whole video much more lively and interesting.
- The End or the Beginning? Another good video shooting trick is to use the end of the story at the start of the video. Again, experts do this all the time. If you're filming a wedding video, you can start off with an interview of the happily married couple on the wedding day before you cut off into scenes of how they met, how they grew up, etc.
- Keep Them Short and Simple. Never, ever, shoot long, drawn out video clips of more than two minutes. It bores the audience to death! Using many short clips of 5 to 10

seconds duration is much more effective. It also brings variety to the audience. Imagine if a television commercial lasted two minutes focusing on the same video scene - wouldn't that be boring?

- Lighting, Lighting, Lighting. Of all the factors listed here, nothing spoils a video shoot as much as poor lighting. I've seen many innovative videos marred by poor lighting. One important tip is to shoot video with the light source behind you, shining on the subject. An example is an outdoor shot where the sun is shining. Make sure the sun shines on your subject and not on your camera lens! Another tip is to use the camera's backlight feature if there is one. It allows you to compensate for overly bright light.
- Change Your Perspective. Don't limit your shots to one angle only. Approach your subject from all angles. Come from behind, come from the top, whatever. Perhaps you can even film yourself walking up the steps and opening a bedroom door before reaching your subject.
- Learn from The Movies. You'd be surprised how much you can learn about shooting video from the movies. Very often, we just sit down and let the movie scenes come at us without considering what went through the director's mind. Try it - for once, when you watch your next movie, consider how the director framed that shot or scene. You will learn a lot and maybe even get interested in directing films!
- Conserve the Battery. I've seen or heard so many people forgetting about the battery that I must emphasize it here. Please buy extra batteries! Preferably two extra batteries. Nothing is worse than shooting a video outdoors and having the battery die on you. Also, always bring the AC adapter too - so that you can charge your current battery whenever time permits.
- Check The Sound. This tip may not be important for the average home user. But if you're a serious videographer, you'll have a microphone attached to your camcorder. Microphones are an essential tool for sound focusing and result in better audio quality during video shoots.

2.5. MAPS, PLANS AND ELEVATIONS

Map

A map projection (15) is any method of representing the surface of a sphere or other shape on a plane. Map projections are necessary for creating maps. All map projections distort the surface in some fashion. Depending on the purpose of the map, some distortions are acceptable and others are not; therefore different map projections exist in order to preserve some properties of the sphere-like body at the expense of other properties. There is no limit to the number of possible map projections.

Plan

Plans are a set of two-dimensional diagrams or drawings used to describe a place or object, or to communicate building or fabrication instructions. Usually plans are drawn or printed on paper, but they can take the form of a digital file (16).

A plan view is an orthographic projection of a 3-dimensional object from the position of a horizontal plane through the object. In other words, a plan is a section viewed from the top. In such views, the portion of the object in above the plane is omitted to reveal what lies beyond. In the case of a floor plan, the roof and upper portion of the walls may be omitted.

Roof plans are orthographic projections, but they are not sections as their viewing plane is outside of the object.

A plan is a common method of depicting the internal arrangement of a 3-dimensional object in two dimensions. It is often used in technical drawing and is traditionally crosshatched. The style of crosshatching indicates the type of material the section passes through (17).

A floor plan, or floorplan, in architecture and building engineering is a diagram, usually to scale, of the relationships between rooms, spaces and other physical features at one level of a structure.

Dimensions are usually drawn between the walls to specify room sizes and wall lengths. Floor plans will also include details of fixtures like sinks, water heaters, furnaces, etc. Floor plans will include notes to specify finishes, construction methods, or symbols for electrical items.

It is also called a "plan" in architectural terms, as opposed to "elevation" which means how the object will look when seen from a side or a "cross section" where the building is shown cut along an axis to reveal the interior (18).

- Elevation

An elevation is an orthographic projection of a 3-dimensional object from the position of a horizontal plane beside an object. In other words, an elevation is a side-view as viewed from the front, back, left or right. An elevation is a common method of depicting the external configuration and detailing of a 3-dimensional object in two dimensions. Building façades are shown as elevations in architectural drawings and technical drawings (19).

2.6. SIMILAR APPLICATION ENVIRONMENTS

Semantic modeling of events has emerged recently as a very promising research direction in business applications and business intelligence (50) (38) (35) (36).). Event Processing Architectures are becoming very important in business processing applications (50). Although event Processing Architectures may seem to contrast to Service Oriented Architectures (SoA), researchers mostly agree for their complementarities, with the Event Processing Architectures covering better the asynchronous requirements of business processing. Event Driven SoA is a combination form of Service Oriented Architectures combining the intelligence and pro-activeness of event driven architecture with the organizational capabilities of SoA (51), (50). Semantic modeling of events is also an active research area in multimedia. To represent the process of multimedia capturing we need to model such aspects as camera parameters, camera location, camera movement, subject movement, light and sound locations, etc. The use of such parameters for capturing quality multimedia is studied in cinematography, which is an area where rich bibliography exists (41), (42), (43), (44), (47). These principles can be taken into account when producing guidelines for taking specific shot types. Our system allows the integration of such help facilities.

Some research related to cinematography tries to capture the rules for good cinematography into language or expert system constructs often with the objective to automate the presentation of virtual reality scenes in games and elsewhere (45), (46).). Whereas our objective is not to automate the cinematography of virtual environments, to the extent that these tools could be used for guidance to “one-man-crew” they could be also useful additions to our system.

A part of the research done was to find similar environments with the application implemented. The search was done by looking for applications or web pages with similar features with the application implemented on the current thesis. Next there will be a list with some of the environments that was found to have similar, but not the exact, functionality with the application implemented on the current thesis.

○ Semantic MPEG-7

MPEG-7, formally named "Multimedia Content Description Interface", is a standard for describing the multimedia content data that supports some degree of interpretation of the information meaning, which can be passed onto, or accessed by, a device or a computer code. MPEG-7 is not aimed at any one application in particular; rather, the elements that MPEG-7 standardizes support as broad a range of applications as possible.

The Semantic Model of Mpeg-7 is based on the modeling of events and includes modeling of actors that are participating in the event as well as the time and the place of the event. The Semantic Model of Mpeg-7 is in a high level, but the extensibility mechanisms of Mpeg-7 can be used to give powerful domain specific semantic descriptions of events in specific domains while still remaining completely within Mpeg-7 (38). The semantic multimedia event model that we present in this thesis can also accommodate domain specific descriptions for the events. Although the purpose of the model is not to concentrate thoroughly in the technical parameters of the audio visual content but to describe multimedia content in a more abstract manner. Moreover the semantic multimedia event model concentrates on the site map, the topography of the locations the capturing took place along with the systematic capturing of the multimedia content.

The Semantic Mpeg-7 descriptions can also be transformed in logic based languages such as OWL for further processing and inference (38). A related recently expanding area of research is the automatic extraction of semantic events from video sequences (37) (39).

While rich literature exists in the area of semantic event modeling and multimedia modeling based on events, as well as in cinematography principles, there is virtually no research in generic tools and methodologies for systematically capturing multimedia events and their context including the spatial context and the context of capturing. Such tools should be based on models associating composite events with the spaces where the events took place and the event evolution in time. In addition they should facilitate the modeling of the capturing processes in space and time and its associations with the event evolution. This will facilitate the visualization of the events, the shots capturing processes and the actual multimedia captured. Without such tools the capturing process is very expensive and slow, and the visualization possibilities limited, overtaxing the cognitive capabilities of the users.

Our work is further motivated by the fact that today’s multimedia capturing devices can automatically capture a lot of contextual information (for example with the help of GPS, compass, azimuth) which can be manipulated to automatically register the 3D natural environment of the picture or video (48), (49). This removes the need for extensive manual editing for the systematic capturing of multimedia events and their context.

Requirements for multimedia event capturing tools include:

1. Capturing the spatial aspects of the event such as integrating maps and diagrams with additional details relevant to the event.

2. Capturing and representing the actors of the event, their spatial representations, locations and movements during the event

3. Capturing the various aspects of the reality which relate to the event using multimedia (photos, video, and sound).

4. Capturing the time of the event. Absolute and relative times with respect to other events. Demarcating the beginning and the ending of shots that relate to a specific event for easy event browsing and extraction.

5. Representing the multimedia capturing process on space and time (camera location, direction, angle, movement, etc.) on top of the spatial representations for facilitating the visualization of the spatial context.

6. Representing event workflows and mechanisms to facilitate the scheduling and preparation for the event capturing and the complete coverage of all the events composing a complex event.

7. Associating guidelines for what to capture and how to do the capturing (such as abstract scene descriptions, camera location and movement, shot taking, lighting conditions, etc.) with each event type.

The mobile multimedia event capturing model and system described here aim to support such functionalities.

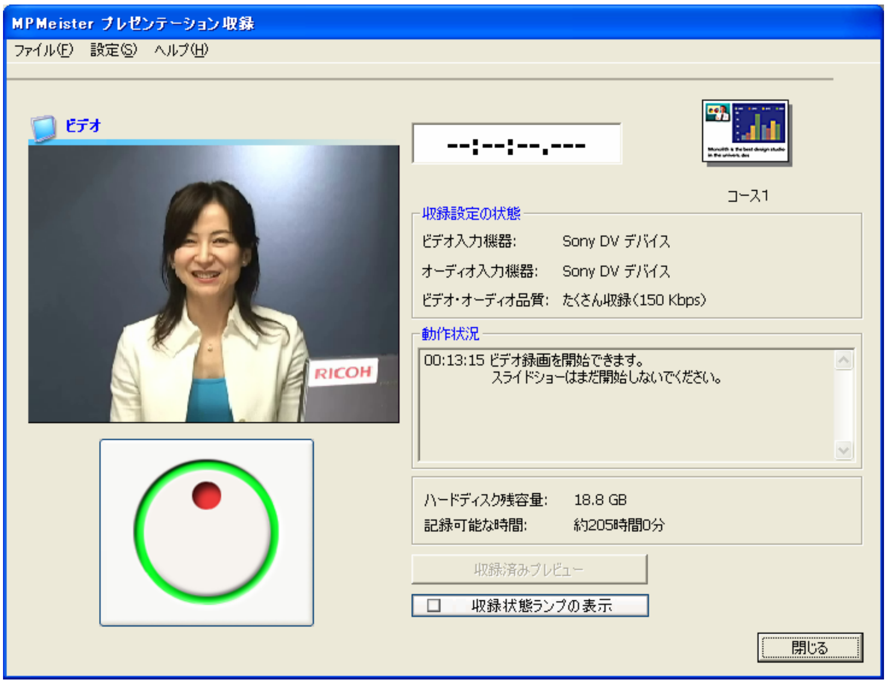
- MP Meister

MP Meister is an automatic content generation tool. It combines video, slides and Table of Contents into a visual format. This kind of software is used for presenters who want to disseminate the presentation information. The contents are typically distributed on CD or shared on the Internet. Contents generated by MP Meister synchronize video with slides. Therefore, content can be integrated with other content and deletions are possible while keeping the video and slides synchronized. MP Meister provides editing functions that specialize in presentation contents, setting it apart from the general video edit tools found in generic video editors.

Support for large halls is also provided by the MP Meister. For content generation in halls, users need two PCs. One PC with video camera attached should be used to record the presenter. The other PC should be used by the presenter for the presentation. Beyond that, with MP Meister, the content can be generated if the two PCs are connected to each other after the presentation ends. Content generation using only one PC is also possible for small conference rooms.

MP Meister is using MPEG-7 as metadata of content. This feature is not readily apparent in MP Meister. The purpose is to deposit metadata of the content. The data contains synchronizing information between the video and slide, textual information etc, and the format is based on the international MPEG-7 standard.

MP Meister is a content management tool with functionality which is similar to the functionality we describe in our thesis, due to synchronization of video with slides and textual information and the usage of MPEG-7 as metadata content. Nevertheless, the mobile multimedia event capturing model and system described here aims to support the functionalities mentioned above.



ビデオ入力機器: Sony DV デバイス
オーディオ入力機器: Sony DV デバイス
ビデオ・オーディオ品質: たくさん収録(150 Kbps)

00:13:15 ビデオ録画を開始できます。
スライドショーはまだ開始しないでください。

ハードディスク残容量: 18.8 GB
記録可能な時間: 約205時間0分

□ 収録状態ランプの表示

Figure 2.4 Mp Meister Capturing Environment Example

○ Adobe Director

Adobe Director (formerly Macromedia Director) is a multimedia application authoring platform created by Macromedia—now part of Adobe Systems. It allows users to build applications built on a movie metaphor, with the user as the "director" of the movie. Originally designed for creating animation sequences, the addition of a powerful scripting language called Lingo made it a popular choice for creating CD-ROMs and standalone kiosks and web content using Adobe Shockwave. Adobe Director supports both 2D and 3D multimedia projects.

Director's has been used to mockup Graphical user interfaces and create prototypes of applications, as well as create self-running Kiosks on CDs and DVDs. It has also been used to author interactive games with rich graphics and embedded QuickTime movies. Director supports many different bitmap, audio, and video formats. Director 11 added the ability to import, manipulate, and display 3D objects. It also supports vector graphics and 3D interactivity (via Shockwave 3D). Since Version 8, Director also natively incorporates Flash animation files.

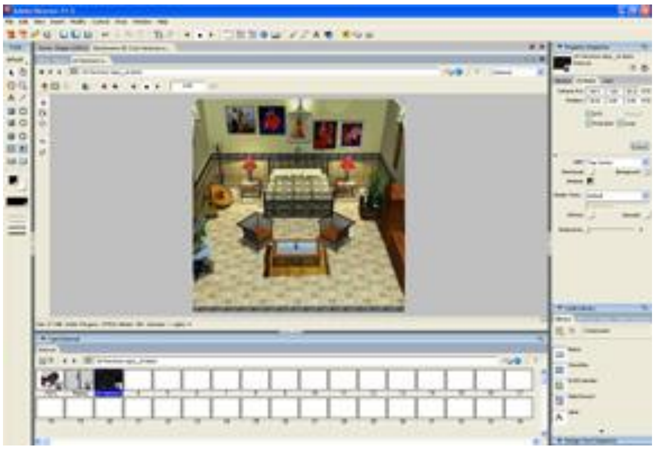


Figure 2.5 Adobe Director Environment's example

○ Similar representation web pages

There were also a number of web pages that have similar features with the application implemented on the current thesis related to the representation part of the application. These web sites provide a virtual tour of a specific location by combining a video or a set of photographs with a specific video.

An example of these web pages is the web page of the National Foundation of Research and Study “Eleutherios Venizelos” that hosts a virtual tour of the house where Eleutherios Venizelos lived his life. This virtual toor contains a video tour of the house and a floor plan of the house that shows, at any time, where the camera is and what is shown (20). leutherios Venizelos Home Virtual Toor

However, the “Eleutherios Venizelos” virtual tour web page similarities with the mobile multimedia event capturing and visualization application stop at the synchronization of the video with the floor-plan. To start, the “Eleutherios Venizelos” virtual tour is a static web page which focuses only to display the interior of the house. On the other hand the presentation of the video along with a site map is only one feature of our application which supports the mobile multimedia event capturing model. Our system supports the real time capturing of complex multimedia events of different types, the recording of the capturing process and the metadata associated with the events, and the learning and preparation of the one-man-crew that will do the multimedia event capturing.

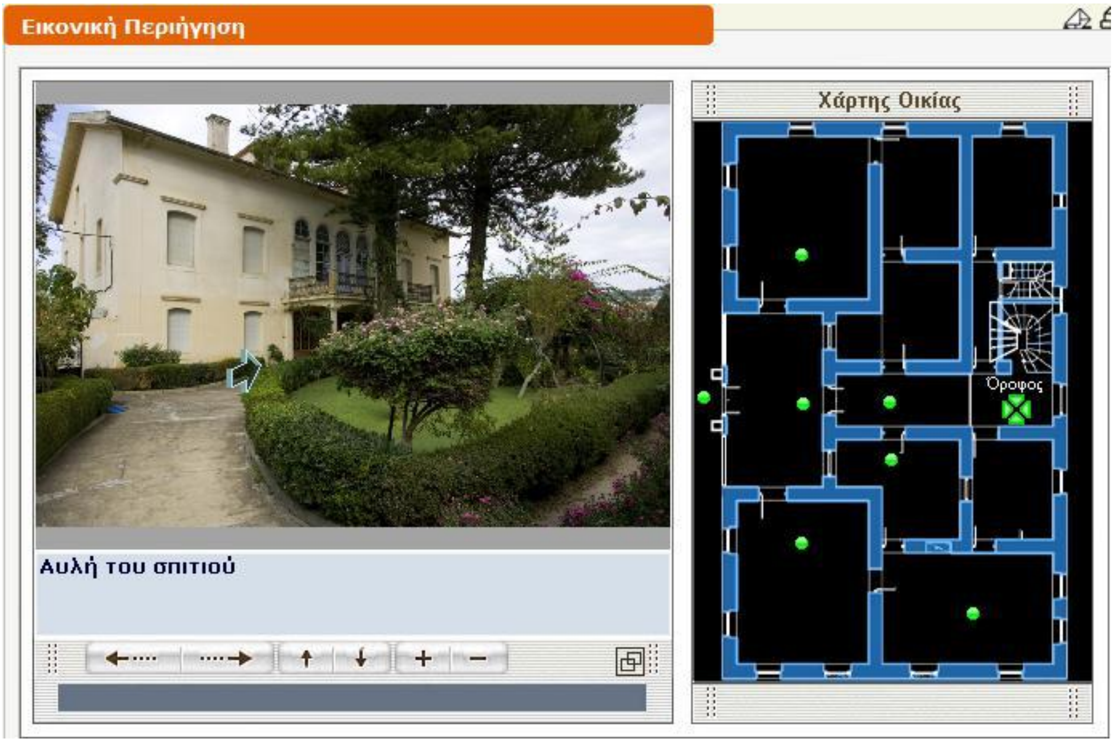


Figure 2.6 Eleutherios Venizelos virtual tour example

3. TECHNOLOGICAL BASE

In the current section, we will describe the technology that was used in the implementation of the application. The application was created in the eclipse platform, which is a platform used to program in the java language, and on the same time other technologies, such as the Java Media Framework (JMF) for the video reproducing, FMJ, LTI-CIVIL and FOBS4JMF that completes the video support that the JMF API provides, JavaGEOMM that helps the site map drawing and sql to java connector that provides the communication of the application with the relational database.

3.1. ECLIPSE PLATFORM

Eclipse (22) is a Java-based, extensible open source development platform. By itself, it is simply a framework and a set of services for building applications from plug-in components. Fortunately, Eclipse comes with a standard set of plug-ins, including the well-known Java Development Tools (JDT).

While most users are quite happy to use Eclipse as a Java integrated development environment (IDE), its ambitions do not stop there. Eclipse also includes the Plug-in Development Environment (PDE), which is mainly of interest to those who want to extend Eclipse, since it allows you to build tools that integrate seamlessly with the Eclipse environment. Because everything in Eclipse is a plug-in, all tool developers have a level playing field for offering extensions to Eclipse and providing a consistent, unified IDE for users.

This parity and consistency isn't limited to Java development tools. Although Eclipse is written in the Java programming language, its use isn't limited to the Java language. For example, plug-ins are available or planned that include support for programming languages like C/C++ and COBOL. The Eclipse framework can also be used as the basis for other types of applications unrelated to software development, such as content management systems.

Additional plug-ins

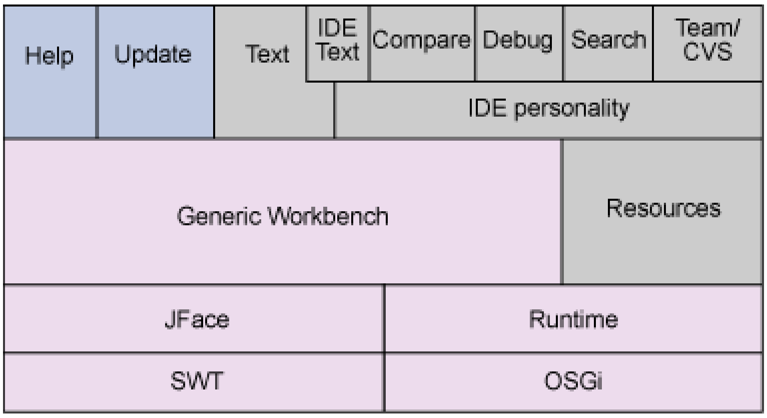
In addition to plug-ins like the JDT for editing, compiling, and debugging applications, plug-ins are available that support the complete development process from modeling, build automation, unit testing, performance testing, version control, and configuration management.

Eclipse comes with a plug-in for working with the open source Concurrent Versions System (CVS) for source control. The Team plug-in connects to a CVS server, allowing the members of a development team to work on a set of source-code files without stepping on each other's changes. Source control from within Eclipse won't be explored here further because it requires setting up a CVS server, but the capability for supporting a development team, not just stand-alone development, is an important and integral feature of Eclipse.

Eclipse Platform architecture

The Eclipse Platform comes with a powerful set of plug-ins (see Figure 8) that supports projects, such as JDT and the PDE.

Figure 2.7 Simplified Eclipse Platform architecture



Simplified Eclipse Platform architecture

The dark blue pieces signify components that are the core part of Eclipse's Rich Client Platform (RCP). The concept of RCP itself is outside the scope of the thesis, but think of RCP as just a set of plug-ins that Eclipse people can use to develop applications, such as Lotus Notes® 8. The light blue pieces are optional (but recommended) pieces to be included in RCP-based applications. And the gray pieces are completely optional. The platform consists of several components, of which I'll cover several:

Runtime

The runtime is the code that defines Eclipse's plug-in model, based on the OSGi specification, and notion of extensions and extension points. The runtime also provides additional services like logging and concurrency.

JFace/SWT

The Standard Widget Toolkit (SWT) is the widget set that gives Eclipse its look and feel. JFace is simply a layer on top of SWT that provides some Model-View-Controller (MVC) classes to make it easier to develop graphical applications.

Workbench

The workbench gives Eclipse its personality. The concept of views, perspectives, and things like editors are defined at this level.

Help (User Assistance)

The Eclipse component allows you to provide assistance to your users. This can be done via the help system, which allows users to search for help documentation or through cheat sheets, which can be thought of interactive task lists for end users.

Update

The update component of Eclipse provides the facilities to allow you to update your applications from version to version.

Team

The team component consists of a framework to allow vendors to plug in their own version-control systems. An exemplary implementation of a provider is the CVS plug-in provided out of the box in Eclipse.

3.2. PROTÉGÉ PLATFORM

We created an abstract (core) ontology according to the specifications of the mobile multimedia event capturing and visualization conceptual model to assist the user during the capturing process.

OWL (Web Ontology Language) is an RDF-based language used to create Semantic Web ontologies. Ontologies indicate the hierarchies and relationships that exist between different resources within a specific domain. That enables us to create scenarios/ontologies, using an abstract ontology, for every different real-world scenario. When RDF resource descriptions are associated with an ontology defined somewhere on the Web, intranet, or extranet, it’s possible for machines to retrieve the semantic information associated with each resource to find, present, and act on data in meaningful ways. The use of a platform that would give us the ability to create the abstract (core) ontology along with the scenarios/ontologies for every different type of real-world scenarios was necessary. Therefore we use the Protégé platform.

Protégé (21) is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. Further, Protégé can be extended by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications.

An ontology describes the concepts and relationships that are important in a particular domain, providing a vocabulary for that domain as well as a computerized specification of the meaning of terms used in the vocabulary. Ontologies can range from taxonomies and classifications, database schema's, to fully axiomatized theories. In recent years, ontologies have been adopted in many business and scientific communities as a way to share, reuse and process domain knowledge. Ontologies are now central to many applications such as scientific knowledge portals, information management and integration systems, electronic commerce, and semantic web services.

The Protégé platform supports two main ways of modeling ontologies:

- The Protégé-Frames editor enables users to build and populate ontologies that are frame-based, in accordance with the Open Knowledge Base Connectivity protocol (OKBC). In this model, an ontology consists of a set of classes organized in a subsumption hierarchy to represent a domain's salient concepts, a set of slots associated to classes to describe their properties and relationships, and a set of instances of those classes - individual exemplars of the concepts that hold specific values for their properties.

- The Protégé-OWL editor enables users to build ontologies for the Semantic Web, in particular in the W3C's Web Ontology Language (OWL). "An OWL ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms" (see the OWL Web Ontology Language Guide).

3.3. JENA

Jena (23) is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. Jena is java API to handle ontology (OWL) model and reasoning. They used RDF and N3 model to represent knowledge in model. Combine with java; it can be used to handle context aware application.

Jena is open source and grown out of work with the HP Labs Semantic Web Programme.

The Jena Framework includes:

- A RDF API
- Reading and writing RDF in RDF/XML, N3 and N-Triples
- An OWL API
- In-memory and persistent storage
- SPARQL query engine

3.4. JMF, FMJ, LTI-CIVIL, FOBS

JMF

The Java Media Framework (JMF) is a Java library that enables audio, video and other time-based media to be added to Java applications and applets. This optional package, which can capture, play, stream, and transcode multiple media formats, extends the Java Platform, Standard Edition (Java SE) and allows development of cross-platform multimedia applications (24).

JMF abstracts the media it works with into DataSources (for media being read into JMF) and DataSinks (for data being exported out). It does not afford the developer significant access to the particulars of any given format; rather, media is represented as sources (themselves obtained from URL's) that can be read in and played, processed, and exported (though not all codecs support processing and transcoding).

A Manager class offers static methods that are the primary point-of-contact with JMF for applications.

JMF provides the basic and necessary functionality for video handling in the implementation; however it does not support all available video, audio formats or methods of encoding. Moreover, the capturing devices that the current version of JMF supports are quite limited. For these reasons, the following java APIs are used to complete the JMF functionality and support devices and formats.

FMJ

FMJ (Freedom for Media in Java) is an open-source project with the goal of providing an alternative to Java Media Framework (JMF), while remaining API-compatible with JMF. It aims to produce a single API/Framework which can be used to capture, playback, process, and stream media across multiple platforms (25).

FMJ is API-compatible with latest JMF, so it can be used with existing JMF codes and run them. FMJ also has two sub-projects and one sister project. The sub-projects, FFMPEG-Java and Theora-Java, are Java wrappers for FFMPEG and Vorbis respectively. The sister project is LTI-CIVIL and it is used as the primary video capture device library.

LTI-CIVIL

LTI-CIVIL is a Java library for capturing images from a video source such as a USB camera. It provides a simple API and does not depend on or use JMF! The FMJ project integrates LTI CIVIL into the JMF architecture by providing a civil: datasource (26).

FOBS and FOBS4JMF

Omnividea Multimedia and the computer vision group at UPM-DISAM present FOBS: Ffmpeg OBjectS. Fobs project is a set of object oriented programming interfaces over the ffmpeg library. FFMpeg is the most complete open source library to work with media files. It is coded in C and completely multiplatform. Its only drawback is the great effort needed to build the most simple application: very complex API and few documentation. Fobs offers developers a simplified Object Oriented API. Actual bindings are: C++ and JMF. It relies in the ffmpeg library, but provides developers with a much simpler programming interface.

Fobs4JMF (Fobs for JMF) is part of the Fobs project and implements the java side of FObs. Fobs4JMF binds the ffmpeg native library of codecs to the Java Media Framework, the Java API to work with media files. Using Fobs4JMF reports two advantages: the support of video and audio codecs supported by JMF is greatly improved and all the heavy computation processes involved in the encoding/decoding are performed natively (27).

3.5. JAVA 2D, JAVA 3D, JAVA GEOMM

JAVA 2D

The Java 2D™ API (30) provides two-dimensional graphics, text, and imaging capabilities for Java™ programs through extensions to the Abstract Windowing Toolkit (AWT), in a single comprehensive model. This comprehensive rendering package supports line art, text, and images in a flexible, full-featured framework for developing richer user interfaces, sophisticated drawing programs, and image editors. Java 2D objects exist on a plane called user coordinate space, or just user space. When objects are rendered on a screen or a printer, user space coordinates are transformed to device space coordinates.

The Java 2D API provides following capabilities:

- A uniform rendering model for display devices and printers
- A wide range of geometric primitives, such as curves, rectangles, and ellipses, as well as a mechanism for rendering virtually any geometric shape
- Mechanisms for performing hit detection on shapes, text, and images
- A compositing model that provides control over how overlapping objects are rendered
- Enhanced color support that facilitates color management
- Support for printing complex documents
- Control of the quality of the rendering through the use of rendering hints

JAVA 3D

THE Java 3D API is an application programming interface used for writing three-dimensional graphics applications and applets. It gives developers high-level constructs for creating and manipulating 3D geometry and for constructing the structures used in rendering that geometry. Application developers can describe very large virtual worlds using these constructs, which provide Java 3D with enough information to render these worlds efficiently.

Java 3D delivers Java's "write once, run anywhere" benefit to developers of 3D graphics applications. Java 3D is part of the JavaMedia suite of APIs, making it available on a wide range of platforms. It also integrates well with the Internet because applications and applets written using the Java 3D API have access to the entire set of Java classes.

The Java 3D API draws its ideas from existing graphics APIs and from new technologies. Java 3D's low-level graphics constructs synthesize the best ideas found in low-level APIs such as Direct3D, OpenGL, QuickDraw3D, and XGL. Similarly, its higher-level constructs synthesize the best ideas found in several scene graph-based systems. Java 3D introduces some concepts not commonly considered part of the graphics environment, such as 3D spatial sound. Java 3D's sound capabilities help to provide a more immersive experience for the user.

Java 3D allows a programmer to specify a broad range of information. It allows control over the shape of objects, their color, and transparency. It allows control over background effects, lighting, and environmental effects such as fog. It allows control over the placement of all objects (even non-visible objects such as lights and behaviors) in the scene graph and over their orientation and scale. It allows control over how those objects move, rotate, stretch, shrink, or morph over time. It allows control over what code should execute what sounds should play, and how they should sound and change over time.

Java 3D provides different techniques for controlling the effect of various features. Some techniques act fairly locally, such as getting the color of a vertex. Other techniques have broader influence, such as changing the color or appearance of an entire object. Still other techniques apply to a broad number of objects. In the first two cases, the programmer can modify a particular object or an object associated with the affected object. In the latter case, Java 3D provides a means for specifying more than one object spatially.

JAVA GEOM

Java Geom (29) is a java library for geometric computations. It provides a general framework for manipulating and creating geometric primitives (points, lines, polygons, circles and more general concs, bezier curves...), computing intersection of shapes, composing them to create new shapes, and performing some measurements. It contains some algorithms to operate efficiently on shapes, like convex hulls or kd-tree.

The aim of the library is to provide a collection of shapes that can be manipulated and processed in many ways: clipping, affine transforms. The different families of shapes planned are:

- Points, and collection of points;
- Straight objects like straight lines, line segments, or rays;
- Polygons, simple and complex (multiply connected and with holes);
- Conic curves: ellipses, parabolas, hyperbolas, and their respective arcs;
- Polynomial curves, like Bezier curves or B-Splines;
- Planar domains, delimited by a set of continuous oriented curves

In addition to these features, composition of several shapes is planned, to manipulate collection of shapes, or complex boundaries composed of different types of curves.

The library will include classes which are not shapes, like

- Vectors
- geometric transforms: affine, projective, inversions, and projections
- A class 'Angle2D' is also provided with utility static methods
- A class Box2D, used for clipping shapes

Many geometrical primitives are already provided. The Geometrical shapes are classified according to their inner dimension:

- Points, and Point sets
- Curves, which can be continuous or oriented. Implemented curves are:
- Straight lines, line segments, rays, polylines, closed polyline
- Conic curves: ellipses, parabolas, hyperbolas, and their respective arcs;
- Polynomial: only the CubicSpline and QuadSline are implemented.
- Planar domains, whose boundary is an arbitrary oriented curve or set of oriented curves can be created
- Polygons implementation will be rewritten

3.6. MySQL CONNECTOR/J

MySQL provides connectivity for client applications developed in the Java programming language via a JDBC (Java to DataBase Connectivity) driver, which is called MySQL Connector/J.

MySQL Connector/J is a JDBC Type 4 driver. Different versions are available that are compatible with the JDBC 3.0 and JDBC 4.0 specifications. The Type 4 designation means that the driver is pure-Java implementation of the MySQL protocol and does not rely on the MySQL client libraries (28).

4. SUMMARY

In this chapter we presented a review of the related research that has been done. In particular we described concepts related to standards, camera capturing parameters, video shooting definitions and guidelines and other application environments relevant to the one implemented in the contents of the current thesis.

CHAPTER 3
SYSTEM REQUIREMENTS ANALYSIS

1. INTRODUCTION

In chapter 3 we will describe the requirements of the system. For that reason, we first need to describe the different types of users involved in the system. After that we will describe the requirements of the system using the Use Cases Methodology. The use cases are software – engineering technology which provides a formal description of the functionality of the system, and which description id fully comprehendible to anyone who wishes to read it. The description template of the application's use cases which we use in the current chapter is the widely defined by Alistair Cockburn in the "Writing Effective Use Cases" (33).

In the current chapter, we will also describe the reference system modules of the mobile multimedia event capturing and visualization system. The reference modules describe the different packages of the system.

2. SYSTEM USERS

The users involved in the application are not all sharing the same technical knowledge. There are different types of users involved during the different stages (modules) of the system.

User	Description
Event Type Creator	Creates a new Event type, describes its structure, the roles of the participants, what shots should be taken, time and clues to detect the sub-events
Event Creator	Creates a specific Event (e.g. a new Wedding of Mary and John). He describes the specific actors that play the roles (e.g. John is the groom and Mary is the bride), the time, place, he gives to the system maps of areas and all the known specific info about the event and its evolution
Multimedia Event Capturing Personnel	Takes the videos and associates them with the specific event. He captures any additional attribute information in real time (e.g. the names of the friends). He captures the locations and other

	parameters of the video (camera, sound etc). He reviews possible past shots in order to make a better job in the capture.
Multimedia Editors / Viewers	They look at the multimedia captured about a specific event, and they browse to find relevant parts. They may make further editing, and store the multimedia.

2.1. Event Type Creator

First of all, there is the Event Type Creator user. He is the user that creates a new composite event type in order to create a scenario (e.g. a Wedding scenario). He describes its structure, describes the roles of participants, describes what shots should be taken, describes time and clues to detect the sub events. That type of user should be acquainted with the specifications of a specific scenario in order to describe it, At the same time, the event type creator user should be aware of the basic principles of cinematography so that he can use them for providing the appropriate guidelines for every different event types he creates in the context of the specific scenario. This user should also know how to use the protégé platform in order to create the scenario – ontology.

2.2. Event Creator

The second type of user is the Event Creator user. He is the user that creates a specific event (e.g. a new wedding of John and Mary) by using an already created scenario – ontology. He describes the specific actors that play the roles (e.g. John is the groom and Mary is the bride, time, place, he gives to the system maps of the areas and all the known specific info about the event and its evolution). That type of user is not required to know any of the principles of cinematography about the proper capturing methods. He is also not obligated to follow the flow of the scenario given to him by the scenario – ontology. He can create another event that has not been defined before (e.g. in the marriage event, they decide to do a party the night before the wedding, to accommodate the guests that will come from other places) and he can also remove an event if he does not need it (e.g. the custom where the father of the bride puts her shoes on is not applied in the specific area).

2.3. Multimedia Event Capturing Personnel

A third type of user is the Multimedia Event Capturing Personnel. He could be the same as the Event Creator or different person. Takes the videos and associates them with the specific event. He captures any additional attribute information in real time (such as the names of the friends).He captures the location and other camera (sound etc) parameters of the video.

Reviews the possible past shots in order to make a better job in the capture. That user could be a naïve user that is just using the application for capturing the desired video. He takes the events created by the Event Creator and uses them for the capturing and saving the information relevant to the specific application (e.g. the wedding of John and Mary). He is not obligated to follow the guidelines given to him for the capturing. He can treat them as guidance and use them as he considers necessary.

2.4. Multimedia Editors/Viewers

The last type of users is the Multimedia editors or Viewers. They look at the multimedia captured about a specific event, and they browse to find relevant parts, they may make further editing, and store the multimedia. That type of user is not required to have knowledge of any of the above fields of knowledge. He just uses the application for his reasons to retrieve the information he desires.

3. SYSTEM USE CASES

In this section we will describe the system requirements by user. To do that, we will use the Use Cases method. A use case captures a contract between the stakeholders of a system about its behavior. The use case describes the system's behavior under various conditions as it responds to a request from one of the stakeholders, called the *primary actor*. The primary actor initiates an interaction with the system to accomplish some goal. The system responds, protecting the interests of all the stakeholders. Different sequences of behavior, or scenarios, can unfold, depending on the particular requests made and conditions surrounding the requests. The use case collects together those different scenarios. Use cases are fundamentally a text form, although they can be written using flow charts, sequence charts, Petri nets, or programming languages. Under normal circumstances, they serve to communicate from one person to another, often to people with no special training. Simple text is, therefore, usually the best choice.

In our case, to describe the system requirements, we will use the **overall picture** tool along with the **summary table** tool. The overall picture represents the use cases as boxes which are linked together with relations (which use case calls which). The summary table represents use cases categorized by level and contains information about the id, the name, the primary actor, the goal and a small description of each use case.

3.1. Event Type Creator Use cases

The next two figures show the overall picture table for the use cases of the user Event Type Creator.

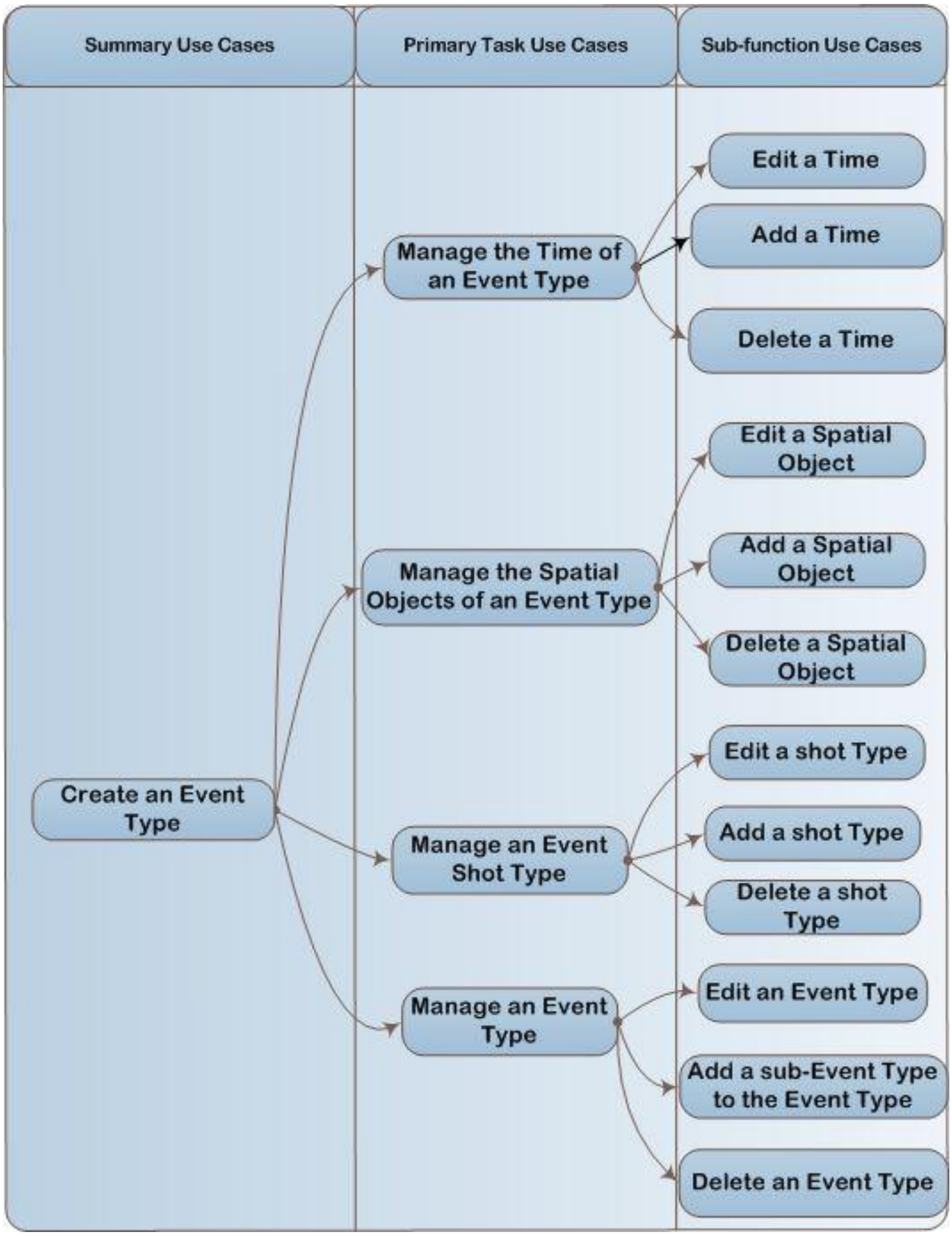


Figure 3.1 Overall Picture of Event Type Creator Use cases Part 1

As we described in the previous section, the event type creator is the user that creates a composite event type in order to create a scenario (e.g. a Wedding scenario). He is the one that manages the location (spatial object) of the event, the time of the event, the roles participating in the event as well as the shots of the specific event.

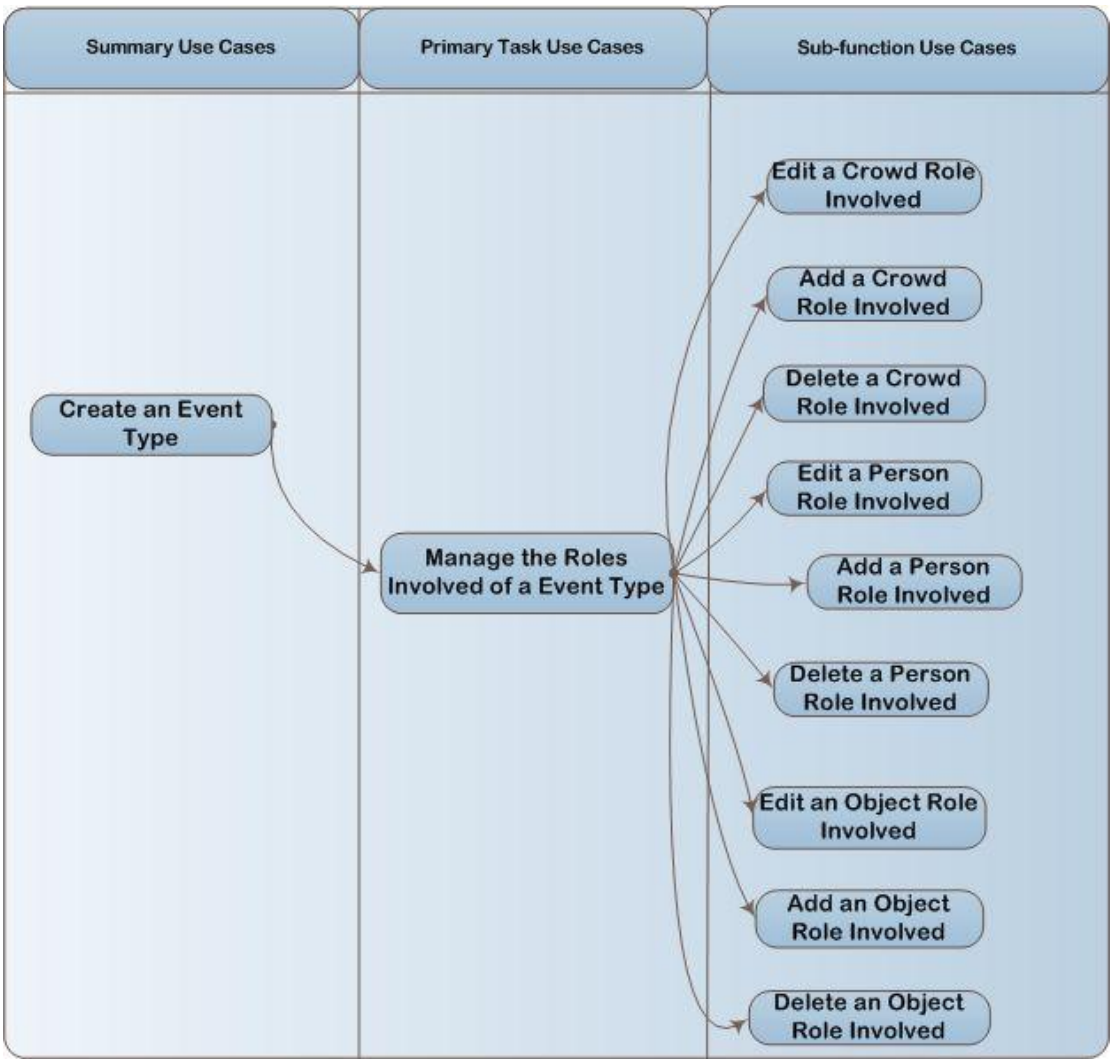

Summary Use Cases
Primary Task Use Cases
Sub-function Use Cases
Create an Event Type
Manage the Roles Involved of a Event Type
Edit a Crowd Role Involved
Add a Crowd Role Involved
Delete a Crowd Role Involved
Edit a Person Role Involved
Add a Person Role Involved
Delete a Person Role Involved
Edit an Object Role Involved
Add an Object Role Involved
Delete an Object Role Involved

Figure 3.2 Overall Picture of Event Type Creator Use cases Part 2

The next table is the summary table of the event type creator use cases which gives a short description of each use case.

#	Level	Primary Actor	Goal	Description
UC1	Summary	Event Type Creator	Create an Event Type	The user wants to create a composite Event Type for a real life scenario.
UC2	Primary Task	Event Type Creator	Manage an Event Type	The user wants to manage the information related to the selected Event Type
UC3	Sub-function	Event Type	Add a sub-Event Type to	The user wants to

		Creator	the Event Type	add a new sub Event Type to the selected Event Type.
UC4	Sub-function	Event Type Creator	Delete an Event Type	The user wants to delete an Even Type.
UC5	Sub-function	Event Type Creator	Edit an Event Type	The user wants to edit some of the details of the Even Type.
UC6	Primary Task	Event Type Creator	Manage the Shot of an Event Type	The user wants to add, delete or edit the shot related to an Event Type.
UC7	Sub-function	Event Type Creator	Add a shot	The user wants to add a shot to the selected Event Type.
UC8	Sub-function	Event Type Creator	Delete a Shot	The user wants to delete a shot from the selected Event Type.
UC9	Sub-function	Event Type Creator	Edit a Shot	The user wants to edit some of the details of a shot of the selected Event Type.
UC10	Primary Task	Event Type Creator	Manage the Spatial Objects of an Event Type	The user wants to add, delete or edit the spatial object related to an Event Type.
UC11	Sub-function	Event Type Creator	Add a Spatial Object	The user wants to add a spatial object to state that this is the location the Event Type took place.
UC12	Sub-function	Event Type Creator	Delete a Spatial Object	The user wants to delete a Spatial object from the selected Event Type.
UC13	Sub-function	Event Type Creator	Edit a Spatial Object	The user wants to edit some of the details of the spatial object.
UC14	Primary Task	Event Type Creator	Manage the Time of an Event Type	The user wants to add, delete or edit the relative time that the Event Type occurred.
UC15	Sub-function	Event Type Creator	Add a Time	The user wants to add a time to state

				that this is the relative time the Event Type occurred.
UC16	Sub-function	Event Type Creator	Delete a Time	The user wants to delete a relative time.
UC17	Sub-function	Event Type Creator	Edit a Time	The user wants to edit some of the details of the time.
UC18	Primary Task	Event Type Creator	Manage the Roles Involved in an Event Type	The user wants to add, delete or edit the participants of the Event Type.
UC19	Sub-function	Event Type Creator	Add a Person Role Involved	The user wants to add a person role involved to state that this person is participating to the specific Event Type.
UC20	Sub-function	Event Type Creator	Delete a Person Role Involved	The user wants to delete a person role involved from a specific Event Type.
UC21	Sub-function	Event Type Creator	Edit a Person Role Involved	The user wants to edit some of the details of a person participant.
UC22	Sub-function	Event Type Creator	Add an Object Role Involved	The user wants to add an object role involved to state that this person is participating to the specific Event Type.
UC23	Sub-function	Event Type Creator	Delete an Object Role Involved	The user wants to delete a object role involved from a specific Event Type.
UC24	Sub-function	Event Type Creator	Edit an Object Role Involved	The user wants to edit some of the details of an object participant.
UC25	Sub-function	Event Type Creator	Add a Crowd Role Involved	The user wants to add a crowd role involved to state that this person is participating to the specific Event Type.
UC26	Sub-function	Event Type Creator	Delete a Crowd Role Involved	The user wants to delete a crowd role involved from a

				specific Event Type.
UC27	Sub-function	Event Type Creator	Edit a Crowd Role Involved	The user wants to edit some of the details of a crowd participant.

3.2. Event Creator Use cases

The next three figures show the overall picture table for the use cases of the user Event Creator.

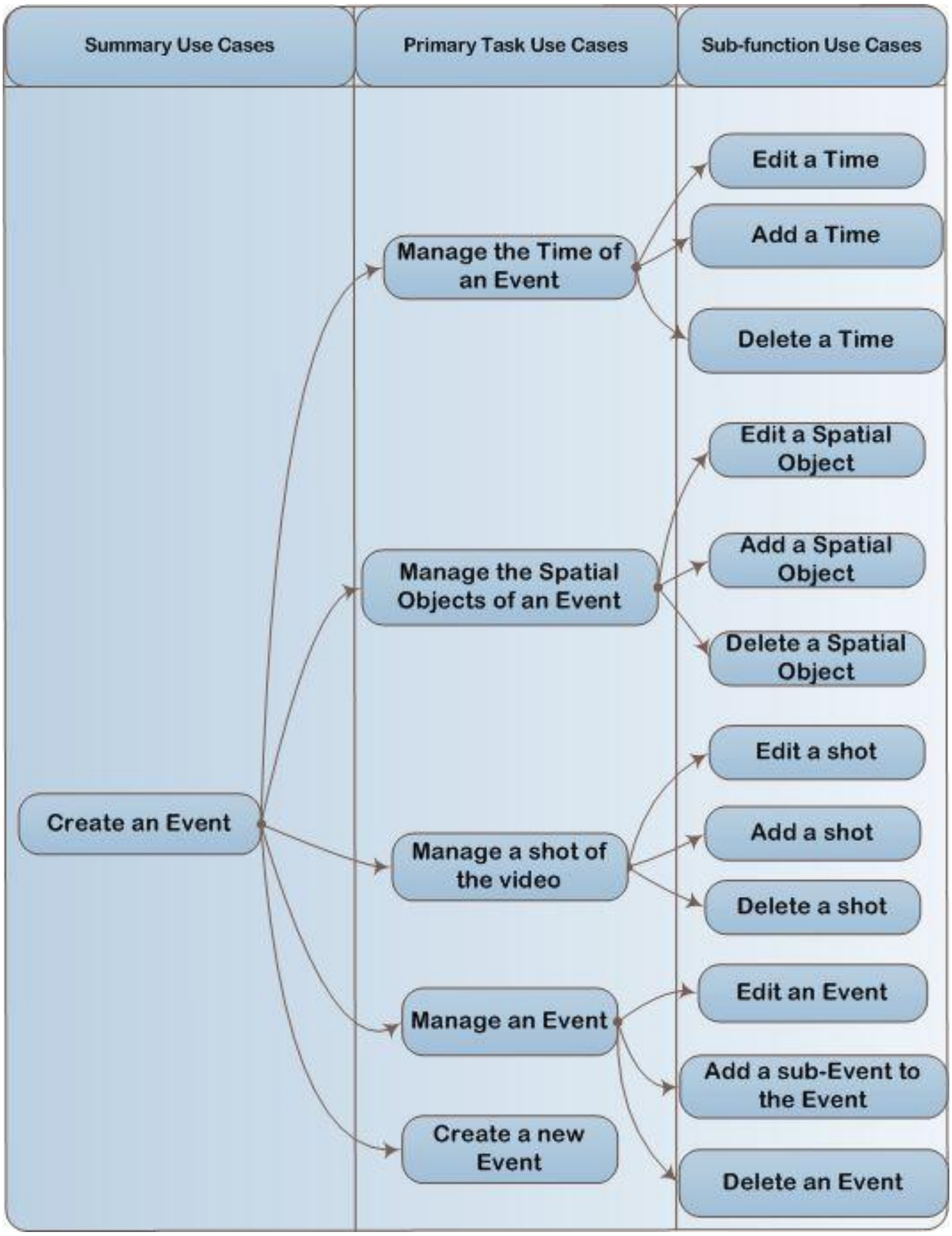


Figure 3.3 Overall Picture of Event Creator Use cases Part 1

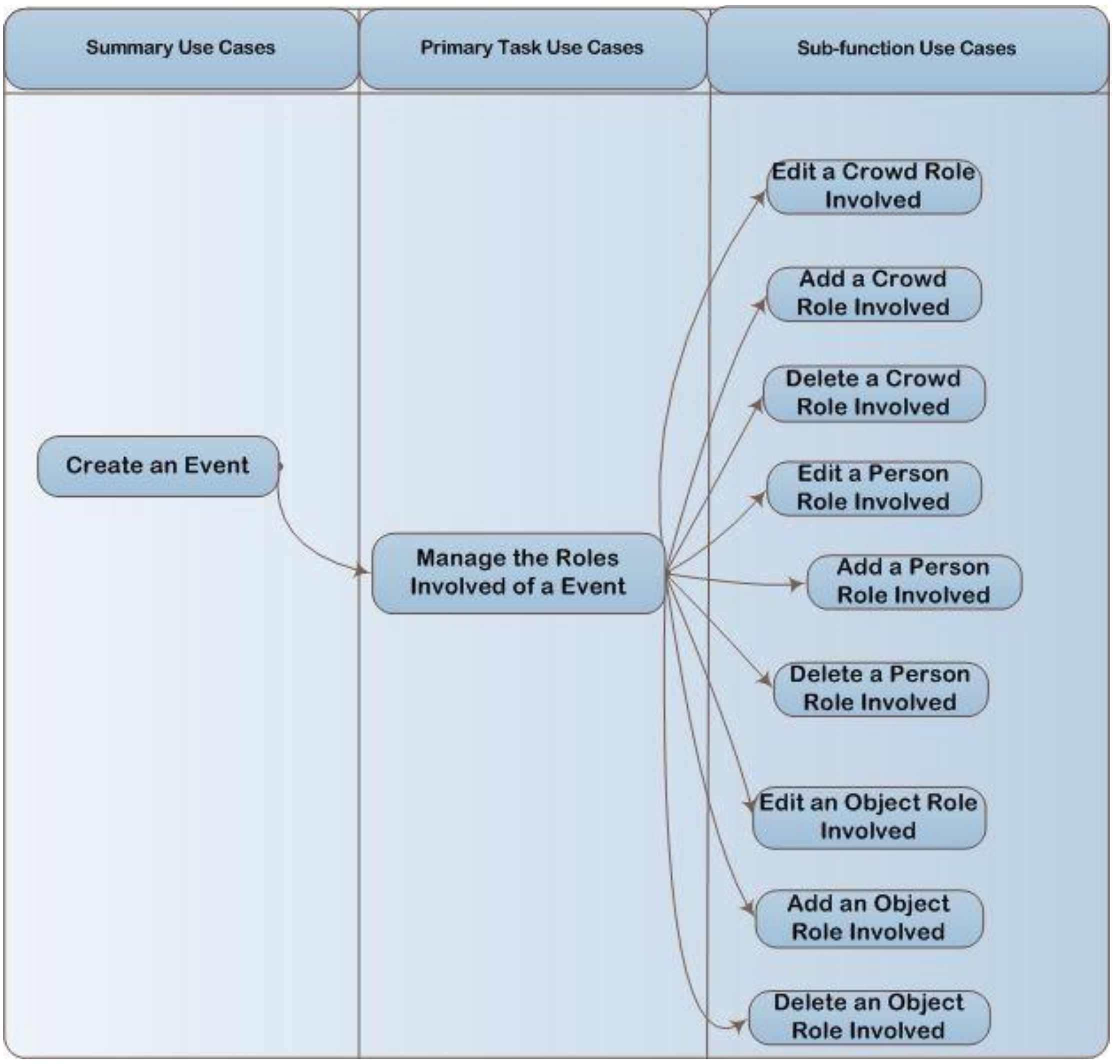

Summary Use Cases
Primary Task Use Cases
Sub-function Use Cases
Create an Event
Manage the Roles Involved of a Event
Edit a Crowd Role Involved
Add a Crowd Role Involved
Delete a Crowd Role Involved
Edit a Person Role Involved
Add a Person Role Involved
Delete a Person Role Involved
Edit an Object Role Involved
Add an Object Role Involved
Delete an Object Role Involved

Figure 3.4 Overall Picture of Event Creator Use cases Part 2

As we described in the previous section, the event creator is the user that creates a specific event in order to create a real life scenario (e.g. a new wedding of John and Mary). He is the one that manages the location (spatial object) of the specific event, the time of the specific event, the roles participating in the specific event as well as the shots of the specific event.

He also manages the site maps of every shot by giving to the system maps of the areas. He manages the participants of each site map, the camera parameters of each shot and all the known specific info about the event and its evolution.

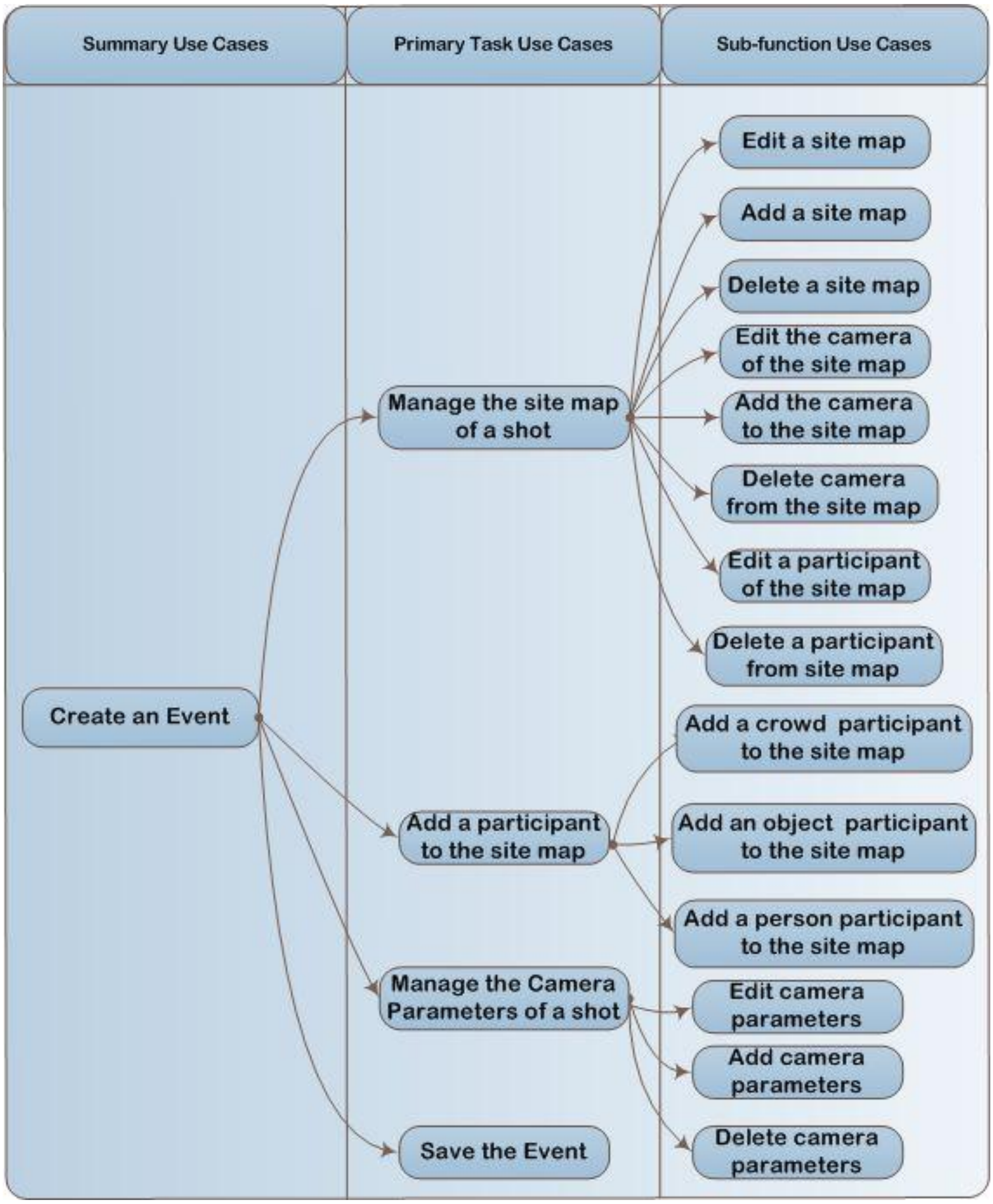

Summary Use Cases
Primary Task Use Cases
Sub-function Use Cases
Create an Event
Manage the site map of a shot
Add a participant to the site map
Manage the Camera Parameters of a shot
Save the Event
Edit a site map
Add a site map
Delete a site map
Edit the camera of the site map
Add the camera to the site map
Delete camera from the site map
Edit a participant of the site map
Delete a participant from site map
Add a crowd participant to the site map
Add an object participant to the site map
Add a person participant to the site map
Edit camera parameters
Add camera parameters
Delete camera parameters

Figure 3.5 Overall Picture of Event Creator Use cases Part 3

The next table is the summary table of the event creator use cases which gives a short description of each use case.

#	Level	Primary Actor	Goal	Description
UC1	Summary	Event Creator	Create an Event	The user wants to create a composite Event for a specific Event Type.
UC2	Primary Task	Event Creator	Manage an Event	The user wants to

				manage the information related to the selected Event.
UC3	Sub-function	Event Creator	Add a sub-Event to the Event	The user wants to add a new sub Event to the selected Event.
UC4	Sub-function	Event Creator	Delete an Event	The user wants to delete an Even.
UC5	Sub-function	Event Creator	Edit an Event	The user wants to edit some of the details of the Event.
UC6	Primary Task	Event Creator	Manage the Shot of an Event	The user wants to add, delete or edit the shot related to an Event.
UC7	Sub-function	Event Creator	Add a shot	The user wants to add a shot to the selected Event.
UC8	Sub-function	Event Creator	Delete a Shot	The user wants to delete a shot from the selected Event.
UC9	Sub-function	Event Creator	Edit a Shot	The user wants to edit some of the details of a shot of the selected Event.
UC10	Primary Task	Event Creator	Manage the Spatial Objects of an Event	The user wants to add, delete or edit the spatial object related to an Event.
UC11	Sub-function	Event Creator	Add a Spatial Object	The user wants to add a spatial object to state that this is the location the Event took place.
UC12	Sub-function	Event Creator	Delete a Spatial Object	The user wants to delete a Spatial object from the selected Event.
UC13	Sub-function	Event Creator	Edit a Spatial Object	The user wants to edit some of the details of the spatial object.
UC14	Primary Task	Event Creator	Manage the Time of an Event	The user wants to add, delete or edit the relative time that the Event occurred.
UC15	Sub-function	Event Creator	Add a Time	The user wants to add a time to state

				that this is the relative time the Event occurred.
UC16	Sub-function	Event Creator	Delete a Time	The user wants to delete a relative time from the selected event.
UC17	Sub-function	Event Creator	Edit a Time	The user wants to edit some of the details of the time.
UC18	Primary Task	Event Creator	Manage the Roles Involved in an Event	The user wants to add, delete or edit the participants of the Event.
UC19	Sub-function	Event Creator	Add a Person Role Involved	The user wants to add a person role involved to state that this person is participating to the specific Event.
UC20	Sub-function	Event Creator	Delete a Person Role Involved	The user wants to delete a person role involved from a specific Event.
UC21	Sub-function	Event Creator	Edit a Person Role Involved	The user wants to edit some of the details of a person participant.
UC22	Sub-function	Event Creator	Add an Object Role Involved	The user wants to add an object role involved to state that this person is participating to the specific Event.
UC23	Sub-function	Event Creator	Delete an Object Role Involved	The user wants to delete a object role involved from a specific Event.
UC24	Sub-function	Event Creator	Edit an Object Role Involved	The user wants to edit some of the details of an object participant.
UC25	Sub-function	Event Creator	Add a Crowd Role Involved	The user wants to add a crowd role involved to state that this person is participating to the specific Event.
UC26	Sub-function	Event Creator	Delete a Crowd Role Involved	The user wants to delete a crowd role involved from a

				specific Event.
UC27	Sub-function	Event Creator	Edit a Crowd Role Involved	The user wants to edit some of the details of a crowd participant.
UC28	Primary Task	Event Creator	Manage the site map of a shot	The user wants to add, delete or edit the maps of the areas of a specific shot to state the topology of the scene during the specific shot.
UC29	Sub-function	Event Creator	Add a site map	The user wants to add a site map related to a shot.
UC30	Sub-function	Event Creator	Delete site map	The user wants to delete a site map related to a shot
UC31	Sub-function	Event Creator	Edit site map	The user wants to edit the details of a site map related to a shot.
UC32	Sub-function	Event Creator	Add the camera to the site map	The user wants to add a video camera representation to the site map to state the location and movement of the camera shooting the video.
UC33	Sub-function	Event Creator	Delete camera from the site map	The user wants to delete the camera from the site map.
UC34	Sub-function	Event Creator	Edit camera on the site map	The user wants to edit the location or any other parameters of the camera on the site map.
UC35	Sub-function	Event Creator	Delete a participant of the site map	The user wants to delete a participant from the site map.
UC36	Sub-function	Event Creator	Edit a participant of the site map	The user wants to edit the location or any other parameters of a participant on the site map.
UC37	Primary Task	Event Creator	Add a participant to the site map	The user wants to add a person, object

				or crowd representation to the site map to state the location and movement of the participant.
UC38	Sub-function	Event Creator	Add a person participant to the site map	The user wants to add a person representation to the site map to state the location and movement of the person.
UC39	Sub-function	Event Creator	Add an object participant to the site map	The user wants to add an object representation to the site map to state the location and movement of the object.
UC40	Sub-function	Event Creator	Add a crowd participant to the site map	The user wants to add a crowd representation to the site map to state the location and movement of the crowd.
UC41	Primary Task	Event Creator	Manage Camera Parameters	The user wants to add, delete or edit the technical parameters of the camera that is used or has been used to capture the video.
UC42	Sub-function	Event Creator	Add Camera parameters	The user wants to add new camera parameters.
UC43	Sub-function	Event Creator	Delete Camera Parameters	The user wants to delete the camera parameters.
UC44	Sub-function	Event Creator	Edit Camera Parameters	The user wants to edit the details of the camera parameters.
UC45	Sub-function	Event Creator	Save scenario	The user wants to finish editing the scenario and save it to the database.

3.3. Multimedia Capturing Personnel Use cases

The next figure shows the overall picture table for the use cases of the Multimedia Event Capturing Personnel.

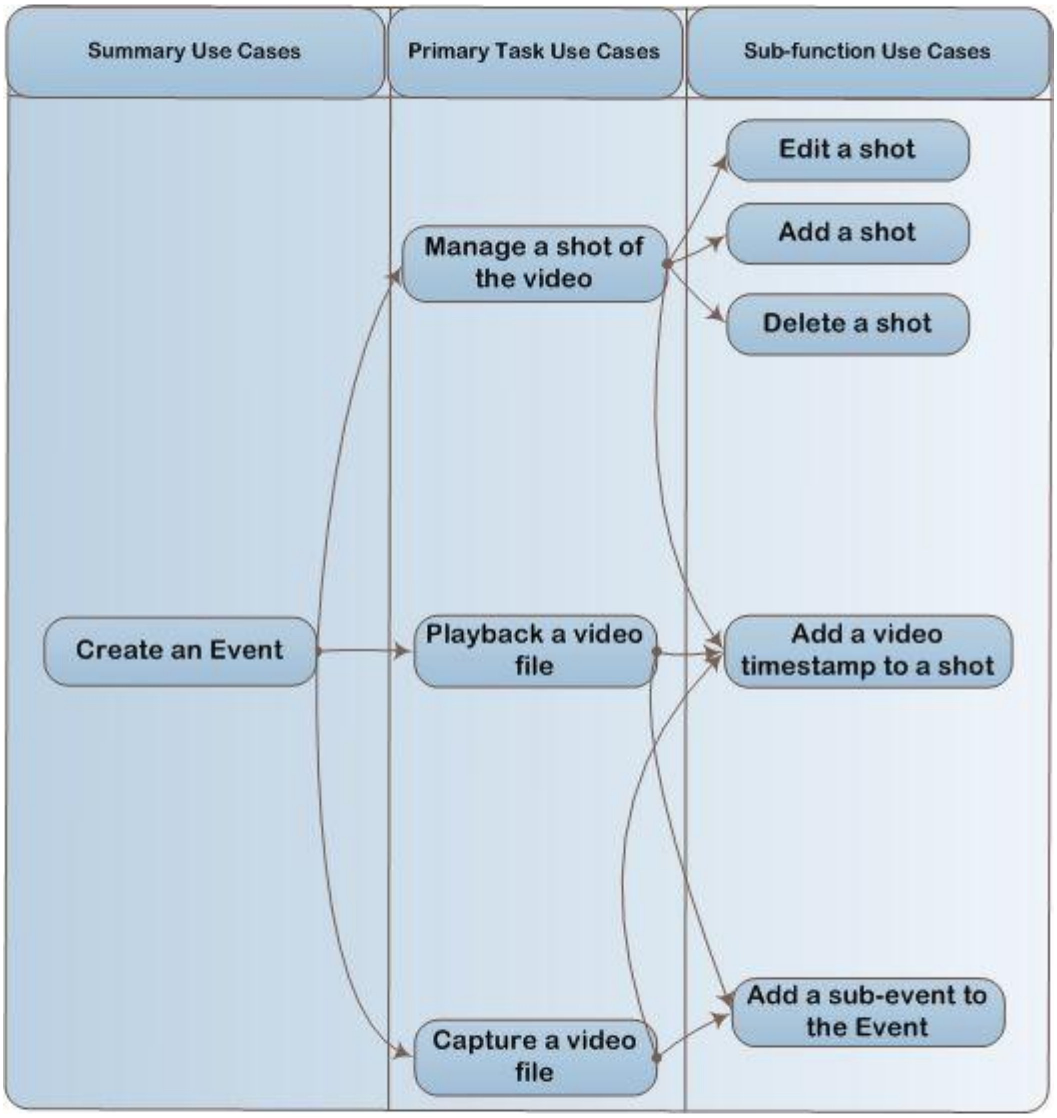


Figure 3.6 Overall Picture of Multimedia Event Capturing Personnel Use cases

The multimedia Event Capturing Personnel and the Event Creator could be the same person. He is responsible to take the videos (either to capture the video in real time or take an already captured video) and associate them with the specific event. He can also manage the shots of a specific event in real time.

The next table is the summary table of the multimedia event capturing personnel use cases which gives a short description of each use case.

#	Level	Primary Actor	Goal	Description
UC1	Summary	Multimedia Event	Create an Event	The user wants to

		Capturing Personnel		create a composite Event for a specific Event Type.
UC2	Primary Task	Multimedia Event Capturing Personnel	Capture a video file	The user wants to capture a video and relate it to an Event and event shots.
UC3	Primary Task	Multimedia Event Capturing Personnel	Playback a video file	The user wants to play an already captured video and relate it to scenes and event shots.
UC4	Primary Task	Multimedia Event Capturing Personnel	Manage a Shot of the video	The user wants to add or edit an event shot related to a specific video.
UC5	Sub-function	Multimedia Event Capturing Personnel	Add a shot	The user wants to add a shot related to a specific video timestamp.
UC6	Sub-function	Multimedia Event Capturing Personnel	Delete a shot	The user wants to delete a shot.
UC7	Sub-function	Multimedia Event Capturing Personnel	Edit a shot	The user wants to edit some of the details of the shot.
UC8	Sub-function	Multimedia Event Capturing Personnel	Add video timestamp to a shot	The user wants to add the time where the shot started in the video.
UC9	Sub-function	Multimedia Event Capturing Personnel	Add a sub-Event to the Event	The user wants to add a new sub Event to the selected Event.

3.4. Multimedia Editor / Viewer Use cases

The next figure shows the overall picture table for the use cases of the Multimedia Editor/Viewer.

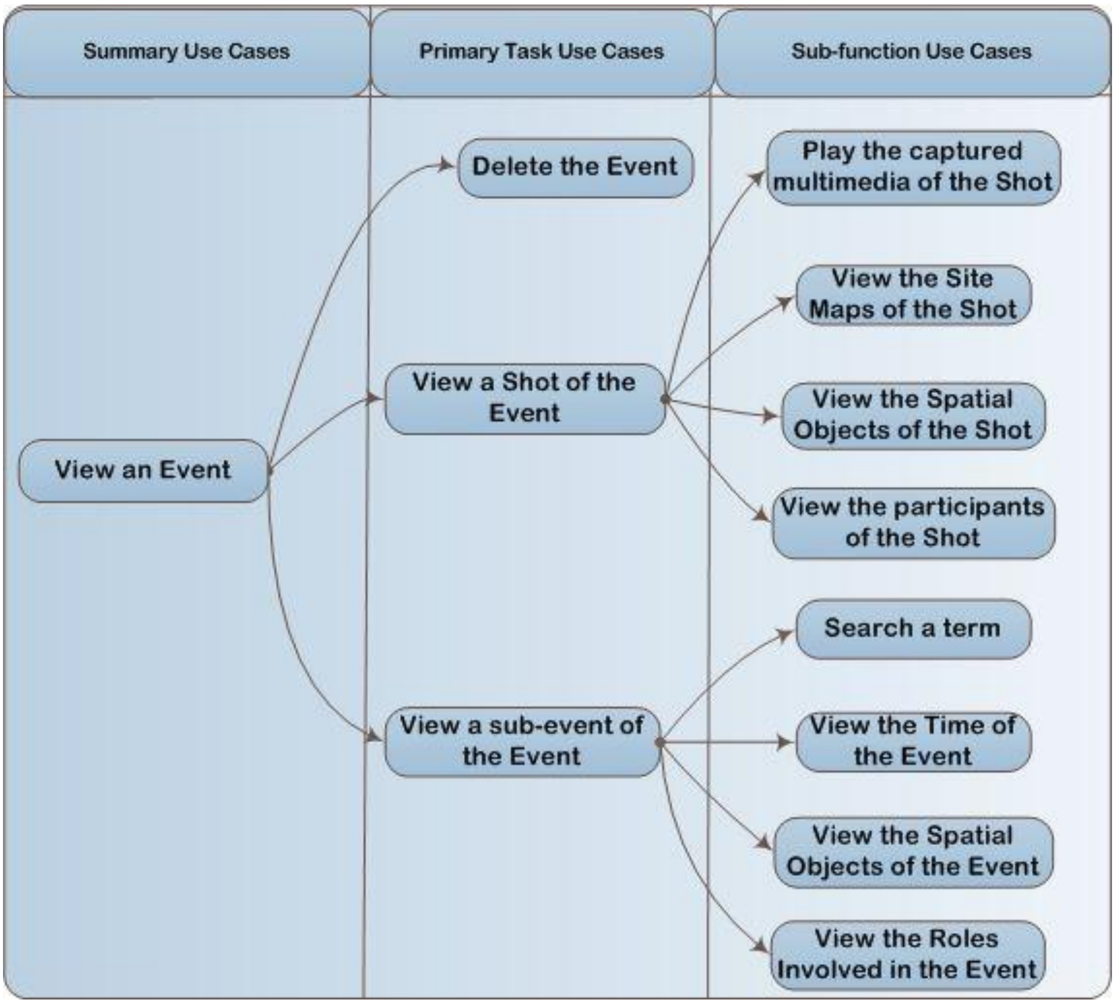

Summary Use Cases
Primary Task Use Cases
Sub-function Use Cases
View an Event
Delete the Event
View a Shot of the Event
View a sub-event of the Event
Play the captured multimedia of the Shot
View the Site Maps of the Shot
View the Spatial Objects of the Shot
View the participants of the Shot
Search a term
View the Time of the Event
View the Spatial Objects of the Event
View the Roles Involved in the Event

Figure 3.7 Overall Picture of Multimedia Editor / Viewer Use cases

The multimedia editor or viewer can view a specific event and its associated detailed information, a specific shot and its associated detailed information as well as he can play the multimedia captured associated with a specific event shot.

The next table is the summary table of the multimedia editor / viewer use cases which gives a short description of each use case.

#	Level	Primary Actor	Goal	Description
UC1	Summary	Multimedia Editor/Viewer	View an Event	The user wants to see the details of an already captured event (e.g. the marriage of John and Mary video

				shots).
UC2	Sub-function	Multimedia Editor/Viewer	Search a term (scenario, scene or shot)	The user wants to search an event or a specific shot of an event.
UC3	Primary Task	Multimedia Editor/Viewer	View a sub-event of the Event	The user wants to view all the sub events of a specific composite event.
UC4	Sub-function	Multimedia Editor/Viewer	View the roles involved in the Event	The user wants to see the information about a role involved in an event.
UC5	Sub-function	Multimedia Editor/Viewer	View the spatial object of the Event	The user wants to see the information about the spatial objects of a specific event.
UC6	Sub-function	Multimedia Editor/Viewer	View the time of the Event	The user wants to see the information about the time that a specific event occurred.
UC7	Primary Task	Multimedia Editor/Viewer	View a Shot of the Event	The user wants to see the details of a specific shot.
UC8	Sub-function	Multimedia Editor/Viewer	Play the captured multimedia of the shot	The user wants to play the multimedia captured related to a specific shot and watch the topology if the event as it was captured by the user.
UC9	Sub-function	Multimedia Editor/Viewer	View the participants of the shot	The user wants to see the information about a participant of a specific shot.
UC10	Sub-function	Multimedia Editor/Viewer	View the spatial objects of the shot	The user wants to see the information about the spatial object of a shot.
UC11	Sub-function	Multimedia Editor/Viewer	View the time of the shot	The user wants to see the information about the time that a specific shot occurred.

4. THE REFERENCE SYSTEM MODULES

In this section we present the reference system modules that can describe the mobile multimedia event capturing and visualization system. Each module describes packages where similar functionalities of the system are kept together. The following figure shows the structure of the general system and the modules in which it can be divided. Next we describe what every different module represents.

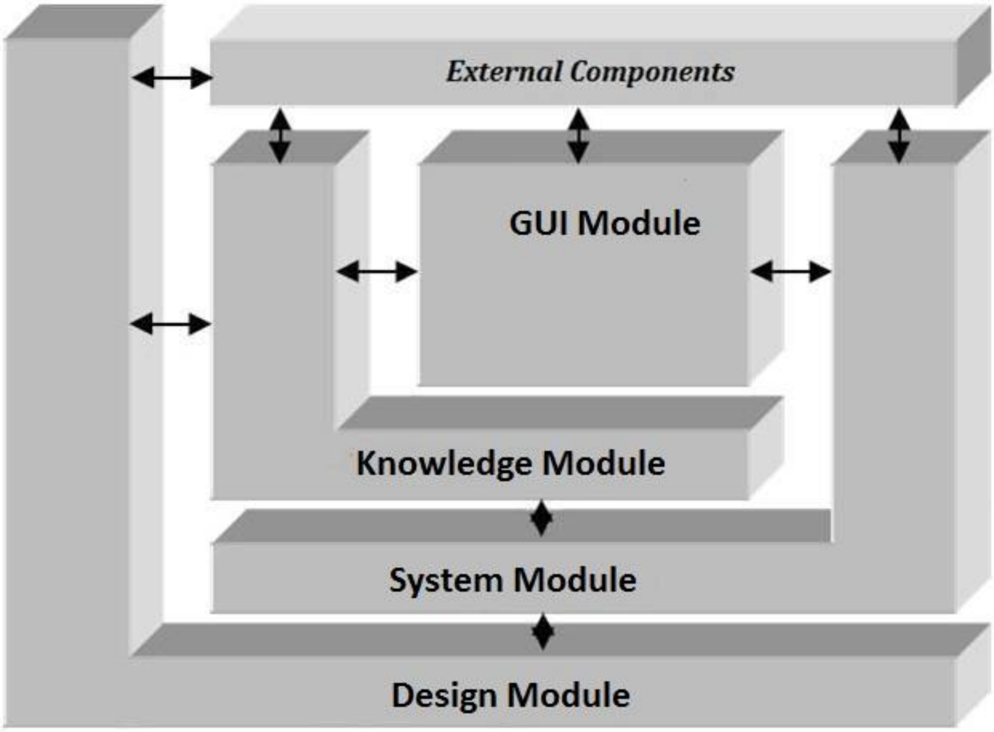


Figure 3.8 Reference System Modules

4.1. Knowledge Module

The knowledge part of the reference architecture shown on figure 6.1 contains the semantic knowledge along with the model upon which the application is built on. Hence, the ontology or ontologies used by the application system is a member of the knowledge part as well as the Complex Event model. The knowledge part communicates with the design part since the design part specifies the ontology or the ontologies of the application. It also communicates with the GUI part since the GUI part uses the ontology and the model to display the information to the user. Moreover it communicates with the system part to use it as interface for its communication with all the other parts of the reference architecture and finally it communicates with the external components part since it uses some of its components to create the ontology or ontologies of the application.

4.2. System Module

Seen from the viewpoint of implementation the system part represents the lowest layer of the application. It implements all system specific tasks needed to run any functionality within any part of the reference architecture. More specifically it contains the relational database that represents the storage of the application, the persistent layer package that implements the communication of the gui part with the storage, the entity layer package that is used to describe the Complex Event model, the ontology package that is used to read a specific ontology and the video connection package that manages the connection with the camera and the video player. The system part communicates with all the other parts as its main task is to act as an interface between the basic operating system and the higher parts of the authoring system.

4.3. Design Module

The design part of the reference architecture contains the needed tools for the concerning application. Thus, the design part is dependent on the specific application that the user wants to implement. For example, if the referred application is architecture related, then the design part will contain the creation of the floor plans of the building in interest. The design part can also contain the creation of the ontology or ontologies that are needed for the specific application. Therefore the design part communicates with the knowledge part. The design part also communicates with the system part because the components of the design part use the system part to communicate with the graphical user interface part. Finally, the design part communicates with the external components part since it uses some of these components, for example for the creation of the floor plans a cad tool of the external components is used.

4.4. GUI Module

The GUI part of the reference architecture shown on figure 6.1 contains the graphical user interface of the application. The graphical user interface is consists of two different sub-systems, the scenario creation sub-system and the scenario presentation sub-system. The scenario creation sub-system is the part of the interface that is used to create a new scenario and capture or play a video and save the new scenario to the database. The scenario presentation system is the part of the interface that is used to show a specific scenario and reproduce the video related to it along with the other information of the scenario. The GUI part communicates with the knowledge part with two different ways. The first way is to use the ontology of the knowledge part to help the user create a new scenario at the scenario creation sub-system. The second way is to use the Complex Event model of the knowledge part to guide the user input the information needed for the creation of the scenario. Furthermore, it communicates with the system part. The communication happens with the access of the gui to the storage of the system part. Moreover, the graphical user interface part communicates with the external components part as it uses some of the external components to display information to the user (e.g. the jmf plugin that is used to play and record video).

4.5. External Components

The external components part contains all the components that are not implemented by the application and are used to perform some specific tasks. As mentioned above the external components communicate with the design part but it also communicates with all the other parts of the reference architecture since some of its components are used in every part of the reference architecture, according the needs.

5. SUMMARY

In this chapter we will described the requirements of the system. First we described the different types of stakeholders involved in the system. After that we described the requirements of the system using the Use Cases Methodology. Finally we described the reference system modules where each of the functionalities of the system is organized and described.

CHAPTER 4
THE CONCEPTUAL MODEL

1. INTRODUCTION

The main objectives of the mobile multimedia event capturing model described in this Chapter, is to describe the functionality of a tool that facilitates creation of multimedia information together with its associated metadata in a systematic manner. The system is capable to support users to systematically capture multimedia information for different event types. The tool is also appropriate for use by naïve users as well as by users that are not expert cinematographers. To do that it provides expert advice for the cinematography practices in general, and the cinematography practices for specific event types. In addition metadata particular to the event types are systematically captured.

In particular, the model presented here, describes events of real world to be captured and classifies the events into Event Types. The elementary events can be captured by shots. The system suggests shot types appropriate for capturing particular event types.

The concepts of the model can be subdivided into four major components:

1. The concepts describing the events and the multimedia shots taken to capture those events.
2. The concepts that are used for the logical spatial representation of events and shots.
3. The concepts that are used for the physical representation of the events and shots described on top of the maps or other spatial representations like diagrams or 3D views.
4. The concepts that are used for supporting learning and preparation.

In the sequence we describe those concepts in brief using UML notation.

2. CONCEPTS DESCRIBING THE EVENTS AND THE SHOTS TAKEN

The UML diagrams involve classes and metaclasses. The metaclasses in UML are represented with UML Stereotypes with the name of the metaclass. Metaclasses in the UML diagram are connected with Composition Associations. Instances of the metaclasses are classes and they are connected with the Dependency Association "instance-of" with the metaclasses.

The composition semantics at the metaclass level imply that the whole structure is copied at instantiation. For example when we instantiate a composite event called "marriage" we also instantiate the events of "ceremony" and "reception" which are associated with the "marriage", as well as the roles of "bride", "groom", "bride's mother", "bride's father", "friend", etc. that are the roles associated with the "marriage". Thus every instance of the model creates a scenario, which describes a primary event of the real world.

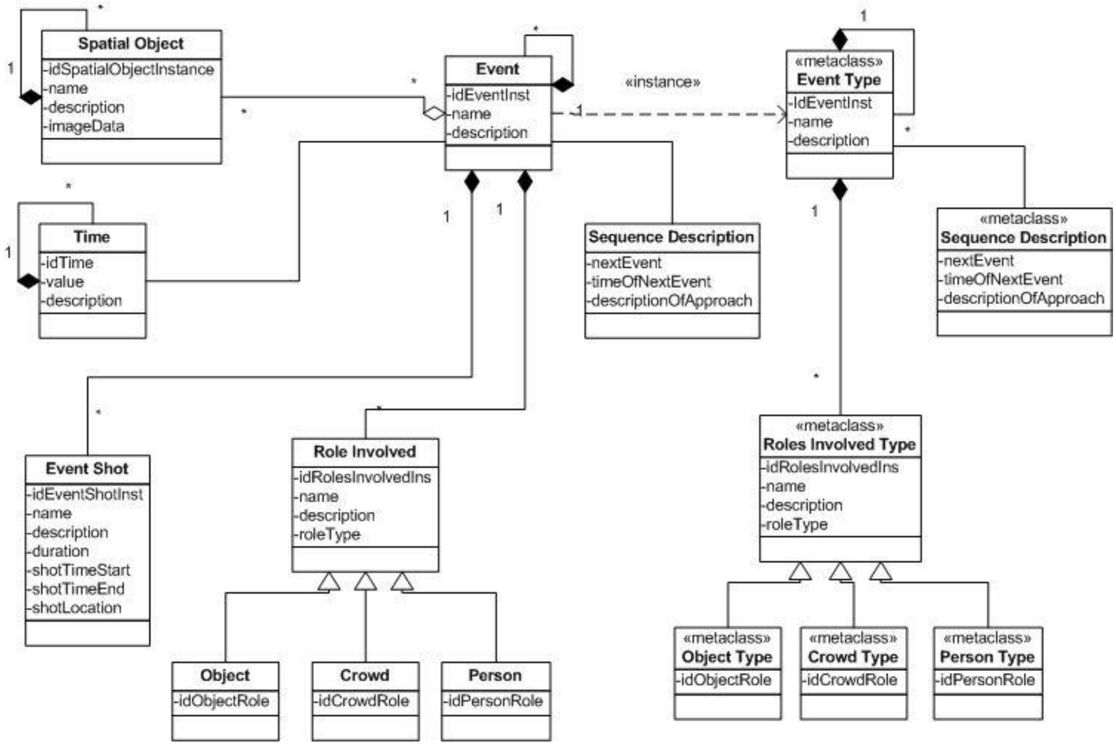


Figure 4.1 Concepts describing the events and the shots taken. The instantiations of an event would mean the instantiation of all the concepts associated with it.

In order to understand better the model of figure 3.1, let's take the marriage example mentioned before. In Figure 3.2 shows an example of an instantiation of an event with the name "Marriage", along with the events which compose the "Marriage" event.

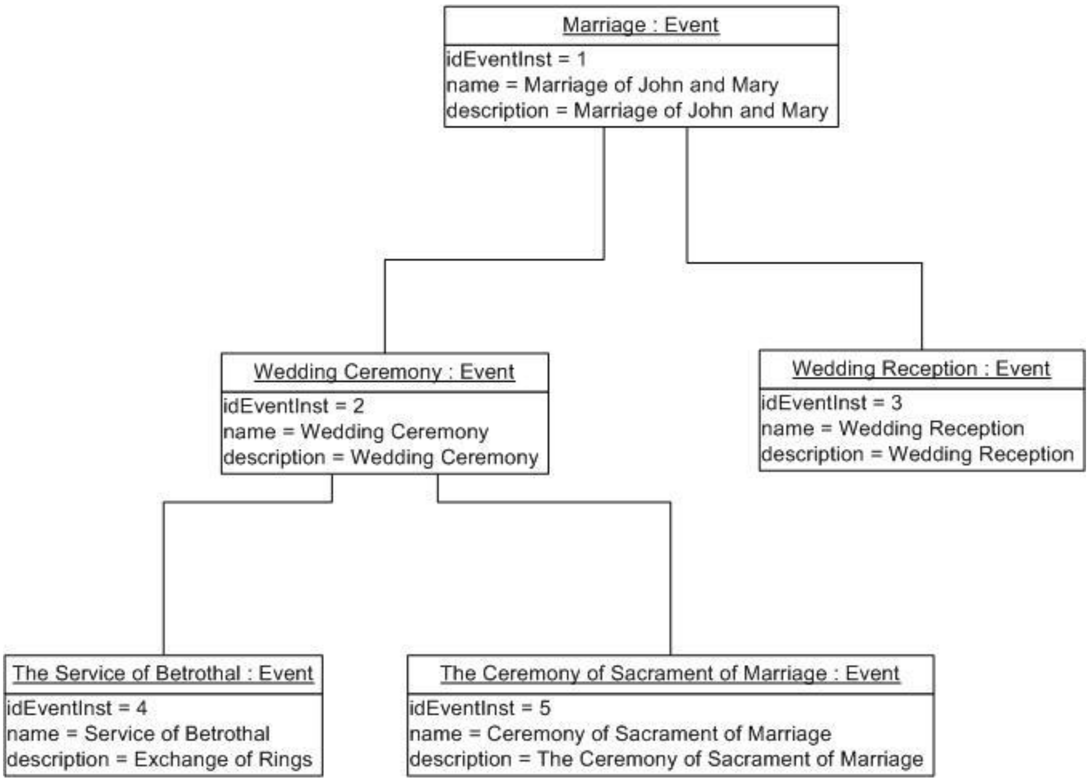

Marriage : Event
idEventInst = 1
name = Marriage of John and Mary
description = Marriage of John and Mary
Wedding Ceremony : Event
idEventInst = 2
name = Wedding Ceremony
description = Wedding Ceremony
Wedding Reception : Event
idEventInst = 3
name = Wedding Reception
description = Wedding Reception
The Service of Betrothal : Event
idEventInst = 4
name = Service of Betrothal
description = Exchange of Rings
The Ceremony of Sacrament of Marriage : Event
idEventInst = 5
name = Ceremony of Sacrament of Marriage
description = The Ceremony of Sacrament of Marriage

Figure 4.2 Example of the instantiation of the event "Marriage" demonstrating the hierarchy of the events.

In the class level specific actors (names of individuals, etc.) are associated with the event.

A specialization of the roles associated with an event, are the actors which are used to model the roles of persons that participate in the event. For example actors participating in the event "The Service of Betrothal" of the composite event "Marriage", should be "bride", "groom", "priest". Other actors, that participating in the wedding scenario may be "father of bride", "best man", "friend", etc. The actors have name and a description in text. An actor name in the marriage scenario could be groom. Note that a specific actor may be played by several different people in a specific event. For example the actor "friend" could be played by several persons in an instance of the marriage scenario. All those persons will be friends of the couple.

Roles other than single human roles can be represented. For example another specialization of the roles associated with an event is the crowd role. It is used to model larger concentrations of people, three persons and more, that play a role in an event without referring to each one in particular. For example in a composite event of the type "demonstration" the crowd roles "demonstrator crowd" and "police force crowd" may be modeled. In a class teaching environment the "student crowd" may be modeled. In the marriage case scenario, the crowd role models the guests attending the wedding ceremony, "attendants". Crowd has a name along with a description in text. Crowd location, role and movement are considered important in cinematography.

A third specialization of the roles associated with an event is the object role. This class is used to model other non-human object types that participate in an event. For example in a composite event of the type "car accident event" the object role "car involved in accident" may be modeled. In a composite event of the type "product announcement event" the object role "product announced" may be involved. Finally in the marriage scenario, in "The service of betrothal" event, participating object roles may be the "rings of engagement". An object role has a name along with a description in text.

The events may also be associated with structured Spatial Objects, Time of the event, and Event Shots that represent the multimedia capturing of the event.

The time associated with an event is referring to a specific time interval and it takes the relative sense of time. For example an instance of time in the marriage example could be "Wedding day". A time interval could contain other time intervals, creating a hierarchy as shown in figure 3.3. For example the time interval "wedding day" could contain the time intervals "wedding day morning" and "wedding day evening", in case there is a need for a more detailed description of the time interval in which a specific event related with the marriage scenario, is taking place. Figure 3.3 shows an example of the Time class hierarchy. An instance of the Time class may have a value along with a description in text.

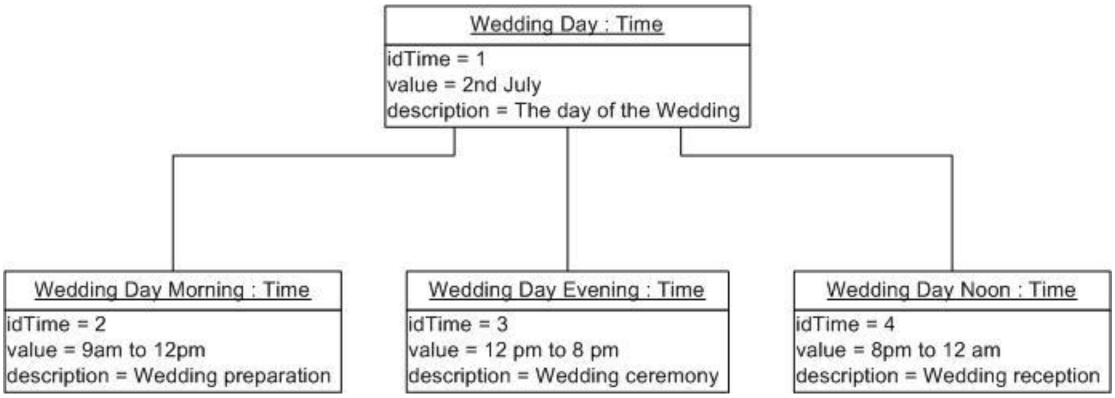

Wedding Day : Time
idTime = 1
value = 2nd July
description = The day of the Wedding
Wedding Day Morning : Time
idTime = 2
value = 9am to 12pm
description = Wedding preparation
Wedding Day Evening : Time
idTime = 3
value = 12 pm to 8 pm
description = Wedding ceremony
Wedding Day Noon : Time
idTime = 4
value = 8pm to 12 am
description = Wedding reception

Figure 4.3 Example demonstrating an instance of time in the marriage example and the hierarchy of the time intervals.

The spatial objects associated with an event are referring to all objects that have a spatial aspect on them and relate to an event, like buildings, rooms, etc. Spatial objects are semantic objects therefore they have a composition hierarchy which shows of what spatial objects, a spatial object is composed. Thus a spatial object can be seen at various levels of detail as one move down the composition hierarchy. For example, a Spatial Object may be a "one floor building". The one floor building Spatial Object may be composed of a Spatial Object of the type "corridor", and some Spatial Object of the type "room". A Spatial Object of the type room could be further decomposed into Spatial Object "door", "window", "wall", etc. An example of the composition hierarchy of the Spatial Objects is shown on figure 3.4.

The class should be able to answer questions related to composition hierarchy in order to facilitate navigation across the hierarchy levels. Figure 3.4 shows a relation between the class Event and the class Spatial Object. This relation is used to describe the location a specific event took place.

A Spatial Object can be described in various ways. Though, since a spatial object is referring to locations and objects with spatial aspect on them, the most common form of description is that of a floor plan or an image in general. Therefore the information that an instance of the Spatial Object class stores is a name for the spatial object, a description in text along with the URL of an image which displays the location, or the spatial information in general, where the Spatial Objects refers to. The model provides a more detailed description of a spatial object with the help of the Logical Spatial Representation class which will be described later.

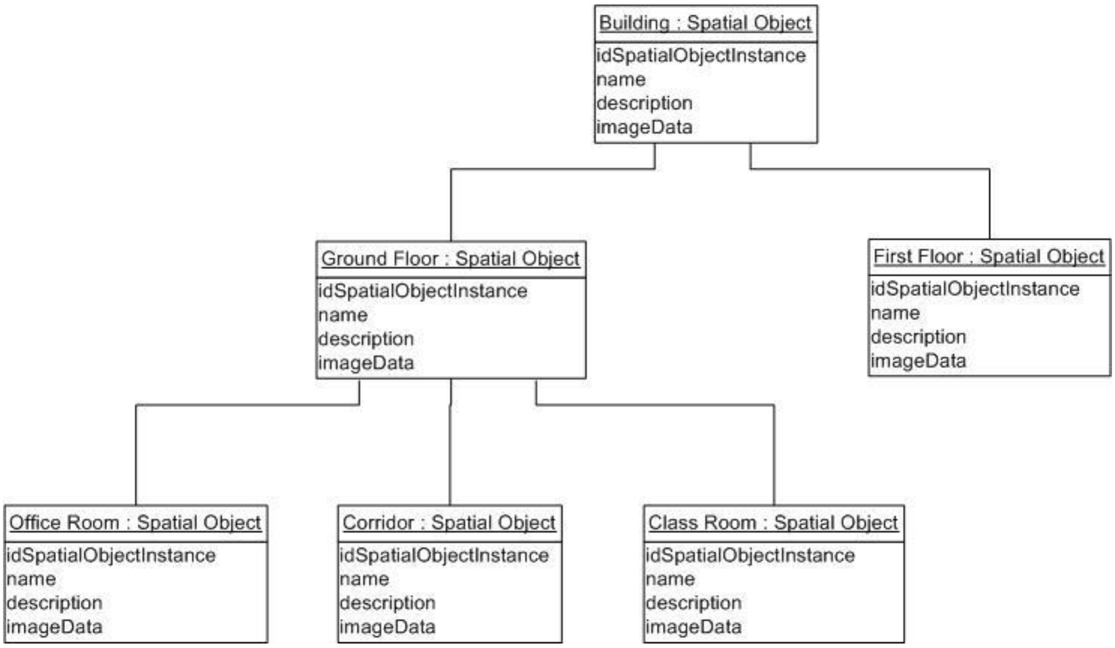


Figure 4.4 Example of a building as a spatial object and the composition hierarchy of the building.

To understand further the model of figure 3.1, a complete example of the marriage scenario is shown on figure 3.5.

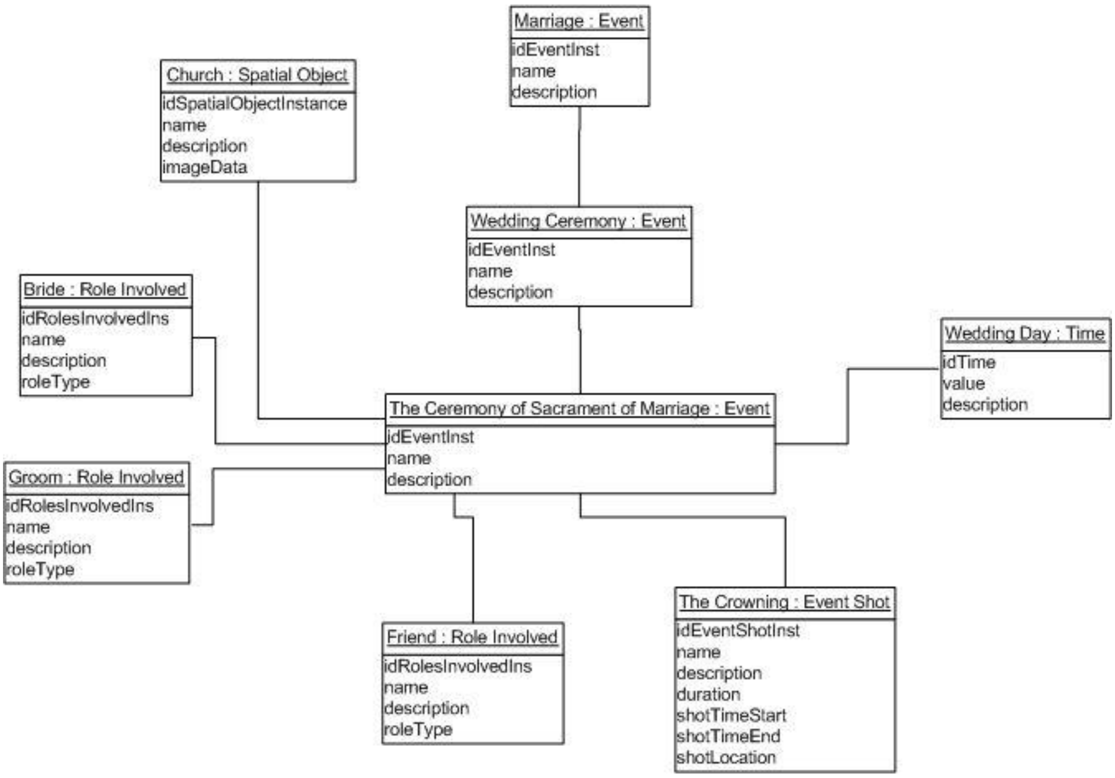


Figure 4.5 A complete example of the marriage scenario demonstrating the various concepts associated the "Ceremony of the sacrament of marriage" event.

3. CONCEPTS FOR THE LOGICAL SPATIAL REPRESENTATION OF EVENTS

Events are associated with physical spaces which may be described by spatial objects. These objects may be represented in the form of diagrams, etc. on a canvass with a semantic meaning provided by the spatial object itself (represented on the canvass with text or otherwise) or without any semantic meaning indication at all.

The events are associated with Roles which describe semantic participants of the event. The Roles are concepts which may describe humans, objects, crowd, etc. Thus they can be seen as an ontology for the Event Type. A concept may have more than one visual representation to represent different states or canvass resolutions. For example, a car may have states crashed or non-crashed. A person may be sitting or standing. Crowd may have different shapes at different times. A room may be represented simply with a square if the resolution is not high or with a detailed diagram if the resolution is high.

It is important to note that roles and objects participating in an event may have specific abstract graphical representations which may be reusable across events of the same type. For example in the recording of an accident event involving two cars, on a map or diagram of the road the location and movement of the cars should be recorded. The model allows reusable representations of cars and movement vectors to be used for that purpose.

The spatial representation of a simple event is described by the original position, size, and orientation of the semantic concepts of the ontology on the canvass and the time that they appear. A visualization of a concept is associated with the time that first appears within the event representation, and with the time that this visualization of the concept finishes as well as the position, size and orientation of the object at the time that it disappears. A trajectory may be associated with each object to describe the line of movement on the canvass that describes the event. Optionally, an object may be declared to be deformable, in which case the visual representation of the object at the time that it disappears has to be specified and the semantics imply that the shape of the object in intermediate positions can be found via interpolation between the original and the final appearance.

For a spatial object at any level of the composition hierarchy there are several different representations. (For example one may be an actual shape described by a polyline, but there should be others to represent the object on a canvass when its actual shape is not important, such as in case where only its presence in a location is important. Other shapes may be needed to describe the state of an object, its emotional state for example, the location and relative direction only, etc.)

Each Spatial Object may have Spatial Object Representations in the Spatial Object Maps (floor plans) of other Spatial Object. For example, a Spatial Object of the type room, in addition to its own Spatial Object Representation, it may have representations in other Spatial Object Maps, like the one floor plan model. The Spatial Object representation has a reference to the Spatial Object that it represents through an ID. The Spatial Object Map has a Shape and a Location on a canvass which will be described later on. The Spatial Object Representations may also have Shape and Locations which they use for their representations in different Spatial Object Maps.

The Logical Spatial Representation class shown on figure 3.6, associates all the different representations with a single spatial object. (The composition hierarchy is ignored at this level). The logical representation does not have actual sizes or measurements associated with the objects, only relative sizes or shapes. For example the shape of a fort is described with a polyline that has the shape of the fort, but not actual sizes in each side of the fort. There is an overall magnification factor associated with the object however. Geometric shapes like circles could be described using, a shape ontology.

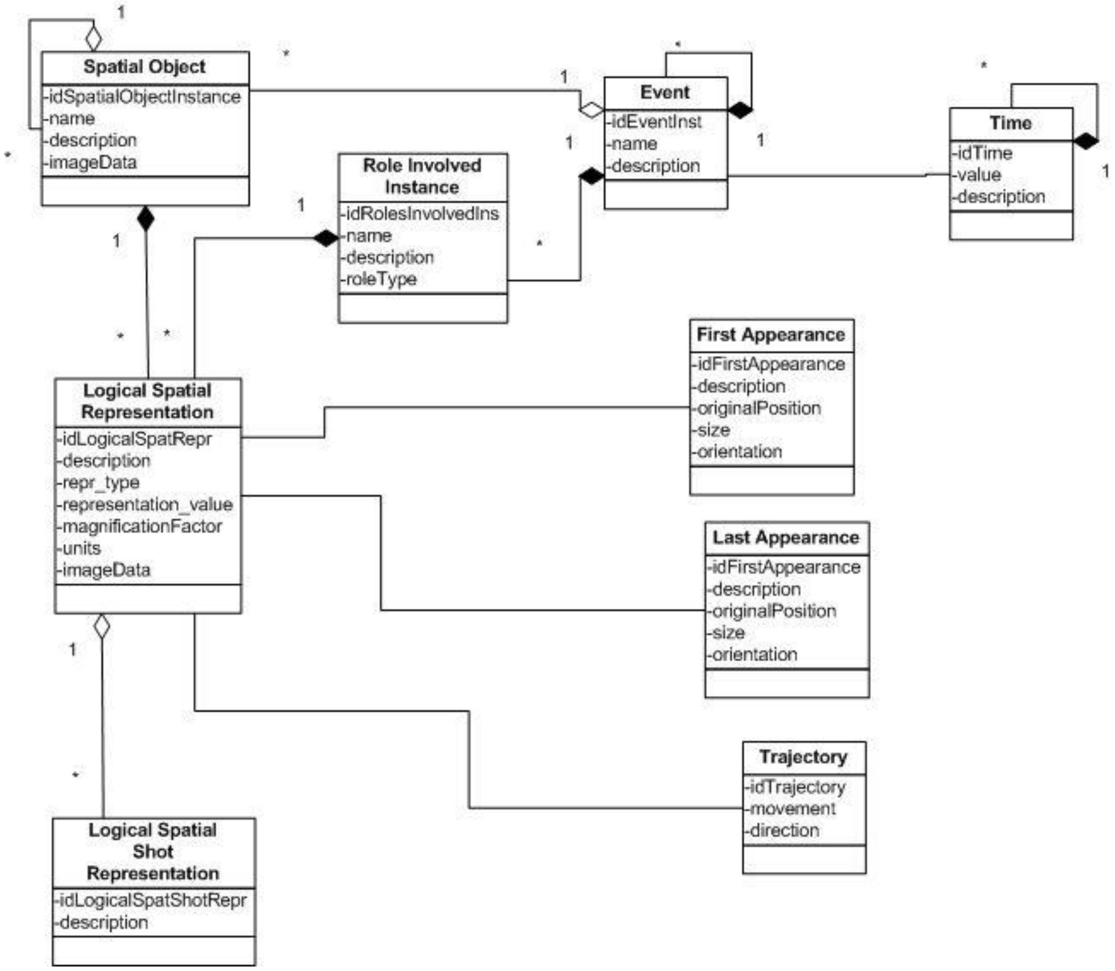


Figure 4.6 Concepts describing the logical spatial representations of events.

For example, in a car accident scenario, we could have a spatial object representation describing the space where the accident happened, which could be a map of the street. The roles would be the cars involved in the accident and an event could describe the cars after the accident. The logical representations, in that case, associated with this event would be the cars involved in the accident. The representation of a car could be a photo of the car crashed or a square posing as the car. The original position and the time of the appearance of the car would also be part of its representation.

In another example, e.g. in the marriage scenario, an event would be the arrival of the bride to the wedding. In that case, we could have a logical representation which would be the bride. Then the first appearance would be when the bride is walking out of the car and the last appearance when she enters the church. Both appearances would have locations and orientations. A trajectory could suggest the movement of the bride as she is walking towards the church.

4. CONCEPTS FOR THE LOGICAL SPATIAL REPRESENTATION OF SHOTS

A Shot in the media industry always refers to a simple event. It may capture the whole event or it may (typically) capture a part of the event. In some cases the event has been executed and the capturing refers to an event that has taken place. For example in an accident report, the event on which one reports has happened before, and the shooting refers to the event, but it does not capture the event itself. In the “marriage” scenario during the “The Ceremony of the Sacrament of Marriage” Event there is a need for more than one Event Shots. One typical Event Shot will show the “The Crowning”, where the couple wear the crowns. Another will show the “The Common Up”. A third Event Shot will show the “The Ceremonial Walk”, where the couple along with the priest and the best man walk in circle three times. Finally a last Event Shot will be “The Removal of the Crowns”. The figure 3.7 shows the Event Shots related to the “The Ceremony of the Sacrament of Marriage” Event described above.

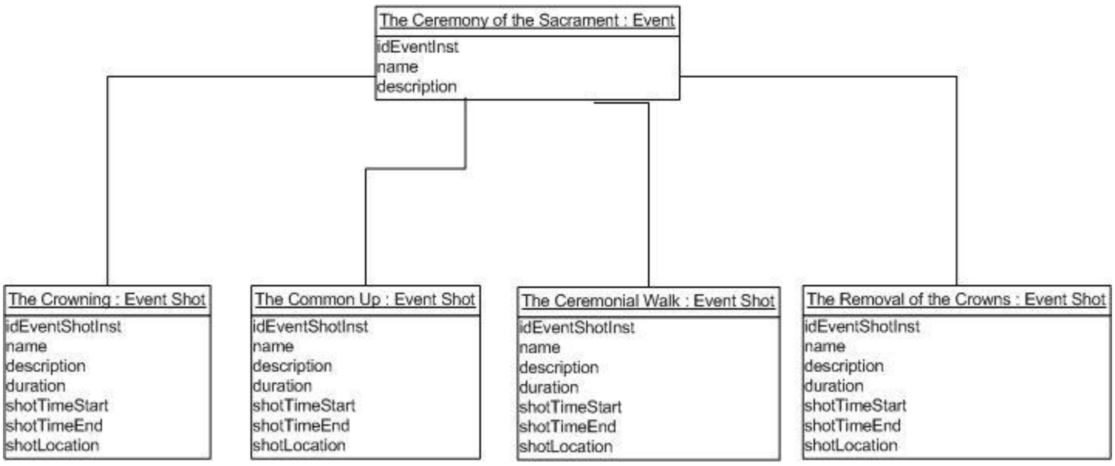


Figure 4.7 Event Shots Example

Each event Shot has a description which utilizes the names of the roles involved, relative time and locations related to the specific Event. In addition, the class Event Shot stores the duration of the specific shot, the timestamp of the video where the shot started, the timestamp of the video where the shot ended and the location from where the shot started capturing.

In the case of shots, in addition to the actors it represents the camera, lighting and sound parameters for the shot and their changes. The camera capturing is modeled with the location of the camera the direction of the capturing and the viewing angle which allows us to represent what objects are viewed by the camera in a specific setting. Similar modeling is used for light and sound sources.

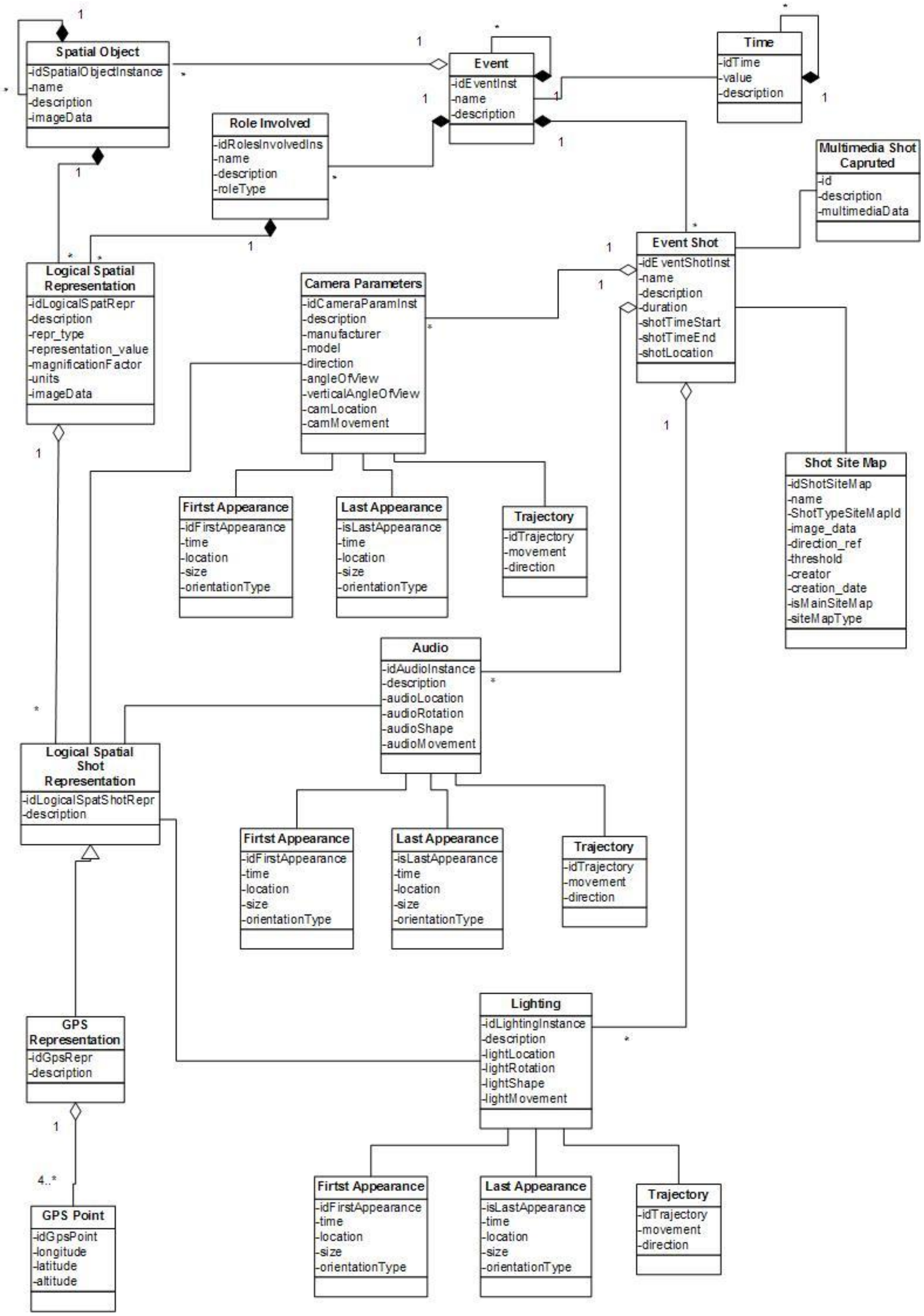


Figure 4.8 Concepts describing the spatial logical representations of shots.

Event Shots are used to represent types of Shots within an event type. For example an Event shot may be a shot of the “bride arrival” event using a zoom in as she enters the church. Event shots have shot spatial representations which represent the event shot on the canvas.

A shot spatial representation describes the original and final locations and parameters of the camera, the sound and the light equipment on top of the canvass, the same way that the event concepts were represented on top of the canvass. In this case the original and final representations of the capturing parameters are described. For example in a camera pan the original and the final position of the direction of the camera are described (together with the camera angle). This can be used to visualize the original and the final footprints of the pictures taken, but also to infer the intermediate positions.

The Logical Spatial Shot representation class shown on figure 3.8 is used to specify the composition of a canvass (shot representation). Although many objects may be in a shot, the representation may include only a subset of them. For example complex composition hierarchies may not show all the objects or object types involved, but keep only the most visible or important objects that have to be represented on a shot. Thus it is an abstraction mechanism that says what is to be represented. In addition, for each object that will be represented, a specific logical spatial representation of the object has to be selected to be included in the canvass. For example, a room may be represented with the actual shape of the room or with the symbol R within a circle or square to indicate a room.

GPS representation is a shot representation where the canvass has specific GPS locations in its four corners, and the object representations may have specific GPS locations associated with the vertices of the polylines that represent them. (This automatically determines location, orientation, and size within the canvass of some of the objects). Some of the objects may have other representations, not polylines, but for them, the location where they will be placed will have to be represented in GPS location.

Naturally, a shot is also associated with multimedia information which is related with a specific event. For instance, in a marriage scenario, this multimedia information will contain the video file or the photos of the wedding concerning a certain event. The multimedia information this class holds is the file URL along with a description of the multimedia content, in text.

For example in a “debate” event (which may involve shots from more than one location), the location of the camera , the viewing angle, and the direction of the shot in relation to the two people debating may be recorded for producing a training example of capturing principles for a debate. In another example, the systematic registration of damages on a house requires that the location, direction, and angle of the camera in relation to the house are registered.

In another example of a “marriage” event, “camera”, “groom”, “bride” may have specific reusable representations which can be placed on a specific diagram representing the spatial setting of the place where the ceremony of the marriage will take place in order to prepare the capturing. In this diagram the movement of the “groom” and “bride” during the shot may be represented. If the “one-man-crew” is moved during the capturing, his move may also be represented. This spatial representation may be reused by the same or another "one-man-crew" that is preparing for the capturing of another marriage event in the same setting.

Note also that the shape of representations may change during the capturing process. For example the shape of the crowd involved may change. The direction of the viewing of the camera as well as the viewing angle of the camera may change as result of panning and zooming operations on the camera by the "one-man-crew'. Interpolation is used to infer the intermediate positions of capturing

One important application of this part of the model is that it allows us to represent the spatial evolution of an event, as well as the spatial evolution of the capturing of those events. For example it may be used for capturing the history (event) of a visit in a museum or a zoo, and the capturing of video that the "one-man-crew" visitor took during the visit.

The parameters captured by the model facilitate the description of the event and the shot taking in an abstract spatial space. For example the relative positions may reflect the different gps locations or relative distance in meters reflected on a drawing canvass. Location in buildings such as museums may also be captured with respect to the location of objects that have been recorded in diagrams of the building for example. This could be automatic, for example by using RFID identifiers for the objects, the location of which have been recorded in electronic diagrams. Note also that direction may be determined with respect to geographic coordinates (by a compass for example) or by recording the position of the camera and the position of another object in the direction of the camera (by recording its gps position or its RFID).

5. THE CONCEPTS USED FOR PHYSICAL REPRESENTATION OF EVENTS AND SHOTS

The logical spatial representations described in the previous section may have to be viewed on top of a map. It is also possible to utilize 3D maps (for example the NASA maps) to represent on them the camera location, movement and angle of view (48) (49).

The concepts of this part of the model are responsible for doing the mapping from the logical representation to other representations as required (one or more). To do this rotation translation and scaling of the logical representations may be required. The logical representations mapped on the site map are called physical representations. The physical representations would have a location on the site map as well as shape and possible movement.

Shot site map is representing a drawing canvass with x and y dimensions where the various physical representations will be represented for a specific event or shot. It is used to describe the geography of the scene where a specific event or an event shot took place. The most common way laying the physical Representations on the canvass is to use the plan (e.g.

the map) of the spatial object associated with the event or shot, as a background and then to put on the canvass the physical representation of the roles involved. For modeling the spatial objects and the roles involved a two dimensional site map is enough. Besides them though, on top of the site map the technical parameters of the shooting are being represented, the camera, audio and lighting parameters. To model the Camera capturing and to take into consideration all the parameters of the camera, there is a need to use more than one site maps for a specific event or shot that will represent elevations of the location associated with it. The model considers four types of elevations, one for the four points of the horizon, north, south, west and east. In this way the model uses two dimensions to model three-dimensional spaces.

A specialization of the shot site map is a geographic map. These kinds of maps are used for exterior locations when a specific map is chosen for usage and the detailed description of the location of the capturing is not needed. The class Digital elevation model is a class used to store the altitudinal data.

For example, in “the crowning” event shot the logical representations of the priest, the best man, the bride and the groom would be mapped to physical representations and put on top of the site map. The spatial object associated with the specific event shot would be the interior of the church. The logical representation of the church would be mapped to a physical representation and this physical representation would be used as a background to the site map. The relative position and movement of the camera capturing the shot will also be represented on the shot site map. These representations present the relative positions and orientations of the various roles, and thus they reveal the geometry of the scene.

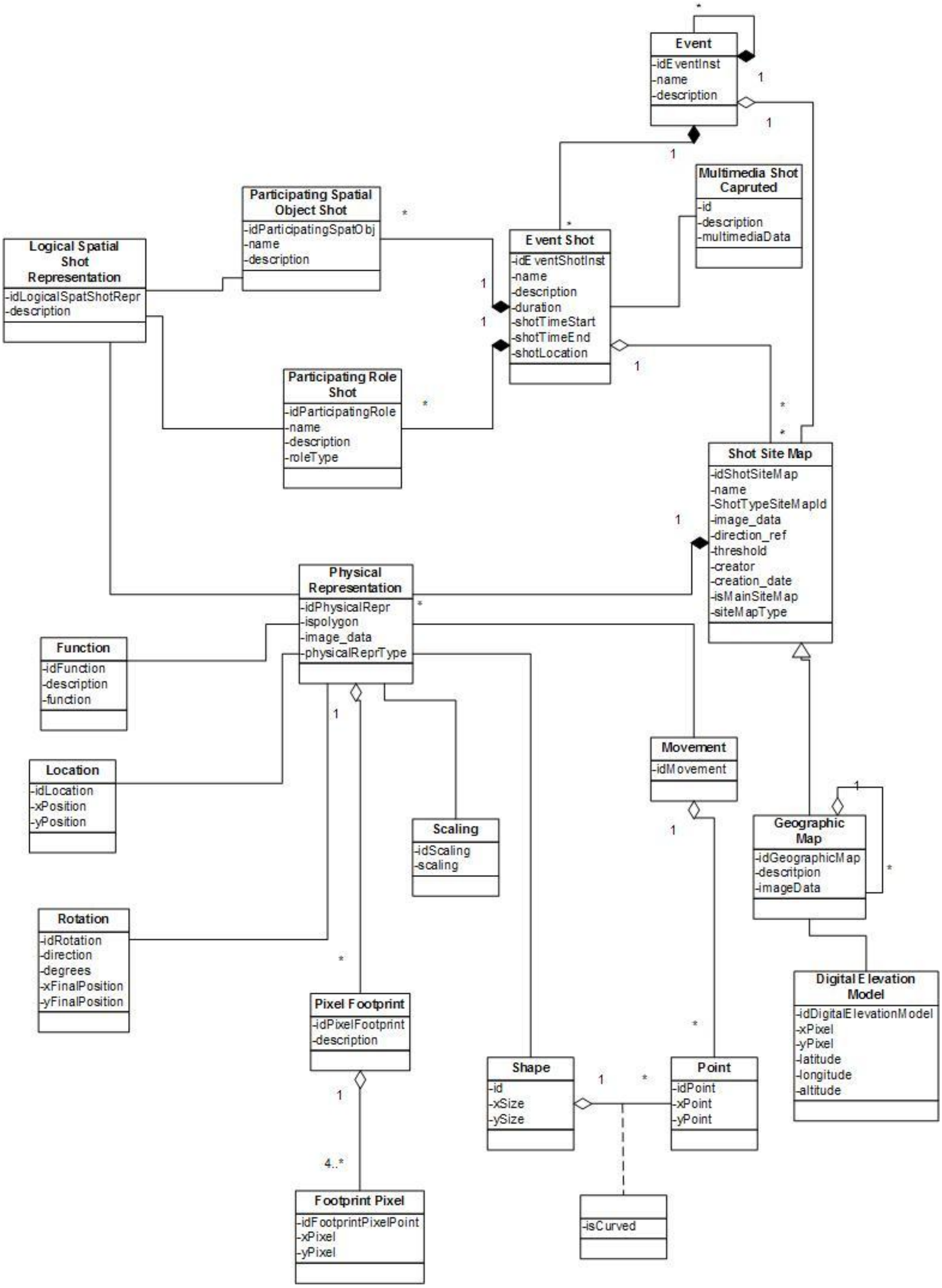


Figure 4.9 Concepts describing the physical representations of events and shot on top of a site map.

Figure 3.9 also shows the Physical Representation class. The Physical representation shows which objects will be represented on the Shot Site Map and with what shape. The Physical representation shows for each one of those objects where it will be represented (location), in which direction with respect to the canvass (rotation) and in what relative size (scaling). In addition, in case that there is no object representation specified, there may be a function that is used to draw the object, or some other input.

To describe the location of an object the Location class is used. The class specifies locations using xPosition and yPosition parameters. Measurements could be in pixels from the bottom left position of a canvass, or other measurements (e.g. GPS, cm, etc).

The Rotation class is used to describe the rotation of the original object shape in order to be placed in the canvass. The values needed to perform a rotation are the direction of the rotation and the degrees of the turn.

The Scaling class is used to show the scaling that an object may need to be put on the Shot Site Map. To perform the scaling, the scaling value is needed.

The function class is used when there is a need to perform a function in order to put an object on the Shot Site Map, when there is no representation provided for the specific object.

The Point class is used to model every point of the Shot Site Map canvass using x and y coordinates (xPoint and yPoint as shown on figure 3.9).

The Shape class is used to model the shape of a specific object. In case the shape can be described by the xSize and ySize parameters, the specializations of the Shape class are not used. However, when the shape of the object is more complicated, the two specializations of the Shape class are used. The Ellipse class is used to model all the objects that have an elliptical shape (e.g. circle). This class needs a point to be the center of the ellipse and an xSize and an ySize parameter. The center reveals and the orientation of the object in that case. The other specialization of the Shape class is the Polyline class. This class is used to model all the objects that arbitrary shape. In this case the shape of the object will be a set of lines. The isCurved Parameter located on the relation of the Shape class with the Point class is used in case the line connecting two points is not straight but curved. This way, even an object with a half moon shape will be represented on the Shot Site Map.

The Movement class is used to describe the movement of the objects during the shot capturing if any. The movement is modeled by a set of points on the Shot Site Map.

As mentioned before, the technical parameters of the capturing are also shown on the Shot Site Map. Because of that, the lighting and audio classes use the Location, Rotation, Shape and Movement classes to completely describe the lighting and audio setup and to specify their position in the drawing canvass. The Camera Parameters class uses the Location and Movement classes to describe the position of the camera and the movement of the camera, either the horizontal or the vertical movement (panning and tilting).

This information about the capturing process may be useful for visualization reasons, for example for understanding what is seen in the video. In some cases this information may be also useful for image registration and 3D reconstruction of Spatial Objects. For example, in capturing the video of a room, the recording of camera location and camera movement (pan) parameters can be used for reconstructing a 3D view of the room.

Note that some of those parameters may be automatically captured in outdoors using for example the EXIF data and GPS and compass information integrated to the camera as we have done it for the Ricoh Caplio 500SE camera.

6. THE CONCEPTS USED FOR TRAINING AND PREPARATION

The concepts used for training and preparation essentially do two things. First help in the recognition of the upcoming events providing contextual information that depends on the event type and second to record principles and guidelines for the capturing that have been found useful for the event type with various shot types.

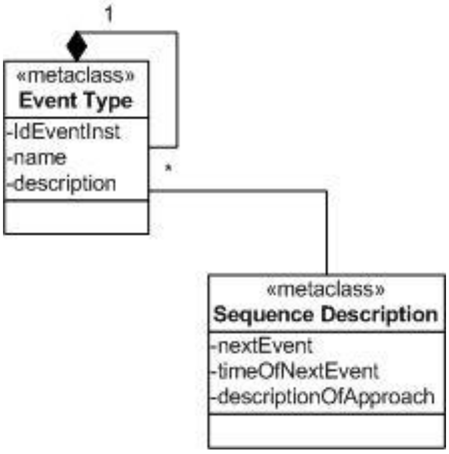


Figure 40 Concepts describing the training and preparation

To help with the preparation and demarcation of the upcoming events a sequence of events is recorded (if a sequence is known to exist). This sequence is used to drive the interfaces not only for the demarcation of the beginning and the end of the events, but also with the preparation of the completion of metadata that is necessary to be captured for the event and the shot. If some clues are known that indicate that the event is about to happen these are recorded for consultation. For example, in a wedding ceremony the time that a particular event is going to happen is rather unpredictable. However, the words spelled by the priest indicate the state of the mystery, and help approximate the time that a particular sub-event that may be appropriate for shooting may happen. This information helps in the preparation of the one-man-crew.

In other cases the order of the event happening and their exact timing may be unpredictable. When the event happens the one-man-crew should be able to identify fast the metadata that need to be completed and complete them as well as be ready to start shooting parts of the event. Again demarcation of the beginning and ending of events is important.

In addition to preparation for specific events and their shooting, generic shot types appropriate for generic training in multimedia capturing are useful. These are known to the multimedia industry and are documented in various books. It is useful to provide those as examples for learning principles and guidelines, especially for people that do multimedia capturing without having some formal training.

These can be viewed from the “one-man-crew” independent on the particular task at hand.

However, it is also very important to be able to explain the recommendations for specific suggestions for specific types of shooting in a given event type with reference to the generic principles and guidelines.

7. SUMMARY

In this chapter we described the mobile multimedia event capturing and visualization conceptual model that we created in the current thesis. We divided the concepts of the model in four different types and described each one of those types.

CHAPTER 5 THE IMPLEMENTATION MODEL

1. INTRODUCTION

The previous chapter describes the mobile multimedia event capturing conceptual model which we created for facilitating the creation of multimedia information with its associated metadata in a systematic manner. However, during the creation of an application scenario, which is also described in the previous chapter, this model is used more than one times, according to the needs of the application.

Next, we describe the abstract (core) ontology we created which is also a part of the knowledge module of the reference architecture. Then, we describe the process of creating an event type in the form of ontology along with two examples of complete composite event types in the form of ontologies.

Finally, we describe the relational database of the system module of the reference architecture. The entity-relation model of the relational database relied on the mobile multimedia event capturing and visualization model. The database is used for the physical storing and retrieving the multimedia and its associated metadata captured by the application.

2. CREATING AN EVENT TYPE

The conceptual model is described in UML which is the standard for software engineering requirements analysis and design. Designers may decide to implement this model in alternative ways, depending on what language they are familiar with. One such language is OWL-DL. We use owl as representation language. As we described in Chapter 2, OWL (Web Ontology Language) is an RDF-based language used to create Semantic Web ontologies. Ontologies indicate the hierarchies and relationships that exist between different resources within a specific domain. That enables us to create event types/ontologies, using an abstract ontology, for every different real-world scenario. We implement the implementation model on top of the Protégé platform which gives us the ability to construct ontologies using OWL – DL.

The purpose of the creation of a (composite) event type, which is a real life application scenario, is to guide the user during the capturing process. Every different event type may have its own capturing conditions, which usually differ from other, different event type's, conditions. Hence, to be able to guide the user during the capturing and feeding him with appropriate instructions, for every different event type, an ontology is created with the help of the owl language and the protégé platform.

2.1. CORE ONTOLOGY

To assist in the event type creation, a core ontology was created. This core ontology was created according to the specifications of the mobile multimedia event capturing and visualization logical model described on the previous chapter. Therefore it contains all the classes belonging to the metamodel described on the previous chapter as well as all the corresponding parameters of the classes.

The creation of the core ontology was implemented in the following way. For every metaclass of the model, an owl class was created. For every parameter of the metaclasses, a data property at the corresponding owl class was created. For every relation of the model, an object property was created at the side of the class with the composition type of the relation. Finally for every hierarchy relation of the model, corresponding owl subclasses were created. The core ontology was also created using the protégé platform.

For every different application scenario, a new event type, which is a scenario specific ontology, is created using the core ontology and specializing its classes and properties. For example if a new application is a "wedding" scenario, an ontology for the wedding scenario is created. In the event type for the application, the composition of events and the roles of participants of the wedding are described. This process corresponds with the definition of a wedding metadata type in the conceptual model of the previous chapter.

The following table contains a list of the classes of the core ontology along with their object and data properties. The description of each class has been given on the previous chapter.

Owl Class	Data properties	Object Properties
Event Type	Id Name Description	happens At takes Place At has Shot has SubEvent has Participant
Roles Involved Type	Id Name	has Role Representation
↳**Person Type**	Id Description	
↳**Object Type**	Id Description	
↳**Crowd Type**	Id Description	
Spatial Object	Id Name Description	composed Of has Spatial Object Representation
Time	Id Value Description	has SubTime
Event Shot	Id Name Description Shot Time Start Shot Time End Duration	shots Participating Roles shots Cam Parameters shots Lighting shots Audio has Site Map shots Participating Spatial Objects
Camera Parameters	Id Description	camera Rotation camera Shape camera Location camera Movement
Lighting	Id Description	lighting Rotation lighting Shape lighting Location lighting Movement
Audio	Id Description	audio Rotation audio Shape audio Location audio Movement

2.2. EVENT TYPE – ONTOLOGY

The core ontology, as described above, is used for the creation of the event types in the form of ontologies that will be used as scenarios for the different types of capturing needs (event type-ontology). An event type – ontology is created using the core ontology and specializing its classes and properties. That is, every class of the event type is created as a subclass of the corresponding core ontology's class. For example at the marriage example that is used on the previous chapter, the "Marriage" event types will be a subclass of the

Event Type class belonging to the core ontology. The events types “Wedding Ceremony”, “Wedding Reception” and every other event of the marriage example, will also become subclasses of the Event Type class of the core ontology. The same will continue for every other class the creator considers necessary. Beside classes, the object properties will also follow the same specialization process. For example, even though there is an object property that describes the relation between the Event Type and the Roles Involved Type, the “Wedding ceremony” event must have an object property that describes its relation with the specializations of the Roles Involved Type such as the “Bride”, the “Groom” or the “Priest”.

In that way, the user creates the desired event type that will later be used for the capturing of the metadata related to the multimedia the user wants to capture.

The classes that are with the most importance in the phase of the creation of the event type – ontology are those that are directly related to the Event class and not those that are related with the representations of the shots as described on the previous chapter. That makes sense because information about the representation of the spatial objects and roles involved in a capturing are not very useful to be given before the capturing process. On the other hand, the information related to Events, the Spatial Objects, the Roles Involved, the Time and the Shots needed for a specific Event are almost necessary to be given before the capturing because in the cinematography that king of planning is necessary for a proper video (or any other kind of multimedia) capturing. For a specific event type, the events that will be included, the location, the time and even the persons and objects involved are more or less specific and known, so a event type – ontology that fully covers the entire event related information can be created beforehand. Also, the user that performs the capturing saves time and completes his work a lot faster especially if the scenario is described entirely in detail. Suppose, however, that an event is not exactly as described by its type. For example, in the marriage event, they decide to do a party the night before the wedding, to accommodate the guests that will come from other places. The user, during the capturing process, can select either to use an already existing event type from the event type – ontology, or to add a new event like the “party at night before” if the event he wants to capture does not exists in the event type – ontology.

Finally, throughout the creation of the event type – ontology, besides the event related information, information related with the capturing guidelines is required. At the previous chapter we described the functionality and usefulness of the classes related to the technical parameters of the capturing. At this point, however, the exact technical parameters are not possible to be provided because they are not known yet, the capturing has not yet happened. So, the classes Camera Parameters, Lighting and Audio are used not to provide the specific technical parameters of the capturing but to provide guidelines to assist the user during the capturing. The guidelines follow the principles described on a previous section and are created based on the principles of the cinematography. The creation of the guidelines is done with the following principle. For every shot class that is created in a specific scenario, a Camera Parameters class, a Lighting class and an Audio class are created. In each one of these classes, besides their name they will have a comment where in text form guidelines for the specific shot will be given.

2.1. AN EVENT TYPE ONTOLOGY – SCENARIO EXAMPLE

The marriage example described above is an event focused scenario example. Event focused scenarios are those that focus more on the event related part of the scene captured than the spatial representation of the scene. These types of event types are used mostly for video capturing that focuses on human integrations just like a marriage, an engagement or even a soccer game and a video describing an accident. Next the event type – ontology created for the marriage example will be presented as a distinctive example of the event focused scenarios.

Owl Class	Data properties	Object Properties
Event Type	Id	happens At
↳ Wedding	Name	takes Place At
↳ Bride Preparations	Description	has Shot
↳ Bride Leaving the house		has SubEvent
↳ Makeup and hair		has Participant
↳ Relatives greeting		
↳ Wedding dress rehearsal		
↳ Wedding Preparation customs		
↳ Bride’s arrival		
↳ Delivery of Wedding dress		
↳ Making of wedding bed		
↳ Wedding ceremony		
↳ Wedding reception		
↳ Wedding pictures		
↳ Wedding portion display		
↳ Wedding validation		
Roles Involved Type	Id	has Role Representation
	Name	
↳ **Person Type**	Id	
↳ Best man	Description	
↳ Bride		
↳ Bride’s father		
↳ Bride’s mother		
↳ Coiffeur		
↳ Cosmetician		
↳ Groom		
↳ Groom’s father		
↳ Groom’s mother		
↳ Priest		
↳ **Object Type**	Id	
↳ Bride’s shoes	Description	
↳ Glass of wine		
↳ Wedding curl		
↳ Wedding dress		
↳ Wedding portion		
↳ Wedding rings		

↳ Crowns		

↳ **Crowd Type**
  ↳ Attendants
  ↳ Bride’s friends
  ↳ Bride’s relatives
  ↳ Groom’s friends
  ↳ Groom’s relatives
	Id
	Description	

Spatial Object ↳ Bride’s parent house ↳ Church ↳ Church’s interior ↳ Church’s yard ↳ Couples new home ↳ Reception hall ↳ Wedding bed	Id Name Description	composed Of has Spatial Object Representation
Time ↳ A week before wedding day ↳ One day before wedding day ↳ Three days before wedding day ↳ Wedding day ↳ Wedding day afternoon ↳ Wedding day morning ↳ Wedding day noon	Id Value Description	has SubTime
Event Shot ↳ Bride breaks a curl in 4 pieces ↳ Best man takes bride to church ↳ Coiffeur making bride’s hair ↳ Cosmetician making up the bride ↳ Bride greeting Friends and relatives ↳ A friend wearing the Dress ↳ Friends write their names on in wedding shoes ↳ Father of bride puts on bride her shoes ↳ Bride walks out of the car ↳ Father delivers bride to the groom ↳ Groom’s friends delivers dress to the bride ↳ Friends throw money on wedding bed ↳ Unmake the bed three times ↳ Child Sits on wedding bed ↳ Service of Betrothal ↳ Ceremony of the Sacrament of Marriage ↳ The crowning ↳ The Common Cup ↳ The Ceremonial Walk ↳ The Removal of the Crowns ↳ End of ceremony ↳ Couple arrives at wedding party	Id Name Description Shot Time Start Shot Time End Duration	shots Participating Roles shots Cam Parameters shots Lighting shots Audio has Site Map shots Participating Spatial Objects

↳ Groom’s relatives dance ↳ Bride’s relatives dance ↳ Couple greet every table ↳ Couple leaves the wedding party ↳ Arrival to the new house ↳ Memory pictures of couple ↳ Wedding portion washing ↳ Wedding portion ironing ↳ Wedding portion display to friends ↳ Couple signing to validate wed		
Camera Parameters	Id Description	camera Rotation camera Shape camera Location camera Movement
Lighting	Id Description	lighting Rotation lighting Shape lighting Location lighting Movement
Audio	Id Description	audio Rotation audio Shape audio Location audio Movement

The second type of scenarios examined in the current thesis, are scenarios that describe the topology of buildings or spatial objects generally, spatial objects type scenarios. Those scenarios focus more on the spatial representations part of the scene captured than the events related part of the scene. These types of scenarios are used mostly for video capturing that focuses on building description. For example, filings reports on building conditions or describe the interior of a structure. Next the ontology – scenario created for the building example will be presented as a distinctive example of the spatial representation type scenarios.

Owl Class	**Data properties**	**Object Properties**
Complex Event Type ↳ Building	Id Name Description	has Event
Event Type ↳ Building ↳ Building entry ↳ Building entry hallway ↳ Class room ↳ Class room hallway ↳ Office room ↳ Office room hallway ↳ First floor ↳ First floor hallway ↳ First floor classroom	Id Name Description	happens At takes Place At has Shot has SubEvent has Participant

↳ First floor office room ↳ Second floor ↳ Second floor hallway ↳ Second floor classroom ↳ Second floor office room ↳ Stairs		
Roles Involved Type	Id Name	has Role Representation
↳ **Person Type**	Id Description	
↳ **Object Type**	Id Description	
↳ **Crowd Type**	Id Description	
Spatial Object Type ↳ Building entry ↳ Building entry hallway ↳ Class room ↳ Class room hallway ↳ Office room ↳ Office room hallway ↳ First floor ↳ First floor hallway ↳ First floor classroom ↳ First floor office room ↳ Second floor ↳ Second floor hallway ↳ Second floor classroom ↳ Second floor office room ↳ Stairs	Id Name Description	composed Of has Spatial Object Representation
Time Type	Id Value Description	has SubTime
Event Shot Type ↳ Building entry shot ↳ Building entry hallway shot ↳ Class room shot ↳ Class room hallway shot ↳ Office room shot ↳ Office room hallway shot ↳ First floor shot ↳ First floor hallway shot ↳ First floor classroom shot ↳ First floor office room shot ↳ Second floor shot ↳ Second floor hallway shot ↳ Second floor classroom shot ↳ Second floor office room shot	Id Name Description Shot Time Start Shot Time End Duration	shots Participating Roles shots Cam Parameters shots Lighting shots Audio has Site Map shots Participating Spatial Objects

↳Stairs shot		
Camera Parameters Type ↳Building entry camera parameters ↳Building entry hallway camera parameters ↳Class room camera parameters ↳Class room hallway camera parameters ↳Office room camera parameters ↳Office room hallway camera parameters ↳ First floor camera parameters ↳First floor hallway camera parameters ↳First floor classroom camera parameters ↳ First floor office room camera parameters ↳ Second floor camera parameters ↳ Second floor hallway camera parameters ↳ Second floor classroom camera parameters ↳ Second floor office room camera parameters ↳Stairs camera parameters	Id Description	camera Rotation camera Shape camera Location camera Movement
Lighting Type ↳Building entry lighting ↳Building entry hallway lighting ↳Class room lighting ↳Class room hallway lighting ↳Office room lighting ↳Office room hallway lighting ↳First floor lighting ↳First floor hallway lighting ↳First floor classroom lighting ↳First floor office room lighting ↳Second floor lighting ↳ Second floor hallway lighting ↳Second floor classroom lighting ↳Second floor office room lighting ↳Stairs lighting	Id Description	lighting Rotation lighting Shape lighting Location lighting Movement
Audio Type ↳Building entry audio ↳Building entry hallway audio ↳Class room audio ↳Class room hallway audio ↳Office room audio ↳Office room hallway audio ↳ First floor audio ↳First floor hallway audio ↳First floor classroom audio ↳ First floor office room audio ↳ Second floor audio ↳ Second floor hallway audio	Id Description	audio Rotation audio Shape audio Location audio Movement

- ↳ Second floor classroom audio
- ↳ Second floor office room audio

↳ Stairs audio

3. IMPLEMENTATION MODEL

The mobile multimedia event capturing and visualization model, we presented in chapter 4, is also used for the physical storage of the information obtained with the application usage. The physical storage is done in a relational database management system (RDBMS). This RDBMS is part of the reference architecture's system module we described in a previous section.

3.1. RELATIONAL DATABASE IMPLEMENTATION

The implementation model of the relational database has been constructed according to the event capturing and visualization model described on the previous chapter. The implementation of the relational database was also in the contents of the thesis and we did not use any automated process for the creation of the database.

The user utilizes the event type – ontology in order to assist him during the capturing process. However, during the capturing the user creates instances of the event type – ontologies and those instances are stored in the relational database in order to retrieve them when necessary. For example, at the "Marriage" scenario, the user utilizes the event type – ontology that concerns the "Marriage" and the stores in the relational database the "Marriage of John and Mary".

The following tables constitute a full description of every table of the database scheme.

<table>
<tr><th>Table</th><th colspan="2">Description</th></tr>
<tr><td><i>Audio instance</i></td><td colspan="2">The table that stores the technical parameters used during the capturing by the audio equipment.</td></tr>
<tr><th>Attribute Name</th><th>Type</th><th>Description</th></tr>
<tr><td>IDAudioInstance</td><td>Int (10)</td><td>The primary key of the table.</td></tr>
<tr><td>Description</td><td>Text</td><td>A description about the audio parameters.</td></tr>
<tr><td>FkLocationID</td><td>Int(10)</td><td>Foreign Key to the Location table.</td></tr>
<tr><td>FkRotationID</td><td>Int(10)</td><td>Foreign Key to the Rotation table.</td></tr>
<tr><td>FkShapeID</td><td>Int(10)</td><td>Foreign Key to the Shape table.</td></tr>
</table>

FkMovementID	Int(10)	Foreign Key to the Movement table.

Table	Description	
Camera parameters instance	The table that stores the technical parameters used during the capturing by the camera equipment.	
Attribute Name	**Type**	**Description**
IDCameraPArametersInstance	Int (10)	The primary key of the table.
Description	Text	A description about the camera parameters.
FkLocationID	Int (10)	Foreign Key to the Location table.
FkRotationID	Int (10)	Foreign Key to the Rotation table.
FkShapeID	Int (10)	Foreign Key to the Shape table.
FkMovementID	Int (10)	Foreign Key to the Movement table.
Manufacturer	Varchar(128)	The manufacturer of the camera used.
Model	Varchar (128)	The specific model of the camera used.
Angle of view	Double	The horizontal angle of view of the capturing.
Coc	Double	Defines how much a point needs to be blurred in order to be perceived as unsharp.
Hyperfocal distance	Double	The distance from the lens beyond which all objects are rendered in acceptably sharp focus.
Longitude	Varchar (45)	The longitude of the location in case the camera supports gps data.
Latitude	Varchar (45)	The latitude of the location in case the camera supports gps data.
Altitude	Int (10)	The altitude of the location in case the camera supports gps data.
Direction	Double	The direction of the capturing in degrees.

Focal length	Double	The focal length.
Fnumber	Double	The calibration of the aperture size of a photographic lens.
Vertical angle of view	Double	The vertical angle of view of the capturing.

Table	**Description**	
Complex event instance	The table that stores the main event of this scenario instance created using the capturing application.	
Attribute Name	**Type**	**Description**
IDComplEventIns	Int(10)	The primary key of the table.
Name	Varchar (128)	The name of the specific scenario instance.
Description	Text	A description of the specific scenario instance.
FkTypeID	Int(10)	Foreign Key to the Semantic Type table.

Table	**Description**	
Crowd	The table that stores the crowd type of roles involved in a specific event. The crowd type roles are three or more persons together.	
Attribute Name	**Type**	**Description**
IDCrowd	Int (10)	The primary key of the table.
Name	Varchar (128)	The name of the crowd.
Description	Text	A description of the crowd.
FkRoleID	Int (10)	Foreign Key to the Roles Involved table. Implements the hierarchy relation between the crowd and the roles involved.

Table	**Description**

Ellipse	The table that stores the shapes that have an elliptic form.

Attribute Name	Type	Description
IDEllipse	Int (10)	The primary Key of the table.
X size	Double	The size of the ellipse on the x axis.
Y size	Double	The size of the ellipse on the y axis.
fkShapeID	Int (10)	Foreign Key to the Shape table. Implements the hierarchy relation between the ellipse and the shape table.

Table	Description
Event instance	The table that stores the event instances that of a specific scenario instance.

Attribute Name	Type	Description
IDEventInst	Int (10)	The primary key of the table.
Name	Varchar(128)	The name of the specific event instance.
Description	Text	The description of the specific event instance.
FkComplexEventInstanceID	Int (10)	Foreign Key to the Complex Event Instance table.
FkTypeID	Int (10)	Foreign Key to the Semantic Type table.
FkSuperEventID	Int (10)	Foreign Key to the Event Instance table. Implements the hierarchy between different events.
fkMultimediaShotCapturedID	Int (10)	Foreign Key to the Multimedia Shot Captured table.

Table	Description

Event roles	The table that represents the relationship between the Event Instance and the Roles Involved. Stores which roles are participating in every event.	
Attribute Name	**Type**	**Description**
FkRolesInvolvedId	Int (10)	Foreign Key to the Roles Involved table.
FkEventInstanceId	Int (10)	Foreign Key to the Event Instance table.

Table	**Description**	
Event shot audio	The table that represents the relationship between the Event Shot Instance and the Audio Instance. Stores which audio parameters belong to every event shot.	
Attribute Name	**Type**	**Description**
FkEventShotInstID	Int (10)	Foreign Key to the Event Shot Instance table.
FkAudioInstID	Int (10)	Foreign Key to the Audio Instance table.

Table	**Description**	
Event shot camera parameters	The table that represents the relationship between the Event Shot Instance and the Camera Parameters Instance. Stores which camera parameters belong to every event shot.	
Attribute Name	**Type**	**Description**
FkEventShotInstID	Int (10)	Foreign Key to the Event Shot Instance table.
FkCameraParametersID	Int (10)	Foreign Key to the Camera Parameters Instance table.

Table	**Description**	
Event shot instance	The table that stores the Event Shot Instances that were captured for every event of a specific scenario instance.	
Attribute Name	**Type**	**Description**
IDEventShotInstance	Int (10)	The primary key of the table.

Name	Varchar(128)	The name of the specific event shot.
Description	TEXT	The description of the specific event shot.
FkEventInstID	Int (10)	Foreign Key to the Event Instance table.
Shot time start	Double	The timestamp in the video captured when the shot starts.
Shot time end	Double	The timestamp in the video captured when the shot ends.
Duration	Double	The duration of the specific shot.
FkLocationID	Int (10)	Foreign Key to the location Table.

Table	**Description**	
Event shot lighting	The table that represents the relationship between the Event Shot Instance and the Lighting Instance. Stores which lighting parameters belong to every event shot.	
Attribute Name	**Type**	**Description**
FkEventShotInstID	Int (10)	Foreign Key to the Event Shot Instance table.
FkLightingInstID	Int (10)	Foreign Key to the Lighting Instance table.

Table	**Description**	
Event shot site map	The table that represents the relationship between the Event Shot Instance and the Shot Site Map. Stores which shot site maps belong to every event shot.	
Attribute Name	**Type**	**Description**
FkEventEhotEnstanceID	Int (10)	Foreign Key to the Event Shot Instance table.
FkShotSiteMapID	Int (10)	Foreign Key to the Shot Site Map table.
Site map type	Varchar(45)	The type of the shot site map. The shot site map can be the main site map or one of the shot elevations (north, south, west and east).

Table	**Description**	
Event spatial objects	The table that represents the relationship between the Event Instance and the Spatial Object Instance. Stores which spatial objects are participating in every event.	
Attribute Name	**Type**	**Description**
FkEventInstanceID	Int (10)	Foreign Key to the Event Instance table.
FkSpatialObjectID	Int (10)	Foreign Key to the Spatial Object Instance table.

Table	**Description**	
Event time	The table that represents the relationship between the Event Instance and the Time. Stores which time intervals are participating in every event.	
Attribute Name	**Type**	**Description**
FkEventInstanceID	Int (10)	Foreign Key to the Event Instance table.
FkTimeID	Int (10)	Foreign Key to the Time table.

Table	**Description**	
Footprint pixel point	The table that stores the pixel points for a pixel Footprint.	
Attribute Name	**Type**	**Description**
IDFootprintPixelPoint	Int (10)	The primary Key of the table.
X pixel	Int (10)	The pixel on the x axis.
Y pixel	Int (10)	The pixel on the y axis.
FkPixelFootprintID	Int (10)	Foreign Key to the Pixel Footprint table.

Table	Description
Function	The table that stores the function that needs to perform a function in order to put an object on the Shot Site Map, when there is no representation provided for the specific object.

Attribute Name	Type	Description
IDFunction	Int(10)	The primary key of the table.
Description	Text	The description of the specific function.
function	Varchar(128)	The function.

Table	Description
Geographic map	The table that stores the geographic map details when that specialization of the Shot Site Map is used.

Attribute Name	Type	Description
IDGeographicMap	Int(10)	The primary key of the table.
Image data	Medium Blob	The photo of the specific geographic map.
Description	Text	The description of the specific geographic map.
FkSuperGeographicMapID	Int(10)	Foreign Key to the Geographic Map table. Represents the hierarchy relations of the geographic maps.
FkShotSiteMapID	Int(10)	Foreign Key to the Shot Site Map table.

Table	Description
Gps point	The class that stores the points of a gps location.

Attribute Name	Type	Description
IDGpsPoint	Int(10)	The primary key of the table.
Longitude	Double	The longitude of the specific gps location.
Latitude	Double	The latitude of the specific gps location.
Altitude	Varchar(128)	The altitude of the specific gps location.
FkGpsReprID	Int(10)	Foreign Key to the Gps Representation table.

Table	Description	
Gps representation	The table that stores the group of the gps points when the gps representation, which is a specialization of the Logical Spatial Shot Representation, is used.	
Attribute Name	**Type**	**Description**
IDGpsRepresentation	Int(10)	The primary key of the table.
FkLogicalSpatialShotReprID	Int(10)	Foreign Key to the Logical Spat Shot Representation table.

Table	Description	
Lighting instance	The table that stores the technical parameters used during the capturing by the lighting equipment.	
Attribute Name	**Type**	**Description**
IDLightingInstance	Int(10)	The primary key of the table.
Description	Text	A description about the audio parameters.
FkLocationID	Int(10)	Foreign Key to the Location table.
FkRotationID	Int(10)	Foreign Key to the Rotation table.
FkShapeID	Int(10)	Foreign Key to the Shape table.
FkMovementID	Int(10)	Foreign Key to the Movement table.

Table	**Description**
Location	The table that stores the location of an object on a Shot Site Map.

Attribute Name	**Type**	**Description**
IDLocation	Int(10)	The primary key of the table.
Description	Text	The description of the specific location
X position	Int(10)	The position of the specific location of the x axis.
Y position	Int(10)	The position of the specific location of the y axis.

Table	**Description**
Logical spat shot representation	The table that stores the Logical Spatial Representations of an object that were used in a specific Event Shot.

Attribute Name	**Type**	**Description**
IDLogicalSpatShotRepresentation	Int(10)	The primary key of the table.
Description	Text	The description of the specific shot logical spatial representation.
FkLogSpatialReprInstID	Int(10)	Foreign Key to the Logical Spatial Representation table.
FkCameraParametersInstID	Int(10)	Foreign Key to the Camera Parameters Instance table.
FkLightingInstID	Int(10)	Foreign Key to the Lighting Instance table.
FkAudioInstID	Int(10)	Foreign Key to the Audio Instance table.

Table	**Description**

Logical spatial representation	The table that stores the logical Spatial Representations of the spatial objects and the roles involved of a specific scenario instance.	
Attribute Name	**Type**	**Description**
IDLogicalSpatialRepresentation	Int(10)	The primary key of the table.
Description	Text	The description of the specific logical spatial representation.
FKCameraParamInstID	Int(10)	Foreign Key to the Camera Parameters Instance table.
FkLightingInstID	Int(10)	Foreign Key to the Lighting Instance table.
FkAudioInstID	Int(10)	Foreign Key to the Audio Instance table.
Representation type	Varchar(128)	The type of the specific representation.
Representation value	Varchar(128)	The value of the specific representation according to the representation type.
Magnification factor	Int(10)	The overall magnification factor.
Units	Varchar(128)	The units used in the specific logical spatial representation.
Image Data	Long Blob	The picture of the object the logical representation reflects.

Table	**Description**	
Movement	The table that stores the movement of an object during the capturing of an event shot.	
Attribute Name	**Type**	**Description**
IDMovement	INT(10)	The primary key of the table.
Description	Text	The description of the specific movement.
Direction	Varchar(128)	The direction of the movement in degrees
Shape	Medium Blob	The shape that the movement may be follows.
X final position	Double	The final position of the

		object moving on the x axis.
Y final position	Double	The final position of the object moving on the y axis.

Table	**Description**
Movement points	The table that stores the points on the shot site map of the movement of a specific object.

Attribute Name	**Type**	**Description**
IDMovementPoints	Int(10)	The primary key of the table.
FKMovementID	Int(10)	Foreign Key to the Movement table.
FkPointID	Int(10)	Foreign Key to the Point table.
Point direction	Double	The horizontal direction of the movement at the specific point.
Vertical direction	Double	The vertical direction of the movement at the specific point.
Time	Double	The timestamp on the video the specific movement happened.

Table	**Description**
Multimedia shot captured	The table that stores the multimedia captured. The table contains the path of the multimedia file captured.

Attribute Name	**Type**	**Description**
IDMultimediaShotCaptured	Int(10)	The primary key of the table.
Multimedia data	Varchar(256)	The path to the multimedia data captured during a specific event.
Description	Text	The description of the multimedia data captured during a specific event.

Table	Description	
Object	The table that stores the object type of roles involved in a specific event. The object type roles are the non human objects participating in an event.	
Attribute Name	**Type**	**Description**
IDObject	Int(10)	The primary key of the table.
Name	Varchar(128)	The name of the object.
FkRoleID	Int(10)	Foreign Key to the Roles Involved table. Implements the hierarchy relation between the crowd and the roles involved.

Table	Description	
Ontology	The table that stores the scenario from which a specific scenario instance was created.	
Attribute Name	**Type**	**Description**
IDOntology	Int(10)	The primary key of the table.
name	Varchar(128)	The name of the scenario – ontology.
description	Text	The description of the scenario – ontology.

Table	Description	
Participating role shot instance	The table that stores the representations of the roles that participate in a specific event shot.	
Attribute Name	**Type**	**Description**
IDParticipatingRoleShotInstance	Int(10)	The primary key of the table.
Role type	Varchar(128)	The type of the role participating on the specific shot (Person, Object or Crowd).
Description	Text	The description of the role participating on the specific shot.

FkEventShotInstID	Int(10)	Foreign Key to the Event Shot Instance table.
FkLogicalSpatShotReprID	Int(10)	Foreign Key to the Logical Spat Shot Representation table.

Table	**Description**
Participating spatial object shot instance	The table that stores the representations of the spatial objects that participate in a specific event shot.

Attribute Name	**Type**	**Description**
IDParticipatingSpatialObjectShotInstance	Int(10)	The primary key of the table.
Name	Varchar(128)	The name of the participating spatial object.
Description	Text	The description of the role participating on the specific shot.
FkEventShotInstID	Int(10)	Foreign Key to the Event Shot Instance table.
FkLogicalSpatShotReprID	Int(10)	Foreign Key to the Logical Spat Shot Representation table.

Table	**Description**
Person	The table that stores the person type of roles involved in a specific event. The person type roles are the people participating in an event.

Attribute Name	**Type**	**Description**
IDPerson	Int(10)	The primary key of thetable.
Name	Varchar(128)	The name of the person.
FkRoleId	Int(10)	Foreign Key to the Roles Involved table. Implements the hierarchy relation between the crowd and the roles involved.

Table	**Description**

Physical representation	The table that stores the representations of the spatial objects and roles involved on the shot site map.	
Attribute Name	**Type**	**Description**
IDPhysicalRepresentation	Int(10)	The primary key of the table.
Is Polygon	Boolean	Shows if the representation id polygon shape or not.
Image data	Long Blob	The photo of the object related to the representation.
FkFunctionID	Int(10)	Foreign Key to the Function table.
FkLocationID	Int(10)	Foreign Key to the Location table.
FkRotationID	Int(10)	Foreign Key to the Rotation table.
FkScalingID	Int(10)	Foreign Key to the Scaling table.
FkShapeID	Int(10)	Foreign Key to the Shape table.
FkMovementID	Int(10)	Foreign Key to the Movement table.
FkShotSiteMapID	Int(10)	Foreign Key to the Shot Site Map table.
FkLogicalSpatShotReprID	Int(10)	Foreign Key to the Logical Spat Shot Representation table.

Table	**Description**	
Pixel footprint	The table that stores the footprint of an object on the shot site map using pixel points.	
Attribute Name	**Type**	**Description**
IDPixelFootprint	Int(10)	The primary key of the table.
Description	Text	The description of the pixel footprint.
FkPhysicalRepresentationID	Int(10)	Foreign Key to the Physical Representation table.

Table	**Description**

Point	The table that stores the various points of the shot site map.	
Attribute Name	**Type**	**Description**
IDPoint	Int(10)	The primary key of the table.
X point	Double	The location of the point on the x axis.
Y point	Double	The location of the point on the y axis.

Table	**Description**	
Polyline	The table that stores the shapes that have an polyline form. They are made up, that is, from a group of lines.	
Attribute Name	**Type**	**Description**
IDPolyline	Int(10)	The primary key of the table.
Is Closed	Boolean	Shows if the polyline shape is a closed shape or not.
FkShapeID	Int(10)	Foreign Key to the Shape table.

Table	**Description**	
Roles involved instance	The table that stores the roles that participate in a specific scenario instance.	
Attribute Name	**Type**	**Description**
IDRolesInvolved	Int(10)	The primary key of the table.
name	Varchar(128)	The name of the specific role.
description	Text	The description of the specific role.
FkEventInstance	Int(10)	Foreign Key to the Event Instance table.
fk_type_id	Int(10)	Foreign Key to the Semantic Type table.

Table	**Description**	
Roles involved logical representation	The table that stores the logical spatial representations of the roles involved in a scenario instance.	
Attribute Name	**Type**	**Description**
FkRolesInvolvedID	Int(10)	Foreign Key to the Roles Involved table.
FkLogicalRepresentationID	Int(10)	Foreign Key to the Logical Spatial Representation table.

Table	**Description**	
Rotation	The table that stores the rotation of the original object shape in order to be placed on the shot site map.	
Attribute Name	**Type**	**Description**
IDRotation	Int(10)	The primary key of the table.
Description	Text	The description of the rotation.
Direction	Varchar(128)	The direction of the rotation.
X final position	Double	The final position of the x axis.
Y final position	Double	The final position on the y axis.
Degrees	Int(10)	The degrees of the rotation.

Table	**Description**	
Scaling	The table that stores the scaling of the original object shape in order to be placed on the shot site map.	
Attribute Name	**Type**	**Description**
IDScaling	Int(10)	The primary key of the table.
Description	Text	The description of the scaling.
Scaling	Int(10)	The scaling.

Table	**Description**	
Semantic attribute	The table that stores the semantic attribute of the scenario from which an attribute of a specific event related data (Complex Event, Event, Roles Involved, Spatial Object, and Time) was created.	
Attribute Name	**Type**	**Description**
IDSemanticAttribute	Int(10)	The primary key of the table.
Name	Varchar(128)	The name of the semantic attribute.
Value	Varchar(128)	The value the semantic attribute can take.
FkSemanticTypeID	Int(10)	Foreign Key to the Semantic Type table.

Table	**Description**	
Semantic type	The table that stores the semantic type of the scenario from which a specific event related data (Complex Event, Event, Roles Involved, Spatial Object, Time) was created.	
Attribute Name	**Type**	**Description**
IDSemanticType	Int(10)	The primary key of the table.
Name	Varchar(128)	The name of the semantic type.
Description	Text	The description of the semantic type.
FkSemanticTypeID	Int(10)	Foreign Key to the Semantic Type table. Represents the hierarchy of the Semantic Types.
FkOntologyID	Int(10)	Foreign Key to the Ontology table.

Table	**Description**	
Shape	The table that stores the shape of a specific object.	
Attribute Name	**Type**	**Description**
IDShape	INT(10)	The primary key of the table.

description	TXT	The description of the shape.

Table	**Description**	
Shape points	The table that stores the points that belong in a specific shape.	
Attribute Name	**Type**	**Description**
idShapePoints	Int(10)	The primary key of the table.
FkShapeID	Int(128)	Foreign Key to the Shape table.
FkPointID	Int(10)	Foreign Key to the Point table.
Is Curved	Boolean	Shows if the line that connects the two points of the shape is curved or not.

Table	**Description**	
Shot site map	The table that stores the shot site map, the canvass that is, of a specific Event Shot Instance.	
Attribute Name	**Type**	**Description**
IDShotSiteMap	Int(10)	The primary key of the table.
Name	Varchar(128)	The name of the shot site map.
Image data	Long Blob	The picture of the shot site map.
Direction ref	Double	In case the picture doesn't have the correct direction, the direction ref is used to fix it.
threshold	Double	
Creator	Varchar(128)	The creator of the shot site map.
Creation date	Date	The date of the creation of the shot site map.
Is Main Site Map	Boolean(Yes or No)	Shows if the specific site map is main or not.

Table	Description	
Spatial object instance	The table that stores the Spatial Objects that participate in a specific scenario instance.	
Attribute Name	**Type**	**Description**
IDSpatialObjectInstance	Int(10)	The primary key of the table.
Name	Varchar(128)	The name of the spatial object.
Description	Text	The description of the spatial object.
FkEventInstanceID	Int(10)	Foreign Key to the Event Instance table.
FkTypeID	Int(10)	Foregn Key to the Semantic Type table.

Table	Description	
Spatial object logical representation	The table that stores the logical spatial representations of the spatial objects that participate in a scenario instance.	
Attribute Name	**Type**	**Description**
FkSpatialObjectID	Int(10)	Foreign Key to the Spatial Object Instance table.
FkLogicalRepresentationID	Int(10)	Foreign Key to the Logical Spatial Representation.

Table	Description	
Time	The table that stores the time intervals that participate in a specific scenario instance.	
Attribute Name	**Type**	**Description**
IDTime	Int(10)	The primary key of the table.
Value	Varchar (128)	The value of the time interval.

Description	Text	The description of the time interval.
FkSuperTimeID	Int(10)	Foreign Key to the Time table. Represents the hierarchy of the time intervals.
FkEventInstanceID	Int(10)	Foreign Key to the Event Instance table.
FkTypeID	Int(10)	Foreign Key to the Semantic Type table.

4. SUMMARY

In this chapter we described the creation of Event types using the OWL/RDF language. We described the core ontology we created in the current thesis and is used as a foundation for the creation of the different application scenarios event types. Next we described the process of an event type creation as an ontology. Finally we described the relational database we created in the current thesis and is used for the physical storing and retrieving of the captured multimedia along with its associated metadata.

CHAPTER 6
SYSTEM FUNCTIONALITY

1. INTRODUCTION

Using the application which was created in the contents of the current diploma thesis, the user can capture a video along with its desired metadata according to the Complex Event Model described on the previous chapter. In addition, the user can choose to not capture real time, but to associate an already recorded video with its desired metadata. Moreover, the application can be used to find and playback the wanted video file along with the information that was stored with the specific video.

The functional aspects of the application are described using the Use Cases methodology. The use cases are software – engineering technology which provides a formal description of the functionality of a system, and which description is fully comprehensible to anyone who wishes to read it. The description template of the application's use cases which is used on the current chapter is the widely known description template defined by Alistair Cockburn in the "Writing Effective Use Cases" (1).

2. USE CASES DESCRIPTION

A use case captures a contract between the stakeholders of a system about its behavior. The use case describes the system's behavior under various conditions as it responds to a request from one of the stakeholders, called the *primary actor*. The primary actor initiates an interaction with the system to accomplish some goal. The system responds, protecting the interests of all the stakeholders. Different sequences of behavior, or scenarios, can unfold, depending on the particular requests made and conditions surrounding the requests. The use case collects together those different scenarios. Use cases are fundamentally a text form, although they can be written using flow charts, sequence charts, Petri nets, or programming languages. Under normal circumstances, they serve to communicate from one person to another, often to people with no special training. Simple text is, therefore, usually the best choice.

A well-written use case is easy to read. It consists of sentences written in only one grammatical form, a simple action step, in which an actor achieves a result or passes information to another actor. Learning to read a use case should not take more than a few minutes of training. Learning to write good use cases is harder. The writer has to master three concepts that apply to every sentence in the use case and to the use case as a whole. Odd though it may seem at first glance, keeping those three concepts straight is difficult. The difficulty shows up as soon as you start to write your first use case. These three concepts are:

- ***Scope***. What is really the system under discussion?
- ***Primary** actor*. Who has the goal?
- ***Level***. How high- or low-level is that goal?

The template that is suggested by Cockburn for the use cases description is shown on the next figure. (1)

Use Case #	**<The name should be the goal as a short active verb phrase>**	
Goal In Context	<A longer statement of the goal, if needed, its normal occurrence conditions>	
Scope & Level	<Design scope, what system is being considered black-box under design & one of: Summary, Primary task, Subfunction>	
Preconditions	<What we expect is already the state of the world>	
Success End Condition	<The state of the world if goal succeeds>	
Failed End Condition	<The state of the world if goal does not succeeds>	
Primary, Secondary Actors	<A role name for the primary or any other actor, or description	
Trigger	<What starts the use case, may be time event>	
Main Success Scenario	**Step**	**Action**
	0	<Put here the steps of the scenario from trigger to goal delivery, and any cleanup after>
	1	2nd Step
	2	3rd Step
Extensions	**Step**	**Branching Action**
	1α	<Put here the extensions, one at a time, each referring to the step of the main scenario>

To understand the use cases description that Cockburn suggested, the understanding of the following summary definitions is essential.

- **Actor**: Anyone or anything with behavior.
- **Stakeholder**: Someone or something with a vested interest in the behavior of the system under discussion (SuD).
- **Primary Actor**: The stakeholder who or which initiates an interaction with the SuD to achieve a goal.
- **Supporting actor**: An external actor that provides a service to the system under design.
- **Use Case**: A contract for the behavior of the SuD.
- **Scope**: Identifies the system that we are discussing.
- ***Preconditions* and *Guarantees**: Say what must be true before and after the use case runs.
- ***Trigger**: States what event gets the use case started. Sometimes the trigger precedes the first step of the use case. Sometimes it is the first step of the use case.*
- **The *main success scenario***: A case in which nothing goes wrong.

- **Extensions**: Describe what can happen differently during that scenario. The numbers in the extensions refer to the step numbers in the main success scenario at which each different situation gets detected.
- When a use case references another use case, the second use case is written in *italics* or underlined.

Both the goals and the interactions in a scenario can be unfolded into finer and finer-grained goals and interactions. This is normal, and we handle it well in everyday life. However normal this is in everyday life, it causes confusion when writing a use case. To avoid this confusion Cockburn defined three different levels of goals which the use cases are called to achieve.

The *user (primary task) goal* is the goal of greatest interest. It is the goal the primary actor has in trying to get work done, or the user has in using the system at all. It corresponds to "elementary business process" in the business process engineering literature. A user goal addresses the question, "Can the primary actor go away happy after having done this? “.

Summary-level goals involve multiple user goals. They serve three purposes in the describing the system:

- They show the context, in which the user goals operate,
- They show life-cycle sequencing of related goals,
- They provide a table of contents for the lower-level use cases, both lower white use cases and blue use cases.

Subfunction-level goals are those required to carry out user goals. Include them only as you have to. They are needed on occasion for readability, or because many other goals use them.

To achieve the complete representation of the use cases used in the requirements analysis of a system, three additional tools are needed:

- **The overall Picture**: Represents the use cases as boxes which are linked together with relations (which use case calls which).
- **The summary Table**: Represents the use cases categorized by level and contains information about the id, the name, the primary actor, the goal and a small description of each use case.
- **The Actors' Skills**: Used to describe certain skill of each actor or some functional part of the system.

The next sections contain the requirements analysis of the application created in the contents of the current thesis, using the tools described above. Next a more detailed use cases description will follow according to the template table suggested by Cockburn.

3. THE OVERALL PICTURE

In this section, the use cases, that the mobile multimedia event capturing and visualization application created in the contexts of this thesis must fulfill, will be described. The use cases will be represented as boxes. These boxes are connected with relations. This type of description of the system provides an overall picture of the use cases that describe the requirements of the system. Moreover, it provides a method of representation of the relations between these use cases, that is, which use case, calls which. In the graph that follows, the use cases are divided in three different categories according to the kind of goal that every use case is requested to fulfill: Summary level use cases, primary task use cases and sub-function use cases. Due to the large scale of the graph, it is divided it more than one smaller graph that are shown in the following figures.

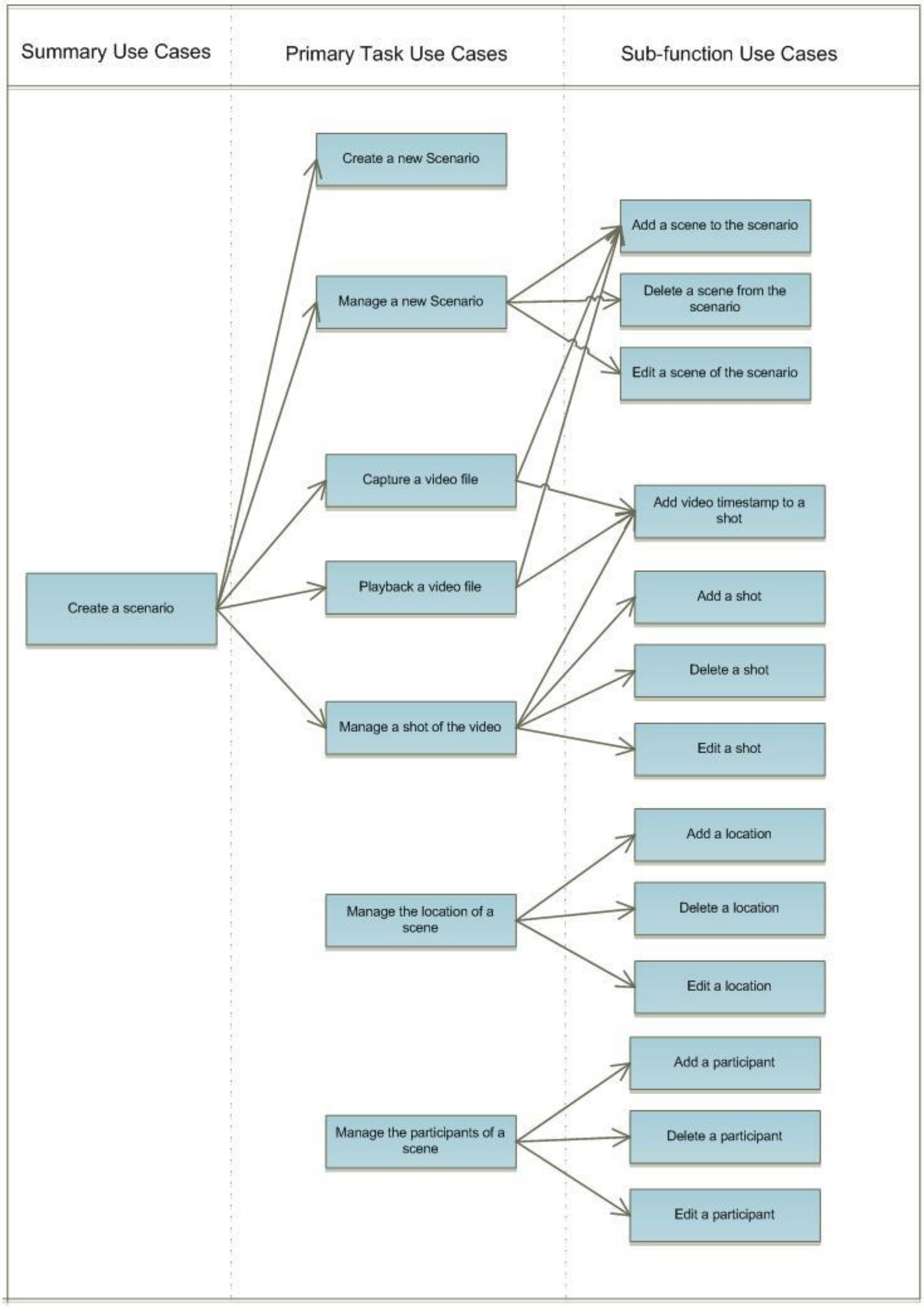


Figure 6.1 Use Cases Overall Picture, part 1

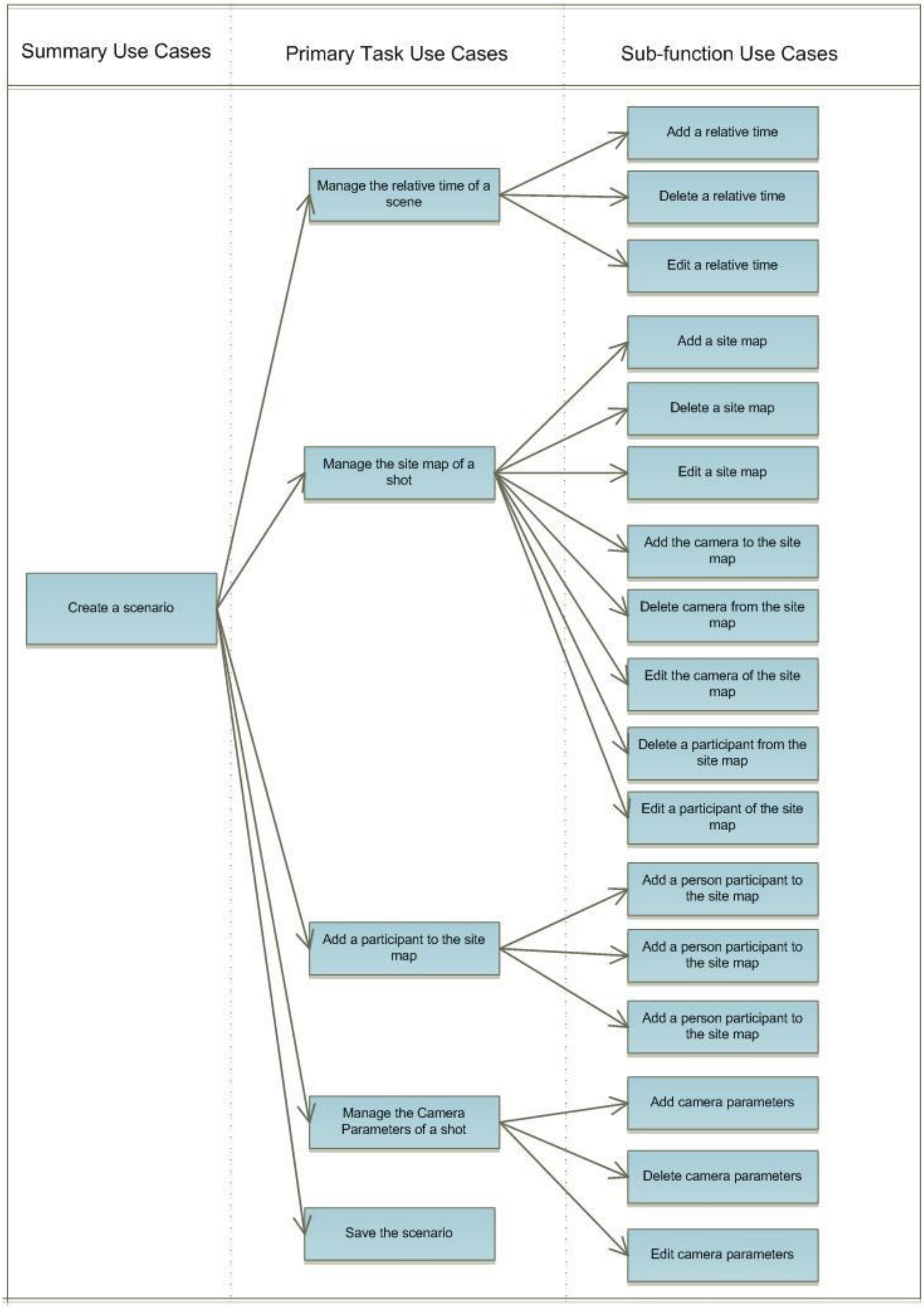

Summary Use Cases
Primary Task Use Cases
Sub-function Use Cases
Create a scenario
Manage the relative time of a scene
Add a relative time
Delete a relative time
Edit a relative time
Manage the site map of a shot
Add a site map
Delete a site map
Edit a site map
Add the camera to the site map
Delete camera from the site map
Edit the camera of the site map
Delete a participant from the site map
Edit a participant of the site map
Add a participant to the site map
Add a person participant to the site map
Add a person participant to the site map
Add a person participant to the site map
Manage the Camera Parameters of a shot
Add camera parameters
Delete camera parameters
Edit camera parameters
Save the scenario

Figure 6.2 Use Cases Overall Picture, part 2

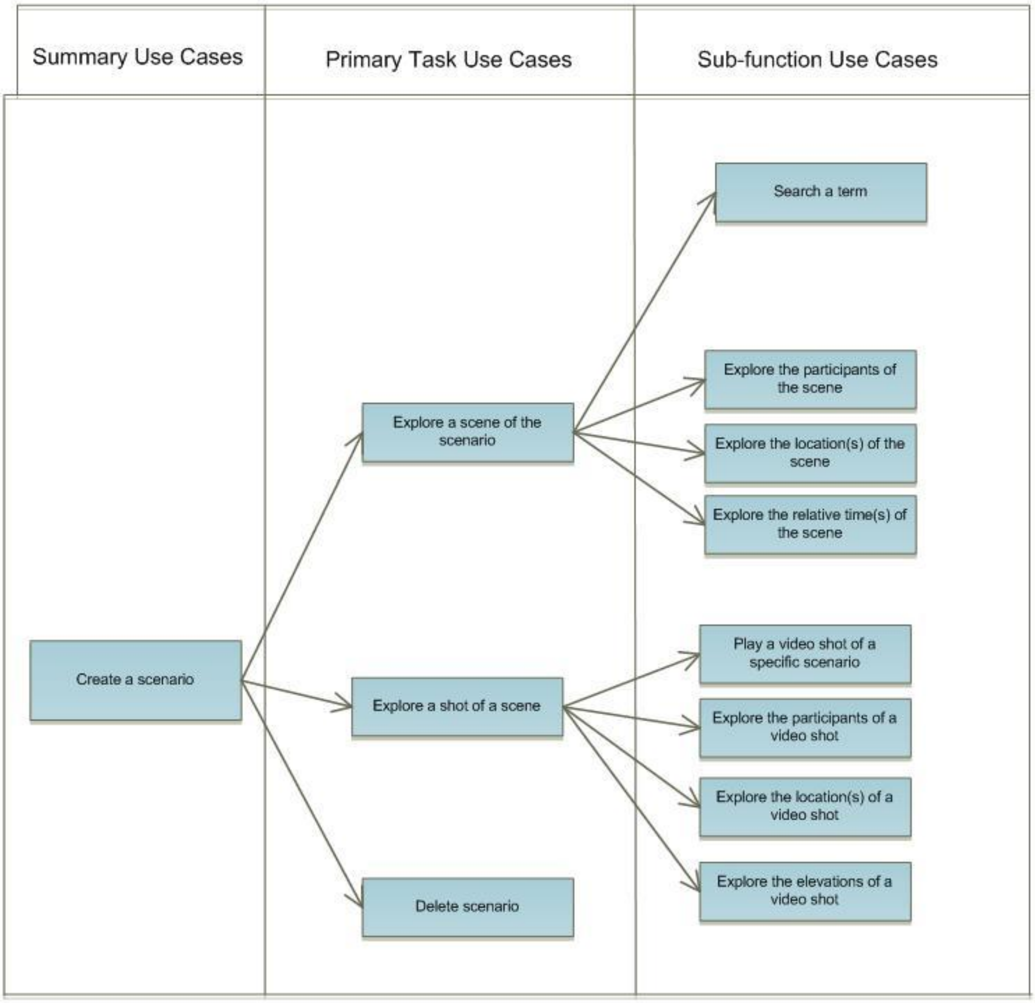

Summary Use Cases
Primary Task Use Cases
Sub-function Use Cases
Search a term
Explore the participants of the scene
Explore a scene of the scenario
Explore the location(s) of the scene
Explore the relative time(s) of the scene
Play a video shot of a specific scenario
Create a scenario
Explore a shot of a scene
Explore the participants of a video shot
Explore the location(s) of a video shot
Delete scenario
Explore the elevations of a video shot

Figure 6.3 Use Cases Overall Picture, part 3

4. SUMMARY TABLE

In this section the summary table of all the use cases that the application must support will be presented. The summary table has the use cases in groups according to the level they belong. It contains information about the id, the primary actor, the goal and a small description for every use case. The table with the necessary information is shown next.

#	Level	Primary Actor	Goal	Description
UC1	Summary	Event Creator	Create a scenario	The user wants to create a new specific scenario.
UC2	Primary Task	Event Creator	Create a new scenario	The user wants to start the creation of a new scenario and enters the basic scenario information (name, description, ontology).
UC3	Primary Task	Event Creator	Manage a scenario	The user wants to manage the information related to the event associated with the specific scenario
UC4	Sub-function	Event Creator	Add event to the scenario	The user wants to add an event related to a specific video file.
UC5	Sub-function	Event Creator	Delete an event from the scenario	The user wants to delete an event.
UC6	Sub-function	Event Creator	Edit an event	The user wants to edit some of the details of the event.
UC7	Primary Task	Event Creator	Capture a video file	The user wants to capture a video and relate it to events and event shots.
UC8	Primary Task	Multimedia Event Capturing Personnel	Playback a video file	The user wants to play an already captured video and relate it to events and event shots.
UC9	Primary Task	Event Creator	Manage a Shot of the video	The user wants to add or edit an event shot related to a specific video.
UC10	Primary Task	Multimedia Event Capturing Personnel	Add the location of a shot	The user wants to add the location of an already existing

				shot to the location of the scene related.
UC11	Sub-function	Event Creator - Multimedia Event Capturing Personnel	Add a shot	The user wants to add a shot related to a specific video timestamp.
UC12	Sub-function	Event Creator - Multimedia Event Capturing Personnel	Delete a shot	The user wants to delete a shot.
UC13	Sub-function	Event Creator - Multimedia Event Capturing Personnel	Edit a shot	The user wants to edit some of the details of the shot.
UC14	Sub-function	Multimedia Event Capturing Personnel	Add video timestamp to a shot	The user wants to add the time in the video capturing the shot happened.
UC15	Primary Task	Event Creator	Manage the location of an event	The user wants to add, delete or edit the locations the scene took place.
UC16	Sub-function	Event Creator	Add a location	The user wants to add a location to state that this is the location the event took place.
UC17	Sub-function	Event Creator	Delete a location	The user wants to delete an event location.
UC18	Sub-function	Event Creator	Edit a location	The user wants to edit some of the details of the location.
UC19	Primary Task	Event Creator	Manage the participants of an event	The user wants to add, delete or edit the participants that appear to the event.
UC20	Sub-function	Event Creator	Add a participant	The user wants to add a participant to state that this person object or crowd is present to the specific video.
UC21	Sub-function	Event Creator	Delete a participant	The user wants to delete an event’s participant.
UC22	Sub-function	Event Creator	Edit a participant	The user wants to edit some of the details of a participant.

UC23	Primary Task	Event Creator	Manage the relative time(s) of an event	The user wants to add, delete or edit the relative time that the event happened.
UC24	Sub-function	Event Creator	Add a relative time	The user wants to add a time to state that this is the relative time the event happened.
UC25	Sub-function	Event Creator	Delete a relative time	The user wants to delete an event timestamp.
UC26	Sub-function	Event Creator	Edit a relative time	The user wants to edit some of the details of the time.
UC27	Primary Task	Event Creator	Manage the plans of a shot (site map)	The user wants to add, delete or edit the plans of specific shot to state the topology of the scene during the specific shot.
UC28	Sub-function	Event Creator	Add a site map	The user wants to add a site map related to a shot.
UC29	Primary Task	Event Creator	Add a participant to the site map	The user wants to add a person object or crowd representation to the site map to state the location and movement of the participant.
UC30	Sub-function	Event Creator	Add a person-participant to the site map	The user wants to add a person representation to the site map to state the location and movement of the participant.
UC31	Sub-function	Event Creator	Add an object-participant to the site map	The user wants to add an object representation to the site map to state the location and movement of the participant.
UC32	Sub-function	Event Creator	Add a crowd-participant to the site map	The user wants to add a crowd representation to

				the site map to state the location and movement of the participant.
UC33	Sub-function	Event Creator	Delete a participant of the site map	The user wants to delete a participant from the site map.
UC34	Sub-function	Event Creator	Edit a participant of the site map	The user wants to edit the location or any other parameters of a participant.
UC35	Sub-function	Event Creator	Add the camera to the site map	The user wants to add a video camera representation to the site map to state the location and movement of the camera shooting the video.
UC36	Sub-function	Event Creator	Delete camera from the site map	The user wants to delete the camera from the site map.
UC37	Sub-function	User Event Creator	Edit camera on the site map	The user wants to edit the location or any other parameters of the camera.
UC38	Sub-function	Event Creator	Delete site map	The user wants to delete a site map related to a shot
UC39	Sub-function	Event Creator	Edit site map	The user wants to edit the details of a site map related to a shot.
UC40	Primary Task	Event Creator	Manage Camera Parameters	The user wants to add, delete or edit the technical parameters of the camera that is used or has been used to capture the video.
UC41	Sub-function	Event Creator	Add Camera parameters	The user wants to add new camera parameters.
UC42	Sub-function	Event Creator	Delete Camera Parameters	The user wants to delete the camera parameters.
UC43	Sub-function	Event Creator	Edit Camera Parameters	The user wants to edit the details of the camera

				parameters.
UC44	Sub-function	Event Creator	Save scenario	The user wants to finish editing the scenario and save it to the database.
UC45	Summary	Multimedia Editor/Viewer	Explore a scenario	The user wants to see the details of an already captured scenario (e.g. the events of the scenario).
UC46	Sub-function	Multimedia Editor/Viewer	Search a term (scenario, event or shot)	The user wants to search a scenario, an event of a scenario or a specific shot of a scenario.
UC47	Primary Task	Multimedia Editor/Viewer	Explore an event of a scenario.	The user wants to see the location of the shots that a specific event contains.
UC48	Sub-function	Multimedia Editor/Viewer	Explore the participants of a scenario event	The user wants to see the information about a participant of a specific event.
UC49	Sub-function	Multimedia Editor/Viewer	Explore the location of a scenario event	The user wants to see the information about the location that a specific event took place.
UC50	Sub-function	Multimedia Editor/Viewer	Explore the relative time of a scenario event	The user wants to see the information about the relative time that a specific event happened.
UC51	Primary Task	Multimedia Editor/Viewer	Explore a shot of a scenario event	The user wants to see the details of a specific shot.
UC52	Sub-function	Multimedia Editor/Viewer	Play a specific video shot of a specific scenario	The user wants to play a video related to a specific shot and watch the topology if the scene as it was captured by the user.
UC53	Sub-function	Multimedia Editor/Viewer	Explore the participants of a video shot	The user wants to see the information about a participant of a specific event.
UC54	Sub-function	Multimedia	Explore the location of a	The user wants to

		Editor/Viewer	video shot	see the information about the location that a specific event took place.
UC55	Sub-function	Multimedia Editor/Viewer	Explore the relative time of a video shot	The user wants to see the information about the relative time that a specific event occurred.

5. DETAILED DESCRIPTION

At this point, there will be a more detailed description of the use cases that the application must support. The most important part of the information that the use cases provide is the main success scenario as well as the extensions, because with the extensions the functionality of every use case is described. Next, the tables of all the use cases that the application supports are shown.

Use Case #1	**Create a scenario**	
Goal In Context	To create or play a video and keep information about the video events and shots	
Scope & Level	Scenario management sub-system, Summary	
Preconditions	The system is running	
Success End Condition	A new scenario is been created and stored	
Failed End Condition	The new scenario is not stored	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to capture or play a video and create a scenario for the specific video	
Main Success Scenario	**Step**	**Action**
	0	The user selects the Creation of a new Scenario
	1	The user selects to perform one of the steps 2-10
	2	Manage a new scenario
	3	Capture a video file
	4	Playback a video file
	5	Manage a video shot
	6	Manage the location of an event
	7	Manage the participants of an event
	8	Manage the relative time of an event
	9	Manage the site map of a shot
	10	Manage the camera parameters of a shot
	11	Save the scenario
Extensions	**Step**	**Branching Action**
	1α	The completion of the scenario is canceled. The user gets informed about the reason

Use Case #2	Create a new Scenario	
Goal In Context	Create the new scenario and show to the user the interface that is used to complete the details of the scenario	
Scope & Level	Scenario management sub-system, primary task	
Preconditions	The system is running	
Success End Condition	The basic information about the scenario is completed and the window of the scenario capturing is popped up	
Failed End Condition	The scenario is not created	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to start the creation of a new scenario	
Main Success Scenario	**Step**	**Action**
	0	The user selects to create a new scenario
	1	The system asks the user to choose an ontology from which the scenario will be loaded
	2	The user chooses ontology
	3	The system asks the user to enter the basic information of a scenario
	4	The user enters the information necessary
	5	The system opens the scenario capturing panel
Extensions	**Step**	**Branching Action**
	2α	2a1. The user cancels the ontology selection 2a2. The system returns to first screen
	4a	4a1. The user cancels the information input 4a2. The system returns to first screen

Use Case #3	Manage a new Scenario	
Goal In Context	Manage the newly created scenario by managing the scenario events.	
Scope & Level	Event management sub-system, primary task	
Preconditions	The scenario capturing panel is open	
Success End Condition	The basic information about the scenario is completed and the window of the scenario capturing is popped up	
Failed End Condition	The details of the scenes are not specified.	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to manage a newly created scenario	
Main Success Scenario	**Step**	**Action**
	0	The user wants to handle the details of a new scenario events
	1	The user executes one of the steps 2-4
	2	Add an event
	3	Delete an event
	4	Edit an event
	5	The system saves the changes the user made

Use Case #4	Add an event to the scenario	
Goal In Context	Add a new event either from the ontology or from user input	
Scope & Level	Event management sub-system, sub-function	
Preconditions	The scenario capturing panel is open	
Success End Condition	The new events is added to the scenario events table	
Failed End Condition	The scene is not added	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add an event to the new scenario	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new event to the new scenario and executes the step 1 or jumps to step 3
	1	The user selects an event from the selected ontology
	2	The user selects to add the event to the scenario's event table
	3	The user selects to add a completely new event to the table
	4	The system asks the user to input some more information about the new event
	5	The use inputs the information
	6	The system add the new event to the event table
Extensions	**Step**	**Branching Action**
	5a	5a1. The user chooses to cancel the addition 5a2. The system does not add the new event and informs the user

Use Case #5	Delete an event from the scenario	
Goal In Context	Delete an event from the scenario event stable	
Scope & Level	Event management sub-system, sub-function	
Preconditions	The scenario capturing panel is open	
Success End Condition	The event is successfully deleted from the table	
Failed End Condition	The event is not deleted from the table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to delete an event from the scenario event stable	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete an event from the scenario
	1	The user selects the event he wants to delete from the scenario event stable
	2	The user selects to delete the event
	3	The system asks the user for delete confirmation
	4	The user confirms the deletion
	5	The system removes the event from the events table
Extensions	**Step**	**Branching Action**
	4α	4a1. The user cancels the deletion 4a2. The system does not remove the event from the events table

Use Case #6	Edit a scene of the scenario	
Goal In Context	Change the details of a specific scenario event	
Scope & Level	Event management sub-system, sub-function	
Preconditions	The scenario capturing panel is open	
Success End Condition	The details of the event are successfully changed	
Failed End Condition	The details of the event are not changed	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to change the details information of an event from the scenario events table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to change some of the details of a selected event
	1	The user selects the event he wants to change
	2	The user double clicks on the event and changes the desired details
	3	The system saves the changes

Use Case #7	Capture a video file	
Goal In Context	Start the capturing of the video	
Scope & Level	Event management sub-system, primary task	
Preconditions	The scenario capturing panel is open	
Success End Condition	The video is captured successfully and the information related to the video events and shots are input	
Failed End Condition	The video is not captured successfully or the information related to the video events and shots are not input	
Primary, Secondary Actors	Multimedia Event Capturing Personnel, System	
Trigger	The user wants to start the capturing of the video	
Main Success Scenario	**Step**	**Action**
	0	The user wants to start the video capturing
	1	The user selects the video capture choice
	2	The system asks the user to select a location where to save the video file
	3	The user selects a location
	4	The system starts the video capturing process
	5	The user executes the next step or jumps to step 7
	6	The user selects Add an event to scenario
	7	The user selects the desired event from the event table and double clicks to associate it to the captured video
	8	The user executes Add video Timestamp to a shot
	9	The user executes the next step or jumps to step 11
	10	The user jumps to step 5
	11	The user stops the capturing
	12	The system saves the captured data
Extensions	**Step**	**Branching Action**
	3α	3a1. The user cancels the selection 3a2. The system does not start capturing

Use Case #8	Playback a video file	
Goal In Context	Start the reproduction of the video	
Scope & Level	Event management sub-system, primary task	
Preconditions	The scenario capturing panel is open	
Success End Condition	The video is played successfully and the information related to the video events and shots are input	
Failed End Condition	The video stopped from playing or the information related to the video scenes and shots are not input	
Primary, Secondary Actors	Multimedia Event Capturing Personnel, System	
Trigger	The user wants to start the reproduce an already captured video	
Main Success Scenario	**Step**	**Action**
	0	The user wants to start the video playback
	1	The user selects the video playback choice
	2	The system asks the user to select a location of the video file
	3	The user selects the video he wants to play
	4	The system starts the video reproduction process
	5	The user executes the next step or jumps to step 7
	6	The user selects Add an event to scenario
	7	The user selects the desired event from the event table and double clicks to associate it to the reproduced video
	8	The user executes Add video Timestamp to a shot
	9	The user executes the next step or jumps to step 11
	10	The user jumps to step 5
	11	The user stops the reproduction
	12	The system saves the captured data
Extensions	**Step**	**Branching Action**
	3α	3a1. The user cancels the selection 3a2. The system does not start playing

Use Case #9	Manage a shot of a video event	
Goal In Context	Handle the shots table of a specific event	
Scope & Level	Shot management sub-system, primary task	
Preconditions	An event from the event table is selected	
Success End Condition	The changes of the shot table are made successfully	
Failed End Condition	No changes are made	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to manage the shots of an event	
Main Success Scenario	**Step**	**Action**
	0	The user wants to handle the details of the selected event's shots
	1	The user executes one of the steps 2-5
	2	Add a shot
	3	Delete a shot
	4	Edit a shot
	5	Add video timestamp to a shot
	6	The system saves the changes the user made

Use Case #10	Add a shot	
Goal In Context	Add a new shot to an already selected event	
Scope & Level	Shot management sub-system, sub-function	
Preconditions	An event from the scene table is selected	
Success End Condition	The shot is successfully added to the shot table	
Failed End Condition	The shot is not added to the shot table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add a new shot to an event	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new shot to a scenario event
	1	The user selects to add a new shot to the table
	2	The system asks the user to input some more information about the new shot
	3	The use inputs the information
	4	The system add the new shot to the selected event's shot table
Extensions	**Step**	**Branching Action**
	3a	3a1. The user chooses to cancel the addition 3a2. The system does not add the new shot and informs the user

Use Case #11	Delete a shot	
Goal In Context	Delete a shot from the scenario's shots table	
Scope & Level	Shot management sub-system, sub-function	
Preconditions	An event from the event table is selected	
Success End Condition	The shot is successfully deleted from the table	
Failed End Condition	The shot is not deleted from the table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to delete a shot from the scenario's shots table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete a shot from the selected event
	1	The user selects the shot he wants to delete from the event's shots table
	2	The user selects to delete the shot
	3	The system asks the user for delete confirmation
	4	The user confirms the deletion
	5	The system removes the shot from the shots table
Extensions	**Step**	**Branching Action**
	4α	4a1. The user cancels the deletion 4a2. The system does not remove the shot from the shots table

Use Case #12	Edit a shot
Goal In Context	Change the details of a specific scenario's shot
Scope & Level	Shot management sub-system, sub-function
Preconditions	An event from the event table is selected
Success End Condition	The details of the shot are successfully changed
Failed End Condition	The details of the shot are not changed

Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to change the details information of a shot from the event's shots table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to change some of the details of a specific event's selected shot
	1	The user selects the shot he wants to change
	2	The user double clicks on the shot and changes the desired details
	3	The system saves the changes

Use Case #13	**Add video timestamp to shot**	
Goal In Context	Set the time on the video that the specific shot starts	
Scope & Level	Shot management sub-system, primary task	
Preconditions	The video capturing is on or the video playback is on	
Success End Condition	The timestamp is successfully related to the specific shot	
Failed End Condition	The timestamp is not successfully related to the specific shot	
Primary, Secondary Actors	Multimedia Event Capturing Personnel, System	
Trigger	The user wants to add the time on the video that the specific shot starts	
Main Success Scenario	**Step**	**Action**
	0	The user wants to relate a video timestamp to a specific shot
	1	The user executes the step 2 or jumps to step 4
	2	The user selects a shot from a specific events shots table
	3	The user double clicks the selected shot
		Go to step 6
	5	The user selects Add a shot
	6	The system saves the video timestamps to the specific shot
Extensions	**Step**	**Branching Action**
	4α	4a1. The user cancels the addition
		4a2. The system does not add the new shot with the desired timestamp

Use Case #14	**Manage the location(s) of an event**	
Goal In Context	Handle the locations table of a specific event	
Scope & Level	Spatial Objects management sub-system, primary task	
Preconditions	An event from the event table is selected	
Success End Condition	The changes on the location table are made successfully	
Failed End Condition	No changes are made	
Primary, Secondary Actors	Event creator, System	
Trigger	The user wants to manage the location(s) of a video event	
Main Success Scenario	**Step**	**Action**
	0	The user wants to handle the details of the selected event's location(s)
	1	The user executes one of the steps 2-4
	2	Add a location

	3	Delete a location
	4	Edit a location
	5	The system saves the changes the user made

Use Case #15	**Add a location**	
Goal In Context	Add a new location to an already selected event	
Scope & Level	Spatial Objects management sub-system, sub-function	
Preconditions	An event from the event table is selected	
Success End Condition	The location is successfully added to the locations table	
Failed End Condition	The location is not added to the locations table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add a new location to a event	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new location to a scenario event
	1	The user selects to the scenario detailed information tab
	2	The user selects to add a new location to the table
	3	The system asks the user to input some more information about the new location
	4	The use inputs the information
	5	The system add the new location to the selected event's locations table
Extensions	**Step**	**Branching Action**
	4a	4a1. The user chooses to cancel the addition 4a2. The system does not add the new shot and informs the user

Use Case #16	**Delete a location**	
Goal In Context	Delete a location from the event's location table	
Scope & Level	Spatial Objects management sub-system, sub-function	
Preconditions	An event from the scene table is selected	
Success End Condition	The location is successfully deleted from the table	
Failed End Condition	The location is not deleted from the table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to delete a location from the event's location table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete a location from the selected event
	1	The user selects to the scenario detailed information tab
	2	The user selects the location he wants to delete from the event's locations table
	3	The user selects to delete the location
	4	The system asks the user for delete confirmation
	5	The user confirms the deletion
	6	The system removes the location from the locations table
Extensions	**Step**	**Branching Action**
	5α	5a1. The user cancels the deletion 5a2. The system does not remove the location from the

		locations table

Use Case #17	Edit a location	
Goal In Context	Change the details of a specific event's location	
Scope & Level	Spatial Objects management sub-system, sub-function	
Preconditions	An event from the scene table is selected	
Success End Condition	The details of the location are successfully changed	
Failed End Condition	The details of the location are not changed	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to change the details information of a location from the event's location table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to change some of the details of a specific event's selected location
	1	The user selects to the scenario detailed information tab
	2	The user selects the location he wants to change
	3	The user double clicks on the location and changes the desired details
	4	The system saves the changes

Use Case #18	Manage the participants of an event	
Goal In Context	Handle the participants table of a specific event	
Scope & Level	Roles Involved management sub-system, primary task	
Preconditions	An event from the event table is selected	
Success End Condition	The changes on the participants table are made successfully	
Failed End Condition	No changes are made	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to manage the participants of a video event	
Main Success Scenario	**Step**	**Action**
	0	The user wants to handle the details of the selected event's participants
	1	The user executes one of the steps 2-4
	2	Add a participant
	3	Delete a participant
	4	Edit a participant
	5	The system saves the changes the user made

Use Case #19	Add a participant
Goal In Context	Add a new participant to an already selected event
Scope & Level	Roles Involved management sub-system, sub-function
Preconditions	An event from the event table is selected
Success End Condition	The participant is successfully added to the participants table
Failed End Condition	The participant is not added to the participants table
Primary, Secondary Actors	Event Creator, System
Trigger	The user wants to add a new participant to an event

Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new participant to a scenario event
	1	The user selects to the scenario detailed information tab
	2	The user selects to add a new participant to the table
	3	The system asks the user to input some more information about the new participant
	4	The use inputs the information
	5	The system add the new participant to the selected scene's participants table
Extensions	**Step**	**Branching Action**
	4a	4a1. The user chooses to cancel the addition 4a2. The system does not add the new participant and informs the user

Use Case #20	**Delete a participant**	
Goal In Context	Delete a participant from the event's participants table	
Scope & Level	Roles Involved management sub-system, sub-function	
Preconditions	An event from the event table is selected	
Success End Condition	The participant is successfully deleted from the table	
Failed End Condition	The participant is not deleted from the table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to delete a participant from the event's participants table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete a participant from the selected event
	1	The user selects to the scenario detailed information tab
	2	The user selects the participant he wants to delete from the event's participants table
	3	The user selects to delete the participant
	4	The system asks the user for delete confirmation
	5	The user confirms the deletion
	6	The system removes the participant from the participants table
Extensions	**Step**	**Branching Action**
	5α	5a1. The user cancels the deletion 5a2. The system does not remove the participant from the participants table

Use Case #21	**Edit a participant**
Goal In Context	Change the details of a specific event's participant
Scope & Level	Roles Involved management sub-system, sub-function
Preconditions	An event from the event table is selected
Success End Condition	The details of the participant are successfully changed
Failed End Condition	The details of the participant are not changed
Primary, Secondary Actors	Event Creator, System
Trigger	The user wants to change the details information of a participant

	from the event’s participants table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to change some of the details of a specific event’s selected participant
	1	The user selects to the scenario detailed information tab
	2	The user selects the participant he wants to change
	3	The user double clicks on the participant and changes the desired details
	4	The system saves the changes

Use Case #22	**Manage the relative time(s) of an event**	
Goal In Context	Handle the relative times table of a specific event	
Scope & Level	Time management sub-system, primary task	
Preconditions	An event from the scene table is selected	
Success End Condition	The changes on the relative times table are made successfully	
Failed End Condition	No changes are made	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to manage the relative time(s) of a video event	
Main Success Scenario	**Step**	**Action**
	0	The user wants to handle the details of the selected event’s relative times
	1	The user executes one of the steps 2-4
	2	Add a relative time
Extensions	3	Delete a relative time
	4	Edit a relative time
	5	The system saves the changes the user made

Use Case #23	**Add a relative time**	
Goal In Context	Add a new relative time to an already selected event	
Scope & Level	Time management sub-system, sub-function	
Preconditions	An event from the event table is selected	
Success End Condition	The relative time is successfully added to the relative times table	
Failed End Condition	The relative time is not added to the relative times table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add a new relative time to a event	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new relative time to a scenario event
	1	The user selects to the scenario detailed information tab
	2	The user selects to add a new relative time to the table
	3	The system asks the user to input some more information about the new relative time
	4	The use inputs the information
	5	The system add the new relative time to the selected event’s relative times table
Extensions	**Step**	**Branching Action**
	4a	4a1. The user chooses to cancel the addition 4a2. The system does not add the new relative time and

		informs the user

Use Case #24	**Delete a relative time**	
Goal In Context	Delete a relative time from the event's relative times table	
Scope & Level	Time management sub-system, sub-function	
Preconditions	An event from the event table is selected	
Success End Condition	The relative time is successfully deleted from the table	
Failed End Condition	The relative time is not deleted from the table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to delete a relative time from the event's relative times table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete a relative time from the selected event
	1	The user selects to the scenario detailed information tab
	2	The user selects the relative time he wants to delete from the event's relative times table
	3	The user selects to delete the relative time
	4	The system asks the user for delete confirmation
	5	The user confirms the deletion
	6	The system removes the relative time from the relative times table
Extensions	**Step**	**Branching Action**
	4α	4a1. The user cancels the deletion 4a2. The system does not remove the relative time from the relative times table

Use Case #25	**Edit a relative time**	
Goal In Context	Change the details of a specific event's relative time	
Scope & Level	Time management sub-system, sub-function	
Preconditions	An event from the scene table is selected	
Success End Condition	The details of the relative time are successfully changed	
Failed End Condition	The details of the relative time are not changed	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to change the details information of a relative time from the event's relative times table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to change some of the details of a specific event's selected relative time
	1	The user selects to the scenario detailed information tab
	2	The user selects the relative time he wants to change
	3	The user double clicks on the relative time and changes the desired details
	4	The system saves the changes

Use Case #26	**Manage the site map of a shot**	
Goal In Context	Handle the site maps table of a specific shot	
Scope & Level	Time management sub-system, primary task	
Preconditions	A shot from a specifics event's the shot table is selected	
Success End Condition	The changes on the site maps table are made successfully	
Failed End Condition	No changes are made	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to manage the site maps of a video shot	
Main Success Scenario	**Step**	**Action**
	0	The user wants to handle the details of the selected shot's site maps
	1	The user executes one of the steps 2-10
	2	Add a site map
	3	Add a participant to the site map
	4	Delete a participant from the site map
	5	Edit a participant of the site map
	6	Add the camera to the site map
	7	Delete camera from the site map
	8	Edit camera of the site map
	9	Delete site map
	10	Edit site map
	11	The system saves the changes the user made

Use Case #27	**Add a site map**	
Goal In Context	Add a new site map to an already selected shot	
Scope & Level	Shot Site Map management sub-system, sub-function	
Preconditions	A shot from the shot table is selected	
Success End Condition	The site map is successfully added to the site maps table	
Failed End Condition	The site map is not added to the site maps table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add a new site map to a shot	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new site to a scenario event's shot
	1	The user selects to the shots detailed information tab
	2	The user selects to add a new site map to the table
	3	The system asks the user to input some more information about the new site map
	4	The use inputs the information
	5	The system add the new site map to the selected event's site maps table
Extensions	**Step**	**Branching Action**
	4a	4a1. The user chooses to cancel the addition 4a2. The system does not add the new site map and informs the user

Use Case #28	Add a participant to a site map	
Goal In Context	Add a new participant to an already selected site map	
Scope & Level	Shot Site Map management sub-system, sub-function	
Preconditions	A site map from the site maps table is selected	
Success End Condition	The participant is successfully added to the site map's participants table	
Failed End Condition	The participant is not added to the site map's participants table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add a new participant to a site map	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new participant to a site map
	2	The user selects one of the steps 3-5
	3	Add a person – participant to a site map
	4	Add an object – participant to a site map
	5	Add a crowd – participant to a site map

Use Case #29	Add a person-participant to a site map	
Goal In Context	Add a new person to an already selected site map	
Scope & Level	Shot Site Map management sub-system, sub-function	
Preconditions	A site map from the site maps table is selected	
Success End Condition	The person is successfully added to the site map's participants table	
Failed End Condition	The person is not added to the site map's participants table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add a new person to the site map	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new person to the site map
	1	The system asks the user to select a person from the event's participants
	2	The user selects a person
	3	The system add the person to the site map's participants table
	4	The user enters the person's location coordinates
	5	The system adds the person to the site map
	6	The use executes the step 7 or the use case ends
	7	The user inputs the user movements during the specific site map's shot
	8	The system adds the movement to the site map
Extensions	**Step**	**Branching Action**
	2a	2a1. The user chooses to cancel the selection 2a2. The system does not add the new participant and informs the user

Use Case #30	Add a object-participant to a site map
Goal In Context	Add a new object to an already selected site map
Scope & Level	Shot Site Map management sub-system, sub-function
Preconditions	A site map from the site maps table is selected
Success End Condition	The object is successfully added to the site map's participants table

Failed End Condition	The object is not added to the site map's participants table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add a new object to the site map	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new object to the site map
	1	The system asks the user to select a object from the event's participants
	2	The user selects a object
	3	The system add the object to the site map's participants table
		The user selects the shape of the new object
	4	The user executes the step 5 or jumps to step 8
	5	The user selects to input an ellipse
	6	The user inputs the desired parameters to create the ellipse
	7	The system adds the ellipse to the site map
	8	The use case ends
	9	The user selects to input a polyline
	10	The user inputs the desired parameters to create the polyline
	11	The system adds the polyline to the site map
Extensions	**Step**	**Branching Action**
	2a	2a1. The user chooses to cancel the selection 2a2. The system does not add the new participant and informs the user

Use Case #31	**Add a crowd-participant to a site map**	
Goal In Context	Add a new crowd to an already selected site map	
Scope & Level	Shot Site Map management sub-system, sub-function	
Preconditions	A site map from the site maps table is selected	
Success End Condition	The crowd is successfully added to the site map's participants table	
Failed End Condition	The crowd is not added to the site map's participants table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add a new crowd to the site map	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new crowd to the site map
	1	The system asks the user to select a crowd from the event's participants
	2	The user selects a crowd
	3	The system add the crowd to the site map's participants table
	4	The user enters the crowd's location coordinates
	5	The system adds the crowd to the site map
	6	The use executes the step 7 or the use case ends
	7	The user inputs the crowd's movements during the specific site map's shot
	8	The system adds the movement to the site map
Extensions	**Step**	**Branching Action**
	2a	2a1. The user chooses to cancel the selection 2a2. The system does not add the new participant and

		informs the user

Use Case #32	**Delete a participant from a site map**	
Goal In Context	Delete a participant from the site map's participants table	
Scope & Level	Shot Site Map management sub-system, sub-function	
Preconditions	A site map from the site maps table is selected	
Success End Condition	The participant is successfully deleted from the table	
Failed End Condition	The participant is not deleted from the table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to delete a participant from a shot's site maps table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete a participant from the selected site map
	2	The user selects the participant he wants to delete from the site map's participants table
	3	The user selects to delete the participant
	4	The system asks the user for delete confirmation
	5	The user confirms the deletion
	6	The system removes the participant from the site map's participants table
Extensions	**Step**	**Branching Action**
	5α	5a1. The user cancels the deletion 5a2. The system does not remove the participant from the site map's participants table

Use Case #33	**Edit a participant of a site map**	
Goal In Context	Change the details of a specific site map's participant	
Scope & Level	Shot Site Map management sub-system, sub-function	
Preconditions	A site map from the site maps table is selected	
Success End Condition	The details of the participant are successfully changed	
Failed End Condition	The details of the participant are not changed	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to change the details information of a participant from the site map's participants table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to change some of the details of a specific site map's selected participant
	2	The user selects the participant he wants to change
		The system shows the details of the selected participant
	3	The user changes the desired details
	4	The system saves the changes

Use Case #34	**Add camera to the site map**
Goal In Context	Add a new camera to an already selected site map
Scope & Level	Shot Site Map management sub-system, sub-function
Preconditions	A site map from the site maps table is selected

Success End Condition	The camera is successfully added to the site map	
Failed End Condition	The camera is not added to the site map	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add a camera to the site map's participants table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add a new camera to the site map
	1	The system asks the user to select a camera from the shot's cameras
	2	The user selects a crowd
	3	The system add the camera to the site map's participants table
	4	The user enters the camera's location coordinates
	5	The system adds the camera to the site map
	6	The use executes the step 7 or the use case ends
	7	The user inputs the camera's movements during the specific site map's shot
	8	The system adds the movement to the site map
Extensions	**Step**	**Branching Action**
	2a	2a1. The user chooses to cancel the selection 2a2. The system does not add the new camera and informs the user

Use Case #35	**Delete camera from the site map**	
Goal In Context	Delete camera from the site map	
Scope & Level	Shot Site Map management sub-system, sub-function	
Preconditions	A site map from the site maps table is selected	
Success End Condition	The camera is successfully deleted from the site map	
Failed End Condition	The camera is not deleted from the site map	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to delete the camera from a shot's site map	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete the camera from the selected site map
	2	The user selects the camera from the site map's participants table
	3	The user selects to delete the camera
	4	The system asks the user for delete confirmation
	5	The user confirms the deletion
	6	The system removes the camera from the site map's participants table
	Step	**Branching Action**
Extensions	5α	5a1. The user cancels the deletion 5a2. The system does not remove the camera from the site map's participants table

Use Case #36	Edit the camera of the site map	
Goal In Context	Change the details of a specific site map's camera	
Scope & Level	Shot Site Map management sub-system, sub-function	
Preconditions	A site map from the site maps table is selected	
Success End Condition	The details of the camera are successfully changed	
Failed End Condition	The details of the camera are not changed	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to change the details information of the camera from the site map's participants table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to change some of the details of a specific site map's selected camera
	1	The user selects the camera from the site maps participants table
	2	The system shows the details of the camera
	3	The user changes the desired details
	4	The system saves the changes

Use Case #37	Delete site map	
Goal In Context	Delete site map from the selected shot's site map table	
Scope & Level	Shot Site Map management sub-system, sub-function	
Preconditions	A shot from the shots table is selected	
Success End Condition	The site map is successfully deleted from the site maps table	
Failed End Condition	The site map is not deleted from the site maps table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to delete a site map from a shot's site maps table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete the site map from the selected shot's site map table
	2	The user selects the site map from the site map's table
	3	The user selects to delete the site map
	4	The system asks the user for delete confirmation
	5	The user confirms the deletion
	6	The system removes the site map from the site map's table
	Step	**Branching Action**
Extensions	5α	5a1. The user cancels the deletion 5a2. The system does not remove the site map from the site map's table

Use Case #38	Edit site map
Goal In Context	Change the details of a specific site map
Scope & Level	Shot Site Map management sub-system, sub-function
Preconditions	A shot from the shots table is selected
Success End Condition	The details of the site map are successfully changed
Failed End Condition	The details of the site map are not changed
Primary, Secondary Actors	Event Creator, System
Trigger	The user wants to change the details information of a site map

	from the shot's site maps table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to change some of the details of a specific site map
	1	The user selects the site map from the site maps table
	2	The user changes the desired details
	3	The system saves the changes

Use Case #39	**Manage the camera parameters of a shot**	
Goal In Context	Handle the camera parameters of a specific shot	
Scope & Level	Shots management sub-system, primary task	
Preconditions	A shot from a specifics event's the shot table is selected	
Success End Condition	The changes on the camera parameters table are made successfully	
Failed End Condition	No changes are made	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to manage the camera parameters of a video shot	
Main Success Scenario	**Step**	**Action**
	0	The user wants to handle the details of the selected shot's camera parameters
	1	The user executes one of the steps 2-4
	2	Add camera parameters
	3	Delete camera parameters
	4	Edit camera parameters
	5	The system saves the changes the user made

Use Case #40	**Add camera parameters**	
Goal In Context	Add camera parameters to an already selected shot	
Scope & Level	Shot management sub-system, sub-function	
Preconditions	A shot from the shot table is selected	
Success End Condition	The camera parameters successfully added to the camera parameters table	
Failed End Condition	The camera parameters is not added to the camera parameters table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to add camera parameters to a shot	
Main Success Scenario	**Step**	**Action**
	0	The user wants to add camera parameters to a scenario event's shot
	1	The user selects to the shots technical detailed information tab
	2	The user selects to add new camera parameters to the table
	3	The system asks the user to input some more information about the new camera parameters
	4	The use inputs the information
	5	The system add the new camera parameters to the selected shot's camera parameters table
Extensions	**Step**	**Branching Action**

	4a	4a1. The user chooses to cancel the addition 4a2. The system does not add the new camera parameters and informs the user

Use Case #41	**Delete camera parameters**	
Goal In Context	Delete camera parameters from the selected shot's camera parameters table	
Scope & Level	Shots management sub-system, sub-function	
Preconditions	A shot from the shots table is selected	
Success End Condition	The camera parameters are successfully deleted from the camera parameters table	
Failed End Condition	The camera parameters are not deleted from the camera parameters table	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to delete the camera parameters from a shot's camera parameters table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete the camera parameters from the selected shot's camera parameters table
	2	The user selects the camera parameters from the camera parameters table
	3	The user selects to delete the camera parameters
	4	The system asks the user for delete confirmation
	5	The user confirms the deletion
	6	The system removes the camera parameters from the camera parameters table
	Step	**Branching Action**
Extensions	5α	5a1. The user cancels the deletion 5a2. The system does not remove the camera parameters from the camera parameters table

Use Case #42	**Edit camera parameters**	
Goal In Context	Change the details of a specific shot's camera parameters	
Scope & Level	Shots management sub-system, sub-function	
Preconditions	A shot from the shots table is selected	
Success End Condition	The details of the camera parameters are successfully changed	
Failed End Condition	The details of the camera parameters are not changed	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to change the details information of a camera parameters from the shot's camera parameters table	
Main Success Scenario	**Step**	**Action**
	0	The user wants to change some of the details of a specific shots camera parameters
	1	The user selects the camera parameters from the site maps table
	2	The user changes the desired details
	3	The system saves the changes

Use Case #43	**Save Scenario**	
Goal In Context	Save the created scenario to the database	
Scope & Level	Scenario management sub-system, primary task	
Preconditions	All the information needed to create the scenario are input	
Success End Condition	The scenario is successfully saved in to the database	
Failed End Condition	The scenario is not successfully saved in to the database	
Primary, Secondary Actors	Event Creator, System	
Trigger	The user wants to save the newly created scenario	
Main Success Scenario	**Step**	**Action**
	0	The user wants to save the newly created scenario
	1	The user selects the save the scenario selection
	2	The system asks for saving confirmation
	3	The user confirms the saving
	4	The system saves the scenario to database
Extensions	**Step**	**Branching Action**
	1α	The user does not confirm the saving
		The system confirms the user that the scenario was not saved

Use Case #44	**Explore a scenario**	
Goal In Context	Explore a scenario and retrieve all the information that it contains	
Scope & Level	Scenario management sub-system, primary task	
Preconditions	The system is running	
Success End Condition	The desired information of the scenario are shown	
Failed End Condition	The desired information of the scenario were not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to explore an already created scenario	
Main Success Scenario	**Step**	**Action**
	0	The user wants to explore an already created scenario
	1	The user executes the step 3 or jumps to step 3
	2	The user executes Search a term
	3	The user selects the scenario he wants to explore from the list
	4	The user executes one of the steps 5-7
	5	Explore an event of the scenario
	6	Explore a shot of a scenario event
	7	Delete a scenario

Use Case #45	**Search a term**	
Goal In Context	Search the scenario hierarchy for a specific scenario, event or shot	
Scope & Level	Scenario management sub-system, primary task	
Preconditions	The system is running	
Success End Condition	The desired term is found	
Failed End Condition	The desired term is not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to search the scenario hierarchy for a specific term	
Main Success Scenario	**Step**	**Action**
	0	The user wants to search the scenario list for a specific

		term
	1	The user inputs the term to the search field
	2	The user selects search
	3	The system selects the desired term from the scenario list
Extensions	**Step**	**Branching Action**
	3α	3a1. The term is not found 3a2. The system prompts the user to the first element of the scenario list

Use Case #46	**Explore an event of a scenario**	
Goal In Context	Explore a scenario's event and retrieve all the information that it contains	
Scope & Level	Event management sub-system, primary task	
Preconditions	The system is running	
Success End Condition	The desired information of the event are shown	
Failed End Condition	The desired information of the event were not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to explore a selected scenario's event	
Main Success Scenario	**Step**	**Action**
	0	The user wants to explore a scenario's event
	1	The user executes the step 2 or jumps to step 3
	2	The user executes Search a term
Extensions	3	The user selects the event he wants to explore from the list
	4	The user executes one of the steps 5-7
	5	Explore the participants of a scenario event
	6	Explore the location(s) of a scenario event
	7	Explore the relative time(s) of a scenario event

Use Case #47	**Explore the participants of an event**	
Goal In Context	Explore a scenario's event participants and retrieve all the information that is stored with the participant	
Scope & Level	Roles Involved management sub-system, primary task	
Preconditions	The event is selected	
Success End Condition	The desired information of the participant are shown	
Failed End Condition	The desired information of the participant were not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to explore a selected scenario's event participant	
Main Success Scenario	**Step**	**Action**
	0	The user wants to retrieve the details of a participant of a scenario's event
	1	The user selects a participant from the participants table
	2	The user selects view or double clicks on the participant
	3	The system pop ups a window with the participant's information

Use Case #48	Explore the location(s) of an event	
Goal In Context	Explore a scenario's event location and retrieve all the information that is stored with the participant	
Scope & Level	Spatial Objects management sub-system, primary task	
Preconditions	The event is selected	
Success End Condition	The desired information of the location are shown	
Failed End Condition	The desired information of the location were not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to explore a selected scenario's event location	
Main Success Scenario	**Step**	**Action**
	0	The user wants to retrieve the details of a location of a scenario's event
	1	The user selects a location from the locations table
	2	The user selects view or double clicks on the location
	3	The system pop ups a window with the location's information

Use Case #49	Explore the relative time(s) of an event	
Goal In Context	Explore a scenario's event relative time and retrieve all the information that is stored with the participant	
Scope & Level	Time management sub-system, primary task	
Preconditions	The event is selected	
Success End Condition	The desired information of the relative time are shown	
Failed End Condition	The desired information of the relative time were not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to explore a selected scenario's event relative time	
Main Success Scenario	**Step**	**Action**
	0	The user wants to retrieve the details of a relative time of a scenario's event
	1	The user selects a relative time from the relative times table
	2	The user selects view or double clicks on the relative
	3	The system pop ups a window with the relative time's information

Use Case #50	Explore a shot of a scenario event	
Goal In Context	Explore a scenario's event shot and retrieve all the information that is stored with the participant	
Scope & Level	Event Shots management sub-system, primary task	
Preconditions	The system is running	
Success End Condition	The desired information of the shot are shown	
Failed End Condition	The desired information of the shot were not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to explore a selected scenario's event shot	
Main Success Scenario	**Step**	**Action**
	0	The user wants to explore a scenario's shot
	1	The user executes the step 2 or jumps to step 3
	2	The user executes Search a term

Extensions	3	The user selects the shot he wants to explore from the list
	4	The user executes one of the steps 5-8
	5	Play a video shot of a specific scenario
	6	Explore the participants of a video shot
	7	Explore the location(s) of a video shot
	8	Explore the elevations of a video shot

Use Case #51	**Play the video of a specific scenario scene**	
Goal In Context	To play the video that was related to a specific shot and reproduce the site map of the shots geometry	
Scope & Level	Event Shots management sub-system, primary task	
Preconditions	A shot is selected	
Success End Condition	The video is played and the site geometry is reproduced successfully	
Failed End Condition	The video is not played or the site geometry is not reproduced successfully	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to play the video that was related to a specific shot and reproduce the site map of the shots geometry	
Main Success Scenario	**Step**	**Action**
	0	The user wants to play the video that was related to a specific shot and reproduce the site map of the shots geometry
	1	The user selects play to video related to the specific shot
	2	The system reproduce both the video and the site geometry of the shot

Use Case #52	**Explore the participants of a video shot**	
Goal In Context	Explore a scenario's shots participants and retrieve all the information that is stored with the participant	
Scope & Level	Roles Involved management sub-system, primary task	
Preconditions	A shot is selected	
Success End Condition	The desired information of the shot's participant are shown	
Failed End Condition	The desired information of the shot's participant were not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to explore a selected scenario's shots participant	
Main Success Scenario	**Step**	**Action**
	0	The user wants to retrieve the details of a participant of a scenario's shot
	1	The user selects a participant from the shot's participants table
	2	The user selects view or double clicks on the participant
	3	The system pop ups a window with the participant's information

Use Case #53	Explore the location(s) of a video shot	
Goal In Context	Explore a scenario's shots location and retrieve all the information that is stored with the participant	
Scope & Level	Spatial Objects management sub-system, primary task	
Preconditions	A shot is selected	
Success End Condition	The desired information of the shot's location are shown	
Failed End Condition	The desired information of the shot's location were not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to explore a selected scenario's shots location	
Main Success Scenario	**Step**	**Action**
	0	The user wants to retrieve the details of a location of a scenario's shot
	1	The user selects a location from the shot's locations table
	2	The user selects view or double clicks on the location
	3	The system pop ups a window with the location's information

Use Case #54	Explore the elevations of a video shot	
Goal In Context	Explore a scenario's elevation shots and retrieve all the information that is stored with the participant	
Scope & Level	Event Shots management sub-system, primary task	
Preconditions	A shot is selected	
Success End Condition	The desired information of the shots elevations are shown	
Failed End Condition	The desired information of the shots elevations were not found	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to explore a selected scenario's shots elevations	
Main Success Scenario	**Step**	**Action**
	0	The user wants to retrieve the details of a shot elevation
	1	The user selects an elevation from the available elevations of a shot
	2	The system pop ups a window with the location's information

Use Case #55	Delete a Scenario	
Goal In Context	Delete a complete scenario as the user considers necessary	
Scope & Level	Scenario management sub-system, primary task	
Preconditions	The system is running	
Success End Condition	The scenario is completed removed	
Failed End Condition	The scenario is not removed	
Primary, Secondary Actors	Multimedia Editor/Viewer, System	
Trigger	The user wants to delete a scenario	
Main Success Scenario	**Step**	**Action**
	0	The user wants to delete a scenario
	1	The user selects the scenario he wants to delete
	2	The user selects the delete scenario option
	3	The system asks the user to confirm the deletion
	4	The user confirms the deletion

	5	The system completely removes the scenario
Extensions	**Step**	**Branching Action**
	4α	4a1. The user cancels the deletion 4a2. The scenario is not deleted

6. SUMMARY

In this chapter we described the functional aspects of the mobile multimedia event capturing and visualization application using the Use Cases methodology. First we presented the overall picture of the use cases. Next we presented the summary table of the use cases where there was a short description of each use case. Finally we provided a full description of the use cases that describe the functional aspects of our application.

CHAPTER 7
MOBILE MULTIMEDIA EVENT CAPTURING AND VISUALIZATION APPLICATION (MOME), USER INTERFACE

1. INTRODUCTION

In this chapter, there will be a detailed description of the user interface implementation of the mobile multimedia event capturing and visualization application. First, we present a description of the ten usability Heuristics principles that were written by Jacob Nielsen (31) and how are they applied to the user interface of the application. Next, with the use of screenshots, we will describe the user interface of the application that every user that we described in Chapter 3 uses in detail.

2. USER INTERFACE PRINCIPLES AND GUIDELINES

2.1. Visibility of System Status

- The system should always keep users informed about what is going on, through appropriate feedback in reasonable time.

The application lets always the user to know where he is and is going on. That is accomplished by having a status bar, located at the down left bottom of the application that in any moment will give the user his whereabouts. Furthermore, it is achieved by always giving accurate titles at the application dialogs and panels.

2.2. Match between System and the Real World

- The system should speak in user’s language with words, phrases and concepts familiar to the user, rather than system oriented terms. Follow real-world conventions, making information appear in a natural order.

The application uses terms and phrases that are commonly used in the real world and in the field of cinematography and video shooting and are explanatory so the user can always know what is happening and what every term means without being necessary to be a professional or an expert of the cinematography and video field.

2.3. User Control and Freedom

- Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state, without having to go through an extended dialogue.

In the most cases, the user can make the step he would like without being forced to follow a specific path. However, if he tries to follow a wrong path, error messages are shown as necessary, to prevent the user from taking that wrong path. Also, the user can always return to the main dialog in case he is in an unwanted state.

2.4. Consistency and Standards

- Users should not have to wonder whether different words, situations, or actions mean the same thing. Formulate platform conventions.

In every panel and dialog of the application, the same design logic (e.g. same colors, the same names in buttons that perform the same actions etc) is used.

2.5. Error Prevention

- Even better than good error messages is a careful design, which prevents a problem from occurring in the first place.

The user is prevented from making mistakes, because in every step the application provides him with all the necessary actions only, to prevent him from making wrong decisions and taking wrong actions.

2.6. Recognition rather than recall

- The user should not have to remember information from one dialogue to another.
  - Make objects, actions, and options visible.
  - Instructions for use of the system should be visible or easily retrievable whenever appropriate.

The user is not obligated to remember, in a given time, information from one dialog to another. In every point of the application there are sufficient elaborations in order to direct the user to do what he needs to do. The elaborations are short explanatory messages (tooltips), or even short labels that explain the process that can be followed.

2.7. Flexibility and Efficiency of Use

- Accelerators -unseen by novice users - often speed up the interaction for the expert user, such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

The application can be used by expert users or even naïve users without discrimination. In most cases the application is efficient and the calculations or the work load are not shown to the user, except from the first usage of the video technology were the application gives the user the status of the work in order to always know what is going on.

2.8. Aesthetic and Minimalistic Design

- Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

The look and feel of the application is the same with the operating system. That way, the application is closer to what the user is used to. Furthermore, an effort is made in order to not give unnecessary information to the user and to make the messages and labels as minimal as possible.

2.9. Help Users Recognize, Diagnose, and Recover from Errors

- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and then constructively suggest a solution

As the principle suggests, the error messages of the application are in plain language, they precisely indicate the problem and then a solution is suggests constructively.

2.10. Help and Documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide and documentation. Any such information should be easy to search, focused on the user's task, list concisely the steps to be carried out, and not be too large.

The next section, will describe in details the graphical user interface of the application and the functionality that it provides.

3. USER INTERFACE AREAS DESCRIPTION

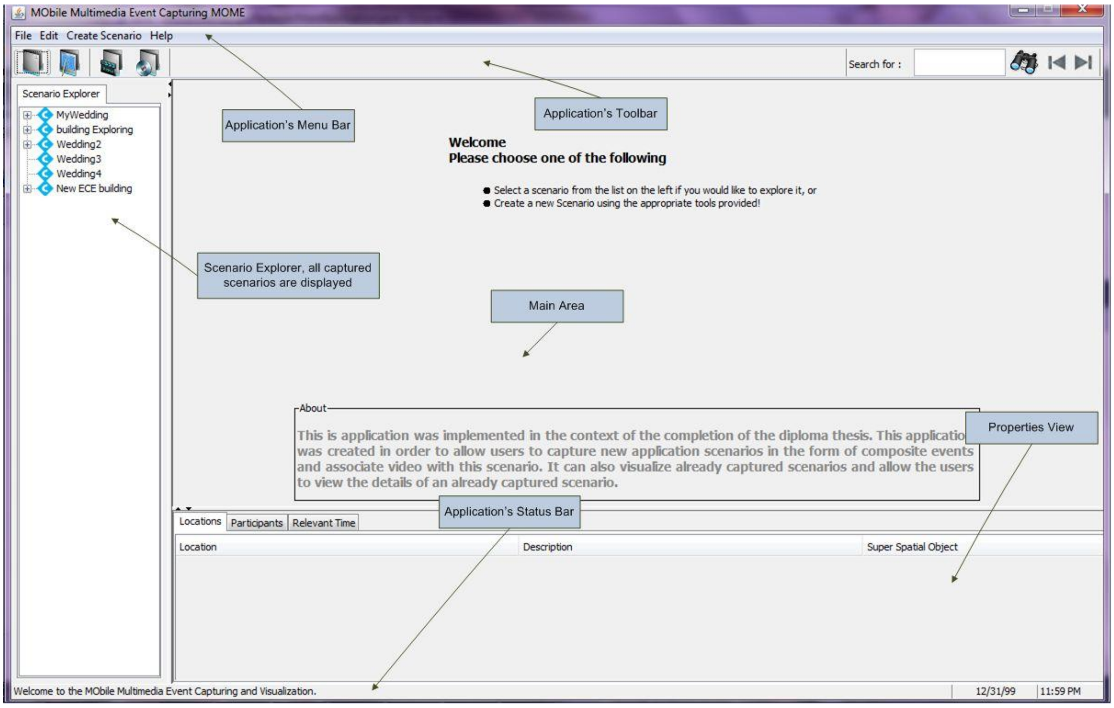
Most of the users we described in Chapter 3, in order to perform their various tasks, use the mobile multimedia event capturing and visualization (MOME) application. In this section we will describe which functionality of the MOME application each of the users uses.

The application is divided in two main sub-systems. The sub-system that captures the events and the sub-system that visualizes the events and the information that it contains. The Event Creator along with the Multimedia Event Capturing Personnel uses the capturing sub-system of the application. The Multimedia Editor/Viewer uses the visualization sub-system. The Event Type Creator does not use the application directly. He is the one that creates the various Event Types that the capturing sub-system uses as an input to create the specific Events.

Even though both sub-systems belong to the same application, the dissection is necessary for the comprehensible description of the application.

3.1. USER INTERFACES FOR EVENT CREATOR

When the application starts running, a welcome window is displayed. In this window someone can see the main areas of the application. The main areas of the application are three, the scenario explorer, the properties view and the main area of the application. Besides the main areas, the application contains a menu bar, a toolbar and a status bar. The functionality of each area will be described later. Figure 3.1 shows the welcome window of the application along with the different areas it contains.



Welcome
Please choose one of the following

- Select a scenario from the list on the left if you would like to explore it, or
- Create a new Scenario using the appropriate tools provided!

About

This is application was implemented in the context of the completion of the diploma thesis. This applicatio was created in order to allow users to capture new application scenarios in the form of composite events and associate video with this scenario. It can also visualize already captured scenarios and allow the users to view the details of an already captured scenario.

Welcome to the MObile Multimedia Event Capturing and Visualization.

Figure 7.1 Welcome Window

➢ *Menu Bar*

The application defines its own menu in the menu bar, which provides the basic functionalities of the application. The standard menus provided by the eclipse platform are also maintained.

The possible options and functionalities that are provided by the application through its graphical user interface menu bar are the following:

Besides the above options, the application provides the default options:

File → Exit, this selection exits the application

Edit→ Copy, Cut, Paste, these selections perform the basic copy, cut and paste operations accordingly. The menu item **Delete Scenario** appears only when a scenario from the scenario explorer is selected. This selection will allow the user to delete an already existing scenario from the list of the scenario explorer.

The selection **Create Scenario** → will be described in the next section.

The selection **Create Scenario** → will be described in the next section.

Help→ About, this selection gives the user information about the application.

➢ *Status Bar*

The status bar of the application is a panel bar where messages about the situation, that the user is in, are provided. The messages may refer to where the user is at a certain point, or they can be error or warning messages.

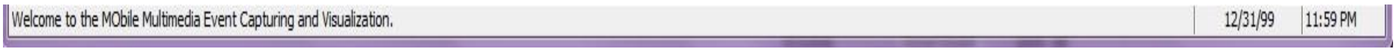

Welcome to the MObile Multimedia Event Capturing and Visualization.
12/31/99
11:59 PM

Figure 7.2 Status bar

The status bar has no more functionality than to provide informative messages about the status of the application.

3.2. USER INTERFACES FOR EVENT CREATOR

The event creator as we described in Chapter 3, is the one that creates a new specific composite event (e.g. the wedding of John and Mary) representing a real life scenario. In our application we call this a “Scenario”.

- *Create a new Scenario*

The event creator can create a new scenario, by either using the menu “Create Scenario” from the Menu Bar or using one of the options provided in the Toolbar of the application.

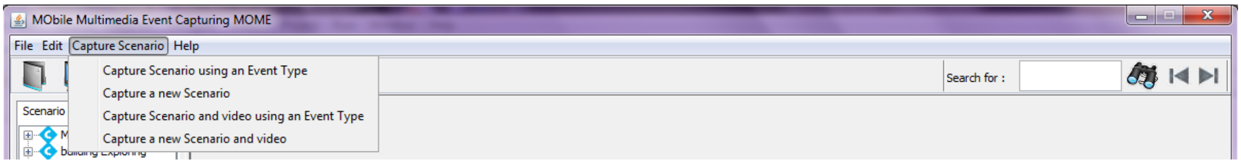

MObile Multimedia Event Capturing MOME
File
Edit
Capture Scenario
Help
Capture Scenario using an Event Type
Capture a new Scenario
Capture Scenario and video using an Event Type
Capture a new Scenario and video
Search for :
Scenario

Figure 7.3 Menu bar

The selection **Capture Scenario** → Capture Scenario and video will allow the user to start a video capturing and associate it with a new scenario, using the application.

The selection **Capture Scenario** → Capture Scenario will allow the user to create a new scenario and associate it with an already captured video, using the application.

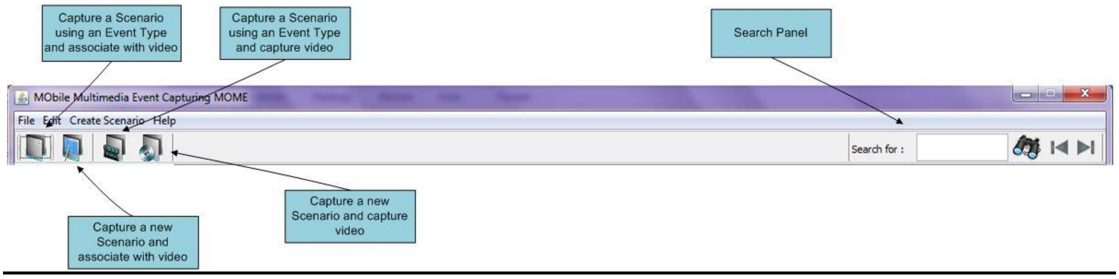

Capture a Scenario using an Event Type and associate with video
Capture a Scenario using an Event Type and capture video
Search Panel
MObile Multimedia Event Capturing MOME
File
Edit
Create Scenario
Help
Search for :
Capture a new Scenario and associate with video
Capture a new Scenario and capture video

Figure 7.4 Toolbar

In the toolbar that the eclipse platform provides, action buttons are added for the completion of certain basic functionalities.

The possible options and functionalities that are provided by the MOME application through its graphical user interface toolbar are the following:

Capture a new Scenario. Video Capturing: The selection will allow the user to start a video capturing and create a new scenario, from scratch, using the application.

Capture a new Scenario using Event Type – Ontology. Video Capturing: The selection will allow the user to start a video capturing and create a new scenario from imported Event Type – ontology, using the application.

Capture a new Scenario. Video associating: The selection will allow the user to create a new scenario from scratch and associate it with an already captured video, using the application.

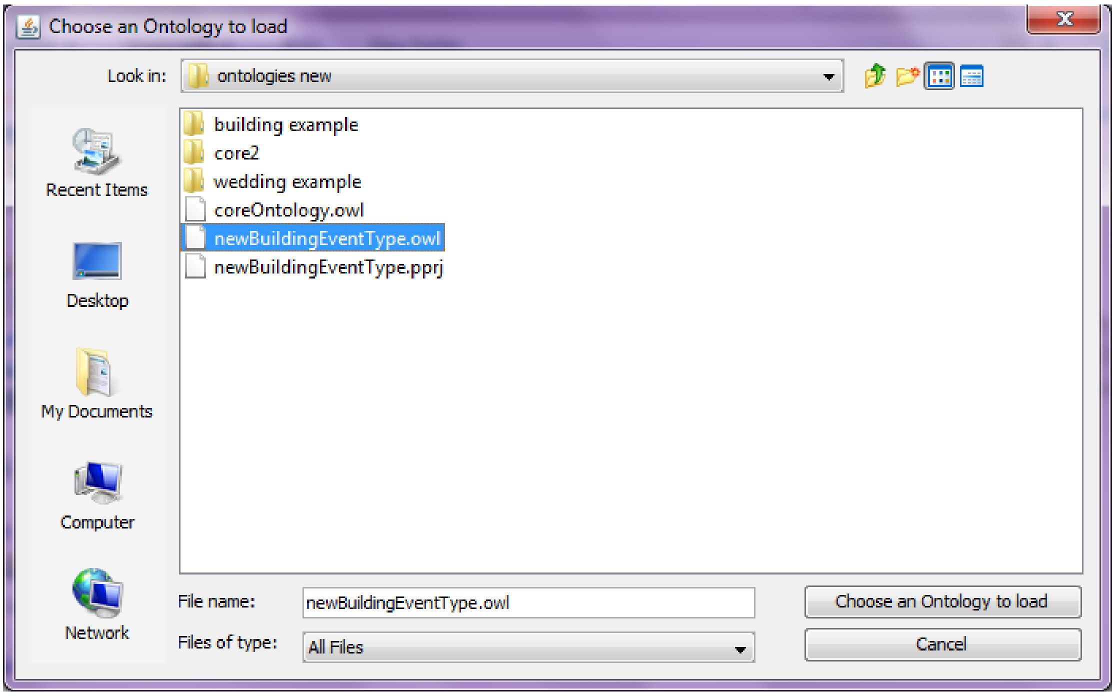
Capture a new Scenario from Event Type – Ontology. Video associating: The selection will allow the user to create a new scenario from imported Event Type – ontology and associate it with an already captured video, using the application.

Search: This functionality is for the Multimedia Editor/Viewer and it will be explained later.

○ *Manage a Scenario*

When the user selects the creation of a new scenario, the first action from the application's view is to prompt the user to select an event type ontology that the application can use to guide and help the user during the creation of the new scenario.

The following figure shows the window that is displayed to the user to prompt him to choose an event type ontology.



Look in: ontologies new

building example
core2
wedding example
coreOntology.owl
newBuildingEventType.owl
newBuildingEventType.pprj

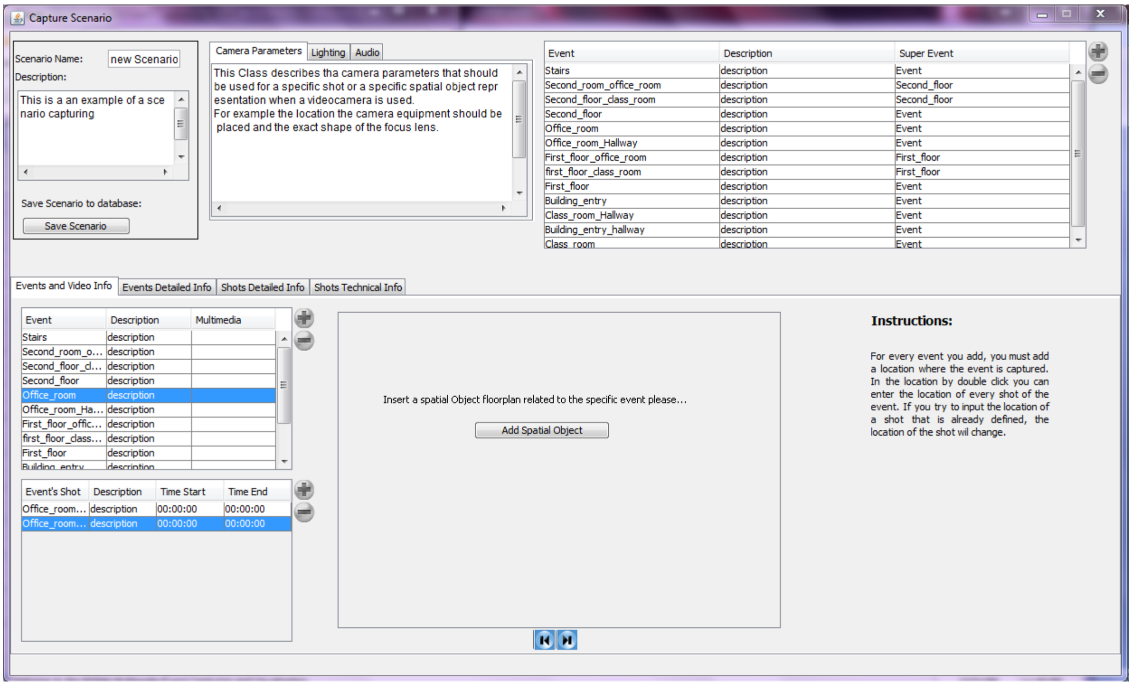
File name: newBuildingEventType.owl

Files of type: All Files

Figure 7.5 The Ontology selection window

If the user chooses to use an ontology, then the application loads the ontology's events, shots and all the other classes it considers necessary. If the user chooses not to use an ontology, the application starts the creation of the new scenario from the scratch guiding the user to complete the necessary information.

After the ontology selection option, the application starts by displaying a new window that will guide the user through the creation. The following figure shows the first screen of this window.



Scenario Name: new Scenario

Description:

This is a an example of a sce nario capturing

Save Scenario to database:

This Class describes tha camera parameters that should be used for a specific shot or a specific spatial object repr esentation when a videocamera is used.
For example the location the camera equipment should be placed and the exact shape of the focus lens.

Event	Description	Super Event
Stairs	description	Event
Second_room_office_room	description	Second_floor
Second_floor_class_room	description	Second_floor
Second_floor	description	Event
Office_room	description	Event
Office_room_Hallway	description	Event
First_floor_office_room	description	First_floor
first_floor_class_room	description	First_floor
First_floor	description	Event
Building_entry	description	Event
Class_room_Hallway	description	Event
Building_entry_hallway	description	Event
Class_room	description	Event

Event	Description	Multimedia
Stairs	description	
Second_room_o...	description	
Second_floor_cl...	description	
Second_floor	description	
Office_room	description	
Office_room_Ha...	description	
First_floor_offic...	description	
first_floor_class...	description	
First_floor	description	
Building_entry	description	

Event's Shot	Description	Time Start	Time End
Office_room...	description	00:00:00	00:00:00
Office_room...	description	00:00:00	00:00:00

Insert a spatial Object floorplan related to the specific event please...

Instructions:

For every event you add, you must add a location where the event is captured. In the location by double click you can enter the location of every shot of the event. If you try to input the location of a shot that is already defined, the location of the shot wil change.

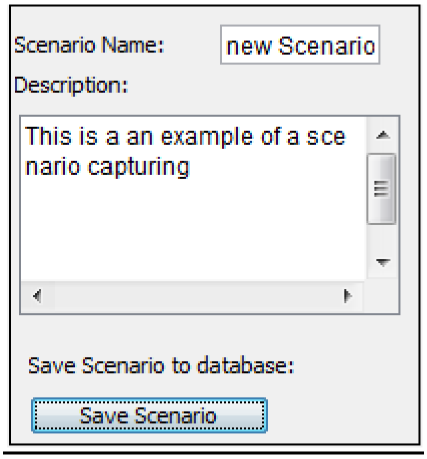
Figure 7.6 Scenario creation main window

This window is composed by four different areas. The first up and the left corner displays the basic scenario details (name, description). The one next to it displays the guidelines that were described on the chapter 4 and are loaded from the ontology. The next area shows the ontology loaded scenes. Finally the area below all the above is the scenario processing area. This area contains the important information to be completed about the new scenario. These areas are also shown on the previous figure. Next follows a more detailed description of each area shown in the figure and the functionality every one of these areas provides.

- ### *Scenario Information*

The scenario information area is the area where the user will input the basic information about the new scenario. These are the name of the scenario, and a description if he would like.

This area also contains a Save Changes button. When the user selects to save the changes, the changes he made to the basic scenario information will be saved.



Scenario Name: new Scenario

Description:

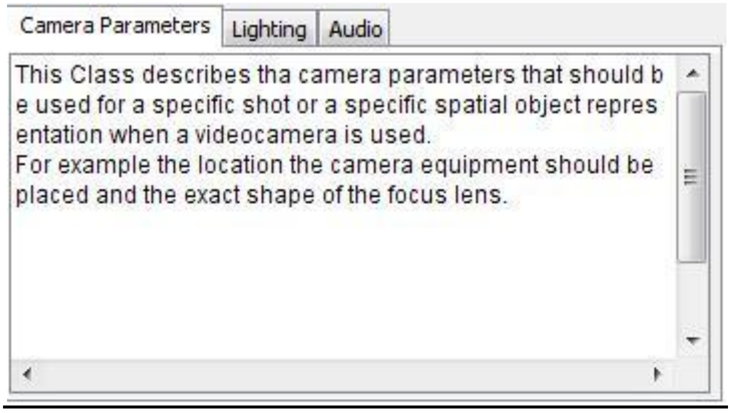
This is a an example of a sce
nario capturing

Save Scenario to database:

Figure 7.7 Scenario basic Information panel

- *Guidelines*

The guidelines area is the area where the guidelines loaded from the ontology for each different shot are shown. When a shot of the shot table is selected, the system shows any available shot guidelines. The guidelines are shown as a text.



This Class describes tha camera parameters that should b
e used for a specific shot or a specific spatial object repres
entation when a videocamera is used.
For example the location the camera equipment should be
placed and the exact shape of the focus lens.

Figure 7.8 Scenario Capturing guidelines

There is no further functionality provided by this area.

- *Ontology navigator*

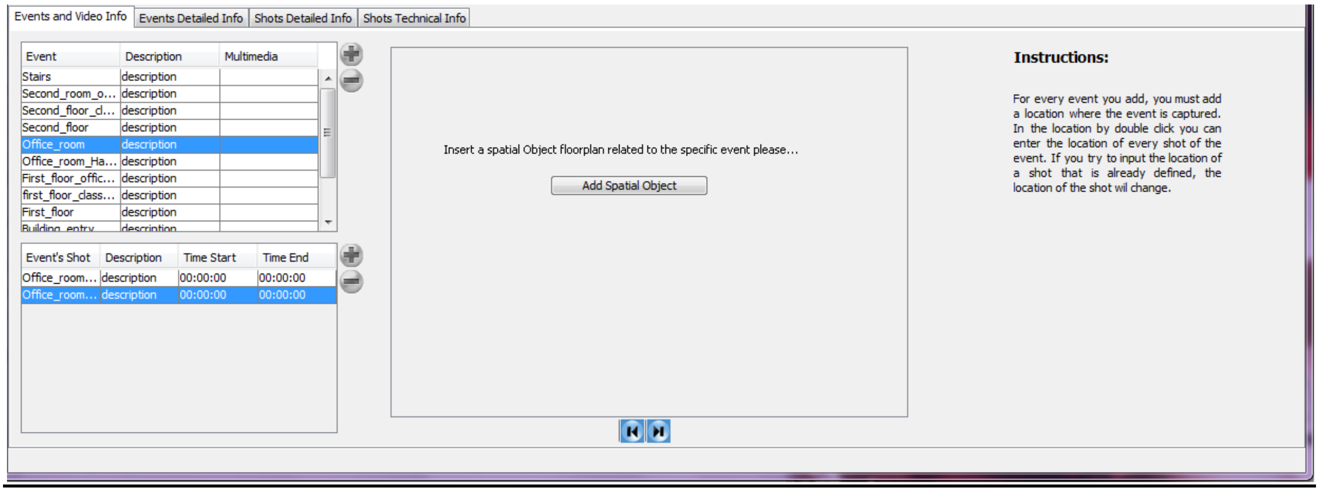
The ontology navigator area is the area where the event type classes from the selected ontology are shown in a list form.

The ontology navigator provides events that are more likely the user would want to add them to his scenario. That functionality provides the add button that is located to the up right side of the ontology navigator area. When the user selects to add an ontology event type to the scenario, the specific event is added to the scenario's events table.

- *Main scenario processing area*

The main scenario processing area is the area where all the scenario information is provided, by the user.

First it's the Scenes and Video Information tab. This tab contains the information about the events and shots of the scenario. At this tab the information can be provided either during the recording of the video or after the recording.



Event	Description	Multimedia
Stairs	description	
Second_room_o...	description	
Second_floor_cl...	description	
Second_floor	description	
Office_room	description	
Office_room_Ha...	description	
First_floor_offic...	description	
first_floor_class...	description	
First_floor	description	
Building_entry	description	

Event's Shot	Description	Time Start	Time End
Office_room...	description	00:00:00	00:00:00
Office_room...	description	00:00:00	00:00:00

Insert a spatial Object floorplan related to the specific event please...

Instructions:

For every event you add, you must add a location where the event is captured. In the location by double click you can enter the location of every shot of the event. If you try to input the location of a shot that is already defined, the location of the shot wil change.

Figure 7.9 Scenario scenes and shots Information

This panel contains a list of the events the user adds to the scenario. The user can either add or delete an event from the list. When he chooses to add an event, a dialog window pop ups and requests from the user to fill in the events basic information. However there is another way to input information to the list of events. When the user starts capturing or playing a video, he combines the specific video to one or more events from the already existing events' list. Besides adding an event, the user can select to delete a events or to edit a event. When the user chooses to delete an event, the system requires confirmation of the deletion and performs or not the deletion accordingly. When the user chooses to edit an event, he double clicks on the specific event and the information becomes editable. The changes of a event's information are applied as long as the user stops editing.

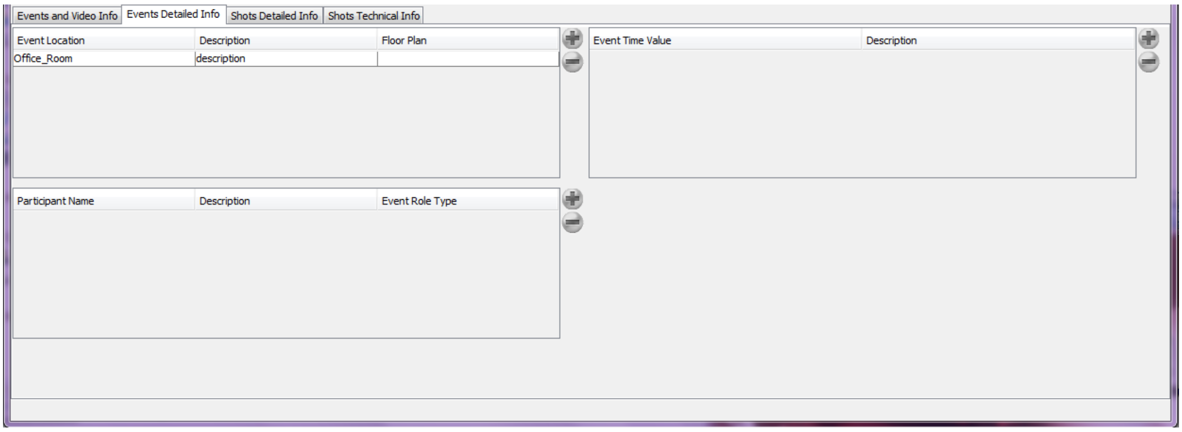
- ### *Manage a Shot of an Event*

The Scenes and Video Information tab also contains a list of the shots of each of the previous events. Again the user can either add or delete a shot from the list. When the user chooses to add a shot, a dialog window pop ups and requests from the user to fill in the shot's basic information. Besides adding a shot, the user can select to delete a shot or to edit a shot. When the user chooses to delete a shot, the system requires confirmation of the deletion and performs or not the deletion accordingly. When the user chooses to edit a shot, he double clicks on the specific shot and the information becomes editable. The changes of a shot's information are applied as long as the user stops editing.

Finally this tab's panel contains the maps of the spatial objects (locations) involved in a specific event. When a location is added to the event's information, the map of this location is shown to the right side of this tab. Here the user can input the location and direction of the shots of the specific event, as well as other spatial objects that may be included to the event. The user has to add a location in the map, for every shot he adds to the list of shots. The user can navigate through the different maps of a specific event (if any) with the right and left arrows located to the bottom center of the map panel.

- *Manage the roles involved, spatial objects (locations) and time of an event*

The second tab contains more detailed information about a selected event from the list of events. This tab contains three different lists a list of all the participants of the selected event, a list of all the locations of the selected event and a list of all the relative times that the selected event occurred at. For each of these lists the operations add, delete and edit that were described above can also be performed.



Event Location	Description	Floor Plan
Office_Room	description	

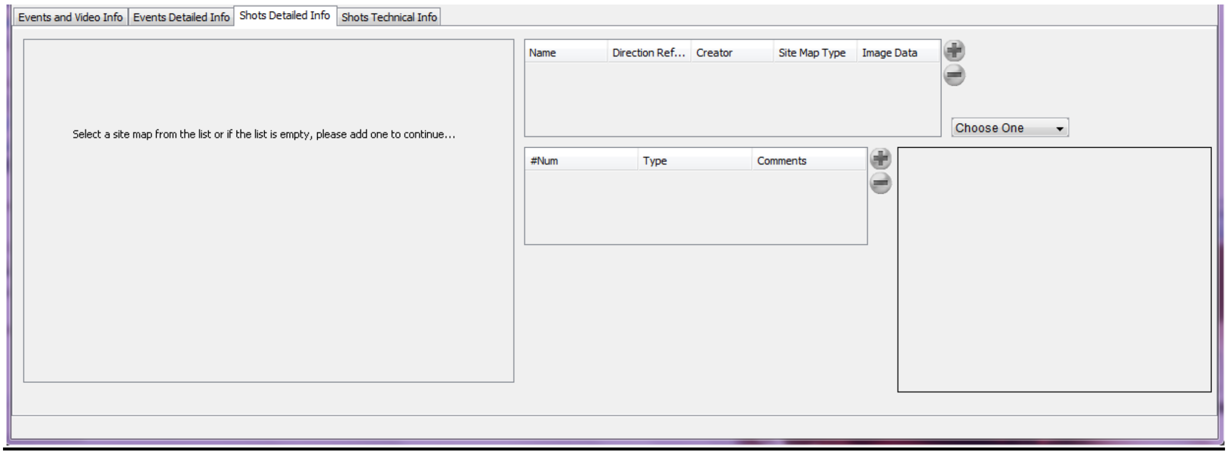
Event Time Value	Description

Participant Name	Description	Event Role Type

Figure 7.10 A selected scene's detailed properties information

- *Manage the site maps of a shot*

The next tab contains more detailed information about the site map of a selected shot from the list of shots. This tab contains all the information about the site geometry of a shot. That is the site maps of the shot, the participants of the shot, the spatial objects (locations), the location and movement of every participant in the site map during the shot and the location and the movement of the camera during the shot.



Select a site map from the list or if the list is empty, please add one to continue...

Figure 7.11 A selected shot's site map information

In more detail, this tab's panel contains a list of the site maps of the selected shot. The user can select to add, delete or edit a site map from the list. When the user chooses to ad a site map, a dialog window pop ups asking him to fill in the basic information about the specific site map. If the user, in this dialog chooses that this is the main site map of the shot (the site map types are described on chapter 2), then the map of the shot's location that was decided in the first tab, will be displayed in the site map's panel. In any different case, the system will ask the user to input the map of the site map that they choose. The delete and edit operations are the same as described before. When a site map is added, the user can add participants to the site map.

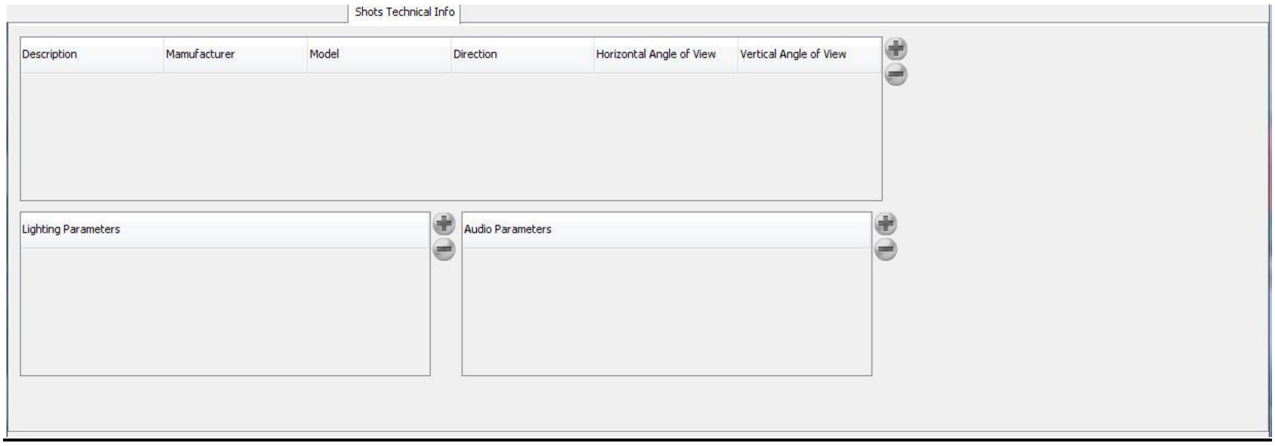
To add anything to the site map the user should first select from the drop down menu what it is he wants to add. If he selects to add an object, then he would have to choose one of the participating objects that have already been added to the scene's list. If he selects to add a person or a crowd, then he would have to choose one of the participating persons or crowds that have already been added to the scene's list. If he selects to add a spatial object (location), then he would have to choose one of the participating spatial objects that have already been added to the scene's list. Also, the user can select to add the location and movement of the camera, the audio and the lighting of the shooting.

For every of the above that he would choose to add, at the details panel, the appropriate forms will appear in order for the user to fill in the location and/or the movement of each of the above. Next an example where the user chooses to add an object to the site map is shown.

Every object that is added to the site map is also added to the shot's participants list. When an object belongs to the shot's participant list, the user can delete or edit its contents. For the deletion operation the same process that was described before is followed. For the editing although, when a user selects an object from the list, the details panel shows the information of that object. In the details panel the user can edit the information that will be stored immediately after the user stops editing.

- *Manage the camera parameters of a shot*

The last tab contains the technical parameters information of a selected shot. This tab contains the cameras technical parameters that need to be completed by the user and two lists of the audio and lighting technical parameters that can be added to the selected shot. The camera parameters that need to be input by the user are in the top panel and are editable text fields. When the user wants to edit the camera parameters the system automatically saves and displays the parameters that are available to the shot's site map on the previous described tab.



Description	Mamufacturer	Model	Direction	Horizontal Angle of View	Vertical Angle of View

Lighting Parameters

Audio Parameters

Figure 7.12 A selected shot's technical parameters information

The lists of the audio and lighting parameters are handled in the way the scene's detailed information tab is handled. That is, for each of these lists the operations add, delete and edit that were described above can be performed.

3.3. USER INTERFACES FOR THE MULTIMEDIA EVENT CAPTURING PERSONNEL

- *Capture/play a video*

In both cases where the user wants to capture a video in order to associate it with the events of the scenario or to associate an already captured video with the events of the scenario the same windows is displayed which gives the user both the options.

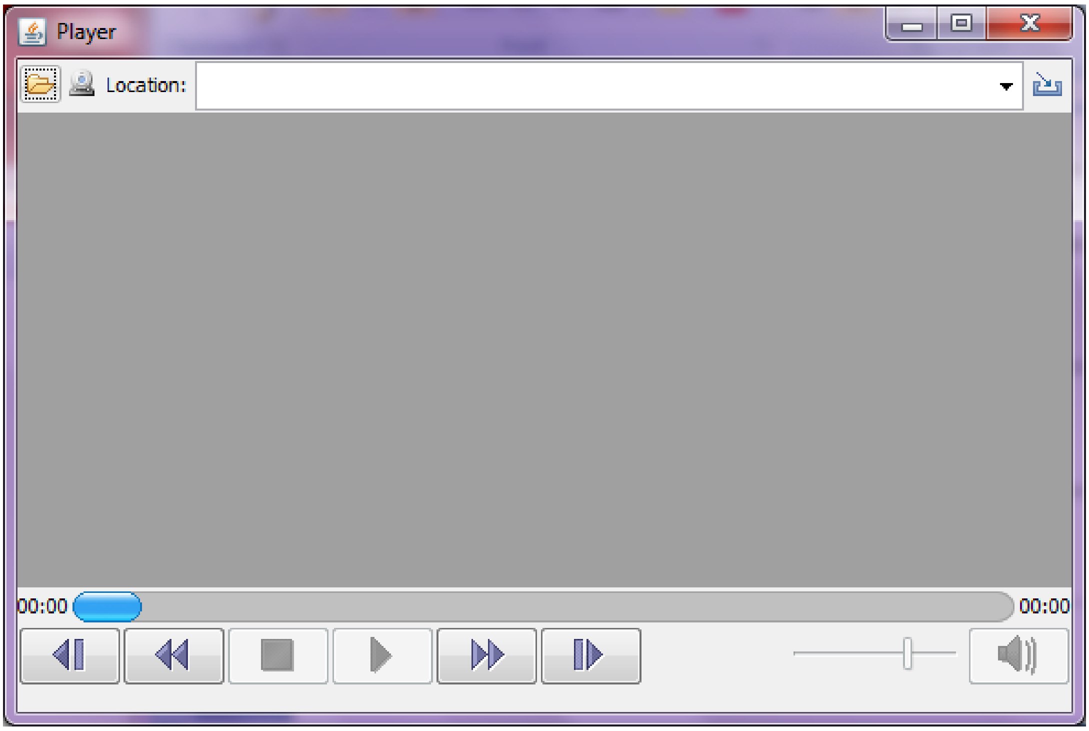


Figure 7.13 Video Player Used by the application for Capturing and playing videos

- *Manage a shot of an event*

This user can manage the event shots in the same way as we described for the event creator. However, when the user is on the same time also recording or reproducing a video, along with the information input by the user, the timestamp of the video where the user chose to relate the new shot will also be added. If the video recording/reproducing has not yet started or it has finished, the information for the timestamp will be requested to be added manually by the user. The user can choose to add the video timestamp to a shot that has already been added to the list of shots. When that happens the user will double click the shot he chose while the video is recording/reproducing, and the application will add the video timestamp to the selected shot.

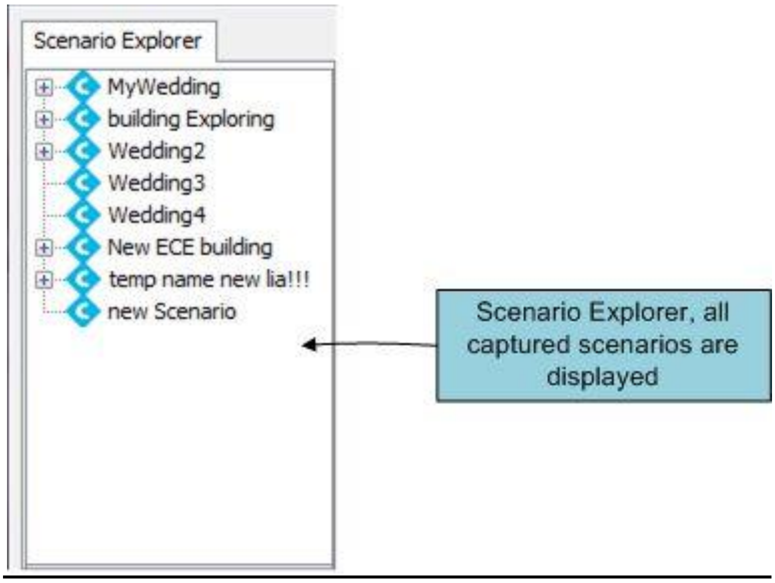
3.4. USER INTERFACES FOR MULTIMEDIA EDITOR/VIEWER

- *Explore a Scenario*

- *Scenario Explorer*

The scenario explorer provides a scenarios tree representation method. The scenarios displayed were created using the application’s event capturing sub-system.

The scenario explorer is used to allow the user to navigate through the various scenarios that are currently available. The user selects a scenario and has the option to navigate through the scenarios events and the various shots that it contains.



Scenario Explorer

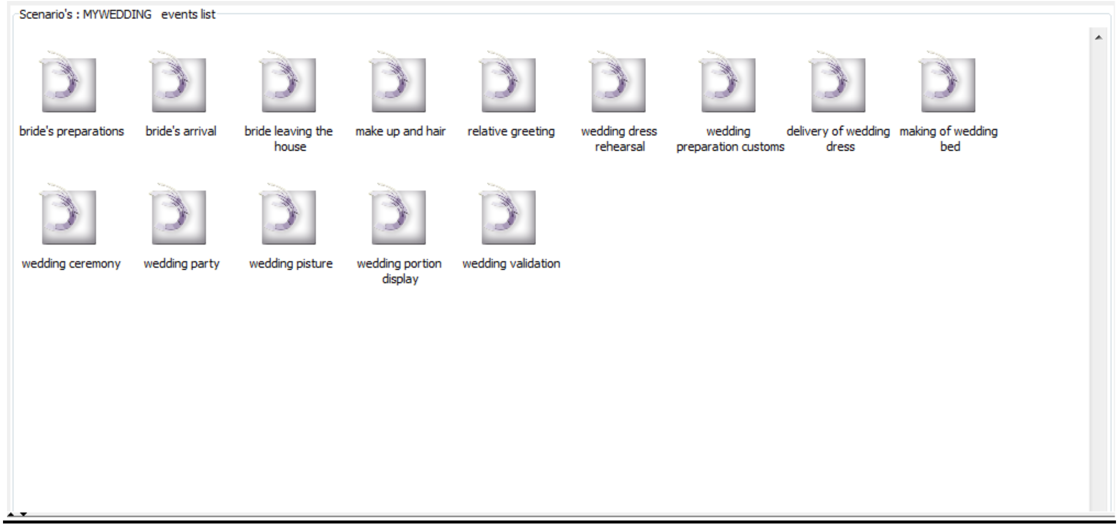
MyWedding
building Exploring
Wedding2
Wedding3
Wedding4
New ECE building
temp name new lia!!!
new Scenario

Scenario Explorer, all captured scenarios are displayed

Figure 7.14 Scenario Explorer

- *Main Area*

When a specific Scenario is selected, the main area displays the events of the specific scenario. The next figure shows the state of the main area, when a scenario from the scenario explorer is selected.



Scenario's : MYWEDDING events list

bride's preparations
bride's arrival
bride leaving the house
make up and hair
relative greeting
wedding dress rehearsal
wedding preparation customs
delivery of wedding dress
making of wedding bed
wedding ceremony
wedding party
wedding pisture
wedding portion display
wedding validation

Figure 7.15 Main Area, Display of the scenes of a selected scenario

In this state, the main area displays the events of a selected scenario. When a user selects a specific event, the properties view shows more details about the event concerning the participants, locations and relative time of the event.

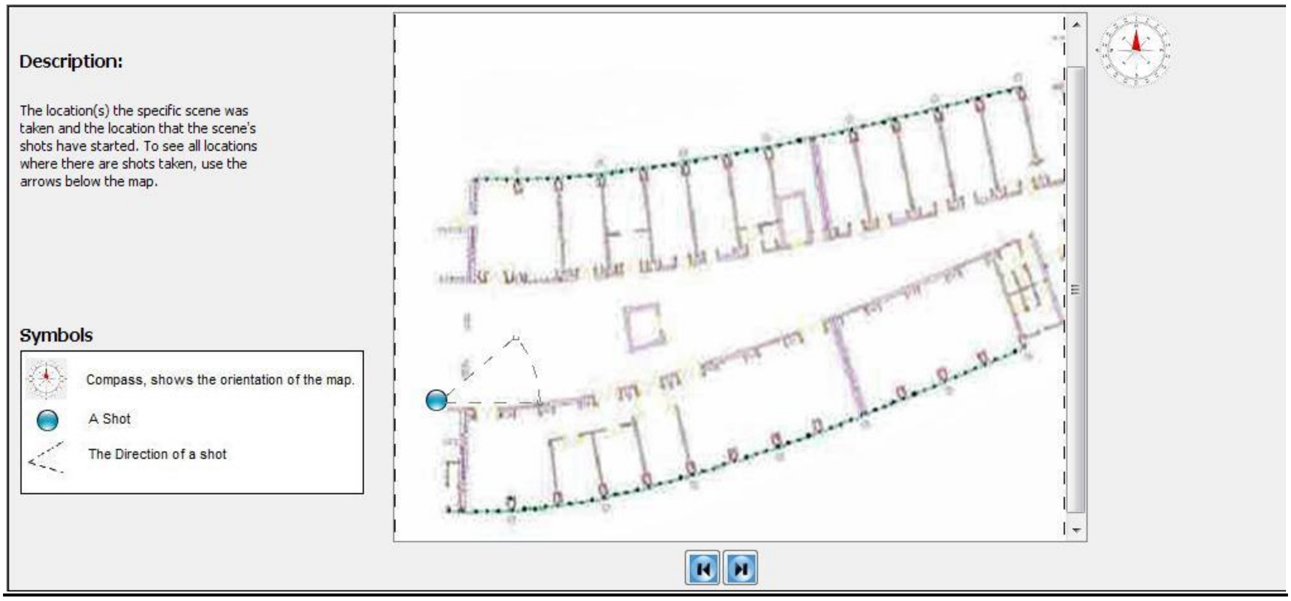
○ *Search a term*

The user can search through the scenario explorer by using the Search option provided in the toolbar. The toolbar also contains a search mechanism that is used to search for a specific item in the scenario explorer. The term will be search to be found in the scenarios names, the events of the scenarios or the shots of all the events. The next and previous buttons are used to navigate through the scenario explorer in order to traverse all the items that contains the desired term. When there term is not found, or when the top or the bottom of the list is found, the application informs the user through messages in the status bar.

○ *View an event of a Scenario*

When a specific event is selected the main area displays the details of that event. An event can be selected by two different ways. First the user can double click on a specific event of the current state. Second, the user can select a event from the scenario explorer. By doing one of the above, the main area jumps to the next state where the location map of the event along with the shots of the current scene is displayed.

The next figure shows an example of a selected event and what the main area displays.



Description:

The location(s) the specific scene was taken and the location that the scene's shots have started. To see all locations where there are shots taken, use the arrows below the map.

Symbols

Compass, shows the orientation of the map.

A Shot

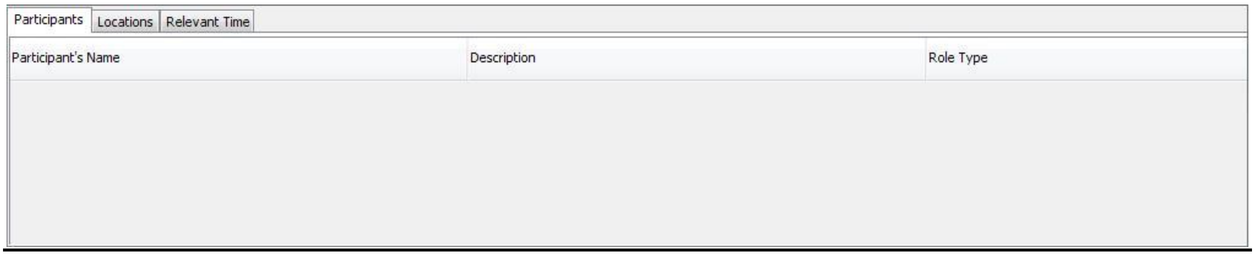
The Direction of a shot

Figure 7.16 Main Area, Display of the shots of a selected scene

In this state, the user can perform one of the following. By using the next and previous arrows, the user can move along the locations that the current event took place and see where the available shots of the selected event were shot. Besides that, the user can select a specific shot that he wants to see in detail. By double clicking on a specific shot of the selected event, the main are jumps to the next state where the video and the site map of a specific shot are displayed. Just like before, the user can go to the shot state of the main area by selecting a specific shot from the scenario explorer.

- *Properties View*

The properties view area of the application is used as its name suggests. It displays the properties of a specific item selected. Particularly, when a specific event is selected, the properties view shows the properties of the event (the participants of the event, the location and the relative time the event took place).



Participants | Locations | Relevant Time

Participant's Name	Description	Role Type

Figure 7.17 Properties View

When an item of the properties view is selected, the user has the option to see more information about the specific item by selecting **More Info**. Then a dialog window pop-ups that shows the information about the specific item that are available.

- *View a shot of an event*

When a specific shot is selected either from the Scenario explorer or from the detailed scene of an event the next panel is displayed in the main area of the application.

The next figure shots an example of a selected shot and what the main area displays.

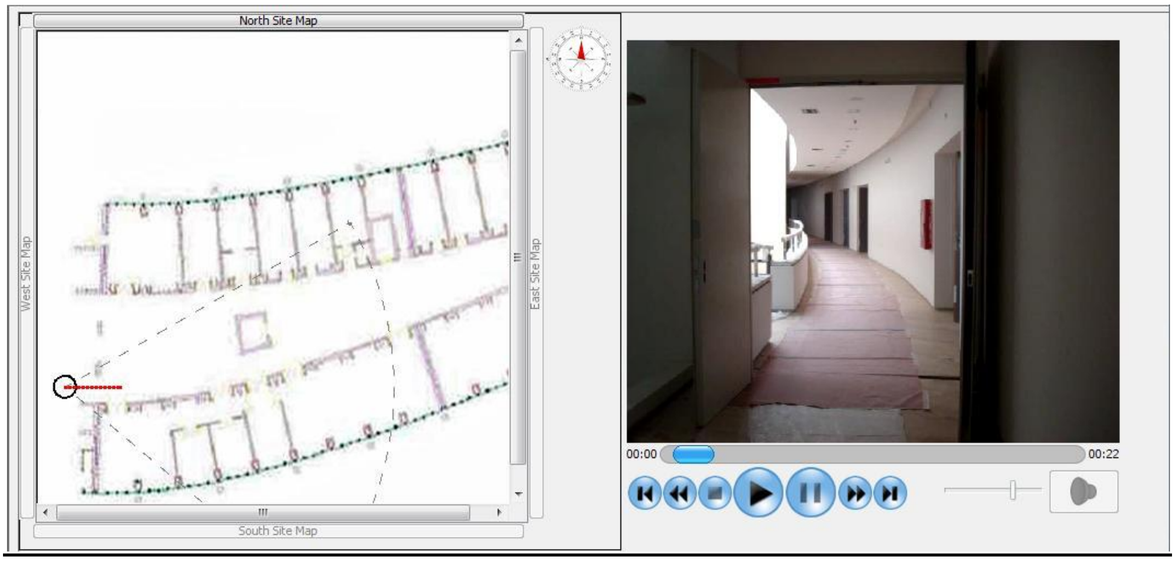


Figure 7.18 Main Area, Display of the video and virtual presentation of a selected shot

In this state the user can see a virtual representation of the space where the selected shot took place. This virtual representation shows the geometry of the location and the relative positions of the various participants of the shot. The site map shows the position, direction and movement of the camera, all the technical parameters available and all the participant's location and movement. This way, the user can have a complete picture of the surroundings of the shot. Besides the site map, when a specific shot is selected, the properties view shows more details about the specific shot concerning the participants, locations and relative time of the scene.

The actions the user can perform in the current state are the following:

The user can select to reproduce the video related to the specific shot. By doing this the user can watch the video and, in the same time, a virtual representation of the video shooting on the site map of the shot.

The user can select a specific point of the camera's movement on the site map. When this happens, the video jumps to the point where the real camera is at the same location as the virtual one. This option is available so that the user is not forced to watch the complete video if he is interested in only a part of it.

Finally in the properties view described before, the user can view the properties of the shot (the participants of the shot, the location and the relative time the shot was captured).

- *Delete a Scenario*

By using the menu item Delete Scenario from the menu bar we described before, the user can complete delete a scenario from the list or delete an event or event shots that are no longer needed.

4. SUMMARY

In this chapter we described in detail user interface implementation of the mobile multimedia event capturing and visualization application. First, we presented a description of the ten usability Heuristics principles that were written by Jacob Nielsen (31) and how they are applied to the user interface of the mobile multimedia event capturing and visualization application. Next, with the use of screenshots, we described the user interface of the application that every user, we described in Chapter 3, uses in detail. For every use case the user has, we described the steps in the mobile multimedia event capturing and visualization application he needs to take in order to complete the use case.

CHAPTER 8
FUTURE WORK

1. FUTURE WORK

In the current thesis, the Complex Event Model was created, along with an application that uses this model and helps a naïve user capture a video and the desired metadata and to present the captured video with the metadata that follow it. However, as it has been mentioned before, the application created in the contexts of the current thesis does not cover the complete model provided.

A future work that this thesis suggests is:

- To build a richer application that will provide more functionalities than the current version. Functionalities that cover all the aspects of the Complex Event Model as well as functionalities that provide more automatic ways of input information. That way the user who captures the video would not be obligated to have knowledge of the technical or other parameters that could be filled in automatically.
- An alternative way of providing guidelines to the user capturing the video, that would give more sufficient instructions to the user for every scene he tries to capture and that could give him a way of planning the capturing before it really happens in order to provide better quality videos.
- The application provided by the current thesis is a java application. A more convenient form could be to transfer all the application's functionality into a web site. That way every user who has access to the internet could use the application without the need of installation of the application in every PC he might uses. The remote access always gives to the user more flexibility.
- A far more future work could be the reform of the application in order to use it as embedded software in a video camera.

2. SUMMARY

Event processing is very important to business. We presented here a model and a system that automates to a large extend the capturing of multimedia related to events together with the context of the events and the context of capturing. This allows powerful visualizations of the spatial and temporal evolution of the events and the capturing processes. The prototype system built supports this functionality. The system is flexible to accommodate maps and diagrams of various types. We are currently in the process of integrating google maps and 3D maps with our previous automatic registration algorithms (14), (15) and the integration of more extensive contextual information captured from sensors. We are also investigating the capturing and visualization of multiple concurrent events.

Bibliography

1. **Cockburn, Alistair.** Writing Effective Use Cases. *Writing Effective Use Cases.*

2. **Equipment, Technical Standardization Committee on AV & IT Storage Systems and.** *Exchangeable image file format for digital still cameras: Exif Version 2.2.* s.l. : JEITA, 2002.

3. *Nikonians.* [Online] http://wiki.nikonians.org/index.php/Angle_of_View.

4. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Angle_of_view.

5. *Digital Photography Review.* [Online] http://www.dpreview.com/learn/?/key=focal_length.

6. *Integrated Publishing.* [Online] http://www.tpub.com/content/photography/14209/css/14209_39.htm.

7. *Integrated Publishing.* [Online] http://www.tpub.com/content/photography/14209/css/14209_37.htm.

8. *Nikonians Wiki.* [Online] http://wiki.nikonians.org/index.php/F_Numbers.

9. *Photography Dictionary.* [Online] http://www.jafaphotography.com/photterm.htm.

10. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Geographic_coordinate_system.

11. *A complete Training Manual of Videotaping and Editing Techniques.* [Online] http://www.angelfire.com/ar2/videomanual1/index.html.

12. *Cybercollege.* [Online] http://cybercollege.com/12pts.htm.

13. *Cybercollege.* [Online] http://www.cybercollege.com/bio.htm.

14. *Desktop Video Guide.* [Online] http://www.desktop-video-guide.com/shoot-video.html.

15. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Map_projection#Azimuthal_.28projections_onto_a_plane.29.

16. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Plans_(drawings).

17. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Plan_view.

18. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Floor_plan.

19. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Elevation_(view).

20. *Venizelos Foundation.* [Online] http://www.venizelos-foundation.gr/.

21. *Protege.* [Online] http://protege.stanford.edu/overview/.

22. *IBM.* [Online] http://www.ibm.com/developerworks/opensource/library/os-eclipse-platform/?S_TACT=105AGX59&S_CMP=GR&ca=dgr-eclipse-1.

23. *Sourceforge.* [Online] http://jena.sourceforge.net/.

24. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Java_Media_Framework.

25. *FMJ.* [Online] http://fmj-sf.net/.

26. *LTI-Civil.* [Online] http://lti-civil.org/.

27. *Omnivideа FOBS.* [Online] http://fobs.sourceforge.net/index.html.

28. *MySQL.* [Online] http://dev.mysql.com/doc/refman/5.0/en/connector-j.html.

29. *JavaGeomm.* [Online] http://geom-java.sourceforge.net/index.html.

30. *Sun.* [Online] http://java.sun.com/docs/books/tutorial/2d/overview/index.html.

31. *Jacob Nielsen.* [Online] http://www.useit.com/.

32. **Zett, Herbert.** *Television Production Handbook.*

33. **Hedgecoe, John.** *COMPLETE VIDEO COURSE a step-by-stop, self-instruction guide to making great videos.*

34. **Chrisa, Tsinaraki.** *A Semantic-Based Framework for Multimedia Management and Interoperability.* Chania : TUC, 2008.

35. **Mylopoulos, John.** What Do Events Mean? Trento (Italy) : s.n., 2009.

36. **Kantere, V. Kiringa, I. Mylopoulos, J.** Supporting Distributed Event-Condition-Action Rules in a Multidatabase Environment. s.l. : World scientific, 2007. Vol. vol. 16, no. 3, pages 467-506.

37. **Carlo Colombo, Alberto Del Bimbo, Pietro Pala.** Semantics in Visual Information Retrieval. *IEEE MultiMedia.* s.l. : IEEE Computer Society Press, July-Sept. 1999. Vol. vol. 6, no. 3, pp. 38-53. doi:10.1109/93.790610.

38**. C. Tsinaraki, S. Christodoulakis**: An MPEG-7 query language and a user preference model that allow semantic retrieval and filtering of multimedia conten ACM /Verlag Multimedia Systems 13, 2: 131-153 (2007)

39. **M. Bertini, A. Del Bimbo, C. Torniai, C. Grana, R. Cuchiara**: Dynamic Pictorial Ontologies for Video Digital Libraries Annotation, Workshop on Multimedia Information Retrieval on the many faces of multimedia semantics, ACM, 2007, pp47-56.

40. **D. Arijon**: Grammar of the Film Language, Silman james Press edition, 1976

41**. M. Schreibman**: Creative Producing from A to Z, Lone eagle PC 2001.

42. **B. Braverman**: Video Shooter, storytelling with DV, HD and HDV cameras, CMP books, 2005.

43**. J. Mascelli**: The Five C's of Cinematography, Silman James Press, 1965.

44**. A. Rosenthal**: Writing, Directing, and Producing Documentary Films and Videos, Southern Illinois University Press, 2007.

45. **S. Drucker, T. Galyean, D. Zeltzer**: CINEMA: A System for Procedural Camera Movements, Proceedings of the 1992 Symposium on Interactive 3D Graphics, ACM, 1992.

46. **M. Gleitcher, R. Heck. M. Wallick**: A Framework for Virtual Videography, Proceedings of International Symposium of Smart Graphics, 2002, ACM.

47. **S. katz**: Film Directing Shot by Shot, Sheridan Books, 1991.

48. **S. Christodoulakis, M. Foukarakis, L. Ragia, H. Uchiyama, T. Imai**: Picture Context Capturing for Mobile Databases, IEEE Multimedia, June 2006.

49**. S. Christodoulakis, M. Foukarakis, L. Ragia, H. i Uchiyama, T. Imai**: Semantic maps and mobile context capturing for picture content visualization and management of picture databases, Proceedings ACM MUM 2008.

50. **D. Luckham**: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems, Addison Wesley, 2002.

51. **A. Michlmayr, F. Rosenberg, P. Leitner, S. Dustdar**: Advanced Event processing and Notifications in Service Runtime Environments, Proceedings ACM DEBS 2008.