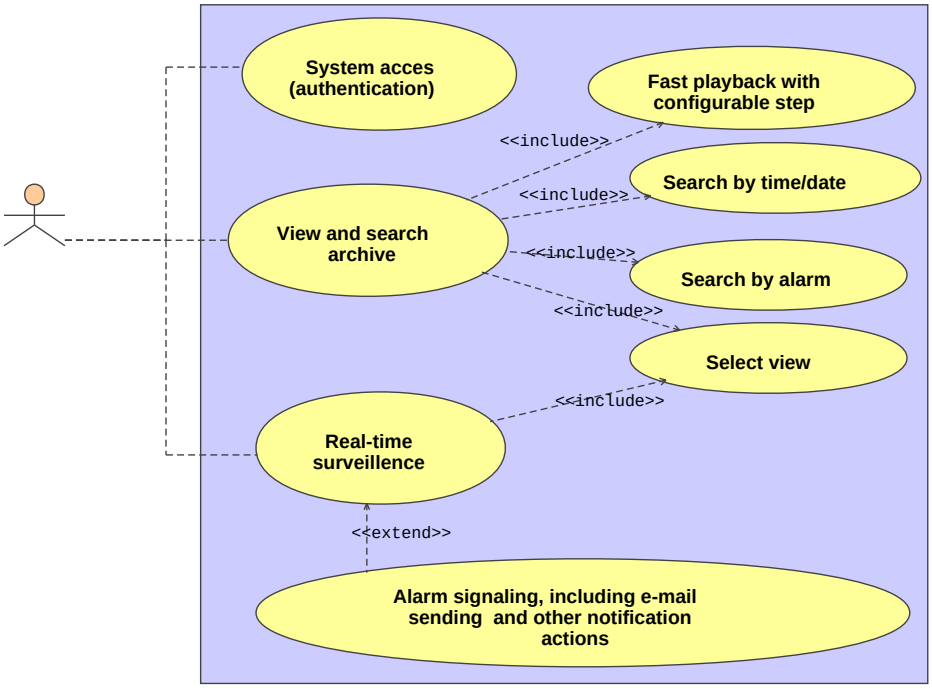
1 Use cases

1.1 Surveillance use case



1.1.1 System access procedure (user authentication)

System access (user authentication) procedure is a starting point of user interaction with system. To get access to the system, user must specify his login name and password. In case of successfull authentication, access to sytem functions is granted, applying restrictions with respect to user access rights. Optional feature may be implemented - logging out user after configurable timeout.

Typical user "log-in" scenario:

- User is prompted for his login name ;
- User enters login;
- User is prompted for his password;
- User enters password;
- Password is checked and in case of successfull authentication, access is granted;

1.1.2 Real-time surveillance

Surveillance can be performed in real-time, when live video streams are transmitted from cameras to the surveillance operator's workstations through a LAN. Cameras are grouped into "views" (views management is performed by the system administrator, and is described in Configuration use case). During real-time surveillance, operator can watch several camera viewpoints simultaneously by selecting pre-configured views. Options available to user during real-time surveillance:

- View(group of camera viewpoints) selection;
- Start watching;
- Stop watching;
- Adjust quality of the recieved videostreams;

In case of alarm, surveillance operators are notified of an incoming alarm signal with specification of occured alarm source and condition. Operator has an oppotunity of viewing list of currently active alarms. An active alarm can be cancelled by an operator. When active alarms are cancelled, the recordings scheduled on them are stopped.

1.1.3 View and search archive

Video and alarm records are stored in the archive. To watch the video recordings from the archive, user should specify the view (group of camera viewpoints) to watch and the recording time period he is interested in. The first available record from the specified time period will be retrieved. During the playback user is provided with VCR-like controls over the playback process: play, pause, ff, rw, step forward, step back, reverse playback, adjustable playback speed. Each video record includes information about the time, when it was shot and reference to source camera viewpoint. Alarm records, stored in the archive can be retirieved by the specifying time period and/or by the alarm source. Alarm records include reference to the alarm source (physical sensor, camera motion detector, etc.), date, time and condition of the occured alarm.

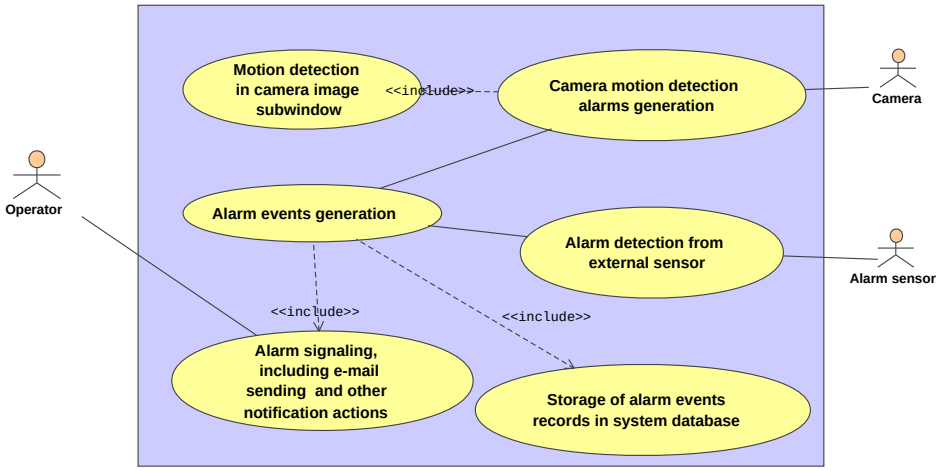
Commands available to user during viewing archive video records:

-Select view (group of camera viewpoints);
-Retrieve records from specified time period;
-Playback controls (play, pause, ff, rw, step forward, step back, reverse playback, adjustable playback speed);

Commands available while viewing archive alarm records:

-Search records by alarm source;
-Retrieve records from specified time period;

1.2 Alarm detection and signalling use case



1.2.1 Generation and handling of Alarm Events

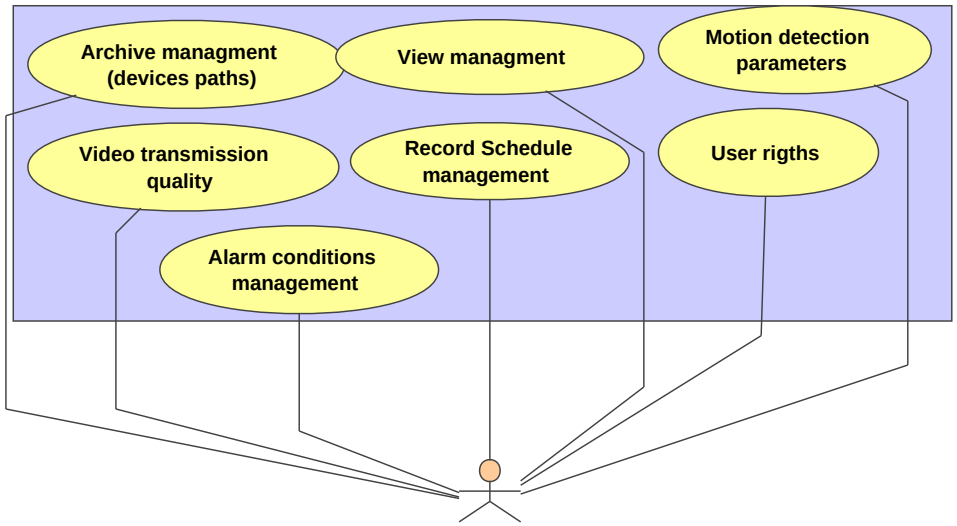
A set of **alarm conditions** is stored in the system (see Configuration use case). In case of an alarm condition is fulfilled, an **alarm event** is generated. Upon regisration of an alarm event an **alarm record** is created and stored to **archive** (alarm date/time, alarm source), and **camera recordings** scheduled on this alarm event are started. Also the alarm be **reported to surveillance operators** working in real-time surveillance mode and sent by email to the specified addresses.

1.2.2 Sources of alarm events

Alarm signals can be generated by **external aClarm sources** (like door open sensor), connected to the system, or can be generated by **motion detection** units, included into Grabber Server machines. The motion detection can be

performed over the whole camera frame, or in defined subwindow (motion detection area). After the motion detection areas are specified, they can serve as alarm sources, initiating camera recordings and etc.

1.3 Configuration use case



1.3.1 Views management

View is a group of camera viewpoints. Views are available to the surveillance officers during real-time surveillance and archive records watching. "View" objects provide ability to perform "multiple screen" camera streams watching. Configuring views is performed by the maintenance team (system administrators). Views management includes the following operations:

- View object creation;
- View naming and renaming;
- View storing and deletion;
- Selection of camera viewpoint objects to include into view;
- Positioning of the camera images inside view.

1.3.2 Users (surveillance officers) rights management

Several levels of priveleges exist in the system, restricting: real-time surveillance access, archive access and administration procedures access. Each user's rights are stored in his personal record, which also holds his login name, password and personal information (name, etc.). System administrator is can add, delete users and control each user's priveleges. Functions of user rights management include:

-Creation of new users;
-Editing existing user attributes;
-Deleting existing users;
-Changing user's password;
-Changing user's priveleges;

1.3.3 Alarms conditions management

A set of alarm conditions can be specified. Alarm condidion defines reference to an alarm source, time period when the alarm condition is active and additional information specific for the defined alarm source. An alarm source can be some kind of physical sensor, camera motion detector, or a specified event like lost camera signal or lost power connection Camera motion detection can be performed over the whole camera image or only in specified subregion. When a signal from a sensor specified in one of the alarm conditions is detected, an alarm is event is registered and an alarm record (including date/time and alarm source) is written to the archive. Upon registration, alarm may be signalled to the surveillace operators and a notification can be sent to the specified email addersses. Alarm conditions are managed by the users granted with system administration rights. Alarms management procedure includes:

-Creation and deletion alarm condition;
-Naming and renaming of alarm condition;
-Selecting alarm source: external alarm sensor, signal lost alarm, motion in specified region of a camera.
-Specifying the actions on alarm registration (surveillance operators notification, email sending)

1.3.4 Recording schedule management

Video streams coming from the cameras can be recorded and stored in the archive. Recording is controlled through recording schedule table management. Recording of the specified camera stream can be started by the specified

date/time or by detection of definite alarm event(s). When setting record on schedule, user specifies the group of cameras that should be recorded (through "view" object), date/time or condition of record start (a definite alarm). Recording schedule table management inlcudes:

-Adding and deletion record on schedule entries;
-Modifying record on schedule parameters (date/time, view, recording condition);

1.3.5 Archive parameters management

Video records archive is stored on a separate network fileserver machine, connected to the video sources through a LAN. The archive capacity can be increased by adding new storage devices, without stopping the system and with no data loss. There is also an option of removing an exisiting storage device by prohibiting recording of new clips to this device and moving the existing clips to free space on other drives. Archive parameters include paths for storage devices, options for storing to network server and parameters, regulating the maximum time period of video clips storing (i.e. 10 days). Administrator is also provided with rights for deletion of some records from the archive.

Options for archive control include:

- Storing records to remote network server;
- Setting maximum time period for videoclips storing;
- Setting paths for new storage devices;
- Prohibiting recording to specified storage device;
- Moving existing records from one device to another;
- Deleting records from the archive;

1.3.6 Camera viewpoints (PTZ) management

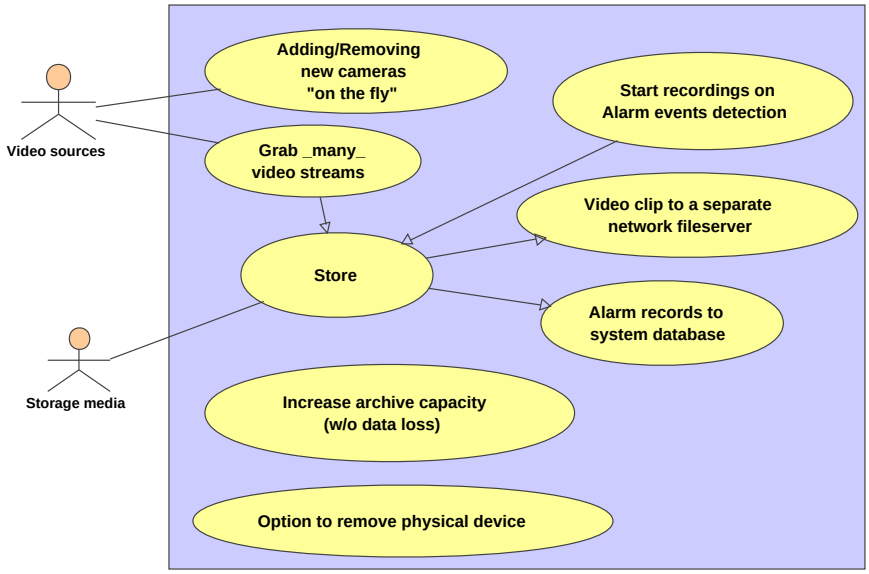
Physical cameras, connected to the system are adderssed to by the named objects called camera viewpoints. Camera viewpoint includes a stored physical camera reference and camera Pan/Tilt/Zoom parameters. Camera viewpoints are grouped to construct views. Camera viewpoints management includes:

-Creating camera viewpoint object;
-Name and renaming camera viewpoint object;
-Camera viewpoint deletion and storing;
-Selecting physical camera associated with this viewpoint;
-Setting camera PTZ parameters.

1.3.7 Video transmission quality

1.3.8 Motion detection parameters

1.4 Archiving



1.4.1 Grabbing videostreams

Possibility to connect up to 2000 physical cameras is to be provided. Video stream, coming from a video camera is grabbed and can be written to system storage devices (in case recording is turned on) and/or transmitted to the real-time surveillance stations. Service information (date and time, camera number) is written to system database.

1.4.2 Adding new cameras “on the fly”

New cameras must be easily added and removed to and from the system.

1.4.3 Store

Video records can be stored in the archive. The recording schedule table shows which camera streams are to be stored. The streams coming from the cameras are written to fixed storage media of system along with service information (date/time and camera number). In case of exhausting of the archive storage capacity, part of the oldest records is deleted and system administrator is notified on the shortage of archive free space.

Incoming alarm records (alarm source, date/time) are written to system database. Option of viewing and searching through stored videoclips and alarms is available to surveillance officers (see "Surveillance use case"). Archiving options are controlled by the maintenance team (see "Maintenance use case").

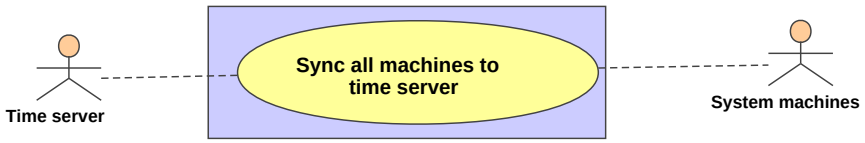
1.4.4 Increasing of archive capacity

New fixed storage device can be added to system to increase archive capacity, without stopping the system and losing any stored data. After physically attaching new storage devices system administrator can include them into st\sytem archive. The maximum possible recording time is recalculated according to new archive capacity.

1.4.5 Option of removing of existing storage devices

To remove an existing storage device system administrator should first prohibit putting new video data on it, then the existing clips can be deleted or moved to other devices by a special utility.

1.5 Services

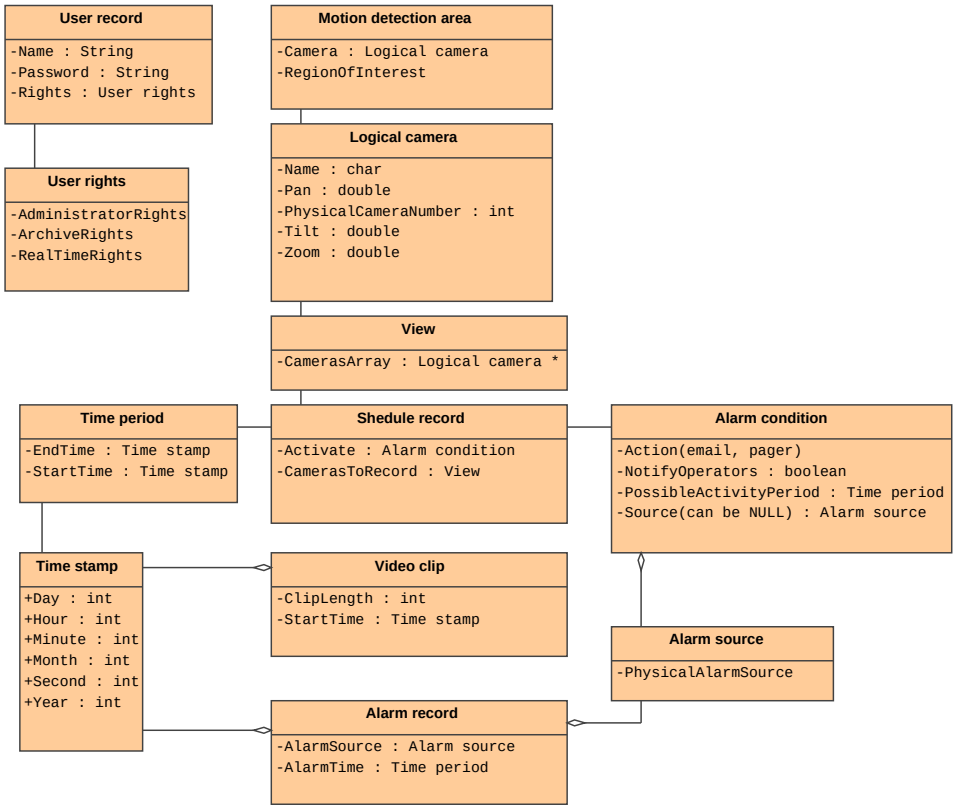


1.5.1 Synch all system components to TimeServer

All machines included into system (fileserver, grabbers, RDBMS, HTTP server), must be syncronized to time server to provide adequate functioning.

2 Data objects description

These data objects are to be used in the system:



Data objects diagram

2.1 TimeStamp

Used for purpose of storing date and time for variuos objects and events in the system. Includes:

-Year
-Month
-Day
-Hour
-Minute
-Second

2.2 TimePeriod

Includes two TimeStamp objects:

-StartTime
-EndTime

and is used for alarm condition and schedule record activity period specification.

2.3 LogicalCamera

Specifies the reference to physical camera object and stores its PTZ paramerers. Used as a named reference to physical cameras.

-Name
-PhysicalCameraNumber
-Camera host name
-Pan
-Tilt
-Zoom

2.4 AlarmSource

Specifies the type of alarm source (motion detector, physical sensor, signal lost alarm, etc.) and reference to a concrete physical sensor or motion detector.

-SourceType {motion_hardware, motion_software, sensor, camera_signal_lost, mux_signal_lost, error)
-SourceNumber

2.5 AlarmCondition

It includes the alarm activity period and alarm source reference. In case the referenced source produces an alarm signal in the specified time period, an alarm is detected. The source may be NULL, which means a special **AlarmCondition** object, used only as a part of **ScheduleRecord** object, specifying recording regardless of alarm events. When alarm is detected, the surveillance operators can be notified with a message (an option is controlled through NotifyOperators field). Also, an email can be sent to the specified addresses (SendEmail field). The active alarm may be cancelled by the signal coming from the alarm source, by a surveillance operator command or by ending of alarm activity period.

-PossibleActivityPeriod
-AlarmSource
-NotifyOperators
-SendEmail

2.6 AlarmRecord

Alarm record is used for storing the information on the occured alarm events. It includes **TimePeriod** of the occured event (when the alarm signal arrived and whan alarm was cancelled) and reference to **AlarmSource** that produced the alarm signal.

-Alarm source
-Alarm time

2.7 MotionDetectionArea

This object is used for specification of motion alarm source. It includes reference to **LogicalCamera** object and rectangle (subregion of the camera image) where the motion detection is to be performed.

-LogicalCamera
-RegionOfInterest

2.8 UserRights

Stores user access rights to real-time viewing, archive and administration functions.

-AdministrationRights
-ViewingRights

-ArchiveRights

2.9 UserRecord

Used to store user account information (login, password) along with user rights and personal information.

-Login name
-Password
-User rights
-Personal information

2.10 ScheduleRecord

Schedule record object is used to specify the camera streams to record. In includes group of cameras (**View** object) to record and a condition of record start (**AlarmCondition** object). Conditions of record start include time period and source of alarm. In case the alarm source is NULL, the recording depends only on the specified time period, otherwise it takes place in the specified period when alarm is active.

-ActivationConditions
-CamerasToRecord

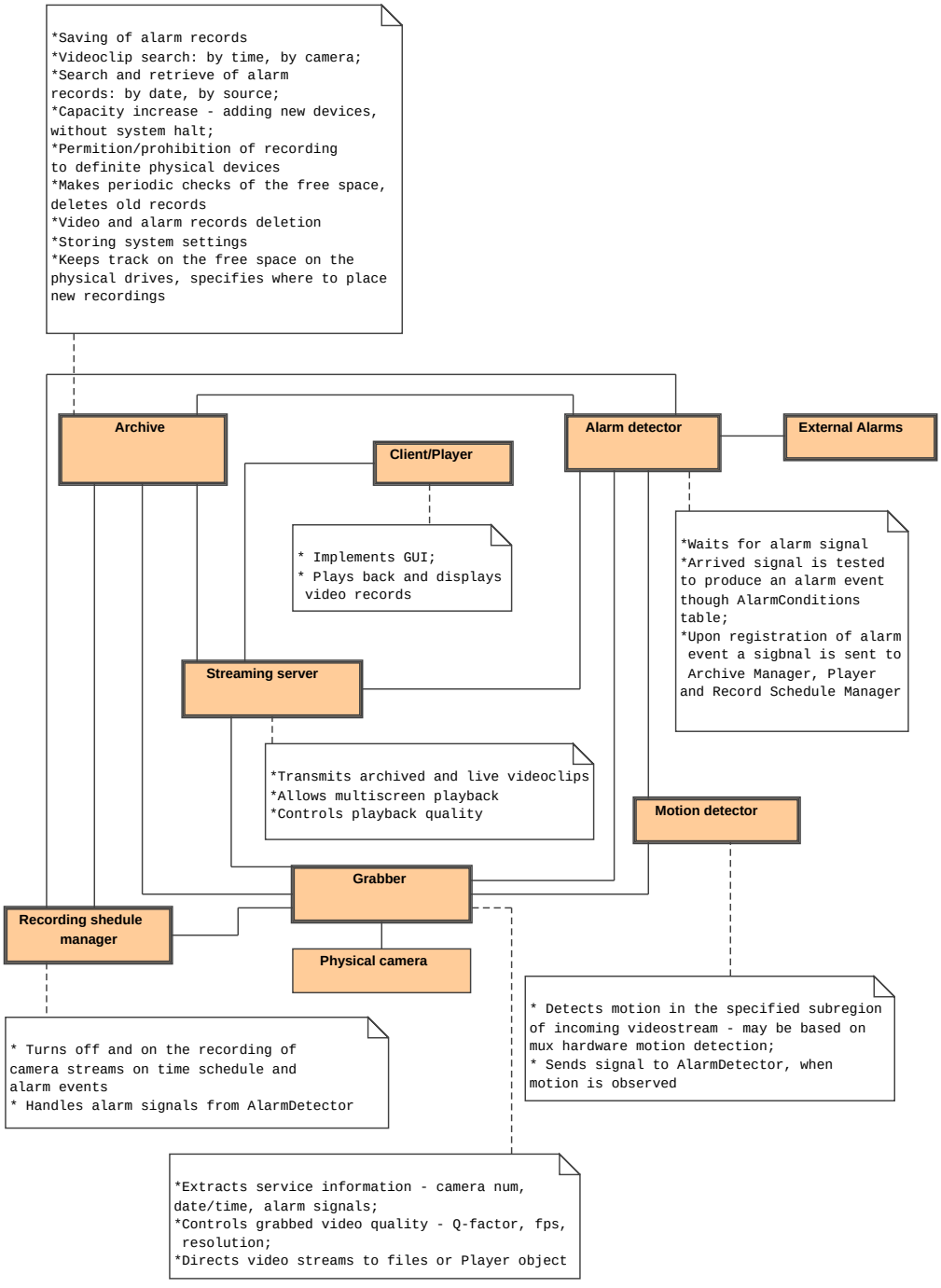
2.11 View

View is an object used for organizing camera groups and setting individual camera positions inside the group. It is used for multiscreen playback implementation and in cases when multiple cameras are referenced to.

-LogicalCamerasList
-LogicalCamerasPostions

3 Active Objects Description

Active objects diagram


*Saving of alarm records
*Videoclip search: by time, by camera;
*Search and retrieve of alarm
records: by date, by source;
*Capacity increase - adding new devices,
without system halt;
*Permition/prohibition of recording
to definite physical devices
*Makes periodic checks of the free space,
deletes old records
*Video and alarm records deletion
*Storing system settings
*Keeps track on the free space on the
physical drives, specifies where to place
new recordings
Archive
Client/Player
Alarm detector
External Alarms
* Implements GUI;
* Plays back and displays
video records
*Waits for alarm signal
*Arrived signal is tested
to produce an alarm event
though AlarmConditions
table;
*Upon registration of alarm
event a sigbnal is sent to
Archive Manager, Player
and Record Schedule Manager
Streaming server
*Transmits archived and live videoclips
*Allows multiscreen playback
*Controls playback quality
Motion detector
Grabber
Recording shedule
manager
Physical camera
* Turns off and on the recording of
camera streams on time schedule and
alarm events
* Handles alarm signals from AlarmDetector
* Detects motion in the specified subregion
of incoming videostream - may be based on
mux hardware motion detection;
* Sends signal to AlarmDetector, when
motion is observed
*Extracts service information - camera num,
date/time, alarm signals;
*Controls grabbed video quality - Q-factor, fps,
resolution;
*Directs video streams to files or Player object

3.1 Archive

Archive object functions include everything that is connected with data storage in the system. Namely:

-Storing and manipulating of system parameters table (alarm conditions, recording schedule table, views table, user records, logical cameras, motion detection areas);

-Storing and retrieving records on alarm events;

-Maintaining the filesystem structure for video information;

-Retreiving and erasing video clip objects;

-Attaching new storage devices to increase archive capacity without system halt;

-Monitoring free space for video storing, deletion of the oldest records in case of free space shortage;

-Permition/prohibition of putting new records on definite storage devices.

-Keeping track of the free space on each physical device and telling **Grabber** where to put new recorded clips.

3.2 Recording schedule manager

Controls the recording of video streams, coming from the **Grabber** objects. The recording can start on a registration of definite alarm event (set by the **AlarmCondition**), and last until the alarm is cancelled (by a timeout or by operator actions). Or, the recording can be performed regardless the alarm events, in the specified time period. Schedule manager performs periodic checks (i.e. each second) whether a new recording must start or an existing recording must be stopped. In case of alarm event is detected by the system, schedule manager is notified to check whether any recording is scheduled on this alarm. Schedule manager functioning is depicted on a separate sequence diagram.

3.3 Grabber

Grabber digitizes and extracts video streams from cameras along with service information (time/date, etc.) It also controls the quality of the generated digital video streams (quality term means fps, Q-factor and resolution). The extracted video streams can be saved directly to specified file and/or transmitted to objects responsible for real-time surveillance functions implementation. Videostream is also passed to **Motion Detector** object. The fact of camera signal loss is also detected by **Grabber**.

3.4 Motion Detector

Motion detector object applies motion detection procedures to the camera streams, coming from the **Grabber** object. The detection is based on procedure implemented directly in the **Motion Detector** object. The motion detection can be done only in a specified subregion (region of interest) of the camera image. The cameras, which must be subjected to motion detection, and the regions of interest are specified by the table of **MotionDetectionArea** objects. In case of motion in the specified camera, **AlarmDetector** is notified.

3.5 Alarm Detector

Alarm detector is responsible for handling alarm signals coming from **Grabber, Motion Detector** and external **AlarmSources** objects. Each signal, coming from an alarm source, generates an alarm event in case there exists an alarm condition referenced to the incoming **AlarmSource** and that can be activated at current system time in (stated in alarm conditions table). When alarm event is generated, **Archive**, **Schedule Record Manager** and **Player** objects are notified on it. Alarm may be cancelled by a timeout or by surveillance operator's actions. Alarm handling scenario is described in detail in a separate sequence diagram.

3.6 Streaming server

Streaming server acquires video streams from Grabber and Archive objects, generates single multiscreen image for each client, transcodes the resulting images to suitable format (H.263, MPEG-4) and transmits it to Player object. It also maintains fps, resolution and quality of the transmitted video.

3.7 Player

Player object organizes survellance operator GUI with the system. It provides functions for watching real-time video from the cameras and recorded streams from the archive. The functions of the player object:

-Playback of the streams taken from the **Grabber** and **Archive** objects;
-Allowing 'multiscreen' playback;
-Modifying quality of the played streams (resolution, Q-factor);
-Implementation of the VCR functions (variable speed playback, reverse playback, pause);

-Notification of the surveillance operators on the alarm events occuring in the system;

-Showing the list of active alarms;

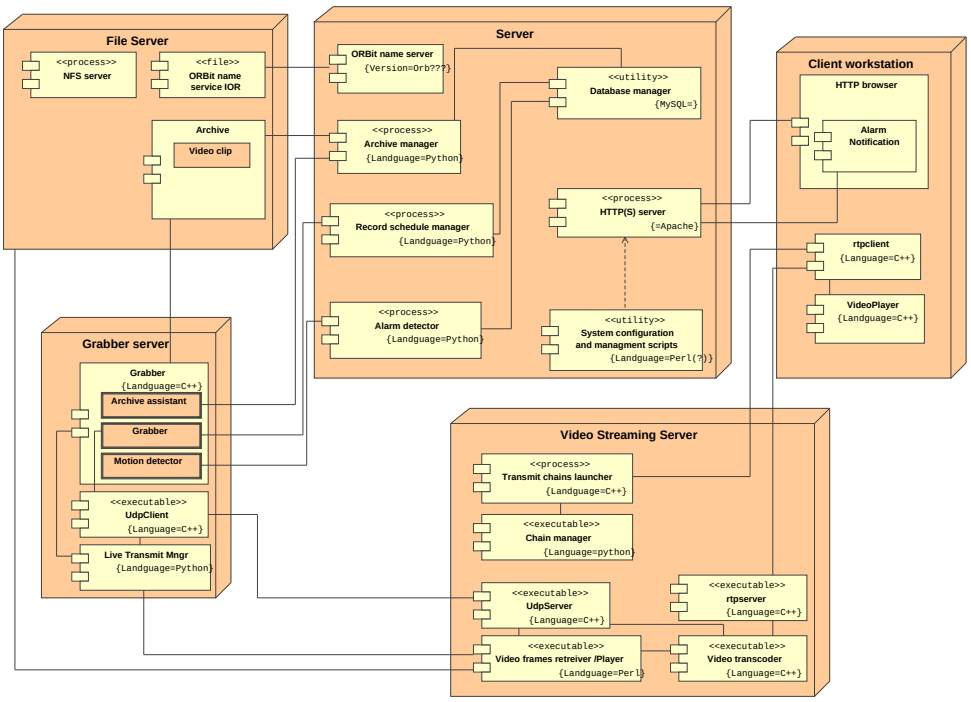
-Cancelling of currently active alarms;

-Implementation system configuration functions (described in **Configuration** use case);

3.8 Time Server

Time server object is used by all other objects of the system, when current time/date is needed.

4 Implementation



4.1 Inter-Components communication implementation

The CORBA ORBit implementation is used. The ORBit name server is runned on the main server, it's IOR is written to a file, which is visible through NFS to all the server (excluding client workstation). The components, when launched register themselves at the name server.

4.2 Archive object implementation notes.

Archive object is decomposed into three entities: one responsible for maintaining the video data storage (**Archive manager**), second for storing all other kinds of information (**DataBase Manager**) and third, keeping track on the free space on different physical devices (**Archive Assistant**). The first is implemented as a filesystem with a specific directory structure, second is a **RDBMS** (MySql), third is integrated into **Grabber** implementation for performance reasons and to gurantee no frames loss.

4.2.1 Videoclips storage

Video information is stored in videoclips of fixed length (in seconds) on a number physical devices, available as regular unix directories. The storage directory tree on each devioce has the following structure:

<DEVICE_ROOT>/Year/Month/Day/Hour/
example:
/opt/video/2001/ 11/02/23/

Each directory is filled with N seconds clips, with the following filenames:

<camera_num>'_'<grab_host>'_'<start_time>'.mjpeg'
example:
cam01_grabber0.sdvdr_11m20s.mjpeg

The filesystem used for videoclips storage is managed by a specific object '**Archive manager**', which performs periodic checks of the devices free space, provides integrity control over the videoclips storage and is capable of adding/removing physical storage devices. The filesystem can occupy several physical devices of different nature (hard drives, NFS servers). The management of those devices is encapsulted into **Archive Manager**. The video clips from the non-removable storage devices can be copied to removable media by means of this component.

The archive manager maintains a list of active **Grabber** objects on different hosts, which is constructed when the component is launched. Each new launched **Grabber** object registers itself in **ArchiveManager.** The **ArchiveManager** functions include:

1. Keeping track of the free space on the physical devices, deleting old clips when the system is low on free space;
2. Handling to administrator's commands on the devices adding, removing, prohibiting and permitting recording on them;
3. Handling user's requests on deletion of clips;

The videoclip search and variable speed playback procedure is implemented through a special table in system database, that stores offset of each frame of every clip inside the clip file. The table structure is as follows:

Frame reference table

Camera Num	Camera Host	Current time	Frame offset	Time offset from clip start	Device Path

These fields allow to reconstruct the full clip filename and allow effectively retreive the desired frame from the file. Usage of this table allows to construct clip search procedure as database query.

4.2.2 System settings storage

The non-video data is stored in the system database in specific tables.

Views table

Logical camera ID	View ID	Camera position inside view	Text notes

Alarm sources table

ID	Sensor number	Sensor type	Sensor host name	notes

Alarm conditions table

ID	Alarm source ID	Activity time period	notes	Email for notification

Record schedule table

ID	Alarm condition ID	View ID

Alarm record table

ID	Alarm source ID	Date/Time of alarm event

Motion detection areas table

ID	Rect Top	Rect Bottom	Rect Left	Rect Right

Active alarms table

ID	Alarm condition ID

Active camera recordings table

ID	Logical Camera ID	Num of alarms referencing this camera

Logical camera table

ID	Camera Host name	Camera physical number	Pan	Zoom	Tilt	Text notes

Several tables store the information related to user accounts and authentication.

Account table

id	Username	Password	Permissions	Active flag

IP table

id	IP of logged users

Session table

id	Session ID	Created Date/Time	Account ID	Last access	IP id	Active flag

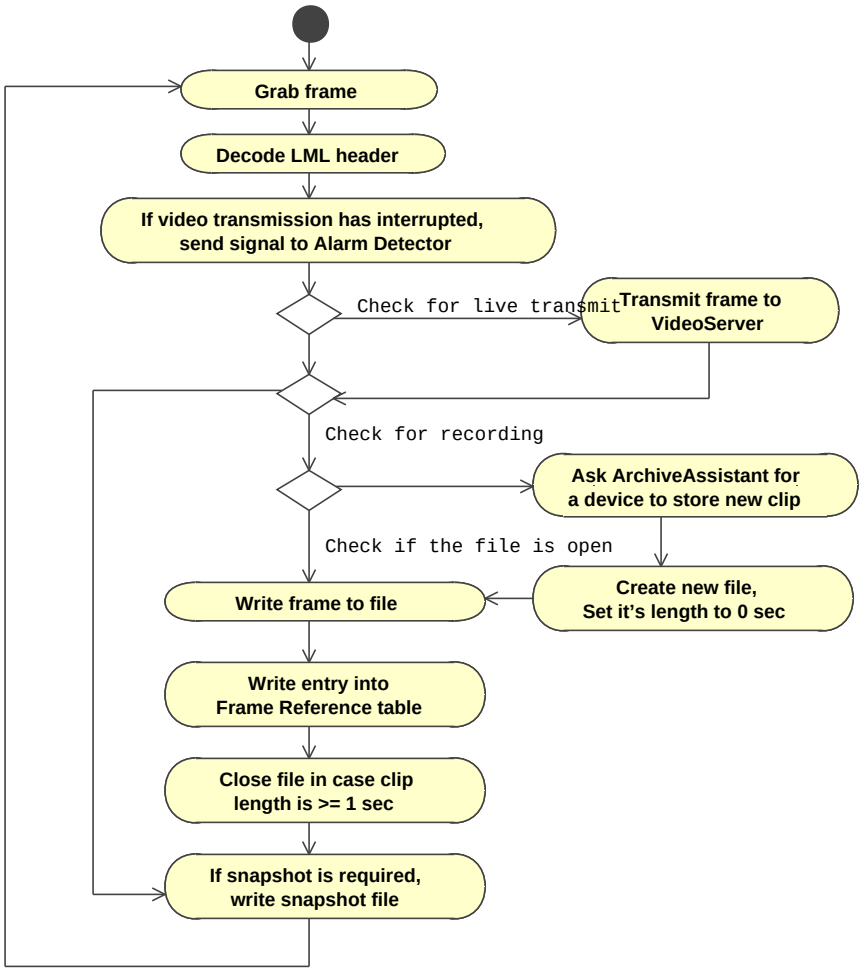
4.3 Grabber, Motion Detector and ArchiveAssistant implementation notes

Are incorporated into one entity for performance reasons. The video is grabbed from the jpeg compressed video steam device, provided by LML33 device driver (i.e. **/dev/mvideo/stream**). The grabbed streams can be subjected to motion detection procedure right after the acquiring. After that, the streams that shoul be recorded are sent to files, the watched streams are sent to **Video Server** objects, the others are not going anywhere. The commands for recording are given by the **RSM** object. When the command to start recording of a definite camera stream is issued, **Grabber** starts creating video clips of a fixed length, until the command to stop recording arrives. The physical devices for video clips storing are chosen by the **ArchiveAssistant** object, integrated into **Grabber**, which keeps track of the free space left on the physical devices. The streams to be viewed in real-time are specified by the player object. Streams to be tested for motion are specified by the table in the system database.

Grabber component funcons include:

1. Recording of clips from cameras and sending them to archive;
2. Transmitting live video streams to client;
3. Maintaining video grabbing parameters (grab devices, FPS, resolution, compression quality);
4. Making static camera snapshot;

Each camera is processed by a separate thread, which is running if camera is being recorded or viewed. The grabbed frames are sent directly to physical/network disks or to components, which implement live video transmit. The video is written in n second clips, to ease the archive maintenance. Recording process is depicted on the following diagram:



Video grabbing parameters are controlled by functions, exported through CORBA interface. The specify the device name used for grabbing (i.e. /dev/mvideo/stream), FPS, resolution and LML jpeg compression quality. Snapshot is taken directly from the device if the camera is inactive, or during the grabbing loop (see diagram) when camera is recorded or viewed.

ArchiveAssistant component functions inlcude:

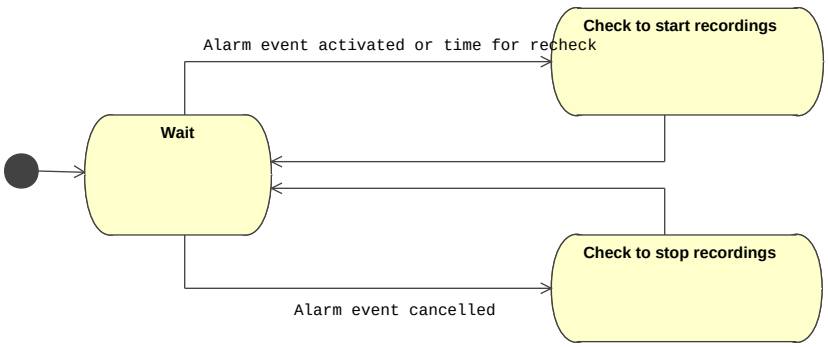
1. Choosing devices for clips recording (based on free space), keeping track of free space;
2. Periodically requesting free space of the devices from **ArchiveManager;**

4.4 Record Schedule Manager implementation notes

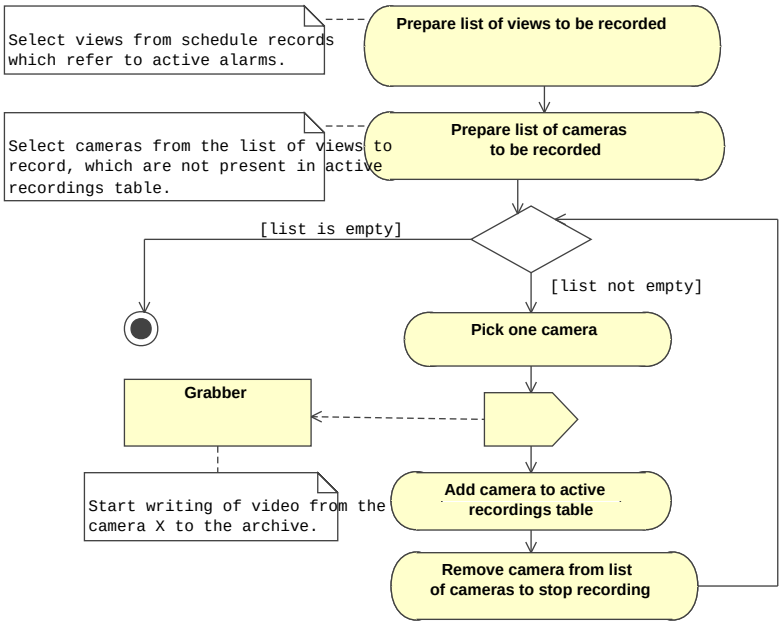
The **RSM** object is implemeted as a separate process. The check of recordings that must be started is made by selecting schedule records which refer to active alarms. The results of this query is a list of **View** objects that must be recorded. The list of cameras recordings of which must be started by selecting cameras that are present in views to be recorded and are not included in **Active camera recordings** table**.** Cameras that must stop recording are also determined by those two tables. Then the commands to start or stop recording of definite cameras are sent to **Grabber** objects. The recordings dependant only on time (regardless of alarm events) are implemented also through **Active alarms** and **Alarm conditions** tables. There exists a number of **Alarm conditions** with NULL reference to alarm sources, specifying only activity time period. These conditions are hidden from the system users, they do not produce **Alarm records** or operators notification. They are activated by the start of the specified period, and cancelled only when it is over.

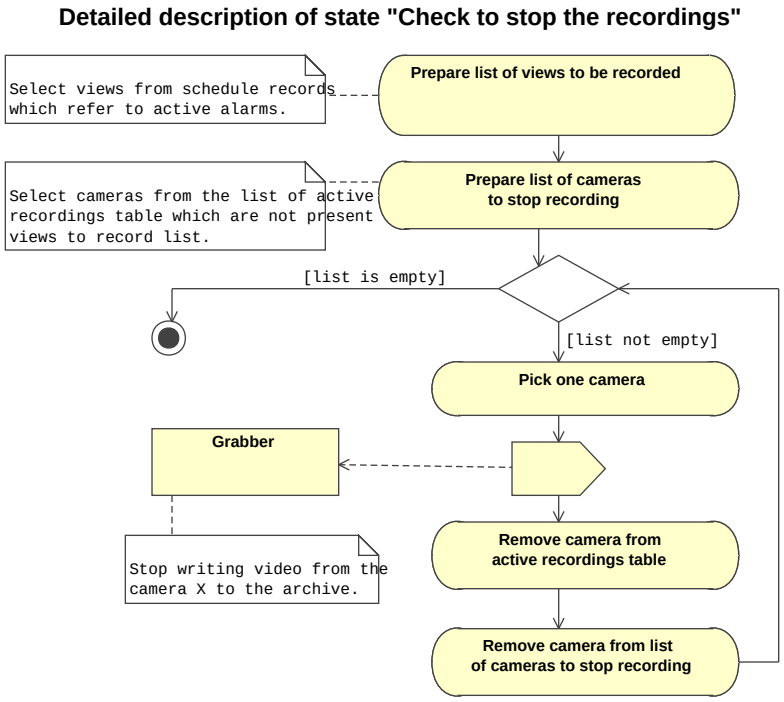
Checking of schedule table is performed in two cases:
1. On timer (each N seconds)
2. On alarm is detected or cancelled (maybe some recording must started or stopped immideately)

General state diagram of schedule record manager



Detailed description of state "Check to start the recordings"





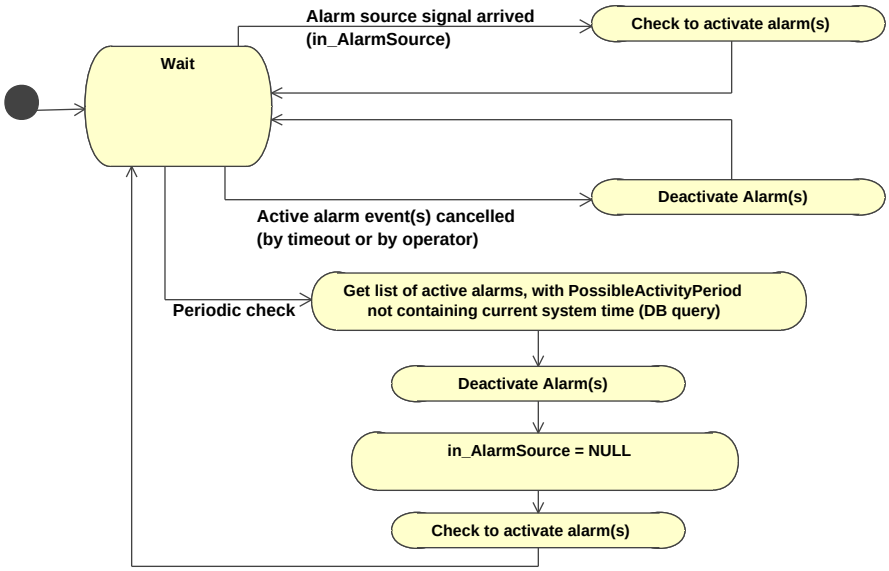
4.5 Alarm Detector implementation notes

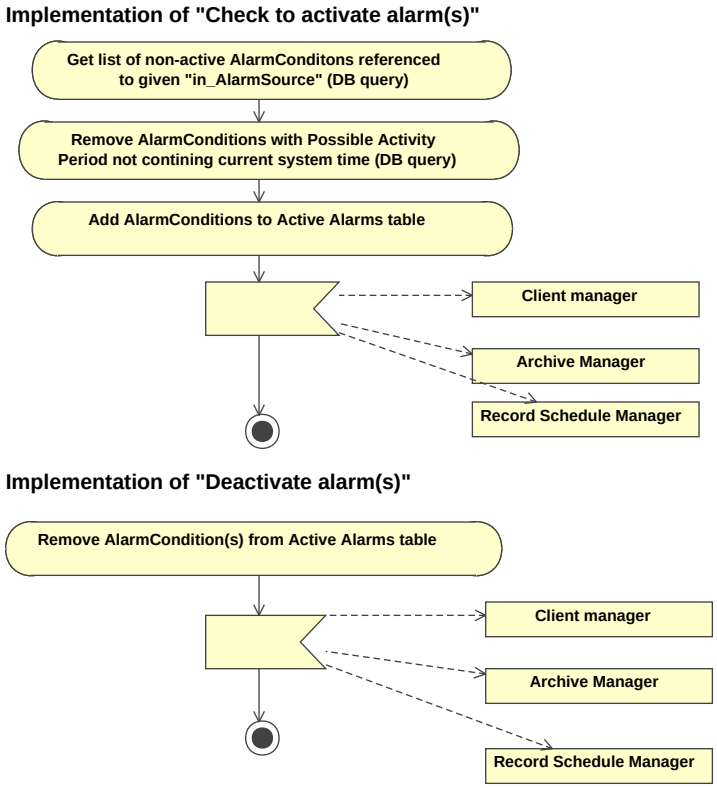
Main role of **Alarm Detector** is handling signals, coming from alarm sensors, **Grabber** and **Motion Detector**. Another fucntion is to make periodic checks of the **Alarm Conditions** table to cancel the timed out active alarms and to activate alarms which have NULL reference to alarm source. These alarm conditions do not produce alarm records in the archive, they are not passed to surveillance operators. They are used solely for time-scheduled recordings support.

The functions of **AlarmDetector:**

1. Wait for alarm signals;
2. Make periodic checks to cancel active alarms and to activate alarms with NULL alarm sources;

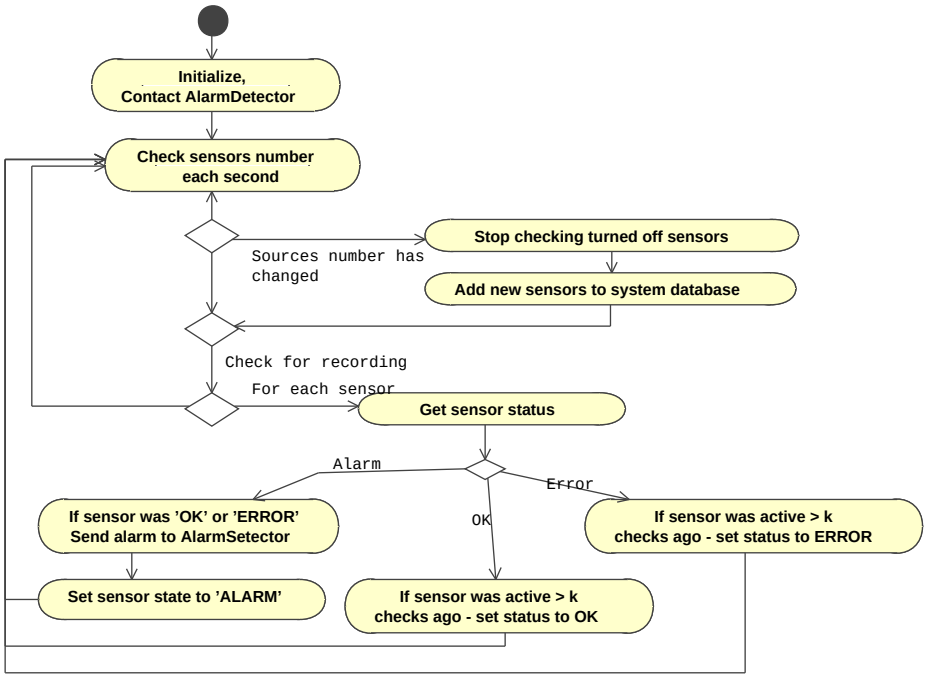
General state diagram of AlarmDetector





4.6 External alarm sources

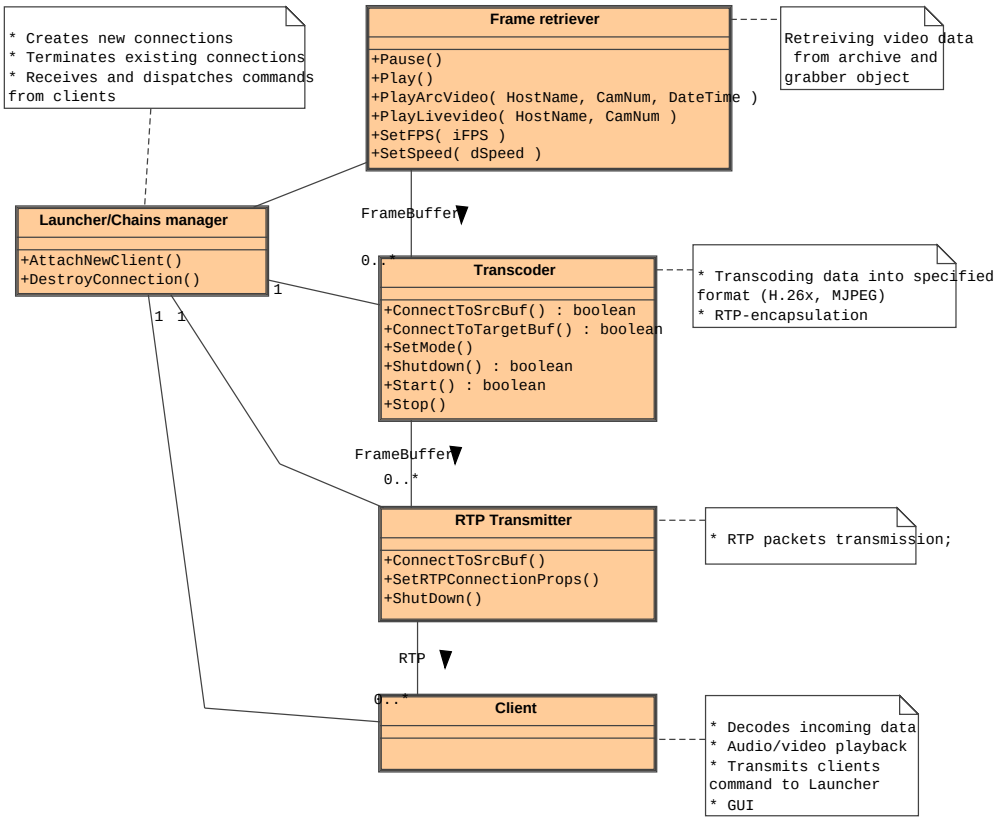
External alarm sensors are attached to the system through parport devices, that can be installed on any server (main, grabber, video). A separate process (**ExtAlarm**) is run on the server where external alarm sources are installed. It contacts **AlarmDetector** component upon start and polls the external alarm sources each n seconds (configurable parameter). In case the alarm signal comes, the appropriate command is sent to **AlarmDetector**.



The **ExtAlarm** component is implemented as an .so library (working with parport driver) and a Perl script, which diagram is shown above.

4.7 Video Server implementation notes

Video server is implemented as several entities. Main control entity is called **Launcher,** its function is to listen the RTP control connection for new clients requests and to process client commands (play, pause, etc.) and pass it to other Video Server components. When a request arrives, **Launcher** creates a **Transmit Chain** (which consists of **Chain Manager, FrameRetreiver**, **Transcoder** and **RTPserver**). The transmit chain components are connected through FrameBuffers. **FrameRetriever** get's a frame from an archive record, or directly from **Grabber**(in case of live transmit) and passes it to **Transcoder**, which implements transcoding of LML MJPEG to format more appropriate for RTP transmission. The **Transcoder** also implements "multiscreen playback" capability and RTP frames incapsulation. The incapsulatd frames are passed to **RTP Transmitter** and sent to the client workstation over the RTP channel. **Client Manager** is a part of **Launcher**, separated for more convinient implementation - its function is to check that all transmit chains components are working properly, in case of component crash or lock up, a singal is sent to Client through **Launcher** and transmit chain is halted.



4.7.1 Archive Play

When achive play is startedm **FrameRetriever** reads the frmaes of the cameras to playback from the storage devices through NFS, consulting the **FrameReference** database table. The FPS and speed of playback is regulated by **FrameRetreiver** component.

4.7.2 Live video

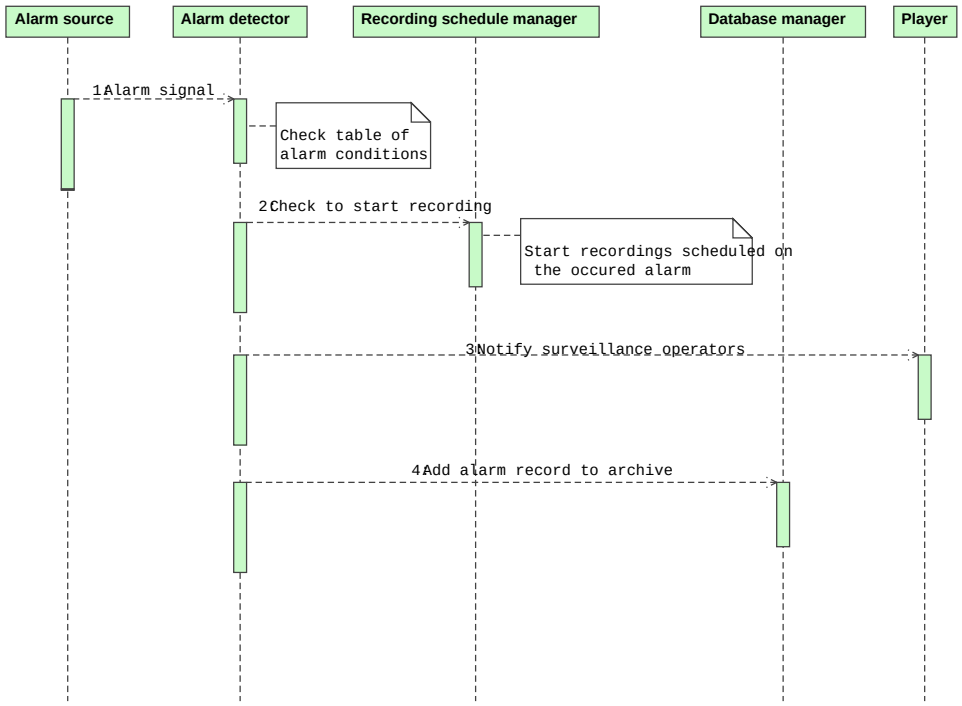
When live video is played, two cases are possible. In both cases the FPS and speed of playback is regulated by **Grabber** components, which aquire commands from the **FrameRetreiver**.
1. Video server components reside on the same station as the **Grabber** for the camera to view;
In this case, the **Grabber** component is directly connected to **Transcoder** through FrameBuffer.

2. Video server components and **Grabber** for the camera to view reside on

the different stations;
In this case, the frames from **Grabber** are transmitted through UDP sockets by **UdpClient** (on **Grabber** side) and **UdpServer** (on**Transcoder** side).

5 Scenarios

5.1 Alarm handling scenario

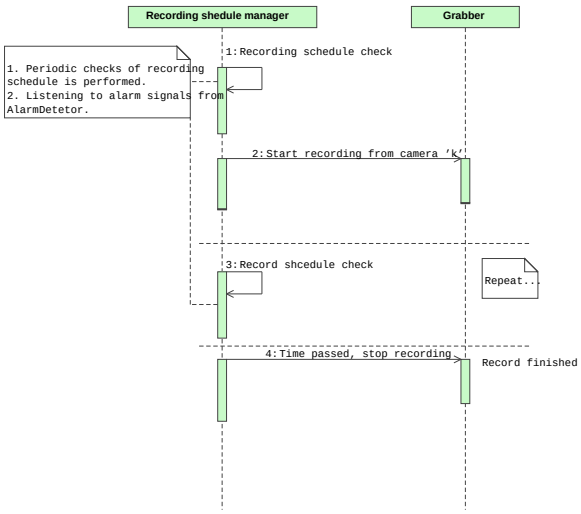


Notes

Alarm source object sends a signal to the alarm detector. Alarm detector checks the Alarm Conditions table to find out whether the incoming alarm can be activated or not. In case it is, messages are sent to objects, which functioning is dependant on the alarm events: **Recording Schedule Manager**, **Player** and **Database manager**.

When alarm is cancelled the objects involved in alarms handling (**Recording Schedule Manager**, **Player** and **Archive manager)** are also notified on the fact that alarm exists no more.

5.2 Recording Schedule Manager functioning



Notes

RSM performs periodic checks (each n seconds) of the Record Schedule table, to find out which recording must be launched or stopped on the current time. Another reason to start or stop recording is the registration or cancelling of alarm event. When recording is to be started, **RSM** sends order to **Grabber** object to start recording data from a definite camera. When the **RSM** detects that recording must be stopped, it sends the signal to **Grabber** to stop recording of a definite camera.