Mika Laitinen

THE DESIGN OF SENSOR LOCATION AND INFORMATION VISUALISATION FOR INDUSTRIAL ENVIRONMENTS

IBM Deutschland GmbH

Thesis
CENTRAL OSTROBOTHNIA UNIVERSITY OF APPLIED SCIENCES
Degree Programme in Information Technology
October 2009

CENTRAL OSTROBOTHNIA
UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

CENTRAL OSTROBOTHNIA UNIVERSITY OF APPLIED SCIENCES, Ylivieska	**Date** 17 November, 2009	**Author** Mika Laitinen
Degree programme Information Technology		
Name of thesis The design of sensor location and information visualisation for industrial environments.		
Instructor Hannu Puomio		**Pages** 41
Supervisor Amadeus Podvratnik, IBM Deutschland GmbH		

This thesis is about visualizing sensory data in 3D. Different sensors provide status and location information of items they monitor within predefined areas. The IBM Location Awareness service for WebSphere Sensor Events provides this information of the monitored items. The purpose of this thesis was to design a monitoring view and visualization elements for monitoring hundreds of sensory items showing their locations and statuses clearly in 3D view.

The user interface layout was designed filling accessibility and information requirements. This created a base for the design of the 3D objects. The objects were designed to be easily distinguishable from each other. The design includes also a way to show location, sensory information and statuses with the tracked sensor representation without disturbing the identification.

The result was a working design that combines 3D sensor representation with identification, location and sensory data, like temperature or humidity, in a clear innovative way. The designed sensor display elements can be used also for 2D visualizing. The designed user interface layout enables further information extensions.

Key words
3D, design, IBM, Location Awareness services, user interface, visualisation, WebSphere Sensor Events

KESKI-POHJANMAAN AMMATTIKORKEAKOULU
MELLERSTA ÖSTERBOTTENS YRKESHÖGSKOLA

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö	**Aika**	**Tekijä/tekijät**
Ylivieska	17 Marraskuu, 2009	Mika Laitinen

Koulutusohjelma
Tietotekniikka

Työn nimi
The design of sensor location and information visualisation for industrial environments.

Työn ohjaaja	**Sivumäärä**
Hannu Puomio	41

Työelämäohjaaja
Amadeus Podvratnik, IBM Deutschland GmbH

Opinnäytetyö käsittelee sensoritiedon esittämistä 3D ympäristössä. Sensorit tarjoavat tila ja paikkatietoa tarkkailemistaan kohteista määrätyllä alueella. IBM Location Awareness service for WebSphere Sensor Events tarjoaa nämä tiedot seuratuista kohteista. Opinnäytetyön tarkoituksena on suunnitella selkeä satojen seurattavien kohteiden seurantanäkymä ja käyttöliittymä 3D ympäristöön.

Käyttöliittymäkomponenttien sijoittelu on suunniteltu tiedon esittämisen ja esteettömyyden ehdoilla. Tämän pohjalle suunniteltiin 3D objektit. 3D näkymässä olevat objektit suunniteltiin helposti erotettavissa toisistaan. Suunnitelma sisältää myös kuinka paikka, sensorien tiedot ja erilaiset tilat esitetään ilman että objektien tunnistus häiriintyy.

Tuloksena oli toimiva suunnitelma joka yhdistää 3D sensori esityksen tunnistuksen, paikan ja sensori tietojen, kuten lämpötilan tai kosteuden, selkeästi. Suunniteltuja sensorien esitys elementtejä voidaan myös käyttää 2D visualisointiin. Käyttöliittymä mahdollistaa lisätietojen lisäämisen.

Asiasanat
3D, IBM, Käyttöliittymä, Location Awareness services, suunnitelma, visualisointi, WebSphere Sensor Events

Table of Contents

1 INTRODUCTION

This thesis was about the designing a visualiser for sensor tracking systems. The requirements for this design were taken from IBM Location Awareness Services for WebSphere Sensor Events. The existing visualiser has limits in terms of accessibility, presenting information and platform dependency, so developing a new visualiser was needed. There is an existing 3D design study that will provide the basis for the practical work.

The focus was accessibility and on how the information is shown to the user. This thesis will provide a design to display tracked items and all required information which belongs to these items in 3D view or in separate panels. The practical work was used only to back up the design.

I will first provide an insight into asset tracking and location awareness, presenting the IBM Location Awareness Services for WebSphere Sensor Events in more detail. Then I will explain the problems of the visualising sensor and location information and give a look on other asset tracking providers that area also struggling with these problems.

After that the basis of the work are presented by introducing the existing 3D design and giving an overview of the used technology. Also I am giving an insight into the accessibility and reasons to build a 3D visualiser and the problems of the 3D view. I will first present the user interface layout, where the information is shown and managed. Furthermore I will continue to describe the objects in the 3D view and finally present the design for the status messages.

The goals of this thesis were to provide a design for the future user interface for IBM Location Awareness Services for WebSphere Sensor Events. Filling the requirements of being accessible, easy to use, clear and informative while supporting customer specific information. The used technologies will allow the interface to be platform independent and presented as a standalone, browser applet or as a Eclipse plug-in.

2 ASSET TRACKING AND LOCATION AWARENESS

As the business grows, also the number of movable assets grow. Security and access control becomes a demanding task. Missing assets generate businesses a notable loss in income, may the loss be because of theft, damage or misplacement. Due to danger or hi-security areas also the monitoring of personnel, guests or patients in facilities that have access restrictions becomes important. A system for asset tracking and location awareness must be developed to solve these problems.
(Aberdeen Group 2004)

The tracked person or asset is equipped with a tag or a device that can be tracked trough sensors which are placed on the monitoring area. The tags can contain also sensors like temperature, humidity or pressure. A location engine observes sensors that monitor these tags and the messages the tags send and calculates the position of the tag. The location techniques vary from RFID, ultrasound, Wi-Fi, UWB, GPS and possible others depending on the location tracking solution. Some of these can reach position accuracy of 15cm while others focus on room-level accuracy.
(Ubisense 2009; Sonitor 2009)

There are numerous companies offering location awareness products and services. As the visibility and user interface issues are common to all solutions in location awareness this thesis presents a few of them but the main focus is on the needs of IBM Location Awareness Services for WebSphere Sensor Events.

2.1 IBM Location Awareness Services for WebSphere Sensor Events

IBM Location Awareness Services for WebSphere Sensor Events is a tool for monitoring and visualising locations of assets and people in predefined areas. The location data is provided by third-party systems, readers, tags and location engine. This makes the

Location Awareness Services for WebSphere Sensor Events independent of the technology used to track items and is therefore suitable for various different business areas, whether it would be security monitoring or asset tracking. Tracked assets and people are referred as an item and Location Awareness Services for WebSphere Sensor Events is referred to as LAS.

The LAS system provides visualisation for the monitored area. Tracked items have real time alerts and access notifications issued to them and for example security breaches are instantly noticed and notified. The LAS system supports events initiated by the defined rules. These event actions can be anything from sending an automatic email to running a scripted action from the server. As the items are put to different classes and groups different kind of security and access zones can easily be made and rules defined and assigned to different item classes or groups.
(IBM 2009b)

2.2 Problems of visualisation

Asset and people tracking is used in various different environments, organisations and businesses. All these have the same need, to be able to see the actions happening on the monitored area and to be able to deal with the possible problems as, or even before, they appear.

However, there are also differences in these areas and these differences create specific needs for the tracking and monitoring. Some monitoring areas have an important necessity to be able to tell at what height the tracked items are while for others, plain two dimensional movement tracking is enough. Also the tracked items may provide dynamic sensor data, like temperature, humidity, pressure or additional statuses along with the general location and alert statuses. These additional sensors can provide more insight on what the situation is on the area if it can be shown with the tracking.

The additional data providing extra status monitoring abilities in addition to the basic information should integrate in to the visualiser interface seamlessly. As the status of the items change it is vital to keep the user aware of these changes. Thus, the visual presentation of the items should tell the user immediately what the item is and what its status is. The available actions the user can do to manage the items should be presented in a clear way. In addition taking into account application accessibility that will enable use of the software by people with disabilities.

As the monitored environments are different so are the monitoring platforms. In the market there are visualisers that work only in some specific browser or client applications that are only targeted for computers running on Microsoft Windows operating systems. Although many provide an open API to their system so clients themselves can build their own visualiser but this cannot really be considered as a support for different platforms.

The old LAS system visualiser has lacks in these presented areas so a design for a new visualiser is needed to keep up with competition. There was an existing 3D design study that had a running 3D visualisation for LAS system and it was mostly focused on technological capability of Java3D. This thesis will be based on the existing study to design a new visualisation and management interface and solving the numerous issues presented.

2.3 Other asset tracking providers

The same visualisation problems are faced also by other software providers. Every provider has come up with its own way of showing and managing the data or specialising on the requirements of one specific business area. To face this competition the new LAS system visualiser must be designed to support high customisation to take on the requirements of different business areas.

Cisco Wireless Location Appliance

Cisco offers WLAN based location tracking based on its own Cisco WLAN controllers and access points. It tracks devices and tags operating on Wi-Fi and provides a 2D management interface for the monitored areas. The application is suitable for various business and government organisations.
(Cisco 2009a; Cisco 2009b)

AeroScout MobileView

AeroScout has a 2D management interface for monitored areas. It can use AeroScout's own Engine to provide location data using Wi-Fi tags. MobileView also supports the Cisco Location Appliance for location data. It is suitable for various business and government organisations.
(AeroScout 2009)

Sonitor Indoor Positioning System

Sonitor provides indoor 2D positioning system using ultrasound tags and receivers. It is used for indoor, room-level location and designed specially for hospitals.
(Sonitor 2009)

Ubisense Location Platform

Ubisense Location Platform provides 2D and 3D visualisation over its location tracking services. The accurate real time position tracking is done via UWB tags. Solutions vary

from military to agriculture.
(Ubisense 2009)

3 BASIS OF THE WORK

3.1 Existing 3D design study

The existing study was to design a basic level visualiser that could show the tracked items in 3D environment. The existing design used a multi-window interface.

In a multi-window interface different tasks of an application are separated into multiple separate dialogs. This is rarely used interface design on Windows systems and also not so common in Linux environments either. It has an advantage over single window interfaces when an application is spread over multiple monitors and therefore this interface design method is commonly used in image and video editing software. It requires some more adaptation from the user to effectively use the window layout than single window interfaces and that may be one of the reasons it is not very common.

The existing work had a working 3D view over the monitored area. It used the item images from the 2D visualiser as textured for single type of item shapes. The item tag id was shown as a text next to the item and had a support for showing if the item had an alarm. The items could be selected, the name and location shown and the trajectory of their movements drawn. In addition, there is a support for basic clustering.

By redesigning the user interface layout, adding support for proper information management and control and extending the features of the 3D view, the basis of the existing design can be used for the new visualiser design.

3.2 Java

Java is an object-oriented programming language developed by Sun. A program written in Java is run with the Java virtual machine and therefore is totally platform independent.
(The Java Language Environment 2009)

3.3 Java3D

The Java3D is a 3D graphics programming API for Java written originally by Sun. After the current stable release (1.5.2), the development was ceased by Sun and the source code and development responsibility was given to the community. The main parts of the source code were released under GPLv2 license with CLASSPATH exception.
(Java3D 2008a; Java3D 2008b; Java3D 2008c)

With Java's platform independence the application can be run on any machine supporting Java and Java3D. There is support for having the program as a standalone or as a browser applet. On unix systems Java3D uses OpenGL and on Windows systems OpenGL or Direct3D to provide hardware 3D support. The basic 3D features are implemented and stable. 3D objects can be loaded with object loaders from numerous different file formats into the 3D virtual universe. The most supported formats are OBJ and VRML97 but loaders exist also for other file formats.
(Java3D 2006)

3.3.1 Java3D thread scheduler

To have control over thread scheduling, Java3D has its own thread scheduler. The Java3D features lots of threads to do the scene graph and optimise it. Threads do geometry organising, control the view and events like picking and collision. The thread scheduler of Java3D is a big loop that runs all the threads and waits for them all to finish before running

them again.
(Doug Twilleager 2006)

3.3.2 Java3D scene graph model

On Java3D objects are formed in a tree like relation model. The 'root' object forms the base and the universe. Branch objects, like tags and zones, are added, and each of these objects may have their own branches. Transformations are done to branch groups as a whole. If the 'root' object is for example rotated, every branch of that object is rotated as well in the relation of the root object. Branch objects, as well as transformation objects can be attached and detached if the rules set to the branch allow it. Branch object can only have one 'root' object.
(Java3D 2006)

3.3.3 Java3D billboard

A basic tag object can have multiple branch objects as well as a special branch called billboard. Billboard is a special branch that is more commonly used to display text over the object it is attached to. Billboard can be attached with special transformation that automatically always translates the billboard to face the camera, or the view. This branch has a special transformation that keeps the objects within always translated towards the viewpoint.
(Java3D 2006)

3.4 Accessibility

When designing user interfaces, the accessibility of the application should not be forgotten. By making software accessible it will not only help people with disabilities to use the application, but might also increase the user base and the marketing value of the product. The Java Swing graphic utilities support automatically accessibility features so by using it the development is easier because the basic accessibility is taken care of by the Java foundation classes.

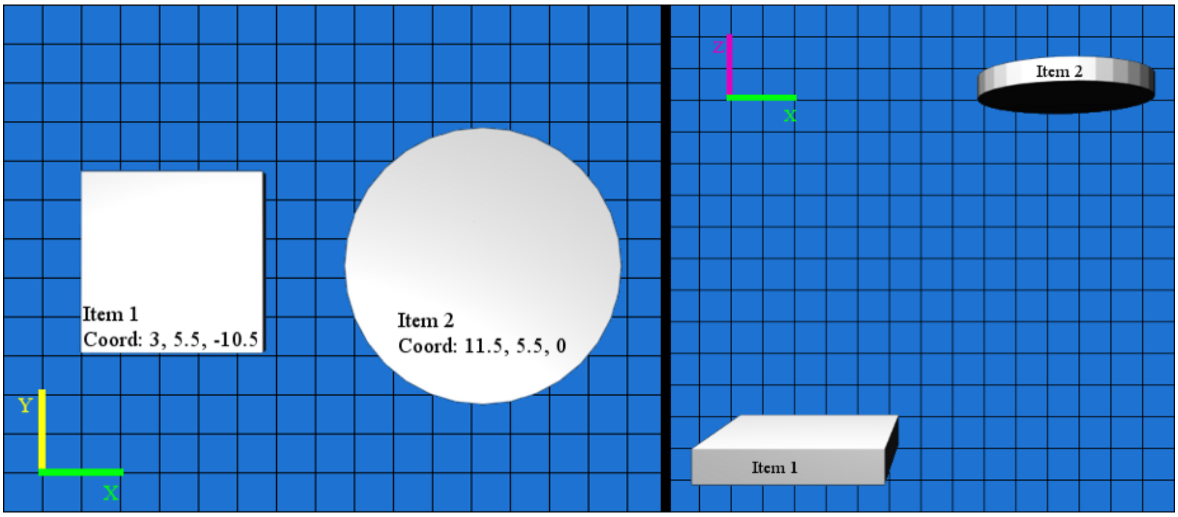
The use of the program can be made easier for people with generally lowered vision by supporting text-to-speech and other similar supportive applications by designing the information displays easily usable by these softwares. In addition, by designing the 3D objects for the items to be simple and clear the application will be generally easier to use.

It has been studied that approximately 10% of all people are somehow colorblind. To take this in account the design requires that when colors are used, additional information should be shown. The same level of information should be included, either by using shapes or text, or by planning the used color palette carefully. The user should also be given the chance to change the used colors throughout the program. Allowing the customisation of the used colors also helps people with other visual defects.

For the people that have mobility disabilities, the controlling of the application must be possible without a keyboard or mouse. Controlling the view and actions, selecting items and showing information should be made possible with either entirely, for example, with a mouse, keyboard, using shortcuts, or by voice control.
(IBM 2009a; Tiresias 2009a; Tiresias 2009b; Archimedes' Lab 2009; Masataka Okabe, Kei Ito 2002; Universal design principles 2007)

3.5 Why the 3rd dimension

As LAS tracks numerous items on the predefined area, the Spatial Management Client can show the movements and locations of the items in real time or as a recording, notify of alerts and item groups and it can be used to generate new zones. However there are some issues with the Spatial Management Client. It works only as a 2D presentation of the area and this generates the following problem.



GRAPH 1. Two items from different angles in 3D universe.

In the graph 1, the same situation is pictured from two different angles. On the left side there is a normal XY-coordinate view of the area with two objects, a default view with 2D interfaces, and on the right, there is the same situation but from a ZX-coordinate view. As seen, the problem of 2D is that the only way to tell that the other object is on different height is the textual information. As this information is only available when the item is selected on the current visualiser, the visual presentation leaves a place for improvement.

The 3D view of the area brings a solution to this with a different viewing perspective and the possibility for the user to control the virtual camera to look at the area from different angles. In addition, the 3D view supports better item or class specific models for the tracked items. This helps to identify items even if no detailed information is available.

Not only the items may have near-realistic models, the area can also be simulated in the view by placing 3D models into it. Displacing the 2D image of the area with 3D presentation of the area is more informative and clear with locations of where the tracked items are within the area. 3D view can be used to locate a stationary item, like a temperature sensor, precisely. These features help the user to get a true image of the actions happening on the area.

Also the Spatial Management Client only works in Internet Explorer. This kind of platform dependency causes limitations in distribution and installation. Thus, there is a real pressure to develop a new user interface that would address the platform dependency and bring the 3rd dimension into the visualisation.

3.6 The problems of 3D

However, there are some problems and things that need more consideration with having 3D visualisation. The controls to manipulate the view of the area are more challenging than on 2D user interface. This is mainly because normal pointing device, a computer mouse, if equipped with a scroll wheel, can only modify the view on 3 axes, X, Y and Z, while full control of a 3D view requires 3 axes more because of the rotation on each given axes. The most usual resolution is to use a keyboard in addition with the mouse to change the effect of the mouse movement to the view. This problem also adds design challenge for graphical controls that are needed to control the view.

The transition from 2D to 3D tracking affects the looks of the tracked items and visualisation of the area. To get the most out of 3D visualisation, the tracked items and the environment must be presented as 3D objects. In most cases only 2D images of the area exist so the objects must be redrawn in 3D or if existing 3D models of the monitored area or items are available, they must be imported, scaled and positioned properly into the viewer.

The solutions of the 2D visualisation, on how to show items that are under or in alert zones or each other must be rethought when moving to 3D. On 2D the item will be just shown on top of the zone it is in, but on 3D presentation the item can actually go inside the zone. This requires that the visual presentation of the zone allows that not only the items within can be seen but also the items that are behind the zone. This also adds requirements to the item representation so that items are equally identifiable while being in or out of the zone. In relation to this, there is also a problem of how to show the items if they get behind of each other or they gather in one place very close to each other.

4 USER INTERFACE

The layout design for the new user interface is kept as simple as possible while being informative of the actions happening in the monitoring area and by the user. The new interface would be a single window application. This clarifies the use on one monitor and provides an interface layout that can be utilised similarly for example as a standalone and browser applet usage keeping the user feel the same. Addressing the accessibility requirements also becomes simpler.

The main features required would be taken from the existing Spatial Management Client and adopted to fit the capabilities and limits of the 3D visualisation. Requirements are the ability to see the exact location of an item in relation to the visualised area and the ability to filter the viewed items and get detailed information of the items within the area. In addition the ability to manage alerts, create zones and record and playback a recording of items movements is required.

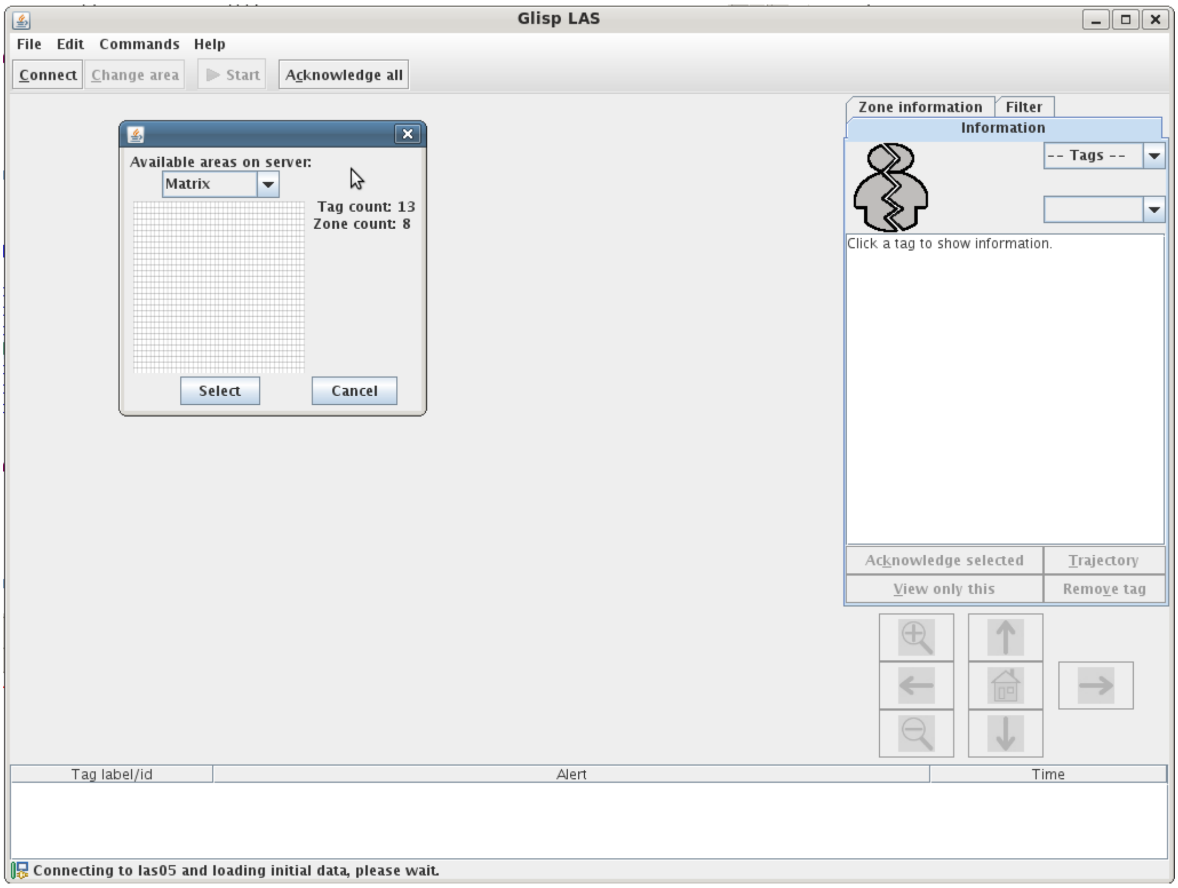
4.1 Location and status updates

The user interface gets updates of the items at certain intervals defined while connection profile to the LAS system is being made. The update contains all location and status changes. The user has a possibility to stop the update process any given time and not receive updates about alerts or movements of the area in case something needs to be observed more closely.

4.2 Interface layout

The application layout follows the design of general software interfaces using as much stock swing components as possible to deliver easy accessibility features and a familiar look and feel. These subjects, however, are not part of this thesis and the focus will be on how the information should be shown.

The interface can be separated into 4 most meaningful parts regarding the information. These parts are the 3D view, information, search and management tools, alert list and view controls. The 3D view is handled in its own chapter.



GRAPH 2. General view of the user interface layout.

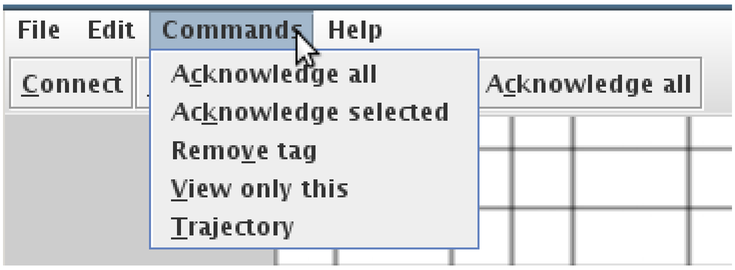
Tool bar

The tool bar features the most common tools; initiating connection to LAS server, getting area selection window, starting and stoping the updates to the view and action to acknowledge all alerts from the area. This provides a fast access to most used actions.

GRAPH 3. Tool bar with the most used commands.

Menu bar

All actions must be found in the menu bar for accessibility reasons as the menu can be easily accessed using for example only keyboard.



GRAPH 4. The menu contains all available actions.

Status bar

The status bar on the bottom of the application shows the current connection status. The shown information can be, for example, the active area name or update interval set to the connection.

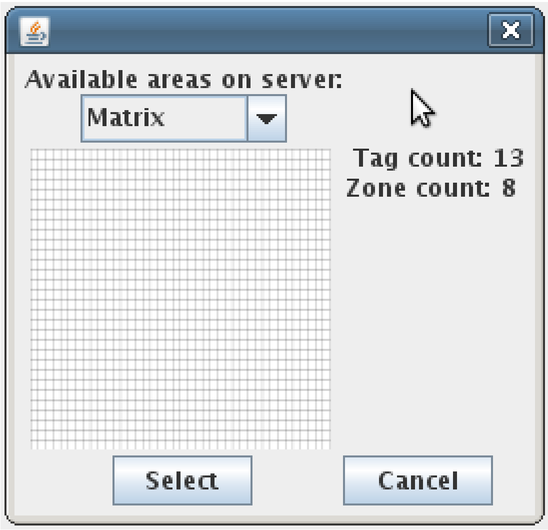
GRAPH 5. Status bar showing connection information.

Connection and area selection dialogs

Connecting to the server is done by first defining connection profiles of the servers. After that the connection is simple with just selecting the profile from connection dialog seen in picture 6.

GRAPH 6. Connection dialog to select connection profile and initiating connection.

A connection to the server is then made and the basic area information is loaded; the area name and image, item and zone counts. This information is then shown as an area selection dialog seen in picture 7. Areas are selectable from the drop down list. Changing the area loads the area image to the area under the drop down list.



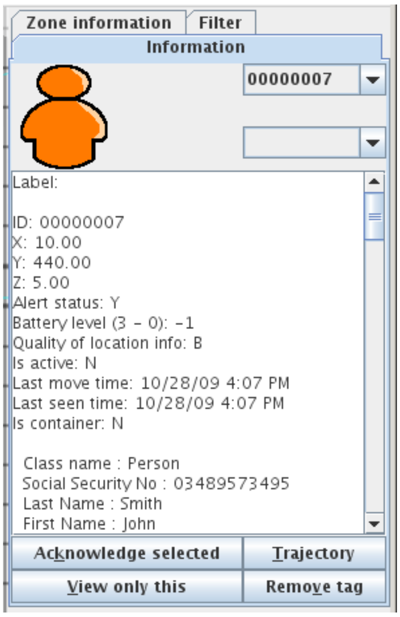
GRAPH 7. Area selection dialog.

4.3 Item information and management

The information display and management of the items are gathered together on the panel, on the right side of the 3D view. The panel is a tabbed panel and has by default three tabs for different actions. However, the tabbed panel can easily be extended with more tabs to feature client specific actions or information displays if the default panels are not enough.

4.3.1 Information display and management

The first tab contains item information. The image icon the item has is displayed and under that, the text panel has the information of the item. This approach to showing the information is better for accessibility reasons than having a separate panel window open when the item is selected. Under the information, there are buttons for management. The user is able to acknowledge alerts the item might have, remove the item from the view, hide all but the selected item and draw a history or trajectory line for the item. A trajectory line is a feature that can be issued to an item. It helps to track an item within the 3D view by drawing a line of the movements of the item. All these actions can be found also in the menu bar.



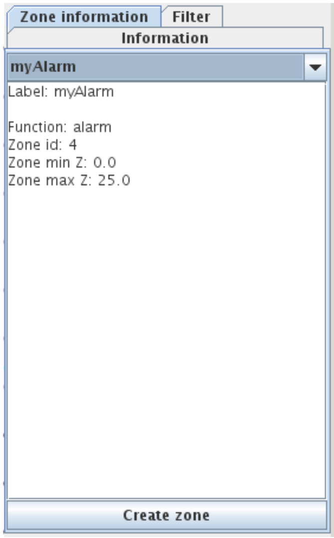
GRAPH 8. Information display and management panel.

Next to the item image, a drop down list of all items on the area are listed, this helps to find a specific item by label. If label is not defined, a tag id is used instead. When a cluster is selected, the name of the cluster is shown and under it, the drop down list is populated with the item labels that are within the cluster. This allows access to the information of the items within the cluster.

Together all these make clean, clear and compact information display and management block. When combining the managed information closely with the management tools, the idea of what the actions do is clearly delivered to the user.

4.3.2 Zone information

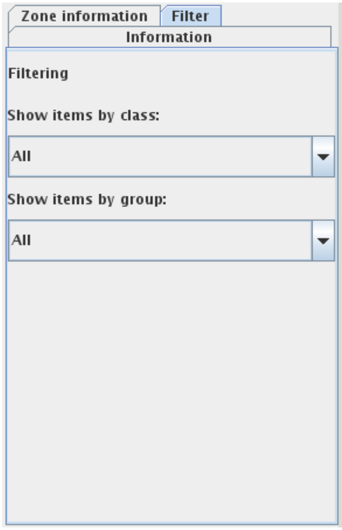
The second tab is for zone information and control. Separating the zone information controls from the item information controls keeps the interface clear. The user is presented a drop down list with the names of the zones within the area. The same area names are shown on the 3D view along the zones. The amount of data the zones have is not much so this does not require a lot of space. The left out space could be used to show a list of items that are inside the zone, provide tools to hide or show zones and give zone creation tools. More about creating zones with 3D view and the problems involved is explained later in the thesis.



GRAPH 9. Zone information panel.

4.3.3 Filtering and searching

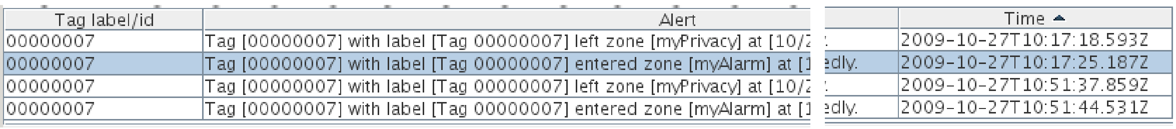
The third and last tab contains tools to filter the view and search for tags. To show only a specific item class or group on the screen so filtering helps to focus on specific types of items. Also when there are numerous tracked items, it is useful to be able to search for items within the area and get their data that way. Again putting this into its own tab helps to keep the user interface clear. Filtering could be done by class or group but also logical filtering is possible, for example by the signal level of the tracked items or by items having alerts.



GRAPH 10. Draft of filtering tab.

4.4 Alert controlling and notification

As alerts occur in the area, noticing them is important. The 3D view can be translated quite freely by the user so some areas and items may be left out of the view. If an item has an alert, the user does not notice it, if the alert is only shown on the 3D view. Also a method to provide an easy and clear, one glance status overlook of all alerts in the area is needed.

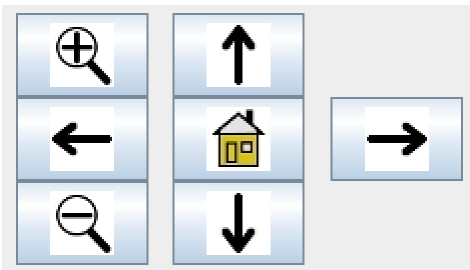


GRAPH 11. Alert list, modified for clarity.

To solve these problems, a list of alerts is kept under the 3D view. The list gets an update when an alert appears on any item in the area, showing the item label and the occurred alert message. The alert list will be sorted by time as new alerts are added. By selecting an alert from the list, the item having the alert is selected and the information shown. Single alerts can be acknowledged using a button on the information panel or by a pop-up window on the alert, accessible with a mouse button. As the alert is acknowledged, it is also removed from the alert list. This kind of a list helps the user to keep track of all the alerts and notifications in the area.

4.5 View controls

As stated earlier, controlling the three dimension view is tricky with normal input devices. The normal method is to use a mouse and keyboard together to translate the view. The view should also be movable with graphical elements, like buttons. A simple button configuration is provided to move the view and for zooming. The controls also feature a reset action to return the view to its original position when the application was started. This feature provides the user easy recovery if accidentally the view has been moved so that the area has disappeared from the view. The picture 12 shows the controls to control the view in 2D. For full 3D movement, keyboard must be used to move the view.



GRAPH 12. Controls to modify the view as 2D.

5 OBJECTS ON 3D VIEW

In the following single problems and requirements of showing the data in the 3D view are separated while presenting solutions to confronted issues. The notifications shown with the items are described in the chapter about item status information.

5.1 Single item

To prevent mis-shipping and possible danger or security issues it is necessary to have clear identification of the items to support the alert system. To separate individual items with the same classes and groups LAS items have a tag id, possibly a label and also they can have their own individual image. The image and the name of the item are a powerful identification couple.

Item specific icons from the existing 2D viewer can be used as a texture for the item's 3D model. If the item does not provide its own icon, a default image is used. The label can be shown by generating a hovering text next to the 3D model to support the identification of the item. If the tracked item does not have a label set, or is an unknown item, the tag id of the item is shown instead of the label. The text may also contain the position of the tag. With the item icon and identification text defined, individual items can be identified with ease. More information of the shape of the item 3D models is given in the chapter regarding the item classes.

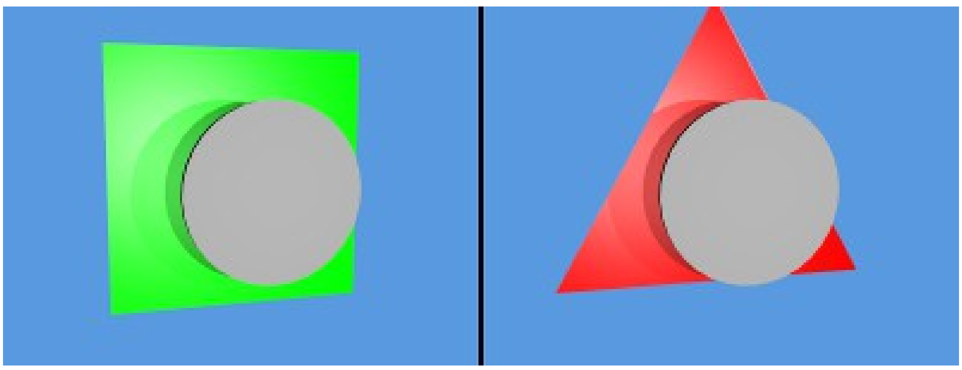
5.2 Item classes

LAS itself provides means to classify and group items and thus helping to separate different kinds of items from each other. Classes add properties and attributes to the items and provide detailed information. Separating the items using classes and groups, supervision can be focused on some specific class or group of items with better efficiency. Therefore in visualisation, different item classes and groups should be clearly distinguishable and combine well with the basic item identification, the image and name.

Each item may have a specific icon image, but the shape of the item's 3D object can be made class specific. The class specific 3D objects could be made to look like the items of that classification or just plain 3D shapes. If a class does not specify its own shape, a default item shape is used. The class name also appears with the item label. As the item icon is used to texture the 3D object, it is recommended that the loaded class models have a texture mapping set so that the icon shows properly on the item.

5.3 Item groups

The grouping of items is used to bound different items and classes together to provide, for example, common rules and searches. The grouping can be enforced to a tree-like hierarchy by the LAS server. When the grouping is enforced, the group can be issued a color to separate items belonging to different groups. On the existing Spatial Management Client, the color is displayed behind the item icons that are part of that group. The group shape is provided by the server as a data model.

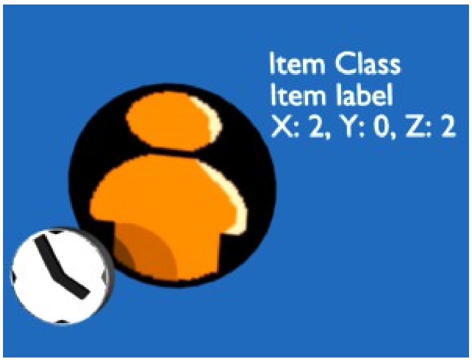


GRAPH 13. Different group separation models.

The same kind of method can also be used in the 3D view by generating a colored shape behind the item like in image 13. The shape with the color helps to separate items belonging to different groups.

5.4 Textual information with the item in 3D

The textual information on 3D view is using Java3D billboard branch. This keeps the text always translated towards the viewpoint keeping it always readable. It is not practical to show all information of an item on the 3D view since there is a chance that the text information of an item gets too long so that it extends over the surrounding items and render their identification data unusable. The location of the text is put to the upper right part of the icon. The static location of the text helps the user to identify which text belongs to which item. The textual information can be almost anything, from item class names to position coordinates as seen in picture 14.



GRAPH 14. An item with some of the information displayed with the 3D presentation.

It should be made possible to change the text size and color to match better the application, personal preference and increase accessibility. To give the user more tools to modify the view, options will allow to simplify the displayed information by turning off or on the displayed information pieces.

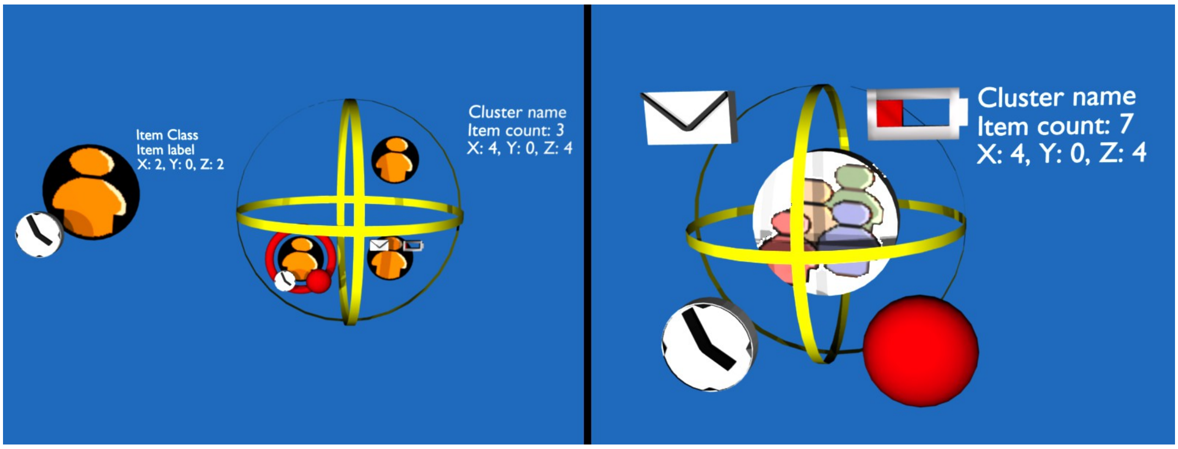
For example, if all items would contain their own icon, the label of the item is not necessarily needed. Then the user could turn the label text off to improve visibility of the displayed area. Also the nature of the 3D view of the area provides a solution. By simply moving the view a bit, the item labels turn to face the changed view and in most cases this solves the visibility problem of the texts. If the texts or the items were moved around the view to get them to show up all the time, the real positions of the items would not be clear.

5.5 Clustering

Turning off the viewed information is not the only way to deal with the items close to each other. Another way is to introduce a special item called ‘cluster’. Cluster is one kind of a container that is automatically generated when two or more items move into predefined approximation of each other, the items are removed from the view and a cluster item is showed instead. Clusters are removed and the normal items are shown as the items separate.

The cluster item has the information of the items it contains and this information can be accessed by selecting the cluster. A pop-up list of the items within the cluster is displayed and by selecting an item from that list, its details will be shown. The items in the cluster are also listed in the drop down list in the information panel.

Each generated cluster is given a name that helps in the search of the items. The cluster name, position of the center of the cluster and a count of how many items are in the cluster are shown as a floating text next to the cluster item. For possible different user requirements, the user is able to change the size of the clustering affect area or turn off the feature entirely.



GRAPH 15. Cluster item presentations with comparison to regular item.

As a graphical presentation, when the item count within the cluster stays small, about four, each item in the cluster could be shown scaled down and the yellow bands, as seen in left side of picture 15, drawn over the items. This is to clarify that a cluster has been generated of these items. The items within do not show their textual information anymore but the alerts and notifications can be seen. Accessing the items within the cluster will behave like described before.

When the number of items in the cluster grow over this limit, the scaling becomes too great to be of any use visually so cluster is show only as one big item telling the user that there are multiple items clustered at one location. The cluster shape can be given a texture image like to the other items by but this is not necessary. The yellow bands are also shown on the bigger cluster item.

Notifications of any item within the bigger cluster generate the same notification on the cluster item with a small exception with the alert that will always use the small alert notification. The red circle of the bigger alert notification is not shown for clusters because it might interfere with the yellow bands, seen in picture 15, that represent the area where the clustered items had been clustered. This does not affect the visibility of the alert since as the clusters are a little bigger than normal items so also the notification icons are scaled bigger as well.

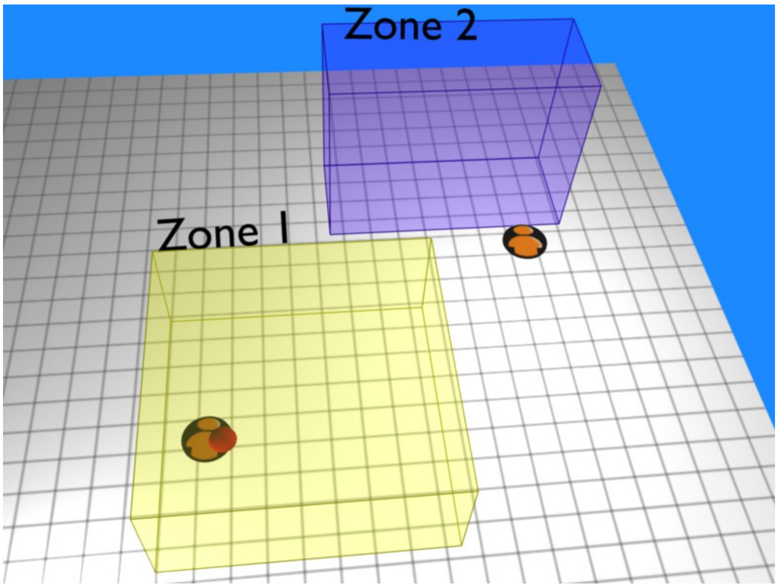
5.6 Zones

In addition to the tracked items, LAS visualisation also includes zones. Zones are areas within the main area that may have rules defined. When an item moves in, out or stays a defined time in a zone, an action is issued. This action may be an alert or other kind of an event set in LAS. As the items can move in and out of zones and stay in them, the zone representation must allow viewing inside the zone at all times.

Also as the zones can overlay each other, have different shapes and meanings, each zone must be identifiable. The zone's label and the name of the rule that has been issued with the zone should be accessible easily. To access the information on what items are within the zone and hiding and showing the zones is done from the zone information panel.

The zone data contains the name, rule name, color and area points of the zone. The area points can be used to generate a 3D object at runtime on to the view to represent the zone. Coloring the generated shape with the given color and making the object transparent allows identification of the zone and the user can see inside and under the zone.

To help the user to distinguish the shape of the zone, the edges of the zones are brought to the display as non-transparent and with darker coloring as seen in picture 16. To access the zone data, the information panel must be used. The same kind of picking that allows the user to select items from the area can not be utilised for the zones. This is because if the zones were selectable, possible tags within or under the zone would become unavailable.



GRAPH 16. Zone presentation.

The creation of a zone is relatively easy on a 2D plane but the 3D space generates its own problems. These problems are again related to the controls over the 3D view and also the visualisation of a 3D view. To allow creation of areas that are freely shaped 3D forms requires much from the user interface and some 3D designing skills from the user so 3D creation tool is not reasonable to have in the general viewer. Instead, allowing the user to generate a freely formed 2D plane on the area and then defining single depth via the dialog window is more reasonable in terms of usability. The zone creation can be done with the tools in the zone creation tab of the user interface.

5.7 Area

The area in the visualiser view is a virtual image of the actual physical location of the monitored area. The existing Spatial Management Client, as all other 2D visualisers, use an overview image of the monitored area to show the tags on. As this image is only 2D it does not provide information on the heights of the area, such as high platforms or ditches. The 3D viewer can use the old 2D images also to show where the items are as in the 2D view but in order to get the best use out of 3D visualisation, the area should be modelled with 3D modelling software or alike.

Having a 3D model of the area helps to clarify where the items are since the objects then really can be monitored by height, and also the visual understanding of the monitored area improves notably. Using the existing 2D images of the area as a base for the 3D models provides an easy way to keep the dimensions and area scaling right.

6 ITEM STATUS INFORMATION

The items have different kind of status information among the other information that should be underlined and the changes in these statuses should be made noticeable even if the item is not in immediate surveillance. The shapes and coloring mirror the statuses they notify from to provide accessibility and clearness.

The position of the status icons stay always in the same place. Thus even when zoomed out to view the area, the user can see difference in one of the corners of the item and after some accustoming the user can immediately tell what notification the item is showing. The alert notification icons take advantage of the Java3D scene graph tree model. The item is a root object to the notification icon "leaves". If connected to the Java3D billboard graph, the notifications will always face the view, enabling viewing the notifications from every angle.

GRAPH 17. Item with all overlaying status icons (battery empty).

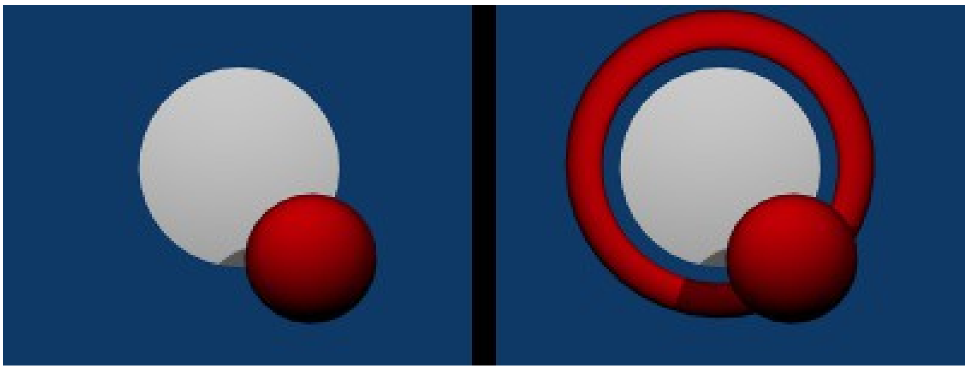
As seen in the image 17, the item identification via the item icon is not disrupted because the status icons are designed not to cover the item fully but to appear around it. Four status icons per item can be kept as a maximum in terms of usability, as more icons would make the representation unclear. The size of the icons are scaled with the item by the Java3D engine so they are always correctly sized to the item.

More status messages and information can be shown also as a text next to the item, like coordinates and temperature information. Showing the text on the 3D view with the item was described in the chapter 'Textual information with the item in 3D' earlier in the thesis.

In the following sections each status icon is examined more thoroughly explaining the problem behind the icon and provided solution.

6.1 Alert

Maybe the most common status change is an alert. A basic alert is generated when an item belonging to a group or a class that has a rule defined for a zone and the item triggers the alert by interacting with the zone. Alerts are important and the appearance of an alert must be noticeable immediately when they occur and also the items having alerts should be distinguishable from the other items.

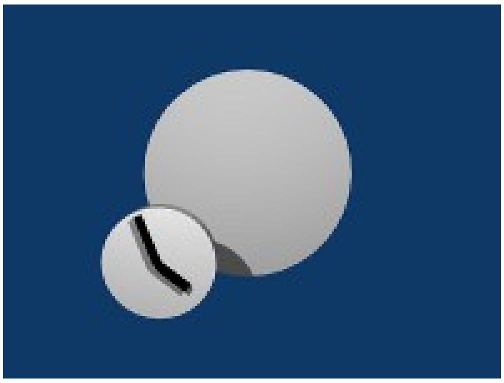


GRAPH 18. Demonstrated alert icons, on left is presented a small alert message and on right a big alert message.

The alert notification differences from the other notifications with two possible alert styles. The alert types can be seen on image 18. The smaller alert type adds a red ball shape on the lower right corner of the tag. The shape is a little bigger than the other notifications and being a single color and form, it distinguishes itself from the surrounding. The user can also select another alert notification type where a red circle is also added around the tag to help it stand out from the environment and other tags. However, this could also be used to notify the alert severity by LAS system.

6.2 Inactivity

When an item stops moving and does not get any updates to its statuses, the item is marked as inactive by the LAS server. This is important information and can be used to track down problems. As this is not an alert, but an notification type of event, it should not be combined with the way alerts are shown but have their own way of notify the inactivity.

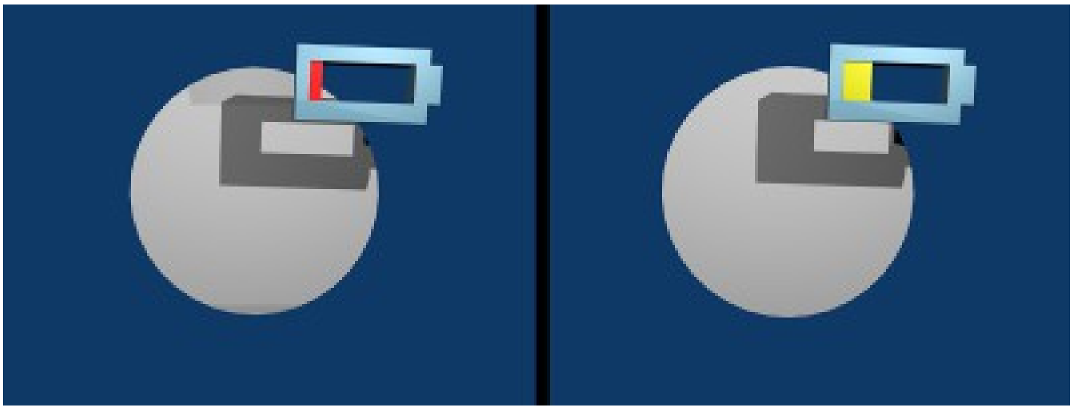


GRAPH 19. Icon representing the inactivity of an item.

When an item ceases to be active, the user is notified of this by adding a clock image on to the lower left corner of the tag icon, shown on image 19. The clock resembles the long time from the last update to the current tag. The black on white design is simple and clean.

6.3 Battery

The item data provides information about possible battery status of the tracked item tag. As the inactivity notification, it is not convenient to notice about low battery by normal alert, but with constant notification that keeps on reminding that maintenance must be done in order to keep the system and tracked items functioning properly. Battery information has four different stages, empty, low, regular and full.



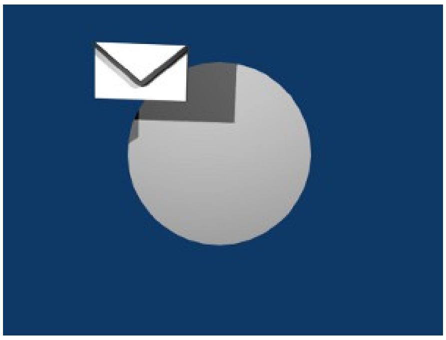
GRAPH 20. Battery empty notification on left and battery low notification on right.

This kind of information is more traditionally shown as a color coded bar that shrinks and grows depending on the battery status. The same principle is applied here, as seen in the picture 20. The battery icon shape resembles battery and within it, a colored bar is telling the status of the battery. The border around the colored indicator bar helps to clarify the information that the battery status of the item is becoming empty or already is.

To cut down the information displayed on the screen at the same time, battery status is displayed only on the tag when it is low or empty. Current battery status can always be checked as a numerical status from the item's details.

6.4 Messages

For the future features, there was a requirement for the possibility to receive, notify and send messages from and to tracked items trough LAS system. The messages may be informative or directions send to persons, possibly pictures or video. When an item sends a message, it would generate a notification for the user.

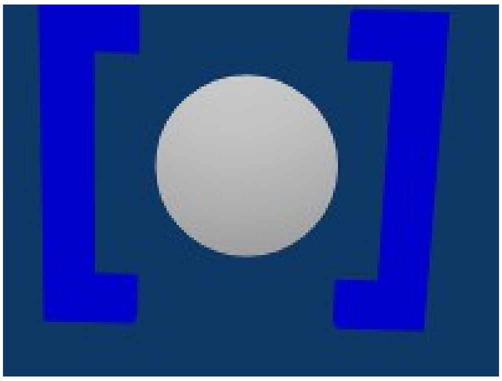


GRAPH 21. Item with unread message.

The envelope shape was a natural selection to notify the user about incoming message. The simple black and white design is easy to distinguish.

6.5 Selection

Detailed information of an item can be accessed by selecting the item from the view with the mouse or using a list of items from the information panel. Also when an alert is selected from the alert list, the item that the alert belongs to is selected and its information shown. The selection needs to be visually shown to have a reference from the information to the view.



GRAPH 22. Selected item notification.

The selection notification is made with opening and closing brackets around the item. This keeps the item icon viewable as well as the possible additional status notifications. As it is bound to the same translation as the item text, always turning towards the view, it is always visible.

7 CONCLUSIONS

The goal of the thesis was to design a new user interface for the LAS system and solve problems of displaying various information and accessibility. Use the existing 3D design study as a base for the practical part to build on a new interface that would be platform independent and movable over different presentation methods, from standalone software to browser applet or as a Eclipse plug-in.

The start of the thesis was difficult. I had no existing knowledge about accessibility, 3D design or 3D programming. Also the Java Swing GUI toolkit and Java3D API were new to me, but working on this thesis, I have gained lots of knowledge and experience of these technologies.

To have an appealing and accessible design all the objects were designed to be simple and at the same time clear on what they mean. I used the existing experience with 3D programs such as games and SecondLife style virtual worlds as a help in design. The design of the view in the 3D visualisation was straight forward. The objects and status messages are clear in the coloring and shape and the user interface is simple and supports customisation. I believe the design has succeeded in filling the given requirements.

The design was taken further than the practical work allowed me to go. Implementing the features into the existing design study proved to be difficult to do properly. On one hand, working on the existing code was time consuming because of missing proper documentation and on the other hand the original program design did not fit into the new design. It turned out to be impossible to integrate all the designed elements. Adding the notification icons on top of the tracked items, for example, caused great performance problems. This was mainly because of how the existing visualiser was designed to function.

The program design of the old visualiser did not take into account the possible lags in fetching the data from the server and parsing it. As the Java3D features its own task scheduler and it is only meant for showing the 3D objects and dealing with different user actions and events it would have been best to separate the data handler from the Java3D. This would help to reduce the performance issues and make some of the designed features possible.

By separating the program controls from the user interface and making the 3D view independent from the data management, the visualiser can be modified easily to get the best out of different possibilities to provide the user interface. Whether it would be a browser applet, standalone software or a eclipse plug-in.

With this time constraint I had for the thesis and because of the subject of the thesis it was not possible to do a program design and implementation for the designed features. Howevver, this thesis shows all the necessary parts for the visualizer and provides an insight into what data is needed to make it possible. Thus, this thesis is a great design plan and a start for the future program design of the visualiser for the LAS system.

REFRENCES

Aberdeen Group 2004. Executive Summary Report: RFID-Enabled Logistics Asset Management: Improving Capital Utilization, Increasing Availability, and Lowering Total Operational Costs . Available:
http://www.aberdeen.com/summary/report/other/rfid_lam.asp. Modified June 2004. Accessed 13 October 2009.

AeroScout 2009. Products Overview on Wi-Fi-based Solutions for RTLS, RFID, Wireless Sensing and More. Available:
http://www.aeroscout.com/content/products-services . Accessed 14 October 2009.

Archimedes' Lab 2009. Ishihara Color Blindness Test, All You Need Know About Color Deficiency . Available:
http://www.archimedes-lab.org/colorblindnesstest.html. Accessed 13 October 2009.

Cisco 2009a. Location Solution Introduction . Available:
http://www.cisco.com/en/US/netsol/ns753/networking_solutions_package.html . Accessed 14 October 2009.

Cisco 2009b. Location Solution Overview. Available:
http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns348/ns753/net_brochure0900aecd8064fe9d.html. Accessed 14 October 2009.

Cisco 2009c. Wireless Location Appliance Introduction . Available:
http://www.cisco.com/en/US/products/ps6386/index.html . Accessed 14 October 2009.

Doug Twilleager 2006. Java3D Internal Architecture. Available:
http://java3d.j3d.org/implementation/architecture.html. Modified April 18, 2006. Accessed 25 October 2009.

IBM 2009a. Understanding accessibility. Available:
http://www-03.ibm.com/able/access_ibm/disability.html . Accessed 13 October 2009.

IBM 2009b. WebSphere Sensor Events, Version 6.2 Documentation. What is Location Awareness Services for WebSphere Sensor Events? Available:
http://publib.boulder.ibm.com/infocenter/pvcsensa/v6r2m0/index.jsp?topic=/com.ibm.wse.doc_6.2.0/las/o_whatis.html . Accessed 16 October 2009.

Java3D 2006. FAQ. Available:
http://java3d.j3d.org/faq/intro.html . Modified April 4, 2006. Accessed 20 October 2009.

Java3D 2008a. Announcement. Available:
https://java3d.dev.java.net/servlets/NewsItemView?newsItemID=5762 . Modified February, 2008. Accessed 14 July 2009.

Java3D 2008b. Forum Announcement. Thread: ANNOUNCE: GPL open source release. Available:
http://forums.java.net/jive/thread.jspa?messageID=261477&tstart=0#261477 . Modified February 28, 2008. Accessed 14 July 2009.

Java3D 2008c. Forum Java 3D Plans. ANNOUNCEMENT: Java 3D plans . Available:
http://forums.java.net/jive/thread.jspa?threadID=36022&tstart=315 . Modified January 29, 2008. Accessed 15 July 2009.

Masataka Okabe, Kei Ito 2002. Color Universal Design (CUD) - How to make figures and presentations that are friendly to Colorblind people.Masataka Okabe , Jikei Medial School (Japan) , Kei Ito, University of Tokyo, Institute for Molecular and Cellular Biosciences (Japan) . Available:
http://jfly.iam.u-tokyo.ac.jp/color/ . Modified November 20, 2002, February 15, 2008, September 24, 2008. Accessed 14 October 2009.

Sonitor 2009. Ultrasound IPS. Available:
http://www.sonitor.com/ . Accessed 14 October 2009.

The Java Language Environment 2009. Available:
http://java.sun.com/docs/white/langenv/Intro.doc2.html. Accessed 23 October 2009.

Tiresias 2009a. Color Blindness. Available:
http://www.tiresias.org/accessible_ict/colour_blindness.htm . Modified August 06, 2009. Accessed 14 October 2009.

Tiresias 2009b. Visual Impairment. Available:
http://www.tiresias.org/accessible_ict/visual.htm. Modified September 22, 2009. Accessed 14 October 2009.

Ubisense 2009. Products. Available:
http://www.ubisense.net/en/products. Accessed 14 October 2009.

Universal design principles 2007. Universal design principles North Carolina State University . Version 2.0. Compiled by advocates of universal design, listed in alphabetical order: Bettye Rose Connell, Mike Jones, Ron Mace, Jim Mueller, Abir Mullick, Elaine Ostroff, Jon Sanford, Ed Steinfeld, Molly Story, & Gregg Vanderheiden . Available:
http://www.design.ncsu.edu/cud/about_ud/udprincipleshtmlformat.html . Modified 1997. Accessed 14 October 2009.