Resource Process Visualization
Technical and User Documentation

version 2.06

Group 2

Christian Mühlroth

Martin Gumbrecht

Jan-Philipp Stauffert

Yao Guo

Kathrin König

Table of contents

1. Introduction

This document serves as both basic technical and user documentation for the project “Resource Process Visualization” at Audi AG, Ingolstadt in close collaboration with the University of Erlangen-Nuremberg.

The AUDI Resource Process Visualization project aims to supply an application of AUDI's global production plants. A navigateable world map allows hierarchical browsing to every production plant in every country. At each level of browsing the aggregated information of the production line is calculated and information about quality and testing are available.

2. Technical Requirements

2.1. Database requirements

The Resource Process Visualization application requires the following database installation:

- PostgreSQL 9.1

The application requires a database with utf8 encoding and a user with the privileges: Insert, select, update and delete

The tables have to be set up with the schema.sql file:

```
DROP TABLE IF EXISTS component CASCADE;
DROP TABLE IF EXISTS device CASCADE;
DROP TABLE IF EXISTS line CASCADE;
DROP TABLE IF EXISTS location CASCADE;
DROP TABLE IF EXISTS hall CASCADE;
DROP TABLE IF EXISTS factory CASCADE;
create table factory(
      id int PRIMARY KEY,
      name char varying(128) NOT NULL DEFAULT '',
      company char varying(128) NOT NULL DEFAULT '',
      city char varying(128) NOT NULL DEFAULT '',
      country char varying(128) NOT NULL DEFAULT '',
      gpslatitude double precision NOT NULL DEFAULT 0,
      gpslongitude double precision NOT NULL DEFAULT 0,
      carmodels char varying(512),
      sizeofstaff int,
      sizeofstaffdate timestamp,
      vehiclesperyear int,
      vehiclesperday int,
      upssystems int,
      upsservers int,
      upsprovider char varying(128),
      parent integer NOT NULL DEFAULT 0,
```

```
      map TEXT NOT NULL DEFAULT ''
);
CREATE TABLE hall (
      id integer PRIMARY KEY,
      name character varying(128) NOT NULL DEFAULT '',
      type character varying(128) NOT NULL DEFAULT '',
      staff integer NOT NULL DEFAULT 0,
      capacity integer NOT NULL DEFAULT 0,
      vehicles char varying(512) NOT NULL DEFAULT '',
      upsServers int NOT NULL DEFAULT 0,
      path character varying(50) NOT NULL DEFAULT '',
      parent integer REFERENCES factory(id) ON DELETE CASCADE,
      map TEXT NOT NULL DEFAULT ''
);
CREATE TABLE line (
      id integer PRIMARY KEY,
      name character varying(50) NOT NULL DEFAULT '',
      series character varying(50) NOT NULL DEFAULT '',
      capacity integer NOT NULL DEFAULT 0,
      path character varying(50) NOT NULL DEFAULT '',
      parent integer REFERENCES hall(id) ON DELETE CASCADE
);
CREATE TABLE location (
      id integer PRIMARY KEY,
      name character varying(50),
      description character varying(100),
      personincharge character varying(100),
      parent integer REFERENCES line(id) ON DELETE CASCADE
);
CREATE TABLE device (
      id integer PRIMARY KEY,
      type character varying(50) NOT NULL DEFAULT 'N/A',
      description character varying(50) NOT NULL DEFAULT 'N/A',
      name character varying(50) NOT NULL DEFAULT 'N/A',
      networkstatus character varying(50) NOT NULL DEFAULT 'N/A',
      ipaddress character varying(50) NOT NULL DEFAULT 'N/A',
      maintainanceinfo character varying(50) NOT NULL DEFAULT 'N/A',
      sector character varying(50) NOT NULL DEFAULT 'N/A',
      serialnumber character varying(50) NOT NULL DEFAULT 'N/A',
      troubleperiod timestamp NOT NULL DEFAULT '2000-01-01 00:00:00.0',
      testfailure boolean NOT NULL DEFAULT false,
      parent integer REFERENCES location(id) ON DELETE CASCADE
);
```

```
CREATE TABLE component (
      id integer PRIMARY KEY,
      name character varying(50) NOT NULL DEFAULT 'N/A',
      value character varying(50) NOT NULL DEFAULT 'N/A',
      sector character varying(50) NOT NULL DEFAULT 'N/A',
      category character varying(50) NOT NULL DEFAULT 'N/A',
      serialnumber character varying(50) NOT NULL DEFAULT 'N/A',
      shiftresponsibility character varying(50) NOT NULL DEFAULT 'N/A',
      troubleoccurrencetime timestamp NOT NULL DEFAULT '2000-01-01 00:00:00.0',
      troubleoccurrencesite character varying(50) NOT NULL DEFAULT 'N/A',
      status character varying(20) NOT NULL DEFAULT 'grey',
      parent integer REFERENCES device(id) ON DELETE CASCADE
);
```

2.2. Software requirements

The Resource Process Visualization Application requires the following software installations:

- Apache Tomcat 6.0.36
- Java JDK 1.6.0_45

2.2.1. Apache Tomcat

The application requires a JDNI datasource that points to the database that should be used. The following configuration has to be added in

```
<tomcat dir>/conf/context.xml:
<Resource
            name="jdbc/postgresql"
            auth="Container"
            type="javax.sql.DataSource"
            driverClassName="org.postgresql.Driver"
            url="jdbc:postgresql://<host ip>:<port>/<database name>"
            username="<username>""
            password="<password>"
            maxActive="20"
            maxIdle="10"
            maxWait="-1" />
```

The following additional libraries are required in

```
<tomcat dir>/lib
```

postgresql-jdbc4.jar	http://jdbc.postgresql.org/download.html
commons-dbcp-1.4.jar	http://commons.apache.org/proper/commons-dbcp/download_dbcp.cgi

2.2.2. JAVA JDK

The JAVA_HOME environment variable has point to the required JDK

2.3. Browser recommendations

The following browsers are recommended:

- Internet Explorer 8 – with constraints (no SVG support)
- Internet Explorer 9+
- Mozilla Firefox 5+
- Google Chrome 11+
- Opera 11+
- Safari 5+

2.4. Application deployment

The only artifact that has to be deployed is the application's war file. All configurations files are included in the archive (see chapter 3 "Configuration").

3. Configuration

3.1. Setting up the hierarchy

Factories, Halls, Lines, Locations have to be inserted into the Database via SQL before the start of the application.

Attributes of an element can be changed at runtime via SQL statements. To apply the changes visit

```
http://<server>:<port>/<applicationname>/config
```

Removing or adding elements at runtime via SQL statements is not allowed and may cause errors.

Testing devices can be inserted via SQL before program start and. Attributes can be changed via SQL update statement at runtime. Testing devices can also be added and configured via XML at runtime.

3.1.1. Factory

Factories can be inserted via SQL insert statement according to the schema of the table factory. The attribute id has to be unique over all tables. The attribute "parent" has to be 0 in all factories. Car models have to be separated by a comma. The attribute map has to be a valid SVG graphic where every path element reflects a graphical representation of a hall. For more details see section Factory and Hall plans.

```
insert into factory values (1, 'Ingolstadt', 'Audi' ,'Ingolstadt', 'Germany', 48.762201,
11.425374, 'Audi A3, Audi S3, Audi S3 Sportback', 35386, '2012-12-31 00:00:00.0', 551889,
2580, 1, 3, 'DSA GmbH', 0 );
```

3.1.2. Production Hall

Production halls can be inserted via SQL insert statement according to the schema of the table hall. The attribute id has to be unique over all Tables. The attribute “parent” has to refer to the id where of the factory where the hall is located. Vehicles have to be separated by a comma. The path element has to refer to the id of the path attribute in the factory map. The attribute map has to be a valid SVG graphic where every path element reflects a graphical representation of a line. For more details see section Factory and Hall plans.

```
INSERT INTO hall VALUES (2, 'G6', 'Assembly+Finish', 709, 6249, 'Audi A4', 2, 'path369',
1);
```

3.1.3. Line

Assembly lines can be inserted via SQL insert statement according to the schema of the table line. The attribute id has to be unique over all tables. The attribute “parent” has to refer to the id of the hall where the line is located. The path attribute has to refer to the id of the path element in the hall map. The attributes series and capacity are currently not used by the application.

```
INSERT INTO line VALUES (3, 'A3 Assembly', '', 0, 'rect217', 2);
```

3.1.4. Location

Locations can be inserted via sql insert statement according to the schema of the table location. The attribute id has to be unique over all tables. The attribute “parent” has to refer to the id of the hall where the location is located. The attribute personinchare is currently not used by the application.

```
INSERT INTO location VALUES (4, 'L3', 'test', ' ', 3);
```

3.1.5. Testing Device

Testing Devices can be inserted via sql insert statement according to the schema of the table device. The attribute id has to be unique over all tables. The attribute “parent” has to refer to the id of the location where the device is located. The attributes networkstatus, ipaddress, mainenanceinfo and sector are currently not used by the application.

Each Testing Device should have 3 Components with the Names “Tests”, “Network” and “Maintenance” in the table component.

```
INSERT INTO device VALUES (5, 'UPS MFT Cradle', 'DSA MFT Cradle for Power Supply and LAN
connection', 'IBNA-LS3X0', ' ', ' ', ' ', 'E', '8885859', '2013-07-01 03:14:11.939',
false, 4);

INSERT INTO component (id, name, value, sector, category, parent) VALUES (6, 'Tests',
'Ok', 'C', 'B', 5);

INSERT INTO component (id, name, value, sector, category, parent) VALUES (7, 'Network',
'IP: 10.0.0.1', 'C', 'B', 5);

INSERT INTO component (id, name, value, sector, category, parent) VALUES (8,
'Maintenance', 'no scheduled offtime', 'C', 'B', 5);
```

3.2. Factory and Hall plans

A factory or hall plan has to be a valid SVG graphic. It contains several path elements that have the path information, an id and an optional style attribute.

In a factory/hall plan there can be two types of halls/line.

- If a hall/line in the plan should be clickable, there has to an entry in hall/line table of the database which refers to the id of this path.
- If a hall/line should not be clickable there must be no entry in the tables which refer to the path. Instead if this a style information should be added so that the element is displayed in a grey color.

In the following example the path with id “polygon5” is clickable and “polygon7” is not.

```
INSERT INTO hall VALUES (1, 'G6', 'Assembly+Finish', 709, 6249, 'Audi A4', 2, 'path369',
10, '<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<svg

   xmlns:dc="http://purl.org/dc/elements/1.1/"

   xmlns:cc="http://creativecommons.org/ns#"

   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

   xmlns:svg="http://www.w3.org/2000/svg"

   xmlns="http://www.w3.org/2000/svg"

   version="1.1"

   width="623.62"

   height="311.81"

   viewBox="0 0 623.62 311.81"

   id="SVGPlan"

   xml:space="preserve"><metadata

   id="metadata4065"><rdf:RDF><cc:Work

       rdf:about=""><dc:format>image/svg+xml</dc:format><dc:type

         rdf:resource="http://purl.org/dc/dcmitype/StillImage"
/><dc:title></dc:title></cc:Work></rdf:RDF></metadata><defs

   id="defs4063">

</defs>
```

```
<g
   id="g7976"
   style="fill:#434343;fill-opacity:1;stroke:none"><path
     d="M 130.101,69.406 H 75.267 v 3.437 h 45.001 l 3.666,2.063 h 4.667 l 14.415,-4.833 h 138.36 c 1.413,0 2.56,1.146 2.56,2.559 v 14.506 c 0,1.413 -1.146,2.56 -2.56,2.56 h -126.36 l -25.415,3.876 H 76.185"
     id="path45"
     style="fill:#434343;fill-opacity:1;stroke:none" /><path
     d="M 71.935,70.073 V 83.53 h 53.167 l 15.946,-5.59 h 136.545 c 0.778,0 1.409,-0.632 1.409,-1.409 v -7.983 c 0,-0.777 -0.631,-1.408 -1.409,-1.408 H 201.444 75.267 l -3.332,2.933 z"
     id="path49"
     style="fill:#434343;fill-opacity:1;stroke:none" /><path
     d="M 82.393,97.97 H 276.28 c 1.105,0 2.001,-0.896 2.001,-2 V 84.634 c 0,-1.104 -0.896,-2 -2.001,-2 H 168.152 141.768 L 125.31,87.655 71.935,87.572 v 8.166 l 3.332,2.232 h 7.126 z"
     id="path51"
     style="fill:#434343;fill-opacity:1;stroke:none" /></g><path
   d="m 523.18597,35.404999 24.318,0 0,36 -24.318,0 z"
   id="rect3"
   style="fill:#434343;fill-opacity:1;stroke:none" />
<path
   d="m 523.186,78.155 0,9.375 4.875,0 0,3 4.875,0 0,2.125 -3.5,0 0,7.75 -6.25,0 0,-7.75 -14.75,0 0,72.749 14.75,0 0,-20.249 24.318,0 0,-42.5 0,-2.25 0,-7 0,-0.75 0,-3.125 0,-1 0,-1 0,-9.375 z"
   id="polygon5"/>
<path
   d="m 547.504,222.322 0,-1.416 0,-4.361 0.017,0 -0.033,-9.973 0.016,0 0,-54.667 -24.318,0 0,54.667 8.484,0 0.016,4.582 -21.5,0 0,28.25 13.75,0 6.666,0 1.084,0 1.25,0 14.585,0 0,-17.082 z"
   id="polygon7"
   style="fill:#434343;fill-opacity:1;stroke:none" />
```

```
INSERT INTO line VALUES (2, 'G6 Assembly', 'Series 2', 86, ' polygon5', 1);
```

3.3. Status aggregation

In the config directory of the application there is a .properties file for every hierarchy level which configures how the status is aggregation in this level.

There are two strategies available:

1. Minimum: The Element will have the minimal status of it's child elements. That means if one child Element is red, the parent element will be red, too. If this strategy should be used the .properties file should look like:

   ```
   strategie=minimum
   ```

2. Percentage: The percentage of sub-elements where an element turns red or yellow can be defined as well as the hierarchy level that should be referred.

In the following example the hall will turn red if at least 20 percent of the devices in this Hall are red, or at least 30 percent are yellow. If at least 10 percent of the devices in the Hall are red, or at least 20 percent are yellow, the hall will be yellow. Otherwise it will be green.

```
strategie=percentage
aggregationLevel=Device
redPercentageForRed=20
yellowPercentageForRed=30
redPercentageForYellow=10
yellowPercentageForYellow=20
```

3.4. Country flags and brand logos

In the config directory of the application there is a folder flag_country and logo_company

Each factory has the attribute country and company that is configured in the database. When displaying the world map with factory information, the application will search for a png in the flag_country and the logo_company folder that has the same name as the database values.

3.5. Configuring testing devices at runtime

At runtime the status of testing devices can be updated, new devices can be registered and devices can be unregistered by xml files.

The XML file has to be sent via html post to

```
http://<server>:<port>/<applicationname>/audi/input
```

The XML File should have the following structure:

```
Root Element: <tasks> can contain multiple <task> elements
```

In the task element the attribute name specifies which task should be executed. Possible options are register, unregister, update.

The task element contains a <device> element with the where the place and the name of the device is specified

If the task is register, the elements <description>, <type> and <serialnumber> specify the data of the device>

If the task is unregister, no additional information is needed.

If the task is update the <component>s have to be specified and the new data should be included.

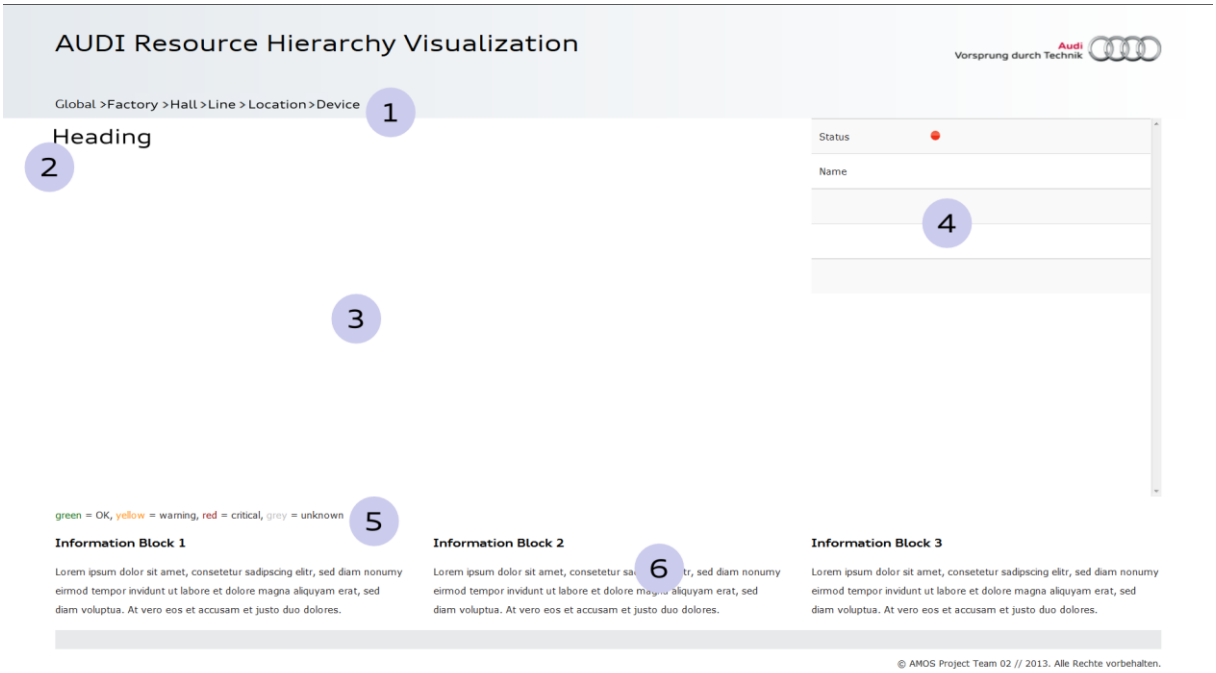
Example:

```
<tasks>
<task name="unregister">
<device factory="Bruessel" hall="C9" line="C9 Finish" location="GD5" name="Test2" >
</device>
</task>

<task name="register">
<device factory="Bruessel" hall="I7" line="I7 Finish" location="AH6" name="Test3" >
    <description> bla</description>
    <type>typ1</type>
    <serialnumber>0183412</serialnumber>
</device>
</task>

<task name="update">
<device factory="Bruessel" hall="I7" line="I7 Finish" location="AH6" name="Test3" >

            <component name="Network">
                <status>green</status>
             <value>10.0.0.1</value>
                <category>A</category>
                <responsibleSector>B</responsibleSector>
            </component>
            <component name="Tests">
                <status>green</status>
             <value>OK</value>
                 <category>A</category>
                <responsibleSector>B</responsibleSector>
            </component>
            <component name="Maintainance">
                <status>yellow</status>
             <value>aa</value>
                  <category>A</category>
                <responsibleSector>B</responsibleSector>
            </component>
</device>
</task>
</tasks>
```

4. User Interface

4.1. General

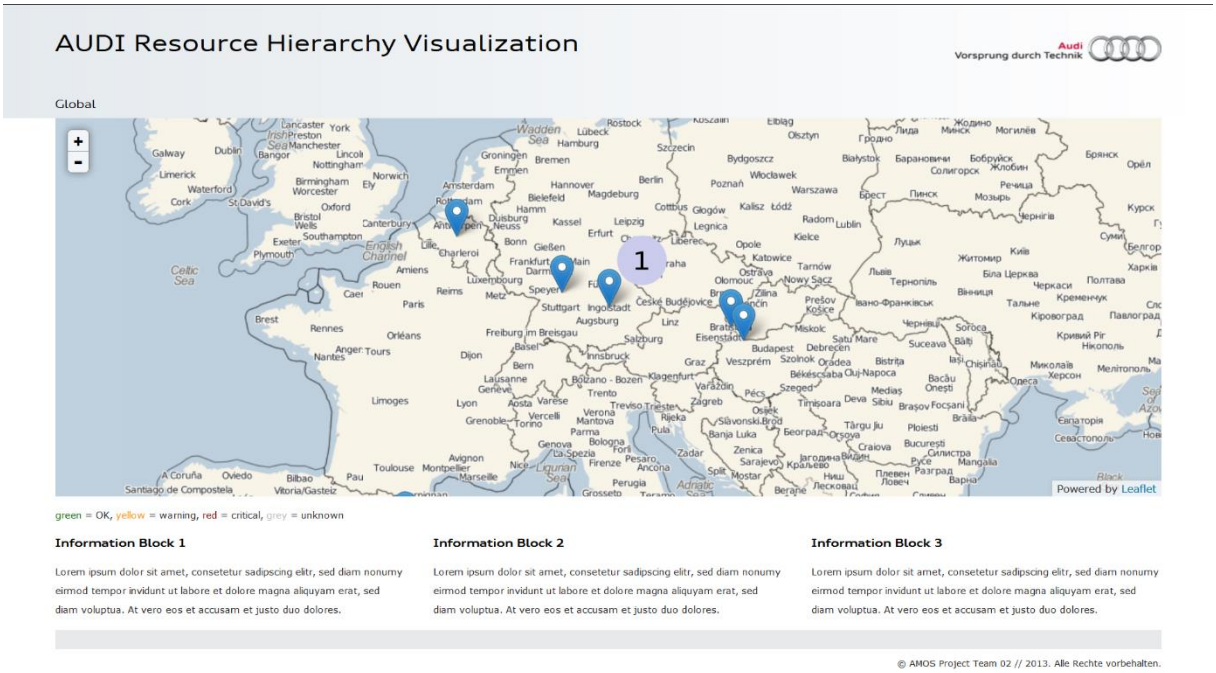
The application is structured in a hierarchical manner. There is for each hierarchy level a screen describing the most important information there. Most of the screens are structured as followed:



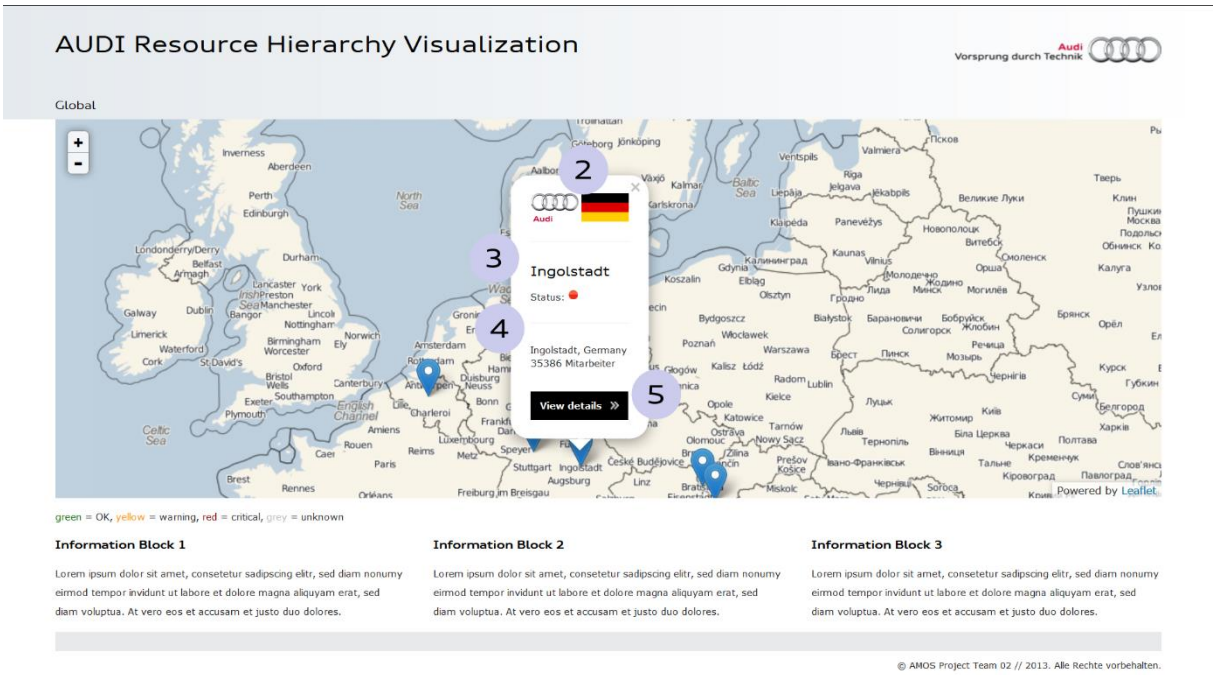
No.	Description
1	**Breadcrumb**: Here, the path taken down the hierarchy is shown. By clicking on entries, going back to the respective higher hierarchy level is achieved.
2	**Heading**: Shows the name of the currently viewed element, preceded by the hierarchy level. For a factory it would say “Factory: Name”.
3	**Nested elements**: While the bottom of the hierarchy is not reached yet, every element has a collection of elements of the next lower hierarchy level. They are listed in the center of the page and can be accessed there. While the factory- and hall view use maps to show where its elements are, the line-, location- and device view employ lists.
4	**Element information**: To the right, the various attributes of the currently viewed element are listed. The status is always listed first.
5	**Key**: Explains what color indicates which status.
6	**Information blocks for future use**: At the bottom there are three text blocks, which are yet to be filled with useful information. Up to now they only hold dummy text.

4.2. Global view

The global view gives an overview over all the factories. The main element is a world map showing the position of every factory with a pin. The information shown is updated every minute.



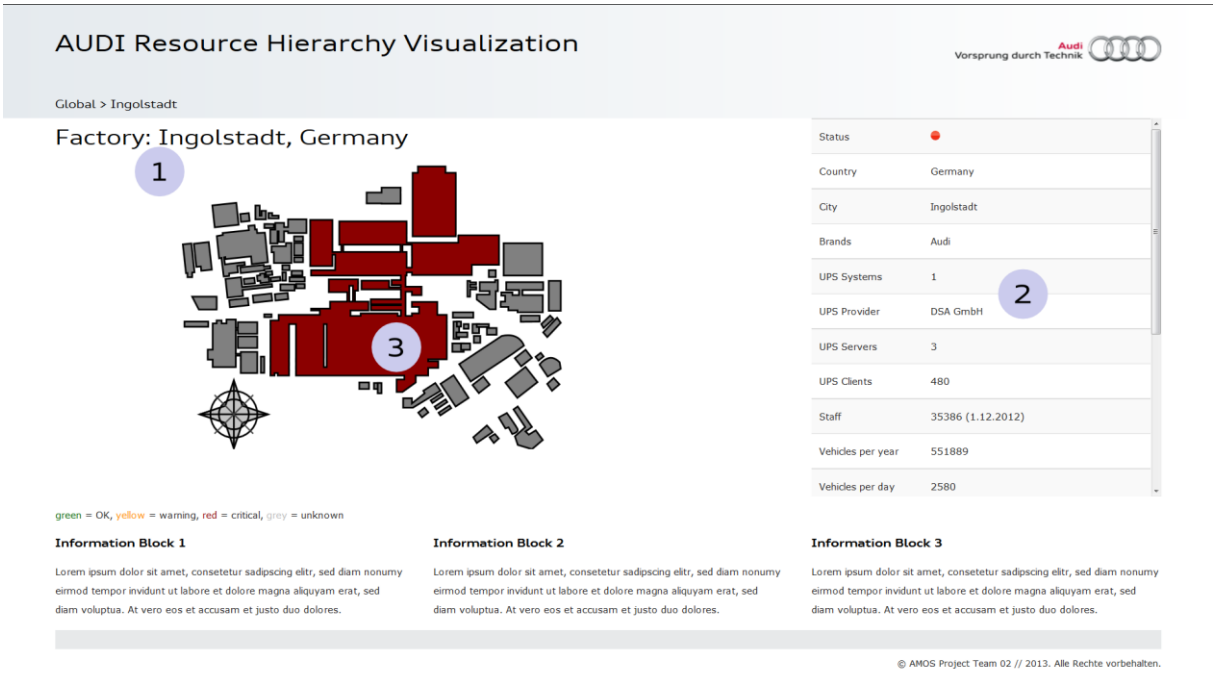
No.	Description
1	**Pins**: Every pin marks the location of a factory. When clicking on a pin detailed information is shown.



No.	Description
2	**Brand and Country:** In the header of the information box there is a picture showing which brand the factory belongs to and the flag of the country the factory is in.
3	**Factory name:** After the header the name of the selected factory is displayed.
4	**General information:** For every factory there is general information that is displayed underneath the name. The current status is shown first, followed by the name of the city it is in and the amount of people working there.
5	**View factory:** At the bottom of the information box there is a button, which leads to the view for the respective factory.

4.3. Factory view

The factory view shows more detailed information about the factory and a plan with the halls colored according to their status.

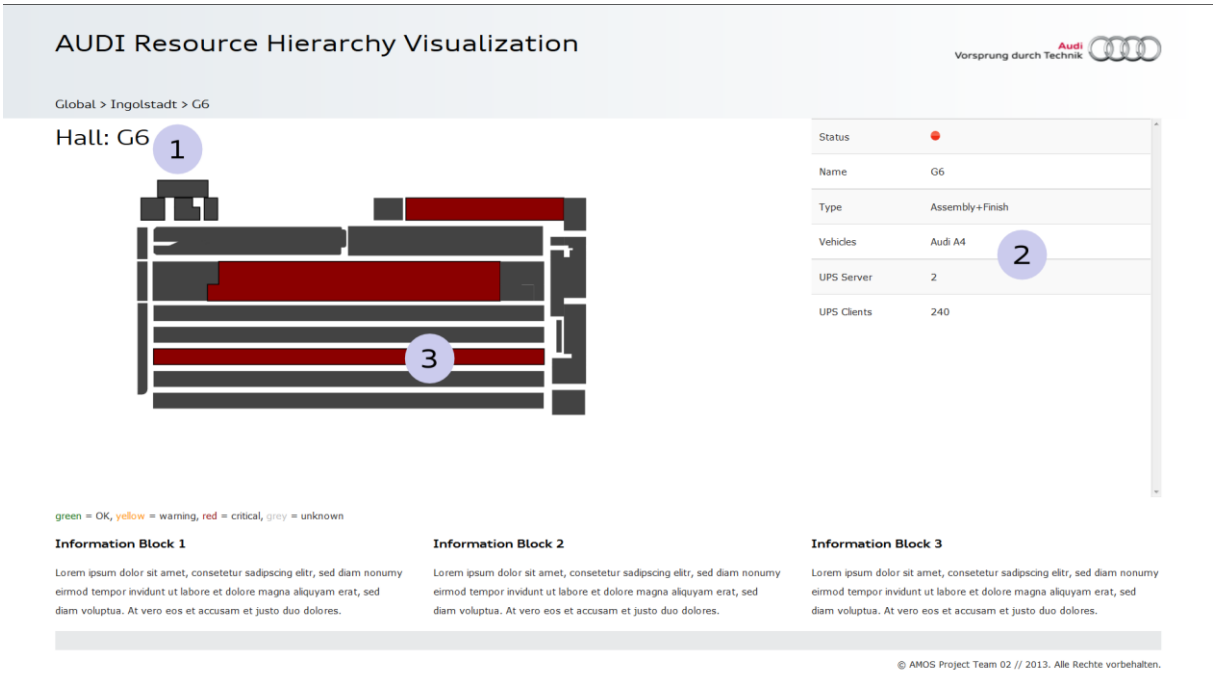


No.	Description
1	**Heading**: The heading shows the name and location of the factory.
2	**Detailed information**: The right there is more detailed information about the factory. To see everything it might be needed to scroll down.
3	**Factory plan**: The plan shows the buildings of the factory. Production halls are colored according to their status. By clicking on a hall the application will change to the hall view displaying more

information for this hall. Other buildings than production halls are not clickable.

4.4. Production Hall view

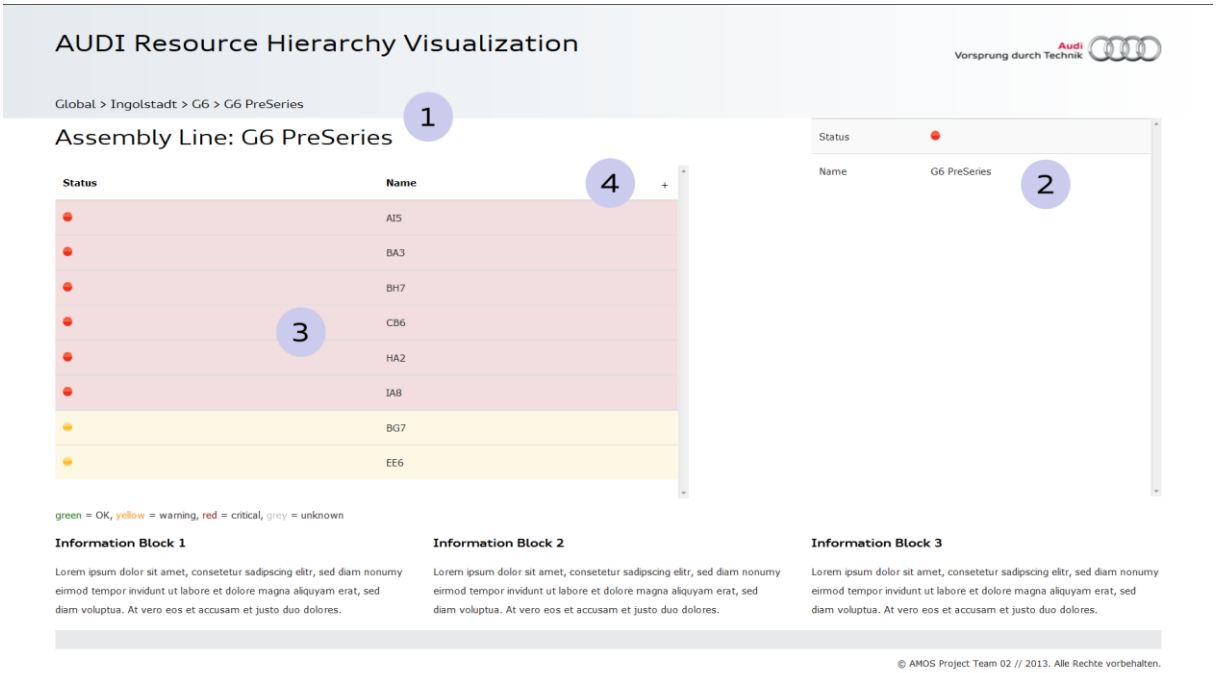
The hall views shows more detailed information about a production hall as well as a plan with the production lines located in the respective hall.



No.	Description
1	**Heading**: The heading shows the name of the current hall.
2	**Detailed information**: To the right there is detailed information about the production hall.
3	**Hall plan**: The plan shows the interior of the production hall. In it there are the production lines colored according to their status. By clicking on a production line the application will change to the line view, displaying more information for the selected line.

4.5. Assembly Line view

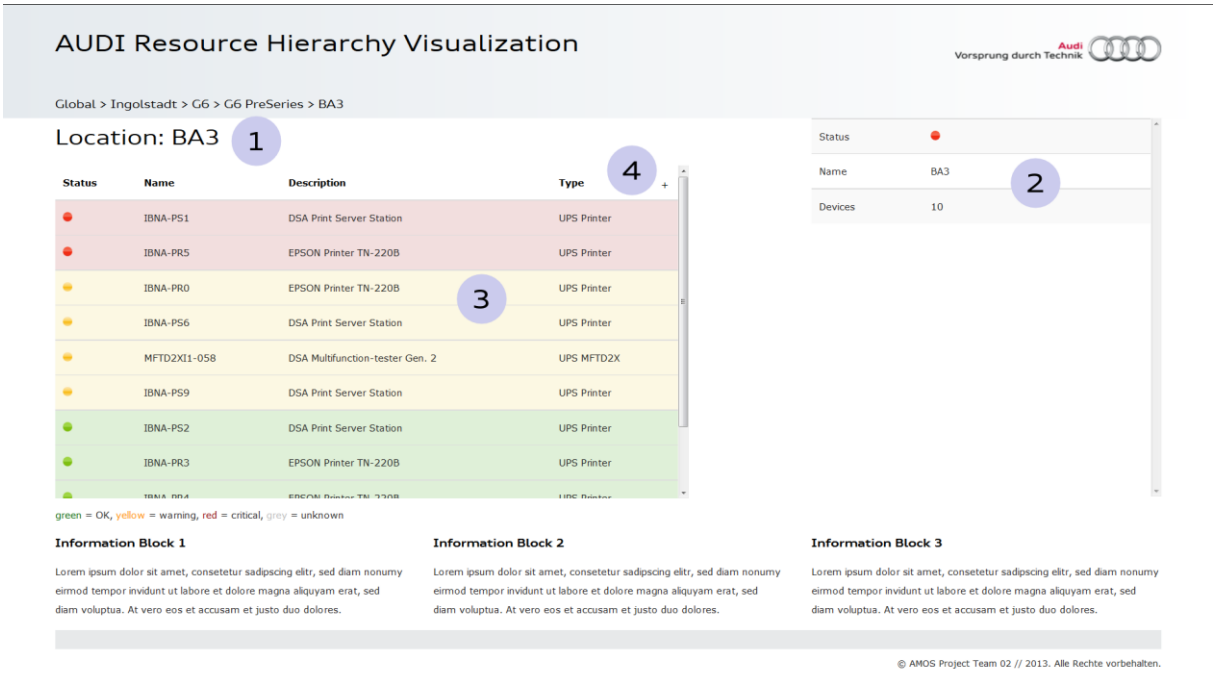
The assembly line view shows more detailed information about a selected assembly line.



No.	Description
1	**Heading**: The heading shows the name of the currently selected assembly line.
2	**Detailed information**: To the right there is detailed information about the assembly line.
3	**Location list**: To the left there is the list of all the locations assigned to the current assembly line. The list is sorted by status, red first, then yellow, green and grey. By clicking on a list item the application changes to the location view for the selected location.
4	**List controls**: By clicking on the column names a menu comes up that allows for sorting and filtering as well as hiding this column. If a column was hidden it can be restored by clicking on the plus sign in the upper right corner of the table.

4.6. Location view

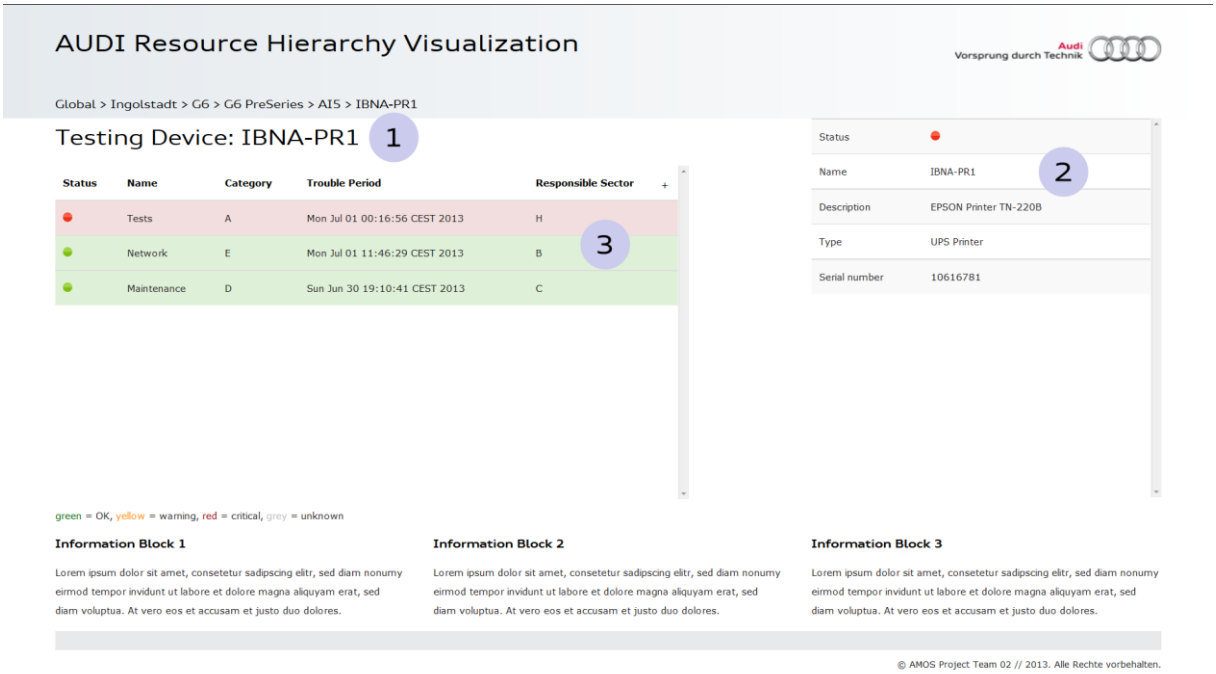
The location view shows more detailed information about a location, located in an assembly line.



No.	Description
1	**Heading**: The heading shows the name of the current location.
2	**Detailed information**: To the right there is detailed information about the location.
3	**Device list**: To the left there is the list of all the devices located at the current location. The list is sorted by status, red first, then yellow, green and grey. By clicking on a list item the application changes to the device view for the selected device.
4	**List controls**: By clicking on the column names a menu comes up that allows for sorting and filtering as well as hiding this column. If a column was hidden it can be restored by clicking on the plus sign in the upper right corner of the table.

4.7. Testing Device View

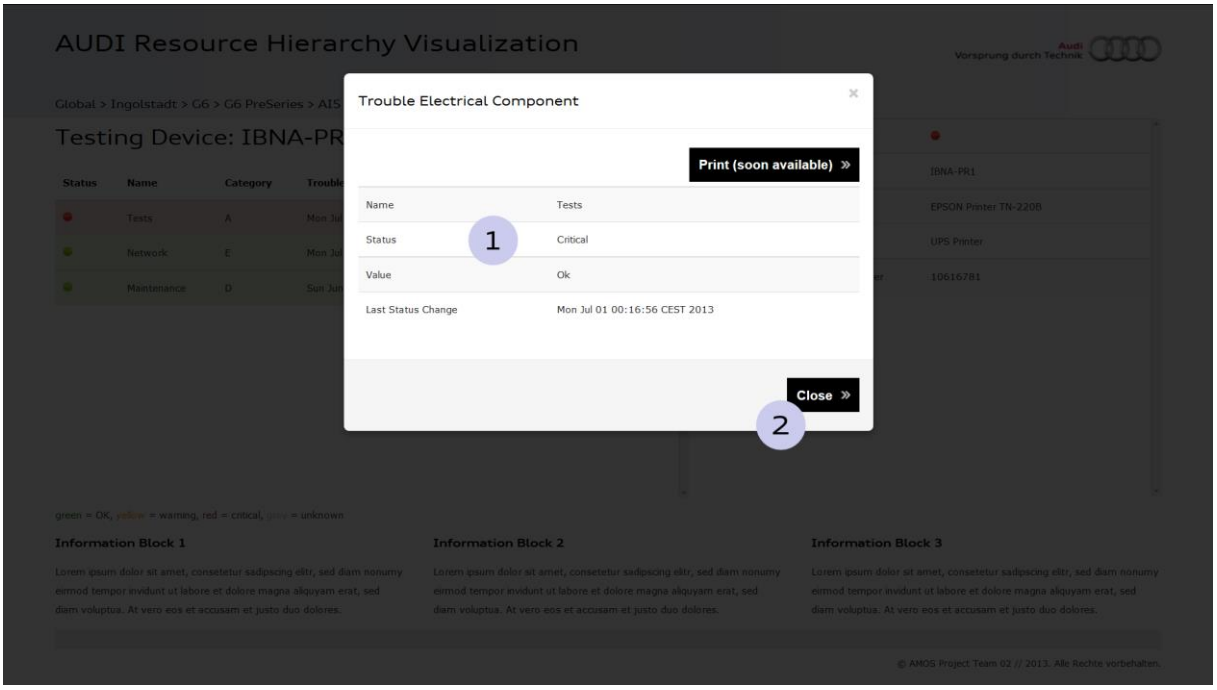
The testing device view shows information about a particular testing device.



No.	Description
1	**Heading**: The heading shows the name of the current testing device.
2	**Detailed information**: To the right there is detailed information about the testing device.
3	**Device list**: To the left there is the list of all the sources, information about the device is gathered from. The list is sorted by status, red first, then yellow, green and grey. By clicking on a list item the application changes to the trouble electrical component view showing further information about that source.

4.8. Trouble Electrical Component view

The trouble electrical component shows information about a device gathered from a specific source.



No.	Description
1	**Detailed information**: The table in the modal box lists all the information for a device gathered from the selected source.
2	**Close button**: The close button closes the modal box, returning to the device view.

5. Closing remarks

We would like to thank all participating roles on this project. The software was created in the frame of the Agile Methods and Open Source (AMOS) project work at the University of Erlangen-Nuremberg, Germany. Information on licensing can be queried from the responsible persons.