Dynamic Configuration of Distributed Systems for Disaster Management

Florian Segor, Igor Tchouchenkov, Rainer Schönbein, Matthias Kollmann, Christian Frey
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
Karlsruhe, Germany
e-mail: {florian.segor, tchouchenkov.igor, rainer.schoenbein, matthias.kollmann, christian.frey}@iosb.fraunhofer.de

Stefan Rilling, Rainer Worst
Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS,
Sankt Augustin, Germany
e-mail: {stefan.rilling, rainer.worst}@iais.fraunhofer.de

***Abstract*—In natural and man-made disasters, it is a necessity for rescue teams to get a quick overview of the situation in place. Robot-supported sensor networks are increasingly used to accelerate surveillance and search operations in complex situations. An appropriate robust system architecture has to support dynamical changes in connectivity as well as in number and type of robots and sensors in action. The proposed solution for a dynamic configuration of a distributed system with heterogeneous sensors and robots for disaster management is based on the Robot Operating System (ROS). The configuration uses an active Information Module with access to the descriptions of the characteristics and capabilities of all relevant system components. The modular descriptions are based on XML standard. Every component has at least one description file with capabilities of the component and their relevant technical characteristics. Descriptions of complex components containing sub-components are hierarchically with references to descriptions of sub parts. Between the system components direct communication links can be established to make the distributed system more robust. External systems may also get information about available capabilities from the Information Module and request needed services directly from the components. The main task of this work is to introduce a dynamic but robust system architecture for controling complex heterogeneous sensor systems to support rescue forces in disaster relive.**

Keywords-Distributed system, disaster management, dynamic configuration, modular, hierarchical, ROS, XML, heterogeneous sensors, robotics

I. INTRODUCTION

Heterogeneous distributed sensor-based surveillance systems have strongly spread during the last years and have grown on account of new demands. A novel concept for such systems for disaster management has been developed by five German Fraunhofer Institutes in the joint research project SENEKA [1][2]. The disaster management in rapidly changing scenarios requires a quick configuration and re-configuration of the system and components with minimal work load on the rescuers. This is a challenge because not only every single component, but normally also the needed communication infrastructure has to be set up and adjusted.

The objective of the SENEKA concept is to network various unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs) and sensor systems usable by first responders. Within the project, possible solutions for a flexible dynamic system reconfiguration during a mission are examined and evaluated [2][3]. In this article, the setup for a dynamic configuration of complex distributed systems for disaster management is described in Section II, as well as a solution concept for automatic configuration with software generation for sensors and sensor modules in Section III. The paper is closed with a final discussion in Section IV.

II. DYNAMIC CONFIGURATION

The concept for the dynamic configuration of SENEKA system is based on a (conceptually redundant) central Information Module (SENEKA-Hub), which has access to data describing the characteristics and capabilities of all relevant system components (see Figure 1). The SENEKA-Hub must always know which components are online, their status and what capabilities and functionalities they can provide.

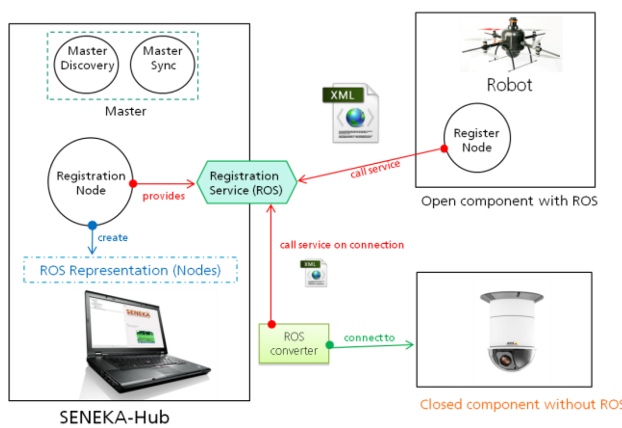


Figure 1: A sample structure of dynamic configurable SENEKA system with heterogeneous components.

To guarantee the automatic “technical” integration of the components in the overall system, the SENEKA-Hub must also “know” relevant technical details about the integration parameters of the abilities (network-addresses, interfaces, protocols, etc.) of all serving components to notify all interested information consumers (human users and technical components or systems). The communication between the

 ISBN: 978-1-61208-399-5

system components as well as to external systems is handled not over the SENEKA-Hub, but directly. This makes the system much more robust – especially under poor communication channel quality in disaster areas.

The implementation of such communication architecture is based on the Robot Operating System (ROS) [4]. It is an Open Source project for distributed control systems. For fast communication a Publisher-Subscriber architecture supported by services is used.

The SENEKA-Hub is acting as a ROS Master and collects information about all connected system components. Four types of components can be used:

1) "Open" components with ROS can be programmed to add functions needed for dynamic configuration.

2) "Open" components without ROS have to be adapted (e.g., programmed) to add ROS interfaces and functions needed for dynamic configuration.

3) "Closed" components with ROS supporting some ROS capabilities, but can't be adapted.

4) "Closed" components without ROS and no ROS support that provide other interfaces only (for example a Web-Cam).

Different types of components must be handled in different ways. "Open" components can inform the SENEKA-Hub after their suitable adaptation when connecting to the system or disconnecting from it, providing their own descriptions and report about changing capabilities, if needed. "Closed" components must be "installed" by introducing their abilities to the SENEKA-Hub before the first usage. During registration, "closed" components with ROS only need to transfer their "call sign". The registration of other components must be ensured on the other suitable way – for example, using other "open" components (e.g., ground control station like [5]) to check if an "installed" component is now alive. The "intelligent" components should normally configure themselves. The special "Configurator" program can set parameters of different components and infrastructure (e.g., computers) as needed.

The central problem of such hierarchical heterogeneous systems is the determination of consistent components' descriptions at different hierarchy levels. The components and their interfaces have very different complexity – from easy temperature sensors with RS232 up to distributed command and control centres equipped with many sensors, sensor carriers, servers and workplaces, but all their descriptions must be standardized. Moreover, the same sub-components can be used many times in different parts of the same system or in different systems with variable parameters, so that the descriptions should be easy adaptable. For example, SENEKA contains multiple mobile sensors mounted on light UAVs or UGVs as well as stationary sensors controlled over an ergonomic user interface from an improved AMFIS ground control station [5][6][7] (Figure 2). After the registration of UAVs and UGVs, different control and data channels for direct communication with the control station or with other components can be used as needed.

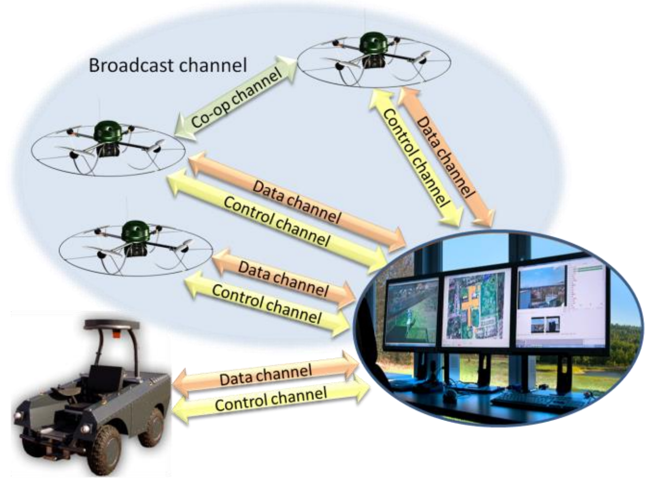
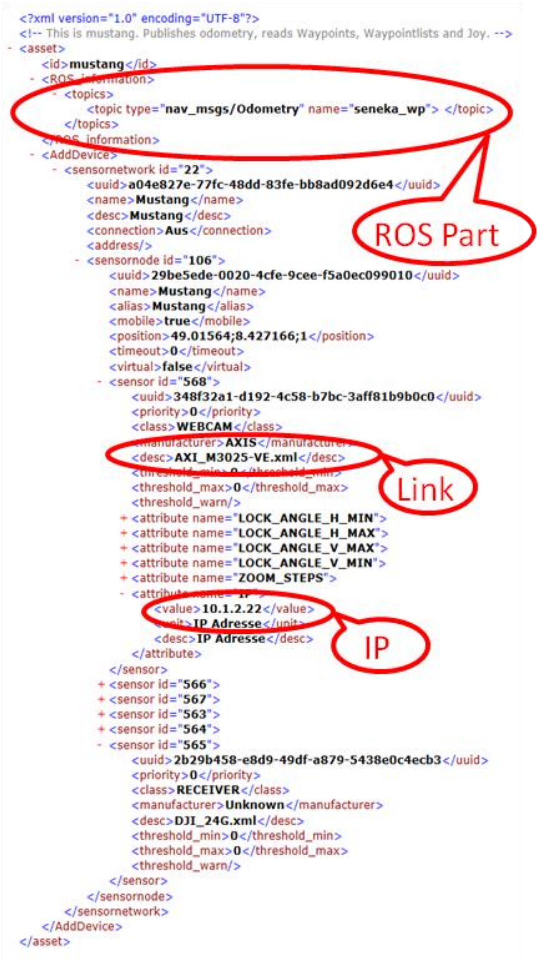


Figure 2: SENEKA sensor network with AMFIS ground control station and mobile sensor carriers (UGV "Mustang" and UAV AirRobot).

The description used is modular and based on XML standard. Every component must have at least one description file that contains two parts: 1) capabilities of the component and 2) their relevant technical characteristics. On account of this information, other subsystems can interoperate directly with this component and use its propagated functions and abilities. The "Configurator" program can also set up the component, if needed. For complex components containing many sub-components and abilities, these descriptions are organized hierarchically and contain references to descriptions of sub parts, e.g., cameras or communication devices. On this occasion, the research about the necessary level of details, which permits an automatic configuration using the descriptions of the single components, is of central relevance. Is the description level too coarse, functions might not be correctly used or actions will lead to unexpected or undesirable reactions. On the other hand, a too detailed description produces abundance in information and needs a lot of expenditure in the generation of the descriptive data.

To explain the configuration concept, a very easy SENEKA system containing only SENEKA-Hub, ground control station and one UGV "Mustang" will be considered. Both, control station and Mustang, are open components and have XML descriptions which can be sent to SENEKA-Hub using the Registration Service in Figure 1. In Figure 3, a very simplified description of UGV Mustang is shown. The UGV contains many different sensors (laser scanner, cameras, magnetometer, GPS, etc.) and WLAN as control/data channel. It can contain additional cameras with additional up/downlinks, so that the actual description file should be sent by each registration.




```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- This is mustang. Publishes odometry, reads Waypoints, Waypointlists and Joy. -->
- <asset>
    <id>mustang</id>
  - <ROS_information>
    - <topics>
        <topic type="nav_msgs/Odometry" name="seneka_wp"> </topic>
      </topics>
    </ROS_information>
  - <AddDevice>
    - <sensornetwork id="22">
        <uuid>a04e827e-77fc-48dd-83fe-bb8ad092d6e4</uuid>
        <name>Mustang</name>
        <desc>Mustang</desc>
        <connection>Aus</connection>
        <address/>
      - <sensornode id="106">
          <uuid>29be5ede-0020-4cfe-9cee-f5a0ec099010</uuid>
          <name>Mustang</name>
          <alias>Mustang</alias>
          <mobile>true</mobile>
          <position>49.01564;8.427166;1</position>
          <timeout>0</timeout>
          <virtual>false</virtual>
        - <sensor id="568">
            <uuid>348f32a1-d192-4c58-b7bc-3aff81b9b0c0</uuid>
            <priority>0</priority>
            <class>WEBCAM</class>
            <manufacturer>AXIS</manufacturer>
            <desc>AXI_M3025-VE.xml</desc>
            <threshold_min>0</threshold_min>
            <threshold_max>0</threshold_max>
            <threshold_warn/>
          + <attribute name="LOCK_ANGLE_H_MIN">
          + <attribute name="LOCK_ANGLE_H_MAX">
          + <attribute name="LOCK_ANGLE_V_MAX">
          + <attribute name="LOCK_ANGLE_V_MIN">
          + <attribute name="ZOOM_STEPS">
          - <attribute name="IP">
              <value>10.1.2.22</value>
              <unit>IP Adresse</unit>
              <desc>IP Adresse</desc>
            </attribute>
          </sensor>
        + <sensor id="566">
        + <sensor id="567">
        + <sensor id="563">
        + <sensor id="564">
        - <sensor id="565">
            <uuid>2b29b458-e8d9-49df-a879-5438e0c4ecb3</uuid>
            <priority>0</priority>
            <class>RECEIVER</class>
            <manufacturer>Unknown</manufacturer>
            <desc>DJI_24G.xml</desc>
            <threshold_min>0</threshold_min>
            <threshold_max>0</threshold_max>
            <threshold_warn/>
          </sensor>
        </sensornode>
      </sensornetwork>
    </AddDevice>
</asset>
```

Figure 3: Simplified XML description of UGV “Mustang” for SENEKA-Hub.

The detailed descriptions of sub-components are in linked XML files (AXI_M3025-VE.xml for camera and DJI_24G.xml for additional wireless video link), so that the main file is not very complex and contains the information at the appropriate hierarchy level only. The UGV receives movement commands, waypoints or waypoint lists and can deliver a video stream from the on board camera – only capabilities on this level are needed for the ground control station. The files for mobile components can contain start coordinates, if needed (<position> in Figure 3), but actual coordinates will be reported dynamically.

The same AXI cameras can be also used as component of SENEKA on the same level as the UGV. Because their IP-addresses can be different, they must be configured. The IP-address is set with parameter “value” in the main file. The camera position is the same as position of UGV – that’s why the parameter <mobile> is “false”. The simplified description of the camera is shown in Figure 4.

```xml
<?xml version="1.0"?>
- <item>
    <name>AXICam</name>
    <category>sensor</category>
    <type>Camera</type>
    <mobile>false</mobile>
    <rotable>true</rotable>
  - <capability>
    - <input>
      - <rotation_command>
          <host port="4475" IP="value"/>
        </rotation_command>
      </input>
    - <output>
      - <video>
          <host port="3740" IP="value" request="axi-cgi/mjpg/video.cgi"/>
        </video>
      </output>
    </capability>
</item>
```

Figure 4: XML description of AXI camera (simplified).

The camera receives rotation commands on port 4775 and can deliver video streams from IP “value” (set to “10.1.2.22” in Figure 3) by setting the next request on port 3740: “AXI-cgi/mjpg/video.cgi”.

To ensure correct configuration, the XML description must contain all information needed to set up wired and wireless communication. After configuration, system components communicate directly, as needed, on the basis of the information from the SENEKA-Hub – mostly with the ground control station.

External systems (for example, command centres) can get information from the SENEKA-Hub about available capabilities and request needed services directly by the components. They have to implement the access to information describing the capabilities available from the ROS SENEKA-Hub – this service has to follow the standard with appropriate descriptions.

III. AUTOMATIC CONFIGURATION CONCEPT

A particular challenge is the specification of modules’ descriptions and reusable software supporting automatic configuration in complex heterogeneous distributed systems. To ensure a generic solution, suitable classifications of all input and output devices as far as software modules are needed. Output devices can be classified on values that they can represent, input devices on generated values. A sample of the sensor classification according to type of their information is shown in Table 1.

TABLE 1: SAMPLE OF SENSORS CLASSIFICATION ACCORDING TO SENSOR OUTPUT.

Dimension	Direction dependent	Sensors/Values
0	No	Temperature, pressure, gas, motion detector, non-directional microphone, voltage,
0	Yes	Distance, velocity, acceleration, force
1	Yes	Line scanning camera, microphone array
2	Yes	Video/IR camera, 2D laser scanner
3	Yes	Radar, 3D laser scanner, PMD, 3D Video/IR camera

 ISBN: 978-1-61208-399-5

To support automatic configuration, the classification of each sensor group must be expanded with possible types of information compression (for camera: uncompressed, MJPEG, MPEG2, MPEG4, H264, etc.), frame rate, resolution and interfaces' classification according to standard/protocol (both input and output, if suitable) like shown in Table 2.

TABLE 2: SAMPLE OF (SIMPLIFIED) 2D SENSORS CLASSIFICATION ACCORDING TO STANDARD/PROTOCOL.

Channel	**Wired**						**Wireless**		
Analogue	RCA						Downlink		
Digital	USB		Ethernet				Digital Downlink	WLAN	
	2.0	3.0	UDP	HTTP	RTSP	RTP	...	...	...

Sensor carriers' classification takes into account all possible movement types (position changing/direction changing/combination, degree of freedoms, etc.), but also possible movement control: position, velocity or acceleration for each degree of freedom and their limits; absolute/relative values; correlations between coordinates etc.

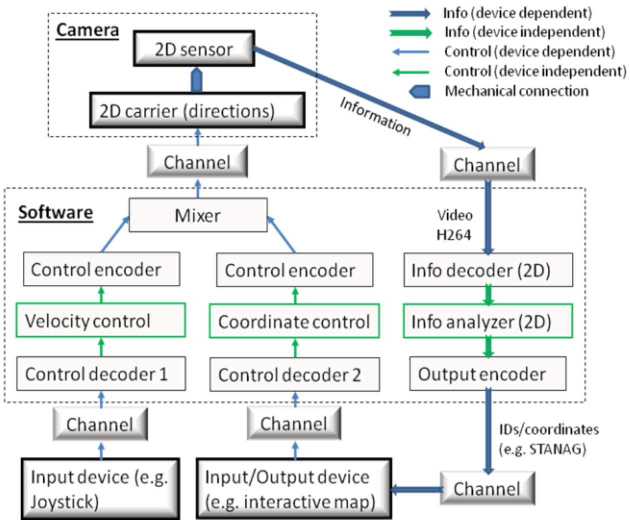


Figure 5: Automatic configurable information analyzing and control system for pan-tilt camera.

Based on the classification, parent classes for each type of components and standard interfaces can be developed. These classes must be device-independent. Device-dependent capabilities can be implemented with decoders and encoders for information, mixers and separators to prepare information for further analysis and transfer. All software modules must be also classified and suitable described with XML. A (simplified) sample of an automatic configurable modular control and information analyzing pan-tilt camera system is shown in Figure 5. Green modules are device-independent.

IV. CONCLUSION

In this article, a solution for dynamic configuration of distributed heterogeneous systems for disaster management has been described. The solution is based on standardized XML descriptions of components containing technical and non-technical details. This concerns a long-term installation, set up and restarting for reconfiguration, especially important in quickly alterable disaster situations where systems must switch fast between certain configurations for different duties or change configuration by exchanging or adding new components.

A sample of an automatic configurable pan-tilt camera control system based on devices' classification and of modular software architecture illustrates the concept of automatic configuration.

To implement adaptive automatic configuration with easy integration of new components and software generation, basic software modules must be defined and implemented. For the automatic software generation, the software has to be described by XML descriptions.

ACKNOWLEDGMENT

The authors would like to thank their colleagues and students, who have contributed to the work presented in this paper.

REFERENCES

[1] H.-B. Kuntze et al., "SENEKA - Sensor Network with Mobile Robots for Disaster Management," The twelfth annual IEEE Conference on Technologies for Homeland Security (HST 2012), Nov. 2012, pp. 406-410, doi:10.1109/THS.2012.6459883.

[2] H.-B. Kuntze et al., "Situation responsive networking of mobile robots for disaster management," Joint Conference of 45th International Symposium on Robotics and 8th German Conference on Robotics (ISR/Robotik 2014), June 2014, pp. 313-320, ISBN:978-3-8007-3601-0.

[3] I. Tchouchenkov, R. Schönbein, F. Segor and M. Kollmann, "Dynamic and Automatic Configuration of Distributed Heterogeneous Surveillance Systems," The 9th Security Research Conference (Future Security 2014), Sept 2014, pp. 164-171, ISBN: 978-3-8396-0778-7.

[4] Robot Operating System. [Online]. Available from: http://wiki.ros.org/ 2015.01.09

[5] A. Bürkle, F. Segor, M. Kollmann and R. Schönbein, "Universal Ground Control Station for Heterogeneous Sensors," Journal On Advances in Telecommunications, IARIA, Volume 3, Numbers 3 & 4, pp. 152 – 161, 2011.

[6] F. Segor, A. Bürkle, M. Kollmann and R. Schönbein, "Instantaneous Autonomous Aerial Reconnaissance for Civil Applications - A UAV based approach to support security and rescue forces," The 6th International Conference on Systems (ICONS 2011), Jan 2011, pp. 72-76, 2011, ISBN: 978-1-61208-002-4.

[7] E. Santamaria, F. Segor, I. Tchouchenkov and R. Schönbein, "Path Planning for Rapid Aerial Mapping with Unmanned Aircraft Systems," The 8th International Conference on Systems (ICONS 2013), Feb 2013, pp. 82-87, ISBN: 978-1-61208-246-2.