Data Acquisition and Processing using a SOA for an automated Inspection System

Keywords: autonomous systems, distributed data acquisition, middleware, CORBA, sewer inspection.

1. INTRODUCTION

Sewer channels are the foundation of public health. Therefore legal guidelines require the inspection of sewer lines to assure their operation. Typically this is accomplished with remote-controlled vehicles equipped with a single TV-camera [1]. Advanced systems include additional sensors or acquire data digitally [2]. As these sewer inspection robots become more complex, computerized data-processing and -acquisition with adequate software support on the middleware layer becomes more important.

The rest of this paper is organized as follows. First, we introduce our inspection robot. We discuss the various sensor systems briefly and characterize the data they produce. Then we motivate the need for a distributed acquisition and post-processing system. Next, we describe a Service Oriented Architecture (SOA) that is specifically tailored for data processing in robotics applications. Finally, we show how the SOA was used to integrate a number of data sources, data processing modules and data consumers. Special attention is paid on supporting the needs of data-processing with multiple stages and sensor-data fusion as well as on the behavior of the system under transient overload conditions.

2. INSPECTION SYSTEMS FOR PARTIALLY FILLED SEWER PIPES

The Emschergenossenschaft based in Germany is currently planning the Emscher sewer system, arguably the largest residential water management project in Europe in years to come. One particularity of this sewer system is its partial filling, 25% at minimum, all the time. Thus walk-through inspection by personnel or conventional inspection robots is impossible in every phase [4]. To detect damages under these difficult conditions, a swimming cable-guided robot has been developed. This robot is equipped with several sensors including laser ranging sensors, cameras, a pan-tilt-zoom video camera and an ultrasound scanner. The data acquired with these sensors is send to a service vehicle where data processing and fusion takes place. The result of the damage detection is presented to a service operator on the car with small delay. Additionally the sensor data can be processed later in the office with more complex algorithms. The huge amounts of data, the high demands on the accuracy of damage detection and the need for fast data fusion in ‘live’ mode necessitate special data processing techniques.

3. SERVICE ORIENTED ARCHITECTURE FOR DATA PROCESSING

We developed a service oriented CORBA-based architecture focusing on data flow and distributed processing. All services implement a generic interface and may be arbitrarily distributed in a network. The interface of the services is data centric. It offers data buffers to other services. Each buffer has a user defined id and represents a continuous time series of specific data objects as shown in Figure 1.

Each service represents a data source (sensor), a processing module or a data consumer (actor, database, GUI). By connecting the services via the publish/subscribe paradigm the data flow of the application is modelled. In this communication model the services play different roles. The module providing data is called the service provider (or briefly service). The data consumer is called the service client (or briefly client).

A client can request individual data objects or arbitrary blocks of data from a buffer. This is achieved by sending structured queries with a simple syntax to the accordant service. The client can then process these data items and offer the results to further clients.

The described architecture has been implemented as a framework in C++. Among many base classes for data handling and communication, we implemented several base services for common problems like configuration or logging. The most important base service is a TimeService for synchronizing the clocks of different computers. The TimeService uses conventional techniques to calculate the offsets between system clocks with an accuracy of <500µs. This accuracy is sufficient to fuse data items acquired on different hardware nodes.

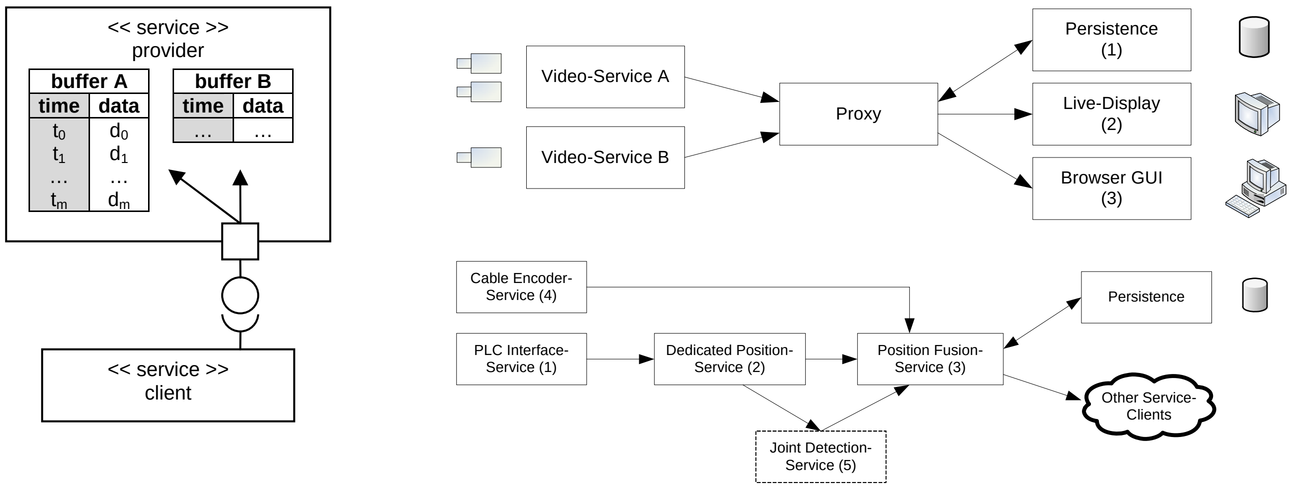

Figure 1: Concept of the service architecture (left) and two settings utilizing the architecture (right)

4. MODELLING DATA FLOW USING SERVICES

The following section is concerned with the realization of two of the software subsystems of the inspection system. The subsystems are briefly introduced. After that will discuss in detail, how these subsystems were organized using distributed services for data acquisition and data processing, how these services interact with each other and how time constraints as well as transient overload of processing stages or communication links are managed.

Digital video subsystem:

The video subsystem consists of multiple cameras at the robot and at the infrastructure that are used for supervision of the inspection process as well as for manual examination of damages. The cameras are attached to two host systems where the video is digitized, compressed and time-stamped and then made available to our distributed system by means of a software service (see the upper right diagram in Figure 1). We choose MJPEG as the compression scheme as it offers good visual quality and produces frame sequences with no inter-frame dependencies, which is beneficial when browsing within the recorded material. Each frame sequence is assigned an observable buffer queue within each video-service. Clients can register for notifications when new frames become available. In the next stage, we deployed a generic service which runs on a node of the computer cluster on board the service vehicle and collects frames from all buffers of the two video-services. Therefore it acts like a proxy that distributes data to other clients. These other clients are: (1) a service that writes frames to disk in a format that is compatible with regular AVI-video files and that offers random access to all persisted data elements. (2) A program, that displays one or more of the frame sequences on a monitor as a live video. And (3) a software, that allows the operator to browse within all collected data.

Position determination and navigation subsystem:

Knowledge of the position is not only required for steering, but also to process and interpret other sensors data. This includes the registration of the images of the photographic camera system, the determination of the magnitude of corrosion and incrustations, or the precise measurement of possible deviation of the position of individual pipe segments. Finally, a seamless image of the whole sewer has to be build and presented to the user. This is also not possible without knowledge of each sensors pose during data acquisition. While only a rough estimation of the systems distance to the walls is required for automatic steering, we need a more precise idea for sensor data processing.

In the lower right part of Figure 1 we show an overview of the services necessary for position determination and how they are arranged. Position is calculated using an algorithm that was specifically developed for this purpose. It is based on the measurements of 15 dedicated laser ranging sensors. See [9] for a discussion on the exact procedure. Please note that the final determination of the position depends also on the result of a service that performs image processing and pattern recognition on the images from the photographic camera system.

5. CONCLUSION

Here we motivated the use of multiple sensors or sensor systems in conjunction with an automated inspection robot for large sewer pipes. With a greater amount of raw data comes the need for intelligent data processing, fusion and visualization in order to present a comprehensive picture of the pipes condition to the operator.

During the development of the inspection system it became evident, that our architecture provided a flexible yet well structured basis for the construction of the complex software system that was necessary for robot in order to perform its tasks. We look forward to applying this scheme to related applications.

6. REFERENCES

[1] Mueller, K.: Strategien zur Zustandserfassung von Kanalisationen. Dissertation; In: Aachener Schriften zur Stadtentwaesserung, Band 7, Institut fuer Siedlungswasserwirtschaft der RWTH Aachen, 2006

[2] Elkmann N., Reimann B., Schulenburg E., Althoff H.: Automated inspection system for large underground concrete pipes under operating conditions; In: International Conference on Field and Service Robotics 2005, p. 567

[3] Frey C. W., Kuntze H.-B.: Neuro-Fuzzy basierte Sensorfusion zur qualifizierten Schadensdiagnose an Abwasserkanaelen. Neuro-Fuzzy Based Multisensor Diagnosis Concept for Qualified Condition Assessment in Sewer Systems; In: tm Technisches Messen 70, p. 386-397, 2003

[4] Althoff H., Martini G., Gruen E., Stemplewski J.: The new Emscher: sewer – 51 km underground sewer for 2.4 million inhabitants in a densely populated area; In: Proceedings of the 24th International NO-DIG Conference, 2006

[5] Kaiser J., Brudna C., Mitidieri C.: COSMIC: A real-time event-based middleware for the CAN-bus; In: Journal of Systems and Software, Volume 77, Issue 1 (July 2005) Special issue: Parallel and distributed real-time systems, p. 27-36, 2005

[6] Gerkey B. P., Vanghan R. T., Howard A.: The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems; In: Proceedings of the International Conference on Advanced Robotics (ICAR'03), p. 317-323, 2003.

[7] Noriaki ando, Takashi suehiro, Kosei kitagaki, Tetsuo kotoku, Woo-Keun Yoon: RT-Middleware. Distributed Component Middleware for RT (Robot Technology); In: International Conference on Intelligent Robots and Systems (IROS'05), p. 3555-3560, 2005

[8] Sziebig G., Gaudia A., Korondi P.: Video Image Processing System for RT-Middleware; In: 7th International Symposium of Hungarian Researchers on Computational Intelligence, 2006

[9] Elkmann N., Kutzner S., Saenz J., Reimann B., Schultke F., Althoff H.: Fully Automatic Inspection Systems for Large Underground Concrete Pipes Partially Filled with Wastewater; In: International Conference Intelligent Robots and Systems (IROS'06), p. 4234-4238, 2006