ADVANCED SOFTWARE ARCHITECTURE
OF AN AUTOMATIC VEHICLE NUMBER PLATE RECOGNITION SYSTEM

A. Atanassov

University of Chemical Technology and Metallurgy
8 Kl. Ohridski, 1756 Sofia, Bulgaria
E-mail: naso@uctm.edu

Received 09 November 2011
Accepted 10 January 2012

ABSTRACT

The paper presents the selection and development of a flexible software architecture intended to an Automatic Number Plate Recognition (ANPR) System. The capabilities of the available on the market competitors’ ANPR systems were analyzed and the requirements to the new generation of ANPR system were recognized and/or formulated. Different software architectures, matching the requirements, were analyzed and the WEB-based one was selected. The appropriate software components corresponding to the requirements were identified. The components were developed using the process-oriented approach based on the Theory of the Communicating Sequential Processes (CSP). All ANPR components were decomposed to the CSP processes exchanging specific messages via channels. Finally, these processes were mapped to the operating system’s processes and threads communicating in parallel. As a result an advanced reconfigurable ANPR System minimizing the execution time was provided to the customers.

Keywords: ANPR system, WEB based UI, traffic control software, CSP.

INTRODUCTION

The automatic number plate recognition (ANPR) systems become of great importance for on-line traffic control. There are different types of ANPR systems deployed in special scopes: traffic control, traffic violation, control of parking lots, highway lots, control of the access to cities centers, etc. Those systems, in general, are based on the Charge-Coupled Device (CCD) infrared or color cameras, radars (optional), some digital inputs and outputs, Optical Character Recognition (OCR) software for number plate recognition, Data Bases (DB) and Graphical User Interface (GUI) for interaction to the operators. Usually, the process (Fig. 1) of number plate recognition [1, 2] includes:

• getting image stream from the camera and sending it to the operator PC;
• converting the image stream to the sequence of images in appropriate image format (TIFF, GIF, JPG, BMP) used by the OCR software. It is done by SW for image handling (IH);
• recognition or not of the number plate via OCR [3];
• verification of the found plate number using plates data in DB;
• and visualizing the found number (and car image, image stream) to the operator.
• In some cases sending some data or signals to the digital outputs or reading additional data from the inputs.

Most of the existing ANPR systems [4] are with specific purposes corresponding to one or more of mentioned above scopes. In most of them the camera hardware and the computer(s) running the software (OCR, GUI and DB) are placed on different places and the connection between them is based on coaxial or optical cables. A great majority of systems are still using this architecture. This is no robust architecture because if the PC controlling the camera fails the whole ANPR system fails. It requires complex installation and start-up. It is necessary to install video and control wire for each one of the cameras, furthermore, it is necessary to provide power supply cable to the cameras. If the distance between the lanes is too long, the signal of the
Software Requirements to ANPR Systems

After intensive investigations and analysis of the leading ANPR systems vendors Tattile, Vitronic [6], Leutron Vision [7], Hi-Tech Solutions [8], etc. the general and specific functional requirements valid to all contemporary ANPR systems were defined.

General requirements:

- Online capturing of the images of the vehicles' number plates.
- Online optical character recognition (OCR) of the captured vehicles' number plates.
- Providing an operator control of the observed by the ANPR system object, including:
  - Supervising the ANPR live video-stream of the controlled road, lane, etc.
  - Visualization of the found number plates
  - Support of databases with information of the vehicle plate numbers (Black and White lists) and providing the search in DB in order to find if there is a hit (match) of the found plate number in the White/Black list.
- Collecting and sending of plain or encrypted information (evidential records (ER)) of each plate number to the specific database servers via internet or GPRS.
- Configuration of the ANPR camera parameters, as brightness, gain, frames per second, shutter, position, angles, etc. Tuning of OCR parameters.

Other ANPR systems are based on cameras with software that implements some IH and OCR procedures. Last generation of ANPR systems (Fig. 2) are embedded systems that incorporate in one body the camera (infra-red, color or both), the illumination and the PC hardware with external interfaces, power supply, the operating system, IH, OCR and communication SW. They are known as all-in-one ANPR Systems. These systems are capable immediately to inform other external systems or interested organization for vehicles data.

The advantages of this architecture are that it is simpler [5] one and all the necessary elements for the ANPR system are integrated in the same housing. The equipment may be connected by Ethernet or serial communication with the client application. It is modular architecture and if an ANPR system, controlling one road lane, does not work, its fall does not affect to the other lanes. Installation and start-up is easier. It is only necessary to provide 220v, Ethernet network or serial communication to each embedded system. The installation and the maintenance of the ANPR system are reduced.

Current paper presents the new flexible architecture of an embedded ANPR system. The system is built by set of Operating System (OS) processes related to the required ANPR functionalities. The proposed architecture is developed on the base of theory of the Communicating Sequential Processes (CSP). The ANPR system provides WEB-based GUI component for remote (Internet) operative control, adjustment and reconfiguration of all HW and SW components forming the ANPR system.
• Restart of the ANPR application or Reboot of the operating system.
• Authentication of logging ANPR users.
• Reporting the status of entire ANPR systems (its specific hardware devices or software components) to the operator.

Specific requirements:
• Configuration of the interfaces to the external systems (DB servers, GPRS, etc.)
• Adjustment of serial lines (COM ports) in order to control parking or highway lots
• Download and upload of configuration files or software updates.
• Support of configurable rules and actions related to found plate numbers, meaning how and where the ERs should be saved or which external system should be triggered.
• Collecting of statistical data (recognized/not recognized numbers per hour, day).
• Collecting of logging and tracing data.
• Support of recording of test decks with live-video images used for further adjustment of the OCR algorithm.

IDENTIFYING CSP PROCESSES

The CSP theory [9] deals with processes that are communicating events or messages over channels. The processes interact with each other sequentially, in parallel or alternatively. Some processes can be decomposed to other (sub) processes and channels and etc. The decomposition approach leads to refining of the inter-processes relations and communications. Each process implements its own internal algorithm which is not subject of the CSP theory.

The application of the process-oriented approach into software development helps to map easily the CSP processes to the OS processes or threads. Mentioned CSP approach decreases the execution time of the implemented software, and as well improves the development of the flexible (open) architecture solutions.

After the analysis of the requirements mentioned above, and taking into account the results achieved in [10] the ANPR system can be presented as one general CSP process composed by the following (Fig 3.) processes:

- **Image Handler (IH) Process** is responsible for getting raw live images from the IR or color cameras and transforming them into GIF, TIF or JPEG formats used by the OCR process. The IH also redirects non-transformed images to ANPR Manager Process.

- **The OCR process** produces the results: – finds plate numbers of the observed vehicles. These results are directed to ANPR Manager Process for further handling.

- **The ANPR Manager Process** (AM) transforms the live images to the image formats used by the Web Server Process and sends them to it. It gets the found number plates data and produces the ER images which are sent to the Web Server and/or to the remote DB Server. AM is also responsible for checking whether the found number plate matches the ones given into the Black or White verification lists, or if the number matches the rules and actions formed via Web UI. It also sends the status data from Status Manager Process to the Web Server and DB Server. AM is responsible for I/O ports control.

- **The Status Manager Process** collects hardware info (temperatures, humidity, voltages, etc.) regarding ANPR cameras and PC. It also gets the status of all software components. The Status Manager Process sends mentioned data via AM to the Web Server or logs it locally.

The **Web Server Process** sends the live stream images and the results to the Web UI browser (operator). It is also responsible for online adjustment of the ANPR system’s parameters. These parameters are exchanged between Web UI browser and AM via WEB Server process.

As can be seen the AM process is very complicated. That’s way it was additionally decomposed to set
of sub-processes – each of them related to the specific activity or communication.

- **The Cameras, I/O Ports and DB Server** can be interpreted as external hardware CSP processes which are connected to the ANPR System process via mentioned channels.

**IDENTIFYING CSP CHANNELS AND MESSAGES**

The arrows on the figure 3 represent the point-to-point CSP channels. Over these channels the processes are exchanging messages. Channels pointing from AM to the surrounding processes are named ToXXX (where XXX corresponds to the targeted process) and channels named FromXXX are targeted to AM, for example FromOCR or ToOCR. Similar is the notation between other software and hardware processes. In general, the messages are of two types – messages with binary image data (in different formats and compressions) and text messages (in plain CSV or XML format). The length of the messages is variable and depends on the size of the images or number of the message parameters.

**DECOMPOSITION OF ANPR MANAGER PROCESS**

As mentioned above the AM process implements the main ANPR logic and it was functionally decomposed (Fig. 4) to the next processes:

- **ANPR Dispatcher Process** is the main process of the AM, which activates all other AM processes depending on the raising events (messages) inside the ANPR system. The ANPR Dispatcher and the external for AM process - IH and OCR processes, have highest priority.

- **Evidential Record Builder Process** creates encrypted evidential image of the captured vehicle containing plate number data, time, place and other specific data.

- **Rule Engine Process** sends ER data to remote DB Server or to COM ports in case some customer defined rules are matched. For example it sends ER to the DB Server in case the plate number matches one in the White list and when the confidence level of the OCR results is greater than 80%.

- **DB Connector Process** stores ER to the remote DB Server. It sends the ER as zip files using FTP protocol.

Fig. 4. ANPR Manager Process sub processes.

```
ANPR = {
  ImageHandler(), OCR(), ANPRDispatcher
  ERBuilder(),
  RuleEngine(), DBConnecter(), SoftwareUpdater
  UIAdapter(),
  WebServer(), StatusManager
}
```

Fig. 5. CSP composition of ANPR System.

- **UI Adapter Process** communicates to the Web Server. It formats the live and result ER images and sends them to the Web Server. It also exchanges and validates ANPR parameters data provided by the Web UI operator.

- **Software Updater Process** is responsible for online software (OS, ANPR) updates and for updates of the ANPR configuration files.

- **Input Output COM Ports Connector Process** manages COM ports.

All mentioned above processes: - Rule Engine, DB Connector, I/O COM Ports Connector, SW Updater and UI Adapter have the next (higher) priority in the system. The WEB Server and Status Manager processes have low priority.

**MAPPING THE CSP PROCESSES TO OS PROCESSES AND THREADS**

After the decomposition the ANPR system was represented of five CSP processes. The AM process itself was split to seven sub-processes controlled by one of them taking dispatching role. These processes are mapped to OS-processes and threads according to their
importance. So OCR, Image Handler, ANPR Dispatcher processes and Evidential Record Builder must be executed with highest priority. Other processes forming AM have middle priority. The Web Server and the Status Manager are the processes with low priority.

Next CSP record (Fig. 5) represents the prioritised parallel composition of all ANPR system processes.

The sign || describes a parallel execution of two processes or group of processes, and the sign ||| a parallel execution of two processes with higher priority than that of the first (left-one) process.

A number of OS architectures [11, 12] were analyzed in order to select the appropriate OS process architecture for ANPR system. Some of them are related to one single OS process implementing sequentially the whole system activities or a number of interacting OS processes with equal priorities where each process implements the activity of one CSP process. Other architectures were related to a number of interacting OS processes with different priorities where each process implements the activity of one CSP process or a number of interacting OS processes with different priorities, coordinated by one special process – Broker. Next architectures [13] are related to one single process composed by a pool of threads with equal priorities, where each thread implements the activity of one CSP process or to one single process composed by a pool of threads with different priorities, where each thread implements the activity of one CSP process. The activities of all threads are dispatched by one thread with highest priority. Last architecture was combination of above given architectures based of interacting processes with different priorities built of threads dispatched by one with highest priority.

Previous author’s investigations related to similar software architectures [14] intended for parcels sorting systems, confirms that first three architectures are not applicable for real-time application as ANPR system. The Broker architecture is frequently used in embedded systems and it reduces the context switches and inter-process communications because the Broker distributes the messages between the processes and controls the common shared process memory. Last three architectures are very suitable for real-time systems because the threads need less time for communications and share common memory inside the process. The architecture with the dispatching thread corresponds fully to the CSP decomposition of the AM process. So the decision was taken to implement AM as one single OS process, built of seven threads. These threads are related to the CSP sub-processes of the AM. The ANPR Dispatcher thread has high priority. It analyses the type of incoming messages from external processes and internal threads and accordingly activates other objects signalling their threads with corresponding messages/events.

Finally, the ANPR System was mapped to five OS processes grouped in two priorities: high for IH, OCR and AM processes and, low for WEB Server and Status Manager Processes. Inside the AM process there are, also, two levels of priorities: highest for AM

Fig. 6. Result view from ANPR System in Web UI.
Dispatcher and ER Builder processes and middle for the rest processes.

The decision to implement IH and OCR as separated OS processes derives from the fact that in most ANPR systems they are third party software (executables with known API) that can not be modified or integrated inside the existing processes.

The Web Server was implemented as a multithreaded process and described in details in separate paper [10]. It is running inside the Apache Server [15] installed on ANPR system PC.

The real ANPR system (Fig. 6) ordered by Siemens is intended to work in Windows and Open Embedded OS environment and all threads’ and processes’ communications are based on the binary or text messages (see the table above) exchanged over TCP/IP sockets.

CONCLUSIONS

Presented in the paper process-oriented approach approach ensures a better way for refining the software architecture of the ANPR system by mapping efficiently the CSP processes to corresponding OS tasks and threads. The developed on the base of this approach ANPR system is flexible and can be easily extended or reduced for different goals - speed control, traffic control (Fig. 6), highway tolling, lots control, etc. by adding or discarding some processes or threads, using simple configuration files. In the future the ANPR system will be extended with additional module (OS process) responsible for on-line self update of the OS and other ANPR processes which will decrease dramatically the time for maintenance.

REFERENCES

10. A. Atanassov, F. Tomova, Web-Based Subsystem For Tuning Of An Automatic Plate Number Recognition System, Sixth International Conference, Challenges In Higher Education And Research In 21st Century, Sozopol, Bulgaria, 2008, 399-402.