XVIII IMEKO WORLD CONGRESS
Metrology for a Sustainable Development
September, 17 – 22, 2006, Rio de Janeiro, Brazil

CONTROL APPLICATION DESIGN OF THERMOWET SYSTEM

Dominik Sankowski [1],*Krzysztof Strzecha* [2], *Michal Janicki* [3], *Tomasz Koszmider* [4]

[1] Computer Engineering Department at Technical University of Lodz, Poland, dsan@kis.p.lodz.pl
[2] Computer Engineering Department at Technical University of Lodz, Poland, strzecha@kis.p.lodz.pl
[3] Computer Engineering Department at Technical University of Lodz, Poland, mjanick1@kis.p.lodz.pl
[4] Computer Engineering Department at Technical University of Lodz, Poland, tkoszmi@kis.p.lodz.pl

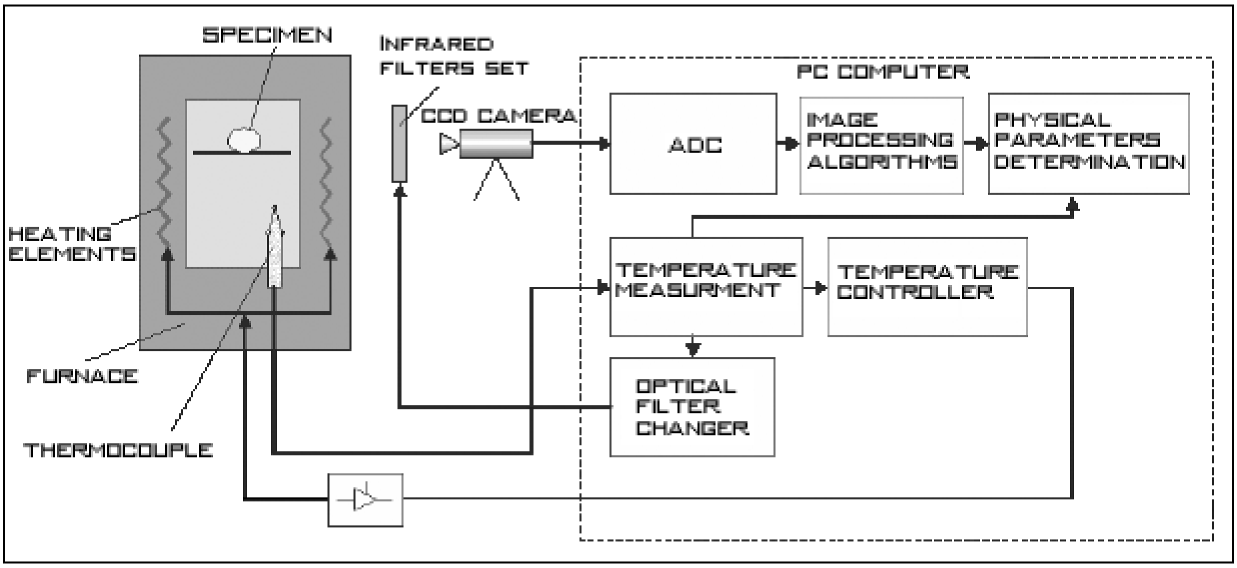

Fig. 1 Block diagram of computerised system with CCD camera for measurement of surface properties.

Abstract: In this paper, a case-study of the THERMOWET measurement system control application is conducted and it's software engineering aspects are described.

Keywords: software engineering, measurement systems, design patterns, analysis and digital images processing.

1. INTRODUCTION

The Computer Engineering Department of Technical University of Lodz with co-operation of Warsaw University of Technology and Industrial Electronic Institute (PIE) in Warsaw has built a computerized device for the automated measurement of surface phenomena occurring during contact of liquid and solid phases [1] (block diagram of the system is shown in Fig. 1, and device's photo in Fig. 2). The system is capable of measuring the surface tension of a liquid and the wetting angle of a solid by a liquid over a wide range of temperatures (up to 1800°C).

Both the surface tension and wettability of solids by liquid at an elevated temperature are essential in many industrial branches: metallurgy, foundry, surface engineering, welding, glass-making industry, etc. The background information about surface tension and wettability can be found in the fundamental work [2].

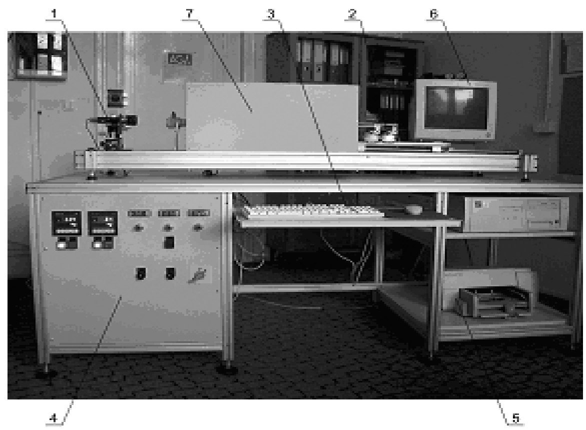

Fig. 2 Computerised system for the automatic measurements of the surface tension and the wetting angle of materials at high temperatures. (1 – the CCD camera of the vision system; 2 – the specimen feeding mechanism, 3 – the computer keyboard and mouse (in a drawer); 4 – the system of electric control and the system of technological gas supply; 5 – the computer and printer; 6 – the monitor enabling the convenient observation of the profile of a specimen to be made; 7 – the heating chamber of the furnace).

In many metallurgical processes, an important and sometimes a predominant role is played by the phenomena occurring at the interfaces of the liquid and the solid state.

These phenomena occurs in welding, making of composite materials with the participation of the liquid phase, sintering of powders, saturation of porous structures, coating, refining of metals to eliminate non-metallic inclusions, foundry or processes of crystallisation from the liquid phase. Thus, the knowledge of physical-chemical processes occurring between a liquid and a solid is a significant technological problem. However, it should be stressed that in some processes it is important to obtain the best possible wettability (e.g. technologies of bonding), while in others interfacial interactions are harmful and should be eliminated (e.g. the reaction between a crucible and a charge).

This article presents project of new software, which takes whole control over THERMOWET measurement system. Main project's structural and functional requirements are oriented to application stability, flexibility, functionality, separate modules re-usability and high measurements accuracy.

2. PROJECT SPECIFICATION

The main requirements set upon the application may be divided into structural and functional requirements.

The structural requirements specify that the core functional modules of the application should be independent of the physical communication media and the communication protocols used. Moreover, the core functional modules are expected to form encapsulated entities, with references to the other ones implementing the "uses" relationship of the object-orientation standard. The latter requirement is legitimate in view of the possible re-usage of the separate modules in other projects. Specific modules span the image acquisition and processing, and serial port communication.

The functional requirements specify that the application allows for experiment planning and archiving, account for dynamic algorithm library linking and conform to the MODBUS serial communication standard. In addition, the application is expected to provide interface appropriate for classes of users having access to the desired scope of functionality reflecting different modes in which the experiments may be conducted.

In view of the above mentioned requirements, the modular structure [3] of the application introduces image acquisition and processing library modules, serial communication module, task scheduling and user interface module.

3. THE MODULAR STRUCTURE

The project distributes the desired functionality between separate modules:

1. The XML processor used as means of description in storing and loading experiment parameters.
2. The task/experiment-scheduling module responsible for task scheduling and experiment planning.
3. The database module responsible for experiment data storage.
4. The graphical user interface module.
5. The serial communication module responsible for RS485 communication with the temperature controller and camera filter adjustment mechanism.
6. The image acquisition module responsible for video camera communication using the frame grabber standard.
7. The image-processing library containing algorithms used for image segmentation and filtering.
8. The image analysis library containing algorithms used for physical and chemical parameter calculation operating on images of the specimen.

The mutual dependency between the modules is depicted in Fig.3.

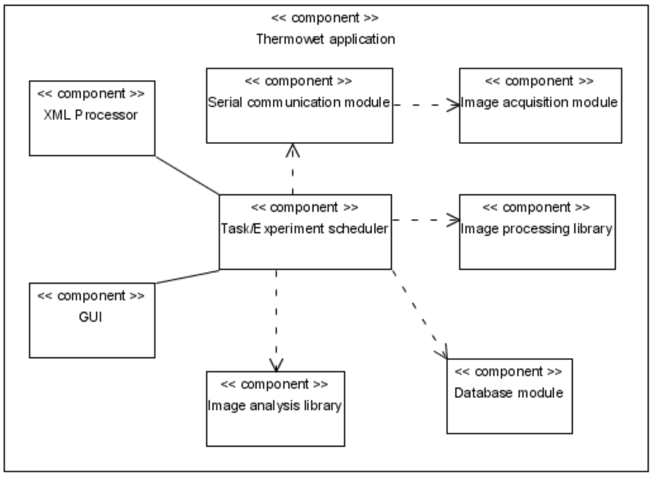

Fig. 3 THERMOWET system's module dependencies.

The task/experiment scheduler realizes the core control functionality of the application.

The two out of the seven designed relationships are collaboration relationships rather than implementation dependencies, reflecting the inner design of the main module being dependent on the inner workings of the processing modules with the input/output and user interface modules being separate entities.

4. THE SERIAL COMMUNICATION

The serial communication module spans the functionality specific to serial port communication issues as well as higher abstraction level data formatting, including protocol-specific frame generation. The classes realizing low-level, binary data transmission collaborate with those responsible for device-specific communication. The latter account for being part of an adapter design pattern provides action methods to other modules.

The core entity of the serial communication module is the abstract device driver, reflecting the functionality of a physical device at logical level. The corresponding class is depicted in Fig.4. The abstract driver uses protocol and frame libraries that incorporate the communication standards at frame communication layer. The abstract device driver is an analogy to the abstract JDBC (Java Database Connectivity) database driver. Therefore, the abstract base driver class defines the universal functionality interface that specific drivers should implement. Moreover, it is expected that there may exist differences between scopes of functionality of drivers controlling different classes of devices, yet using the same low-level communication library.

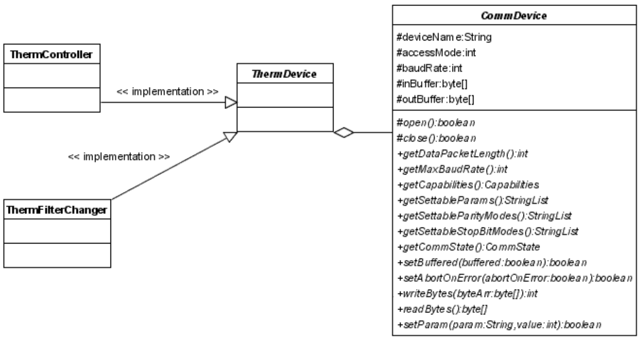

Fig.4 The CommDevice class representing the logical communication device.

The abstract device driver is expected to be part of an adapter design pattern, providing collections of action methods to other modules.

The serial communication module distributes the scopes of responsibility involving physical device communication, bit packing and frame formatting among separate classes, incorporating the "uses" object-orientation relationship. Each entity representing different scope of responsibility is designed as an abstract base class for specific implementations. Each abstract base class is related to a corresponding abstract factory class, resulting in better re-usage possibilities and structural dependencies.

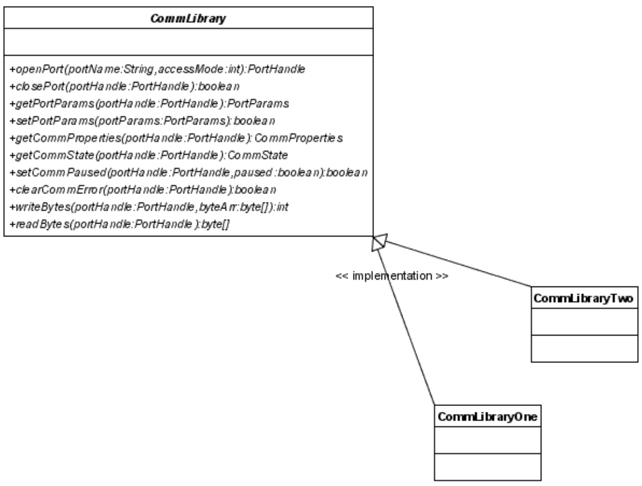

Fig. 5 The CommLibrary class realizing the physical implementation of the communication functionality.

The bridge design pattern is used to separate the logical device entities from the system-specific communication libraries, creating a mutually independent link between the structure and the implementation. The libraries are implemented as subclasses of the main abstract library class, as depicted in Fig.5. This is justified in view of the possible re-usage of the abstract driver classes in projects targeting at different host operating system.

Many of the communication methods at logical device level are designed to be synchronized using an implicit monitor of the enclosing class. The result structure prevents overlapping calls to communication methods.

The system-specific communication libraries collaborate with the class hierarchies responsible for single-character transmission and frame-level character bundle formatting.

The designed single-character transmission protocol factory is depicted in Fig.6.

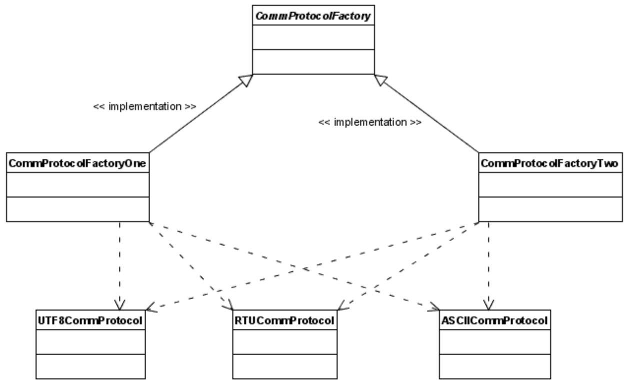

Fig.6 The CommProtocolFactory class incorporating single-character transmission modes.

The class hierarchy incorporates the two most frequently used transmission modes in control applications, ASCII and RTU, and is extensible enough for implementation of additional modes, such as UTF-8.

The designed frame-level character bundle formatting factory is depicted in Fig.7. The class hierarchy incorporates the two most frequently used transmission standards in control applications, MODBUS and JBUS and is extensible enough to incorporate other standards.

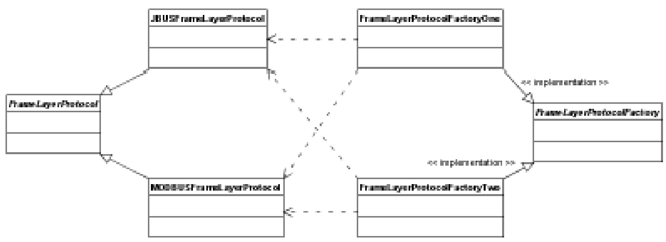

Fig.7 The FrameLayerProtocolFactory class incorporating frame-level character bundle formatting standards.

The abstract class representing the logical communication device driver is designed to be inherited from to incorporate the possible physical devices over which the communication may be implemented, as depicted in Fig.8.

The device factories are introduced for further extensibility of the solution to different operating systems.

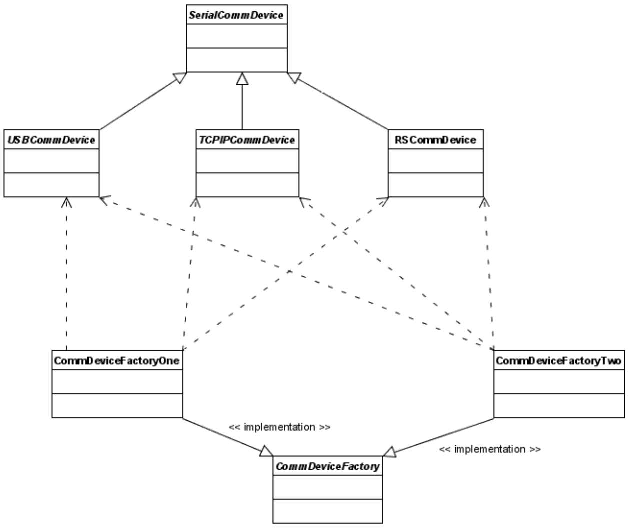

Fig.8 The SerialCommDevice inheriting from the abstract class representing the logical communication device driver.

5. TASK EXPERIMENT SCHEDULER MODULE

Basic requirement set upon the Experiment Scheduler module is to take over whole control over the Thermowet measurement system.

Experiment controlling processes consist of two apart threads, which responsibility concentrates on separate device. Functionality of Task Experiment Scheduler module is depicted in Fig.9.

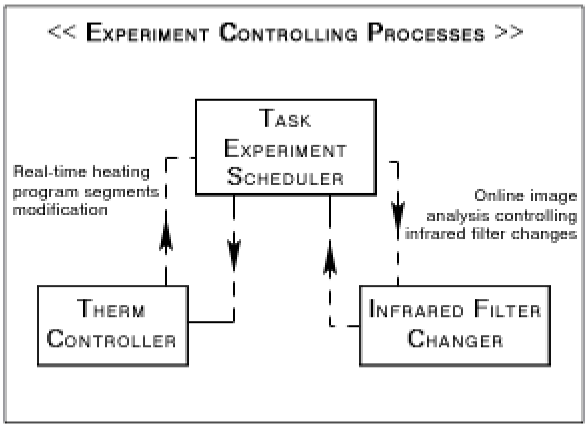

Fig.9 Task Experiment Scheduler functionality.

The use of separate therm controller's thread was dictated by therm controller's tendency to over control the setpoint temperature. This solution gives the ability to modify linear temperature growth or change the setpoint temperature of current program segment.

Online specimen's image analysis affords possibility to change infrared filter automatically, when specimen's brightness became to high to differ it from feeder's table.

Appropriate choice of infrared filter is extremely important for precise surface tension and wettability calculations.

6. CONCLUSION

The use of the bridge and abstract factory design patterns [4] allows for portable separation between the components realizing the logical and physical functionalities. This is particularly important in design of industry-dedicated control applications, an example of which is THERMOWET. The result solution is highly maintainable and flexible enough to possible changes that may be introduced during further project development. In particular, the benefits of this approach are indicated in the design of the serial communication module.

REFERENCES

[1] D. Sankowski, K. Strzecha, S. Jezewski, J. Senkara, and W. Lobodzinski, *Computerised Device with CCD Camera for Measurement of Surface Tension and Wetting Angle in Solid-Liquid Systems,* in *16th IEEE Instrumentation and Measurement Technology Conference IMTC*, Venice, Italy, May 1999, pp. 164-168.

[2] A.W. Adamson, *Physical Chemistry of Surfaces*, Wiley Int. Publ., New York, USA, 1982.

[3] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Massachusetts: Addison-Wesley, 1998.

[4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Massachusetts: Addison-Wesley, 1995.