STUDIA INFORMATICA 2003
Volume 24 Number 1 (52)

Maciej ZYGMUNT, Maciej WNĘK, Pieter-Jan VLOK
ABB, Corporate Research Center

# SELECTED PATTERNS OF INDUSTRIAL [IT] ARCHITECTURE

**Summary**. Seamless enterprise application integration might become reality with the usage of modern and advanced software tools. Tool enabling realization of "Industrial [IT]" vision is called "Aspect Integrator Platform" ("AIP"). In this paper there is a short description of this product to give the reader a general view what we are talking about. From software engineering point of view there are immense advanced techniques and design patterns used inside AIP. One of such pattern called "ASO late binding" is described below.

**Keywords**: software architecture, design patterns, real time.

# WYBRANE WZORCE PLATFORMY "INDUSTRIAL [IT]"

**Streszczenie**. Nieograniczona integracja komputerowych programów wewnątrz przedsiębiorstwa może się stać rzeczywistością dzięki użyciu nowoczesnego oprogramowania. Narzędziem umożliwiającym realizacje wizji Industrial [IT] jest „Aspect Integrator Platform" („AIP"). W tym artykule zawarto krótkie wprowadzenie do „Industrial [IT]", żeby lepiej pokazać, do czego ma zastosowanie poniższa technika. Z punktu widzenia inżynierii oprogramowania wewnątrz AIP jest wiele bardzo zaawansowanych rozwiązań. Jedno z nich, to znaczy wzorzec nazwany „ASO późne łączenie" („ASO late binding"), będzie opisane poniżej.

**Słowa kluczowe**: architektura oprogramowania, wzorce projektowe, czas rzeczywisty.

## 1. Introduction to Industrial [IT]

With advances in current software and hardware it appears that integration between plant floor and top floor is becoming reality. Term Industrial [IT] stays for information technology applied to industry, which means a seamless integration of systems. As shown on Fig. 1

managing whole enterprise from one electronic workplace might become reality soon. More materials on this vision are available on ABB website and in [2].

### 1.1. Motivation for building Industrial IT

- Seamless integration of systems for power, automation, and information
- Built-in tools to install, operate, optimize, and maintain every plant device
- A compatible, open architecture from plant floor to top floor

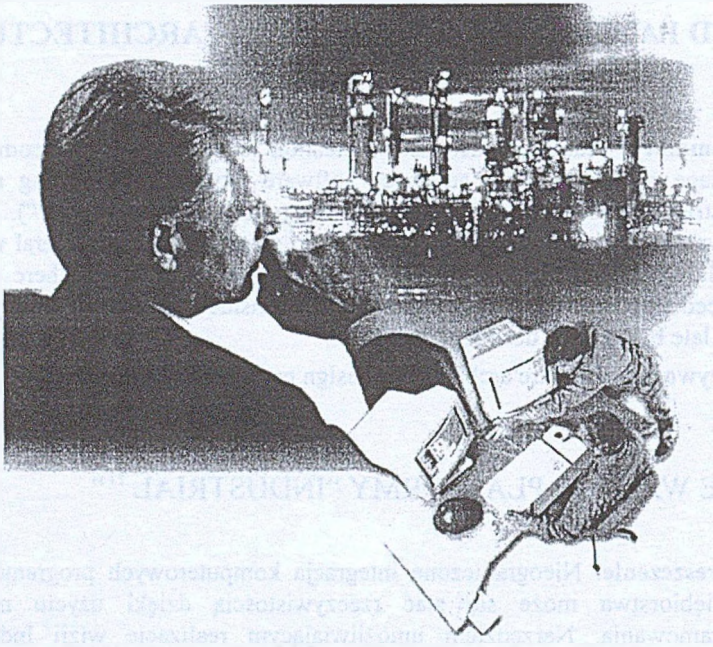

Fig. 1. Industrial IT Vision
Rys. 1. Wizja Industrial IT

### 1.2. Software tools to achieve this vision

#### *1.2.1. Compatible enterprise building blocks*

Through the broadest technology initiative in its history, ABB is "information enabling" every product bundling documentation, configuration and connectivity tools, lifecycle support, and more in consistent electronic format (see Fig. 2).

Using patented ABB Aspect Object™ technology, these electronic characteristics are built into a powerful software shell which represents the physical object in any sense.

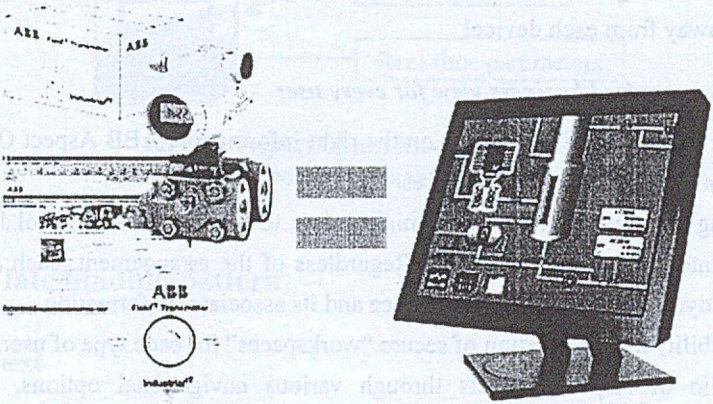

Fig. 2. Enterprise Building Blocks
Rys. 2. Części składowe systemu

### *1.2.2. An open enterprise architecture*

The ABB Aspect Integrator Platform™ lets you monitor, control, optimize, and maintain with one powerful interface. It allows you to evaluate, deploy, and influence plant components just like browsing the files on your PC. It has fast, intuitive navigation from a logical systems perspective.

It is open as a consumer of open solutions and as a supplier of open standards:

- It might use external devices on condition they can communicate with ole for process control (OPC).
- External suppliers can deliver a system compatible with Industrial IT, there is a certification program with clearly defined certification levels and procedures.

### *1.2.3. Asset information that follows the asset*

Physical installation of system devices has long been the easy part. The difficult part is collecting the information to install, operate, and maintain each component – then keeping it up to date.

Industrial IT from ABB changes all this. No matter where each real object or its asset information is deployed, one "click" on the Aspect Object opens the links necessary to configure, optimize, analyze, and influence the real object.

The list of dynamic device characteristics (Aspects) begins with complete electronic documentation, drawings, and instructions.

Depending on the product and its level of Industrial IT Certification, additional Aspects may include integral configuration and maintenance tools, control faceplates and graphic symbols, communication protocols, on-line service, and more.

The result: "Plug-and-Produce" installation, operation, and asset management just a mouse click away from each device!

*1.2.4. A customized business view for every user*

To help every user to find and act on the right information, ABB Aspect Objects may be organized in a variety of logical structures.

Depending on your interest, these might relate to process flow, control logic, physical location, maintenance status, or others. Regardless of the arrangement, each Aspect Object maintains its dynamic links to the real device and its associated information.

This capability allows creation of secure "workspaces" for each type of user, accessing the same portfolio of Aspect Objects through various navigational options, shortcuts and favorites.

### 1.3. Aspect Object Model

The most important thing to start understanding the platform is the "Aspect Object Model". Base for definition of this abstraction is an assumption that the world consists of real objects and each of these real objects has different view for different users perspectives. These views are called aspects of an object. Projection of real world into computer system makes "Aspect Object Model" where objects from the real world are represented as an entity or a container and aspects are some kind of components capable of storing their internal state and doing some actions like presenting it to the user, exchanging information with other aspects.

The Aspect Object Model addresses the issue of presenting information and allowing the user to operate on information in a consistent way. The model also addresses how different functions are integrated into the system in a way natural to the user. The Aspect Object Model is based on Objects and Aspects.

The objects in the model are the objects the user interacts with. For example, an object can be a reactor, a pump or a node (a computer). The object itself is a container that holds different parts, aspects, of the object. This is illustrated in the Fig. 3.

An aspect is one 'piece' of data and operations that is associated with an object. Typical aspects of an object (process) are: its control program, operator faceplate, trend configuration, function specification etc.

From the computer science perspective, the Aspect Object is a meta-object that groups aspects, while aspects from a software engineer point of view are software objects. These software objects are called "Aspect System Objects" ASO.

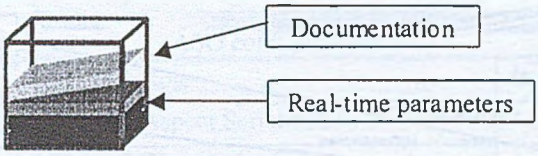


Fig. 3. Aspect Object Model
Rys. 3. Aspektowo-obiektowy model

## 2. ASO late binding pattern

### 2.1. Context

Our team has developed powerful statistical methods for evaluating machines life. This methods are using external data collected by external programs. In each factory they are using different tools for collecting data, this is caused by historical or business specific reasons. There are different programs to collect data, and these observations have a huge statistical value especially if they are collected during several years.

### 2.2. Problem

Changing binary part of the software requires appropriate software engineering cycle with software releasing rules, and is unacceptable for the single factory settings. Further on, changing in core algorithms should be populated back to all locations without the need to rebuild other dependent software parts.

### 2.3. Solution

Solution to this problem is: make more pieces of software than is needed for one site, allow for different runtime configuration for particular needs.

#### *2.3.1. Implementation of Aspects*

The simplest way to implement aspects is to do them as single components. This will give quickest results, but with the sake of reusability. AIP provides powerful mechanisms for making more reusable software. First it is suggested (see Fig. 4.) to separate view and data according to pattern known in literature [1] as "Model View Controller" (MVC )

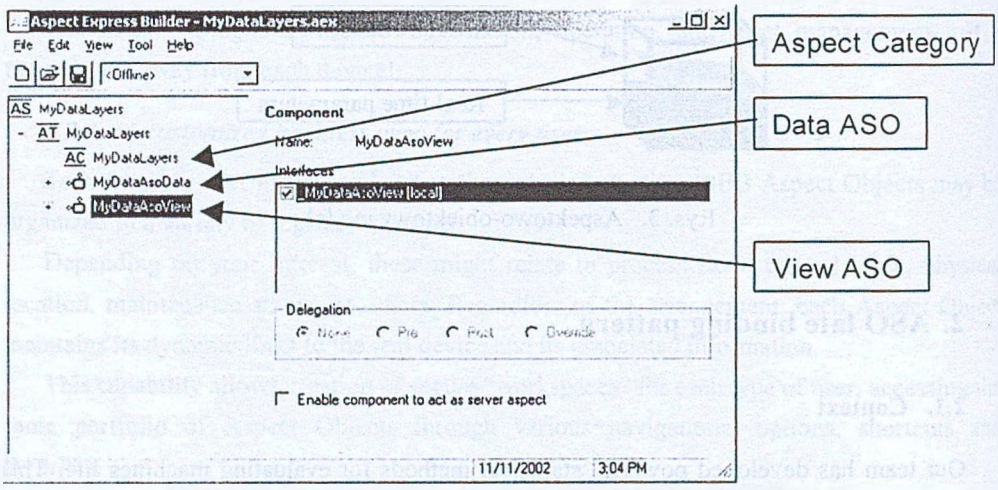


Fig. 4. Basic Aso Binding
Rys. 4. Podstawowe łączenie ASO

Advantages of splitting aspect into separate View ASO and Data ASO are following:

- better reusability,
- better performance

**Better reusability**: one data ASO can be used for several other aspects, therefore it is obvious that it could be that more sophisticated, more robust and better tested. These are the obvious benefits of reusability.

**Better performance**: data ASO are cached in memory so accessing them for consecutive number of times by browsing or by subscribing to their properties is much faster than if they were instantiated every time needed.

### *2.3.2. ASO Crosscutting*

By providing crosscutting of ASO's (components) in late binding one can have a new functionality which is achieved on configuration level without necessity of changing binary implementation of used components. Suppose we have an aspect consisting of Generic data ASO and "Generic View ASO" that for specialized functionality this "specialized view ASO" will access data of this aspect using interface from "Generic Data ASO" (see Fig. 5.). The same interface might be provided by other ASO and Specialized View ASO needs only to find appropriate interface. In Table 1 there are listed several possible runtime configurations an bindings there are four possible aspects (A, B, C, D) build out off one data ASO and three different view ASO's.

Table 1

ASO configurations

| No. | ASO Name | Role | Aspect |
|---|---|---|---|
| 1 | Generic Data ASO | Aspect Serialization | A, B, C, D |
| 2 | Generic View ASO | Bind to data ASO, provide specialized interface | A {1,2} |
| 3 | Specialized View ASO | Bind to specialized interface | B {1,2,3} |
| 4 | Other ASO | Bind to specialized interface | C {1,2,4}, D {1,2,3,4} |

Possibilities for composing aspects during runtime deployment are limited only by the user needs.

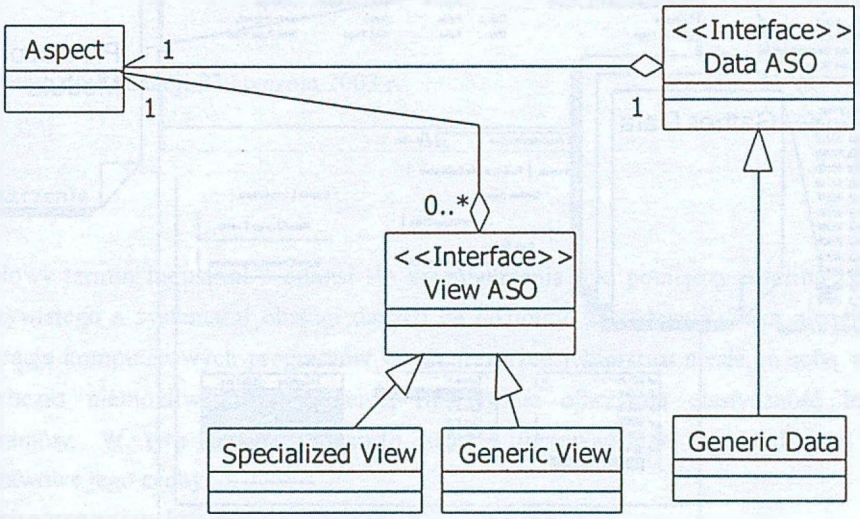


Fig. 5. Aspect Runtime Binding
Rys. 5. Dynamiczne łączenie aspektu

### 2.4. Known usage

This technique was used to implement aspects for lifetime estimation aspect system (LTE). Two Aspects were implemented:

- LTE Data Acquisition Aspect: for collecting data on selected machine.
- LTE Data Processing Aspect: for performing calculations on group of machines.

Both of this aspects share the same implementation of Data ASO, even if they are assembled into one library and only external packaging or enriching with runtime info metadata makes several different aspects out of one set of binary components. This aspects might be configured or glued to gather data from an external maintenance database (like

Smart Advisor), perform calculations and then set alarms used by other platform programs, or work on data from Excel spreadsheet and present results interactively to the maintenance engineer. Collection of aspects assembled for specific collaboration is called aspect system. To perform lifetime estimation on bearings, there has been prepared specialized aspect system. Results of this analysis [3] have shown possibilities of life time estimation approach, but also have verified the concept of highly configurable systems. This aspect system is presented on Fig. 6, arrows with text "Gather Data" are showing that aspects on one machine are taking data from aspects on other machines in the same group.

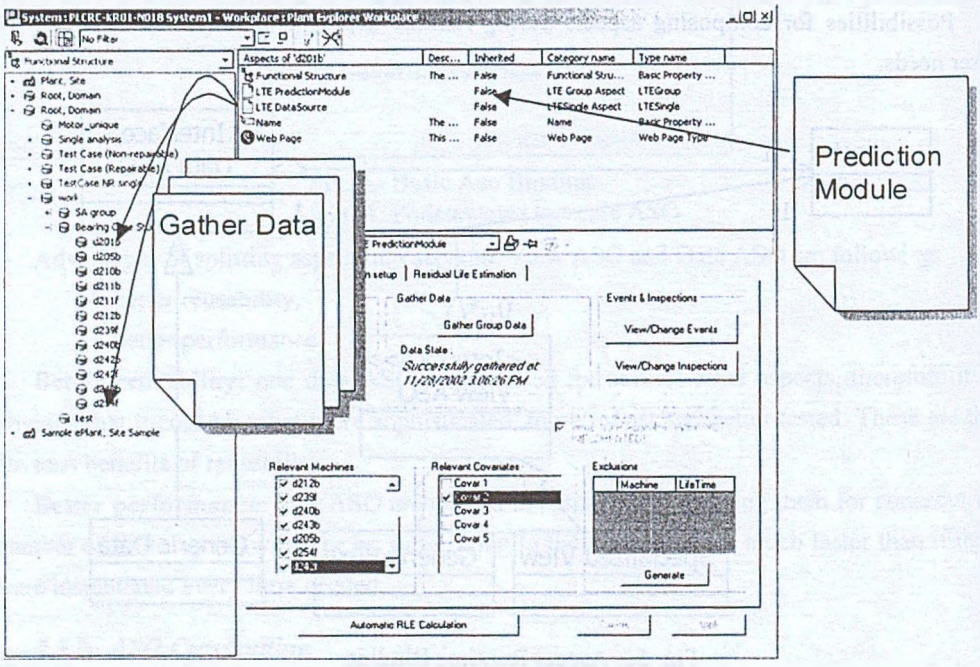


Fig. 6. Life Time Estimation Aspect System
Rys. 6. System aspektowy szacujący czas życia

## 3. Conclusion

Effort which has been applied to divide system into smaller components and then assemble customizable system out of smaller parts is paying off in terms of possibilities to adopt to changes. With help of Aspect Integrator Platform it is possible now to develop more reusable software.

## REFERENCES

1. Gamma E. et al.: Design Patterns Elements of Reusable Object-Oriented Software. Addison-Wesley Publishing Company, Reading Massachusetts 1995.
2. Welcome to the world of Industrial IT, 3BUS092071R0001, http://www.abb.com
3. Vlok P. J., Wnek M, Zygmunt M: Utilizing statistical residual life estimates of bearings to quantify the influence of preventive maintenance actions, Submitted for publication. Journal for Mechanical Systems and Signal Processing, Cambridge, UK.

Recenzent: Dr inż. Przemysław Szmal

Wpłynęło do Redakcji 23 stycznia 2003 r.

**Streszczenie**

Nowy termin Industrial IT odnosi się do zapełnienia luki pomiędzy informacjami czasu rzeczywistego a systemami obsługi danych na poziomie zarządzania. Taka nieograniczona integracja komputerowych programów wewnątrz przedsiębiorstwa niesie za sobą wyzwania dotychczas niemożliwe do spełnienia, mianowicie olbrzymią elastyczność tego typu programów. W tym artykule zawarto krótkie wprowadzenie do „Industrial IT" oraz podstawowe jego cechy:

- nieograniczona integracja systemów;
- zarządzanie wszystkimi urzadzeniami w fabryce;
- kompatybilna otwarta architektura od poziomu sterowania do poziomu zarządzania.

Następnie przedstawiono „Aspect Integrator Platform" (AIP) jako narzędzia programowe realizujące tę wizję.

Z punktu widzenia inżynierii oprogramowania wewnątrz AIP jest wiele bardzo zaawansowanych rozwiązań. Jednym z nich jest wzorzec nazwany „ASO późne łączenie" („ASO late binding"). Wzorzec ten opisano zgodnie z konwencją opisywania wzorców: to znaczy od wyjaśnienia potrzeby takiego rozwiązania po pokazanie rozwiązania i przykładowe zastosowanie.

Przykładowym zastosowaniem jest program szacujący czas życia maszyn opisany w pracy [3]. Dzięki zastosowaniu techniki dzielenia na mniejsze fragmenty i późniejszej konfiguracji aplikacji do właściwych potrzeb uzyskano większą niż tradycyjna elastyczność oprogramowania.

**Adresy**

Maciej ZYGMUNT: ABB Sp. z o.o, Corporate Research, ul. Starowiślna 13 A, 31-038 Kraków, Polska, maciej.zygmunt@pl.abb.com .

Maciej WNĘK: ABB Sp. z o.o, Corporate Research, ul. Starowiślna 13 A, 31-038 Kraków, Polska, maciej.wnek@pl.abb.com .

Pieter-Jan VLOK: ABB Ltd, ABB SERVICE, Daresbury Park, Daresbury Warrington, Cheshire WA4 4BT, United Kingdom, pieter-jan.vlok@za.abb.com .