BLACKBOARD ARCHITECTURE FOR PRODUCT LIFE CYCLE STAGE DEFINITION

Darya Rudenko and Arkady Borisov

Department of Modelling and Simulation, Institute of Information Technology, Riga Technical University
1 Kalku Str., LV-1658, Riga, Latvia
Phone: (+371) 67089530
e-mail: to.darya@inbox.lv, Arkadijs.Borisovs@cs.rtu.lv

Abstract

The product life cycle concept suggests that a product passes through four stages of evolution: introduction, growth, maturity and decline. As the product evolves and passes through these four stages, profit is affected, and different strategies have to be employed to ensure that the product is a success within its market, therefore there is a need to define in time at which of the stages the product is at the moment.

The blackboard architecture is not a new technology and has a lot of benefits; therefore it can be widely used. The blackboard methodology is a complicated system task solving strategy using different knowledge sources communicating by means of common information field.

The blackboard architecture can offer a centralized system, and it is known that system centralization means that the system will be robust without complicating system's parts. Instead, simple parts will make up robust system.

Keywords: blackboard architecture, blackboard benefits, product life cycle, PLC stage definition

1. Introduction

Nowadays product life cycle stage definition issue nowadays is very common. Company's profit depends on which of the stages the product is. It is not easy to tell which stage the product is in. The maturity stage is a good stage for every product, but wrong manager decision can change the stage to decline.

There are different strategies for each product life cycle stage in order to maximize profit and to delay decline stage; therefore there is a need to find a solution for product life cycle stage definition. This would help manager to take correct decisions in order to optimize company's work.

This paper consists of 6 sections: 2) product life cycle definitions, some examples and the main issue with product life cycle, 3) blackboard architecture definitions and benefits, 4) rule-based system used for this issue solution, 5) an example of blackboard architecture application for product life cycle stage definition, 6) ideas how to implement blackboard architecture.

2. Product life cycle

The product life cycle concept suggests that a product passes through four stages of evolution: introduction, growth, maturity and decline. As a product evolves and passes through these four stages profit is affected, and different strategies have to be employed to ensure that the product is a success within its market (see Figure 1). [1]

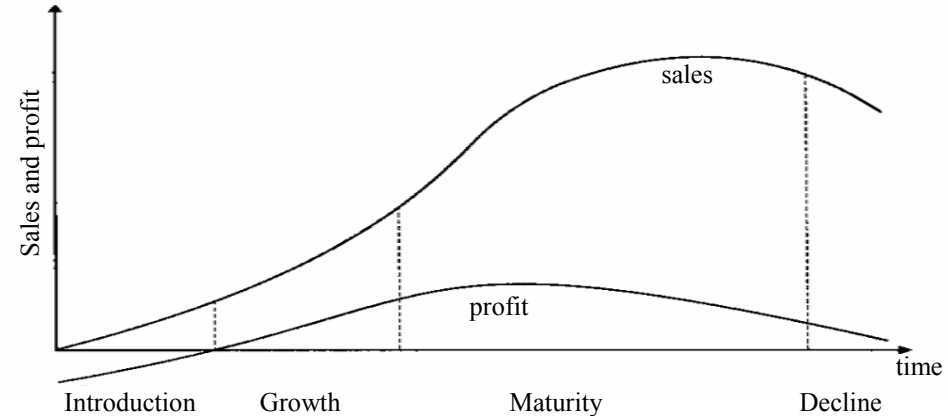


Figure 1. Product Life Cycle diagram

Product life cycle stage characteristics and strategies are listed below:

Product Life Cycle stage	Stage characteristics	Strategies
Introduction	The product is introduced to customers. Sales are low until customers become aware of the product and its benefits. Development and marketing costs are high. Profit is negative because of this reason.	The need for immediate profit is not a pressure. Primary goal is to establish a market and build primary demand for the product. Informative promotion is needed.
Growth	This stage is a period of rapid sales growth. Once the product has proven a success and customers begin asking for it, sales will increase further. Competitors see the opportunity and enter the market. Some just copy the most successful product, or try to improve it to compete better. Profit is positive and grows.	The goal of this stage is to gain customer preference and increase sales. Market share tends to stabilize.
Maturity	Rapid sales growth cannot last forever. Sales slow down as the product sales reach peak as it has been accepted by most buyers. Product lines are widened.	Producers attempt to differentiate products and brands. Price war and intense competition occur. Market share reachs saturation. Producers begin to leave the market due to poor margins.
Decline	Long-run drop in sales and falling demand are the main characteristics of this stage. A few small specialty firms may still manufacture the product. [1, 2, 12]	The organisation may try to change their pricing strategy to stimulate growth, however the product will either have to be re-modified, or replaced within the market. [1, 2, 12] In order to prevent slipping into decline: promote more frequent use of the product by current customers, find new target markets for the product, find new uses for the product, price the product below the market, develop new distribution channels, add new ingredients, delete old ingredients. [7]

Examples

For better comprehension of this subject some examples are presented where the products are currently at different stages of the product life cycle:

1. Introduction: Third generation mobile phones, E-conferencing;
2. Growth: Portable DVD Players, E-mail;
3. Maturity: Personal Computers, Faxes;
4. Decline: Typewriters, Handwritten letters. [3]

Problems with Product Life Cycle

According to [2, 4], in reality very few products follow such a prescriptive cycle. The length of each stage varies enormously. The decisions of marketers can change the stage, for example from maturity to decline by price-cutting. It is not easy to tell which stage the product is in.

Companies often try to use extension strategies in order to try to delay the decline stage of the product life cycle. The maturity stage is a good stage for the company in terms of generating cash. The costs of developing the product and establishing it in the market are paid and it tends to then be at a profitable stage. The longer the company can extend this stage, the better it will be for them. [2, 4]

3. Blackboard architecture

Blackboard metodology is a complicated system task solving strategy using different knowledge sourses communicating by means of common information field.

Blackboard is a global accessible data base which is used for intermediate, partial results of problem solving. As a rule, various sites of a blackboard allow to represent hypotheses at various levels of abstraction and provide joint actions of several experts.

Blackboard system can be imagined as a group of sitting human specialists next to a large blackboard. They are working cooperatively in order to solve the problem and they use the blackboard as a workplace for solution development.

Problem solving begins with announcement of a problem and writing initial data onto the blackboard. The specialists are watching the blackboard looking for an opportunity in order to make contribution for solution development. When a specialist finds this opportunity, he records the contribution on the blackboard in hope that others will use his contribution for final problem solving. This process continues until the problem is solved.

A blackboard system consists of three components (see Figure 2):

1) Knowledge sources (KSs);
2) Blackboard;
3) Control component.

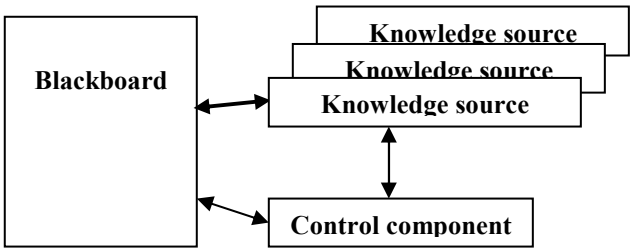


Figure 2. Blackboard architecture

Knowledge sources are independent modules that contain the knowledge needed for problem solving. They don't need to know about the existence of the others, but they have to understand the state of problem-solving process and the representation of relevant information on the blackboard.

Knowledge sources can be represented with different kind of knowledge; they can include rule-based systems, case-based systems, neural networks, fuzzy logic systems, genetic algorithms, legacy software systems (see Figure 3) [3].

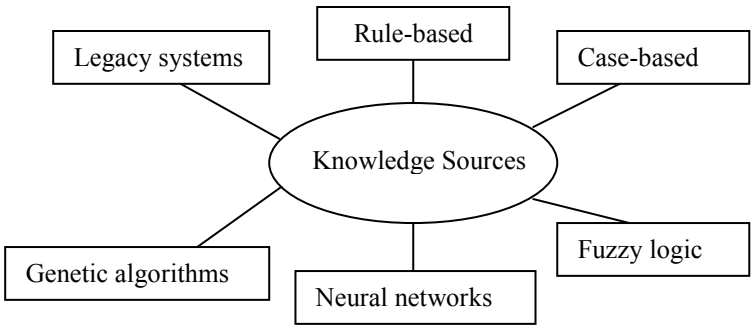


Figure 3. Different kinds of knowledge sources in blackboard system

Blackboard is used as a global database for sharing different information as input data, partial solutions, alternatives and final solutions.

Blackboard applications tend to have complex blackboard structures, with multiple levels of analysis or abstraction.

Control component makes runtime decisions which of knowledge sources to execute next for optimal problem solution.

The blackboard was originally designed as a methodology for handling complex, ill-defined problems. The first famous example is the Hearsay II speech recognition system. A more recent example, the PLAN component of the mission Control System for RADASAT-1, is a blackboard system. [5]

Blackboard benefits

The Blackboard system can offer the following benefits to software developers:

1) Integration of different knowledge sources;
2) Modularity - each knowledge source is independent;
3) Flexibility - the Blackboard architecture allows blackboard applications to adapt to changing requirements much more flexibly than traditional procedural software applications.

4) Software reuse:
   a) The independence and modularity of knowledge sources means that new applications can easily be constructed using existing knowledge sources.
   b) Legacy (traditional procedural) software investments can be preserved because they can be incorporated as knowledge sources.
   c) The Blackboard itself is application independent, and is easily applied to new problem domains.
5) Extensibility - new knowledge sources can be developed and integrated without changing the existing system. [6]

4. Rule-based system

Here the blackboard architecture's knowledge sources are represented with a rule-based system here. The rules for rule-based knowledge sources are based on different product life cycle researches and notes as [7], on diagram (see Figure 1) and on logic as well.

These rules are based on Theater Consultation [8] and this is a mix of rules for different stages:

№	**Condition**	**Action**
1.	IF Pre-Stage is (*Introduction* OR *Growth)* AND profit is negative and grows	THEN the Stage is *Introduction*
2.	IF Pre-Stage is (*Introduction* OR *Growth)* AND high service level	THEN the Stage is *Introduction*
3.	IF Pre-Stage is (*Introduction* OR *Growth)* AND profit grew and reached 0	THEN the Stage is transition from *Introduction* into *Growth*
4.	IF Pre-Stage is (*Introduction* OR *Growth*) AND profit is positive	THEN the Stage is *Growth*
5.	IF price falls	THEN the Pre-Stage is (*Maturity* OR *Decline)*
6.	IF sales fall	THEN the Pre-Stage is (*Maturity* OR *Decline)*

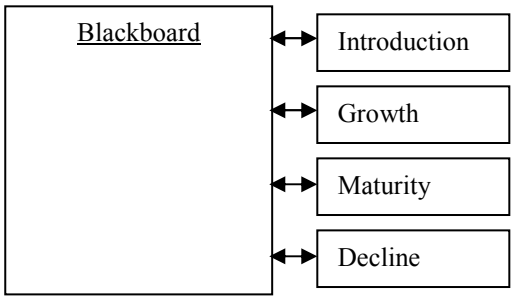
For better understanding let's check up each rule one by one:

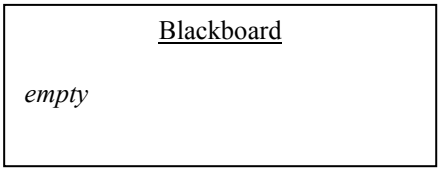
1) IF Pre-Stage is (Introduction OR Growth) AND profit is negative and grows, THEN the Stage is Introduction - this rule is based on [7] and might be seen and checked in Figure 1;
2) IF Pre-Stage is (Introduction OR Growth) AND high service level, THEN the Stage is Introduction – this rule is based on theory, that the offer in Introduction stage is bigger than demand.
3) IF Pre-Stage is (Introduction OR Growth) AND profit grew and reached 0, THEN the Stage is transition from Introduction into Growth – this rule is based in Figure 1, which pictures ideal product life cycle diagram;
4) IF Pre-Stage is (Introduction OR Growth) AND profit is positive, THEN the Stage is Growth – this rule is based on [7] and might be seen and checked in Figure 1;
5) IF price falls, THEN the Pre-Stage is (Maturity OR Decline) – this rule is based on theory that the price used to fall only on Maturity and Decline stage and on [7]
6) IF sales fall, THEN the Pre-Stage is (Maturity OR Decline) – this rule is based on [7]; it can be seen and checked in Figure 1.

5. Example

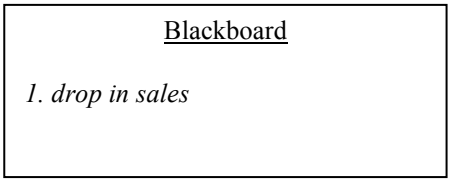
The idea of this example is to show how this blackboard architecture works with this rule-based system for product life cycle stage recognition.

Basically initial facts are recorded on blackboard, and knowledge sources are executed one by one. So it means that at first Introduction stage rules are executed, then Growth stage rules, then Maturity stage rules and then final Decline stage rules are executed. After each stage rules passed, any result (intermediate or final) is written on blackboard. The system continues it work until all stages rules are passed. The final result (the name of the Stage) is printed on the screen.





Let's see an example:
Blackboard is empty;
Initial facts: drop in sales. This fact is recorded on blackboard.



Control component executes first knowledge sours- Introduction stage rule-based system. The rules are being checked and the system asks the user:

Question for user: is the price falling?
Answer: yes, the price is falling.

The rules for Introduction stage are false.

So after this stage this is written on blackboard: price falls;
not Introduction.

Blackboard

1. drop in sales
2. price is falling
3. not Introduction

The next step is to check the next stage- Growth.

The rules from Growth stage use information on the blackboard, so there is no need to ask the same questions on this stage.

All the rules for Growth stage are false. So it is written on the blackboard: not Growth.

Blackboard

1. drop in sales
2. price is falling
3. not Introduction
4. not Growth

The next step is to check the next stage- Maturity.

Question for user: Is profit positive and slowly falling?
Answer: profit is positive and rapidly falling
Question for user: Is the product line widened?
Answer: No, product line is minimal.

All the answers of Maturity stage are written on blackboard:
profit is positive and rapidly falling;
product line is minimal;
not Maturity.

Blackboard

1. drop in sales
2. price is falling
3. not Introduction
4. not Growth
5. profit is positive and rapidly falling
6. product line is minimal
7. not Maturity

The next step is to check the next stage- Decline.

Question for user: is the price the lowest right now?
Answer: yes, the price is the lowest.

The following data is written on blackboard: the price is the lowest;
this is Decline.

And the final result is printed on screen: Decline.

Blackboard

1. drop in sales
2. price is falling
3. not Introduction
4. not Growth
5. profit positive and rapidly falls
6. product line is minimal
7. not Maturity
8. price is the lowest
*9. this is **Decline***

The stage of this example (Decline) was known before rule-based system execution, it means that this example was needed for checking existing system – does this system work correctly? The proposed system gave a correct answer, so these rules work correctly, and the next step is implementation of this system.

6. Implementation

According to [9], Blackboard architecture can be implemented at least in two ways: T Spaces and JavaSpaces:

1) T Spaces is an application of the event-based blackboard architecture. It is a middleware component which provides group communication and event notification services. Its tuple spaces (blackboard) hold data to be exchanged. A set of APIs implements the function of control policies by means of implicit invocation. The data producer and consumer (knowledge sources) do not know each other. Thus, they can be attached to (detached from) the system easily by registering (removing) interests in events.

2) JavaSpaces is designed to provide a simple unified mechanism for dynamic communication, coordination, and data- and object-sharing. It also implements the event-based blackboard architecture. Data or objects are stored in the Spaces, shared by providers and requesters in the way of implicit invocation. [9]

According to [10], nowadays they are experiencing a renaissance of blackboard architectures with various industry-strength tuplespace architectures, such as IBM TSpaces and JavaSpaces. [10]

For more details about JavaSpaces and T Spaces see [11].

For example: Book buyers post their request to a JavaSpaces.
Interested sellers detect requests and send responses to buyers.

Conclusions

Blackboard architecture has a lot of benefits and therefore can be widely used. The main idea is to try to apply the blackboard architecture for such common issue solution - product life cycle stage definition. This original approach has shown successful results; therefore it can be used even for this issue.

The proposed system of blackboard architecture application for product life cycle stage definition can become an effective helper for manager in his work, it can help to maximize profit, to delay decline stage, etc.

The successful results of proposed system persuade into continuing the study of this approach.

Future work

A theoretical part of blackboard architecture application for product life cycle stage definition is successfully finished and tested, so the next step will be implementation with JavaSpaces.

References

1. For Marketing Learners Globally, http://www.learnmarketing.net/productlifecycle.htm, Last visit day May 2008.
2. Marketing teacher, Est 2000, http://www.marketingteacher.com/Lessons/lesson_plc.htm, Last visit day May 2008.
3. The leading global publisher of e-learning resources for Economics, Business, Politics, Enterprise, Law, Sociology, Religious Studies and related subjects, http://tutor2u.net/business/marketing/products_lifecycle.asp, Last visit day May 2008
4. Online postgraduate courses for the electronic industry, Design and Manufacture of Electronic Systems, Engineering Design, http://www.ami.ac.uk/courses/ami4900_ed/u01/unit_1_sec_2/index.asp, Last visit day May 2008.
5. Rudenko D.; Borisov A. "An overview of blackboard architecture application for real tasks", *Scientific Proceedings Of Riga Technical University, Ser. 5, Computer Science, Information Technology and Management Science. Vol. 31.* RTU, Riga, 2007, pp. 50-57.
6. http://www.nb.net/~javadoug/bb.htm, Last visit day May 2008.
7. Rohan Academic Computing, notes, http://www-rohan.sdsu.edu/~renglish/370/notes/chapt11/index.htm, Last visit day May 2008.
8. Department of Computer Science and Information Engineering, Courses, http://www.csie.ntu.edu.tw/~sylee/courses/clips/examples/theater.htm, Last visit day May 2008.
9. Dong, J.; Chen, S.; Jeng, J.-J. **"**Event-based blackboard architecture for multi-agent systems", *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) – Volume II - Volume 02,* Pages: 379 – 384, 2005.
10. Silva O.; Garcia A.; Lucena C., "The Reflective Blackboard Architectural Pattern: Architecting Large Multi-agent Systems", *Software Engineering for Large-Scale Multi-Agent Systems*, Springer, Volume 2603/2003, pp. 73-93, 2003
11. Wikipedia, free enciclopedia, http://en.wikipedia.org/wiki/JavaSpaces#JavaSpaces.
12. NetMBA, Business Knowledge Center, http://www.netmba.com/marketing/product/lifecycle/, Last visit day May 2008.