# THE CONCEPT OF USING HAMILTON CYCLE IN THE DESIGN AND TESTING LOGISTIC PROCESSES ON THE EXAMPLE OF THE MODULE TENDERS AND TENDERING IN B2B SYSTEM OPTIBUD

ŻUKOWICZ Marek, ŁOBAZIEWICZ Monika

*OPTeam S.A., Jasionka, Poland, EU, mzukowicz@opteam.pl; mlobaziewicz@opteam.pl*

**Abstract**

The main purpose of this article is to show how it can be used Hamiltonian graphs and their property in the software designing and testing. In the article there are described Hamiltonian graphs that are necessary to understand the content of the research conducted in a frame of R&D project called "The prototype of an innovative and technologically advanced OPTIbud B2B platform that supports the management of construction processes through the integration of data and information from multiple sources", funded by the National Research and Development Centre under the INNOTECH program, Hi-Tech path. In the article it is shown the application of the traveling salesman problem in testing. The main problem of using Hamiltonian graphs is explained on the example of Polish construction companies, which use tenders and tendering at the pre-investment step and it is shown on the example of using B2B OPTIbud system. Results of the research show the advantages of using Hamiltonian cycle in software development and testing.

**Keywords:** Hamiltonian graphs, Hamiltonian cycle, regression testing, portability tests, software testing, tendering process.

## 1. INTRODUCTION

The designing and testing software systems and applications is not obvious and simple task [6], [8]. In practice, testers very often do not have time to test an application or a system optimally before it enters to a customer [1]. This problem appears in many IT companies [4]. Thus, it can stay the question: How to design a system or an application and how to write test cases to perfectly check that all functions of the software work properly? How to optimize test data portability and many other attributes in the software?

Whereas designing functionalities of software, there are different solutions for implementing the system, the class module and interfaces for subsequent tests [2], [3]. One of the solutions is using mathematical methods like the Hamilton graphs theory [11], [13]. It turns out that the using of Hamilton graphs property and their application in IT systems improves usability and enhances the testing of certain of functional and non-functional attributes. The method described in the article can be applied at every level of implementation and at each level of testing the tendering module that is a part of B2B OPTIbud system.

## 2. HAMILTONIAN GRAPHS

According to F. Harray Hamiltonian cycle is a cycle in the graph where each nodes of the graph is passed only once (except the first nodes). A Hamilton path (called “The Hamiltonian path”) is the path in the graph that visits each of its nodes exactly once [7]. Thus, graphs which contain the Hamiltonian cycle are called Hamiltonian graphs. Hamiltonian graphs considered in the graph theory contain the path which passes through each node exactly once called the path of Hamilton.

Let $G$ be a graph, $V(G)$ set of its nodes, $E(G)$ set of edges, $|A|$ cardinality, $u_i$ one node i of a graph $G$ and $deg(v)$ the degree of the node (number of edges in it). Notation $\{v,u\}$ is used to determine the edge between $v$ and $u$.

**Theorem 1** [12] If the simple graph with $n$ nodes has at least $m$ edges, where:

$$m = \frac{1}{2} * (n-1) * (n-2) + 2 \quad (1)$$

**Theorem 2** [12] Let $G$ be a graph with $n$ nodes and $C(G)$ be the parent graph built to the rule that for every pair $\{u, v\}$ not connected with the edge where we have condition

$$Deg(u) + Deg(v) \geq n \quad (2)$$

is added the edge $\{u, v\}$. Graph $G$ is Hamiltonian graph if and only if $C(G)$ is Hamiltonian graph.

The idea of first claim is that "graph is Hamiltonian if it has a sufficiently large number of edges". The second theorem says that if in the inside of the graph find a Hamiltonian cycle and the other nodes and edges will satisfy the assumptions of Theorem 2, we also expanded graph is a graph Hamilton. However, there are graphs that do not satisfy the assumptions of any of the above theorems, and contain a cycle or a Hamilton path.

## 3. THE TENDERS AND TENDERING MODULE IN B2B OPTIBUD SYSTEM

In B2B OPTIbud system tenders and tendering module contain functionalities related to the processes for tenders records and stages of preparation of the tender. It has the ability to automatically downloading published tenders for public procurement portals with data filtering functionality [10]. An important element of this module is staging, so it is possible to keep the tracking progress, introducing and analyzing the protocols of opening tenders, collecting information about the winnings / losses tenders and contractors. At the level of a creation process it is able to adding estimate / budget to the tendering. The module is linked with the outside system (Public Tender Bulletin) and it makes to be possible to assume building cards of winning the tender. [9]

## 4. USING HAMILTONIAN CYCLE IN A DESIGN OF STATUS CHANGES RELATED TO SOFTWARE FUNCTIONALITIES ON THE EXAMPLE OF B2B OPTIBUD SYSTEM

In many cases the investment process begins from the tendering. According to the tendering process the functional module in the software must be complied with at least the following conditions [10]:

- The tender is added to the system manually or imported from a web page.
- The tender proceeds to project in preparation.
- The project in preparation proceeds to a submitted project.
- The submitted project can be moved to the project in preparation.
- The submitted project can be rejected, cancelled, lost, but the company can win the submitted project too and then it becomes in accepted project.
- The accepted project can be rejected or canceled.
- The canceled, lost or rejected project can be moved into project in preparation.

A graph showing the status of tenders/projects in the OPTIbud system is in Figure 1.

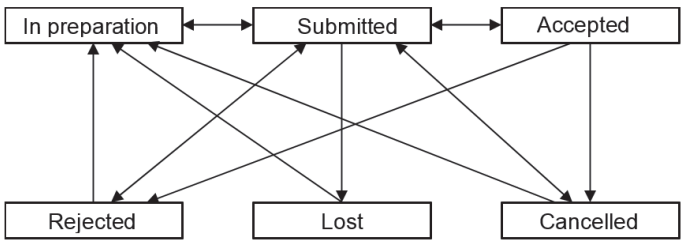


**Fig. 1** Graph showing the status of tenders (1)

This graph is not the Hamiltonian graph. It is not against the law to allow rejecting the project in preparation stage. If one allows such possibility, the graph showing the status of tenders will take the following form (see Fig. 2).

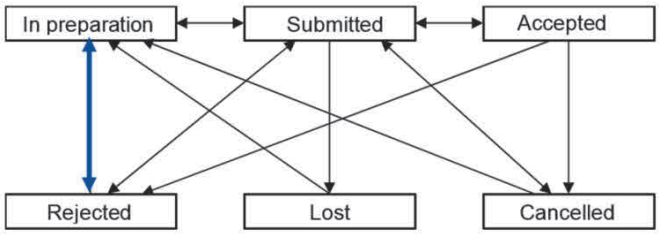


**Fig. 2** Graph showing the status of tenders (2)

Now one should check whether the above graph satisfies the assumptions of theorem 1. A graph is simple. Since there are six nodes it should contain 12 edges (see Equation 3).

$$m = \frac{1}{2} * 5 * 4 + 2 = 12 \quad (3)$$

Taking into account the theorem 1, it can stay that there is in the Hamiltonian cycle. This cycle is the following sequence of steps:

Lost → In preparation → Rejected → Submitted → Accepted → Cancelled.

Now, supposing that a company operates in such a way that in case of cancellation, rejecting and losing of the tender wants to be able to close the tender. Closing of the tender means that it is about the information in the database, but the company will do nothing with this tender for some period of time. After that, however, the company can open tender and return to the project in preparation. It can for example add closing of the project from rejected, lost or cancelled project. Then the data flow graph takes the form as in Figure 3.

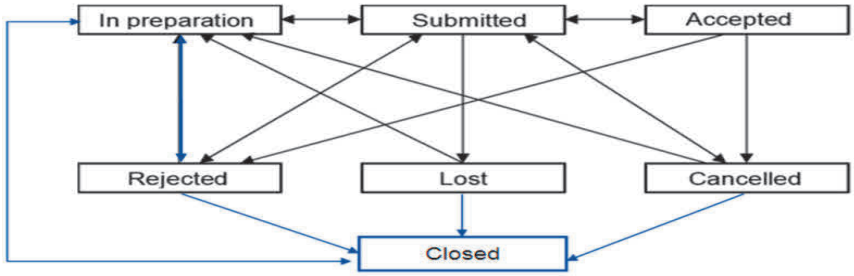


**Fig. 3** Graph showing the status of tenders after adding closing of the tender

It turned out that paths have been added that satisfy the objectives of theorem 2. The extensive graph will remain Hamiltonian graph too. It can be chosen one Hamiltonian cycle:

Lost → In Preparation → Rejected → Submitted → Accepted → Cancelled → Closed

## 5. CONCLUSIONS

Designing of systems and applications for the existence of the Hamilton cycles in certain areas or modules save the time while testing the logistics processes. This remark stems from the fact that the tester must make fewer movements, if it can perform the Hamiltonian cycle in the tests, especially in manual testing. According to Bluemke and Kiermasz a lot of testing around the world are performed manually, thus saving time is an extremely important thing [5]. If the test data flow will be positive, it means that the data transferred properly

and there is no gap or defect between function pass through states. The reduction in run-time or test design and reduces the amount of these tests is valued in companies that place a strong emphasis on testing. This approach is also very beneficial when designing and writing automated tests, because the validation data transfer is written, one can write once, after passing the Hamilton cycle and thus also shortened the time of writing these tests (test cases). Theorem 3, which is provided in chapter related to Hamiltonian graphs, can be used to create new functionality (or some states) to the system or software. It can be created new functions so that the Hamiltonian path will exist after implemented in software. Then this software will have advantages such as are described in the preceding paragraph.

## ACKNOWLEDGEMENTS

***The paper presents the part of R&D project called "The prototype of an innovative and technologically advanced OPTIbud B2B platform that supports the management of construction processes through the integration of data and information from multiple sources", funded by the National Research and Development Centre under the INNOTECH program, Hi-Tech path.***

## REFERENCES

[1] ADAMS B., ENGSTROM W., KHOMH F., MANTYLA M. V. PETERSEN K. On rapid releases and software testing: a case study and a semi-systematic literature review. Empirical Software Engineering, Vol. 20, No. 5, 2015, pp. 1384-1425.

[2] ALSHRAIDEH M., BOTTACI L. Search-based software test data generation for string data using program-specific search operators. Software Testing, Verification and Reliability, Vol.16, No. 3, 2006, pp. 175-203.

[3] ARCURI A., It really does matter how you normalize the branch distance in search-based software testing. Software Testing, Verification and Reliability, Vol. 23, No. 2, 2013, pp. 119-147.

[4] BERNER S., WEBER R., KELLER R.K. Observations and lessons learned from automated testing. Proceedings of the 27th international conference on Software engineering, 2015, pp. 571-579.

[5] BLUEMKE I., KIERMASZ W. Kompozycja i integracja usług w architekturze SOA, Inżynieria oprogramowania w procesach integracji systemów informatycznych. Gdańsk: PWNT, 2011.

[6] HARMAN M, JONES B.F. Search-based software engineering. Information and Software Technology 2001, pp. 833-839.

[7] HARRAY F., Graph theory. Addison - Wesley, 1969.

[8] HUANG C., Performance analysis of software reliability growth models with testing-effort and change-point. Journal of Systems and Software, Vol. 76, No. 2, 2005, pp. 181-194.

[9] ŁOBAZIEWICZ M., Standard architektury modelu systemu B2B wspomagającego zarządzanie procesami budowlanymi. In KOSIUCZENKO P., ŚMIAŁEK M., SWACHA J.: Od procesów do oprogramowania: badania i praktyka. Wydawnictwo Polskie Towarzystwo Informatyczne, Warszawa 2015, Part 8, pp. 111-120.

[10] OPTEAM S.A. Raport 1 z badań „Prototyp innowacyjnej i zaawansowanej technologicznie platformy B2B OPTIbud, wspomagającej zarządzanie procesami budowlanymi, poprzez integrację danych i informacji z wielu źródeł".

[11] ROMAN A., Testowanie i jakość oprogramowania. Warszawa: Wydawnictwo Polskie Wydawnictwo Naukowe, 2015.

[12] WILSON R., Wprowadzenie do teorii Grafów, Warszawa: Wydawnictwo Polskie Wydawnictwo Naukowe, 2000.

[13] ŻUKOWICZ M., Testowanie oprogramowania: Uzasadnienie potrzeby dydaktyki oraz tworzenia strategii testowania wykorzystując wnioski z teorii „No Free Lunch". General and Professional Education, Vol. 3, No. 8, 2015, pp. 46-51.