Aeronautical Journal, Vol. 121, Issue 1242, August 2017, pp. 1051-1065
DOI:10.1017/aer.2017.50

Remote Operation and Monitoring of a Micro Aero Gas Turbine

Michail Diakostefanis, Research Fellow
m.diakostefanis@cranfield.ac.uk
Cranfield University
SATM
Propulsion Engineering Centre
Cranfield
United Kingdom

ABSTRACT

Internet applications have been extended to various aspects of everyday life and offer services of high reliability and security at relatively low cost. This project presents the design of a reliable, safe and secure software system for real time remote operation and monitoring of an aero gas turbine with utilisation of existing internet technology, whilst the gas turbine is installed in a remote test facility

This project introduces a capability that allows remote and flexible operation of an aero gas turbine throughout the whole operational envelope, as required by the user at low cost, by exploiting the available Internet technology. Remote operation of the gas turbine can be combined with other remote Internet applications to provide very powerful gas turbine performance simulation experimental platforms and real time performance monitoring tools, whilst keeping the implementation cost at low levels.

The gas turbine used in this experiment is an AMT Netherlands Olympus micro gas turbine and a spiral model approach was applied for the software. The whole process was driven by risk mitigation.

The outcome is a fully functional software application that enables remote operation of the micro gas turbine whilst constantly monitors the performance of the engine according to basic gas turbine control theory. The application is very flexible, as it runs with no local installation requirements and includes provisions for expansion and collaboration with other online performance simulation and diagnostic tools.

Keywords: remote operation; monitoring; aero gas turbine, spiral model, formal validation methods, risk assessment, Internet application

NOMENCLATURE

ECU	Electronic Control Unit
EGT	Exhaust Gas Temperature
FAA	Federal Aviation Administration
FMEA	Failure Mode And Event Analysis
FTA	Fault Tree Analysis
HAZOP	Hazard And Operability
RPM	Revolutions Per Minute
SSL	Secure Sockets Layer
XML	Extensive Mark Up Language

Symbols

m	Times of execution
n	Number of states
pij	Transition probability
R	Reliability for single execution
E(R)	Overall reliability of a system
si	State
t	Time
θ	Failure Rate
σ	Standard Deviation

1.0 INTRODUCTION

Internet applications are gaining increasing popularity during the last decades and they are used in many aspects of everyday life, such as web banking, online shopping and mail exchange. Internet technology is available worldwide at low cost. The aim of this project is to introduce a software application which will enable safe, secure and reliable remote operation and monitoring of an aero gas turbine situated in a test facility, through the Internet. The end user will be able to perform various tests on the gas turbine from anywhere, without any local installation prerequisites at the user's computer or any other Internet access device.

1.1 Relevant applications and problem definition

In the aviation field, there are applications based on internet technology that allow the management of the technical operations of an airline fleet. Operations as quality monitoring, reliability trends and status reports can be obtained via web services[(1)]. Manage/m, a Lufthansa Technik developed MRO websuite can provide troubleshooting support by using data of the aircraft and its engines condition from on – board computers, even when the aircraft is airborne.

1.1.1 Industrial Applications

In the industrial sector, similar applications based on internet technology can be found in power plant systems management. An example of that is the Siemens SPPA – T3000 Control System for power plant management and control. As presented by Siemens, this system provides real time data at any place and any time, single software system imaging and simplification of the system architecture, rendering it easy to learn and use[(2)]. According to a research study from Karlsruhe University Germany, supervised by Siemens and available in the public domain, SPPA – 3000 is based on Java

and XML and the application server can be accessed by any web browser. The web page of the application server consists of several Java applets.

1.1.2 Academic Applications

Similar internet applications are very common in the Academic Field as useful tools for the remote creation of simulation models and even real time conduction of control experiments with internet operation of mechanical or electronic devices. An example is the NCSLab, developed by Glammorgan University. It is a Control Engineering experiment platform which consists of 6-tier architecture, based on MATLAB – Simulink. Several other control engineering web based platforms have been developed around the world such as a web – based laboratory for control experiments on a coupled tank apparatus by Ko et. al. (2001)[3] and WebLab Development for Kyatera Network by Okajima et. al. (2006)[4]. The last 2 applications use downloaded java applets on the client side in order to operate remotely.

1.1.3 TURBOMATCH WebEngine

The TURBOMATCH WebEngine is an Internet application developed by Cranfield University, for the creation of gas turbine performance simulation models[5]. It utilises TURBOMATCH in the server side, a powerful gas turbine performance simulation tool developed by Cranfield University as well. The WebEngine offers ease of use with no local installations required. It is based on client – server architecture, with TURBOMATCH as the core solver behind the GUI. TURBOMATCH is a software based Gas Turbine performance simulation tool developed by the Propulsion Engineering Centre in Cranfield University. It was developed in a modular zero-dimensional structure, meaning that every existing or conceptual engine can be thermodynamically described by a sequence of pre-programmed components that represent the actual intakes, compressors, combustors, turbines, nozzles, ducts, bleed valves and mixers. Every component unit, referred as Brick, is in reality a routine capable of calculating the thermodynamic process that takes place through the equivalent engine part. Certain inputs are required for every Brick, such as rotational speed, isentropic efficiency, pressure ratio or pressure losses[5].

1.2 Significance

This project introduces a novel application for real-time remote operation and monitoring of aero gas turbines through the Internet, with standard browsers as thin clients and without prerequisites of additional installed software components at the user's end. It demonstrates how the underlying physical risks, Internet security and network performance can be weighted and combined accordingly to provide a reliable tool. The outcome is a flexible system that can be integrated with Internet gas turbine performance simulation applications, providing a powerful experimental and testing platform.

1.3 Olympus Micro Gas Turbine

The gas turbine used for the project is an Olympus micro gas turbine manufactured by AMT Netherlands (Figure 1), used in radio controlled aircraft, gliders, remote heat/power generators or auxiliary power units[6]. It is a miniature single spool turbojet gas turbine with an overall length of 270 mm, maximum diameter of 130 mm and maximum thrust of 190 N at maximum RPM (110,000). The single stage centrifugal compressor delivers a pressure ratio of 4:1 and handles a maximum value of mass flow 400 gr/sec[7]. It is controlled by a closed loop control Electronic Control Unit (ECU).

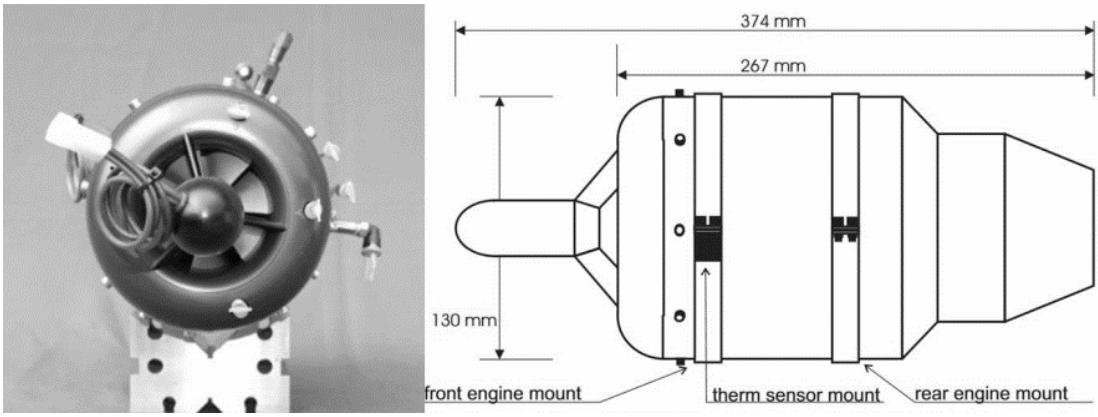


Figure 1. Olympus gas turbine front view and main dimensions[7]

2.0 METHODOLOGY

The development of a software system is described by the software process, which is a group of activities leading to the production of a software product[8]. The software process can be described by appropriate models. The process herein followed the Spiral Model (Figure 2), introduced by Boehm (1988)[9] with the purpose of overcoming the difficulties imposed by the techniques that existed up to then, such as the waterfall model, the evolutionary and the transform model. The spiral differs from the other software development models mainly in the identification and consideration of risk[8].

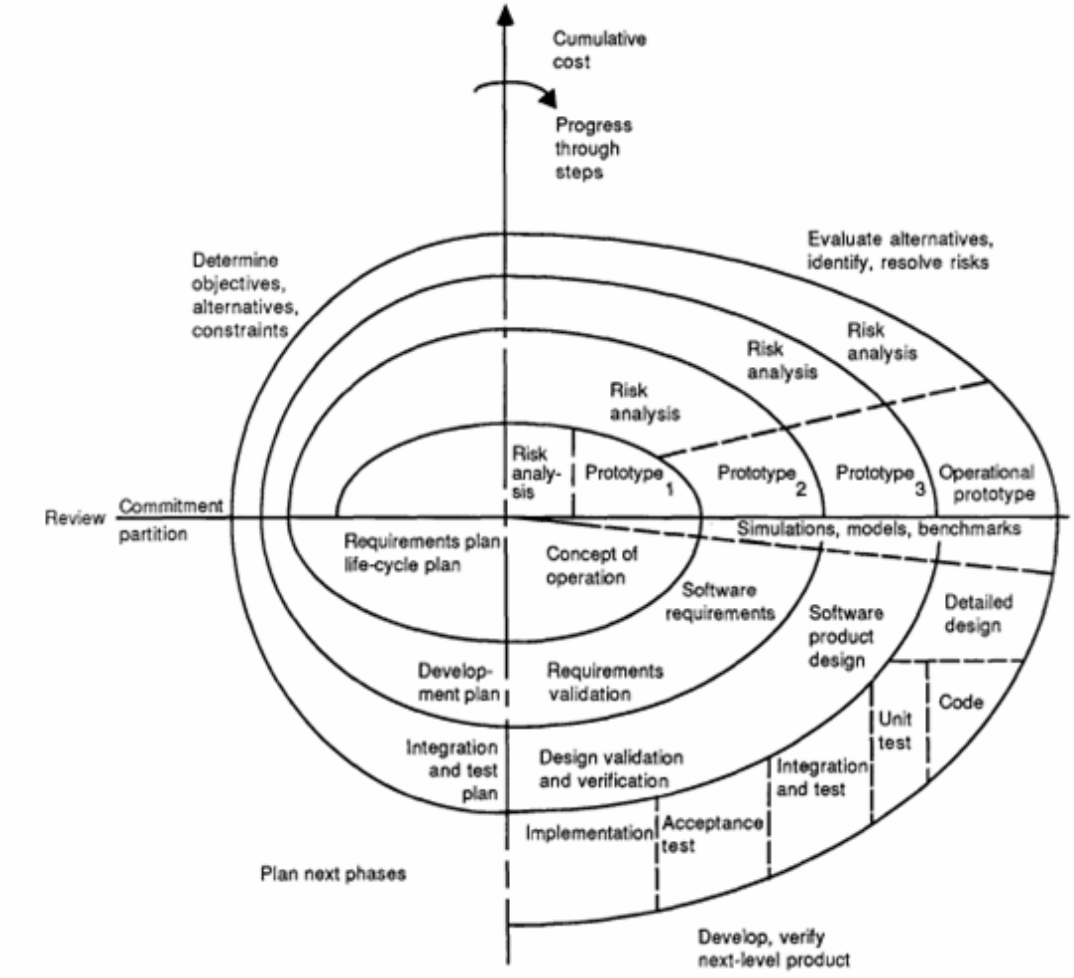


Figure 2. The Spiral Model[9]

For this application there were 6 phases (rounds) considered:

- Round 0: Feasibility study and System Requirements definition
- Round 1: Local Interfacing of the gas turbine
- Round 2: Design of operation control software
- Round 3: Implementation of additional performance and safety features
- Round 4: Internet application design
- Round 5: Acceptance validation

2.1 Definition of the Application and Validation

The requirements elicitation process included use – case scenarios, following the requirement elicitation template of Dependable Product Lines, proposed by Gallina et. al. (2007)[10], which includes among others, fault assumptions and risk analysis. The System Requirements were classified in different modes of operation and expressed in structured natural language, in order to avoid the ambiguities of natural language[11].

Validation took place after the completion of each individual development phase. The criteria for evaluation were set in accordance with the RTCA/DO-178 document guidelines (Software Considerations in Airborne Systems and Equipment Certification), used by software developers in avionics in order to obtain FAA approval[12]. The DO-178 document includes guidelines for the software lifecycle, the planning and development process, verification, configuration management and quality assurance. An important topic within the document is the definition of criticality levels of software, determined according to the extent of the damage caused in the case of a failure, ranging from E (no effect) to A (catastrophic)[13] (Figure 3). For this project criticality level C was considered. Criticality level C implies major damage to components – in this case the gas turbine – and this was determined by applying an analytic Event Tree Analysis (ETA) to investigate the potential outcomes of a situation where the remote operation control capability of the engine was inadequate or totally lost. The criticality level determines the depth of the validation tests required for certification[12] hence the amount and type of the test cases applied are directly affected by this. Criticality level C requires testing of the software to ensure total statement coverage.

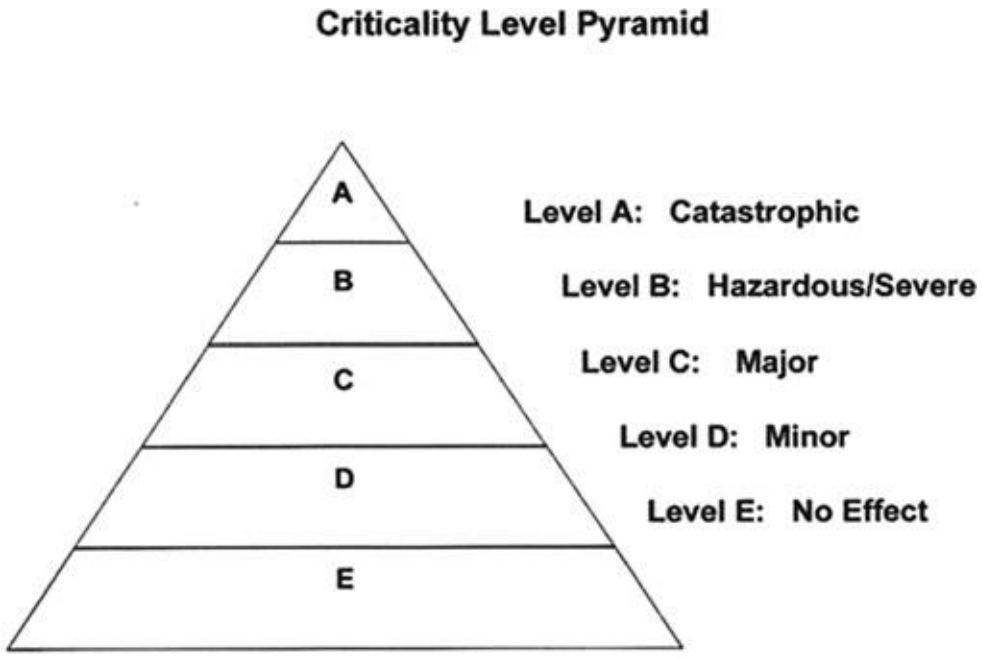


Figure 3. Criticality Levels according to DO-178B[13]

The final validation included two main tasks:

- Functionality validation
- Probabilistic reliability analysis

Functionality validation was obtained with the use of a formal validation method (Model Checking) to define all the possible test cases. Model checking is commonly applied in critical software systems. In contrast to simulation, formal verification methods do not rely upon the dynamic response of the system, but perform static analysis on a formal mathematical representation of the system, in order to check it for correctness with respect to a given specification. Model checking has been gaining popularity in aerospace systems design and verifications over the last few years[14]. It is a powerful approach for the formal verification and validation of software. It automatically provides complete proofs of correctness, or explains, via counter-examples, why a system is not correct[15].

Probabilistic reliability analysis relied on Markov equations, which is an approach based on a state diagram and applicable in risk assessment of power systems[16]. Having assumed reliabilities of individual components to be known, Gokhale et. al. (2002)[17] stated that the overall reliability of the examined system for single execution is given by:

$$R = \prod_{i=1}^{n} R_i^{X_{1,i}} \quad \ldots(1)$$

The number of visits to each component is a random variable, implying that reliability also is a random variable. With the use of Taylor series they concluded that the expected reliability of the system is given by:

$$E[R] = \left[\prod_{i=1}^{n-1}\left(R_i^{m_{1,i}} + \frac{1}{2}\left(R_i^{m_{1,i}}\right)(logR_i)^2\sigma_{1,i}^2\right)\right]R_n \quad \ldots(2)$$

By ignoring the second order architectural effects Equation 2 can be reduced to the following expression:

$$E[R] \approx \prod_{i=1}^{n-1} R_i^{m1,i} R_n \quad \ldots(3)$$

From the results of the study of Gokhale et. al. (2002)[17], it was observed that the difference of reliability estimation was of 3rd decimal magnitude. Thus for the preliminary reliability analysis of the designed application, Equation 3 was considered to be adequate and hence applied. When the failure rate is constant, the reliability function can be expressed exponentially in terms of time t and failure rate θ[18]:

$$R(t) = e^{-\theta t} \quad \ldots(4)$$

2.2 Risk Assessment

A variety of risk assessment methods were implemented, according to the requirements of each phase of development of the application. Risk assessment in the Spiral model development is evident at the initiation of each round. Fault Tree Analysis[19] and HAZOP[20] were applied at the early stages of the development. FTA (Figure 4) is often found in power plant, aerospace, chemical and nuclear systems (FTA)[21][22]. This method is based on the identification of a top undesirable event that corresponds to a particular failure mode and consists of the individual events that contribute to the top event. FTA is a qualitative method in its nature but it can also provide a quantitative aspect as well[23].

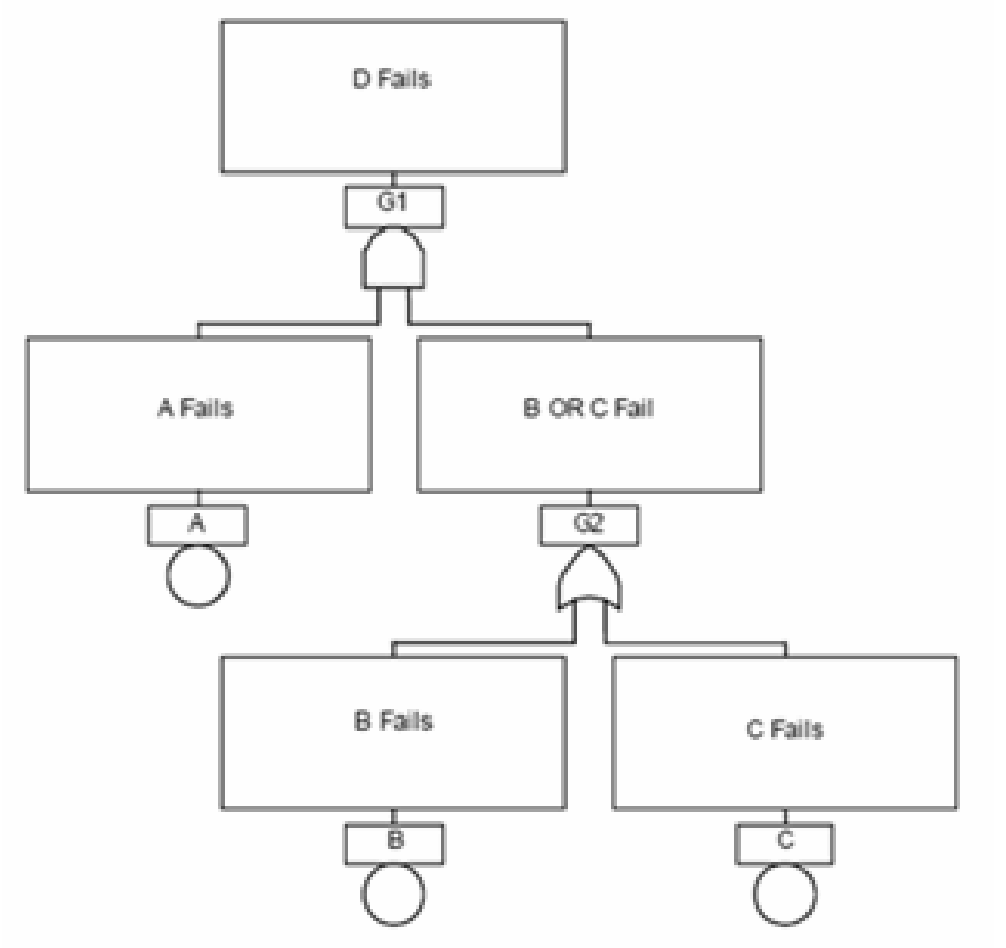


Figure 4. A simplified FTA tree[23]

Hazard and Operability (HAZOP) analysis is a method mainly used by the chemical industry[21] and it deals with the identification of deviation from the design intent, also investigating the causes and assessing their consequences[20]. It was used as an additional tool to explicitly address the risks identified by FTA.

In the first phase of the development (round 0), a general risk assessment was conducted with FTA. Two major undesirable situations were considered: Total loss of command of the gas turbine and sending erroneous command orders. All the individual events that could contribute to the major undesirable situations were identified and Boolean algebra was applied in order to determine the most likely combinations to occur. The major risks that were identified and mitigation actions are listed in **Table 1**.

.

Table 1
Mitigation of major risks

Risk source	Mitigation action
Loss of power supply	Power stabiliser and monitoring of mains supply
Network connection loss	Automated action upon detection of connection loss
Internet threats	Form-based authentication and encrypted information over SSL, main application hidden behind firewall via reverse proxy
Misuse	Prevention of abrupt user inputs by server, acknowledgment of command reception by server
Loss of intercommunication or camera image	Automated shutdown if system inactive for more than 3 minutes, 2 observers placed to monitor the system

In the subsequent phases Failure Modes and Effect Analysis (FMEA) was applied for physical and procedural risk assessment. FMEA is a method for the analysis of a system in order to identify the potential failure modes, their causes and any effects on the system performance[24]. FMEA is extremely efficient when it is applied to the analysis of elements that cause a failure of the entire system or of a major function of the system. It can be used alone or as a systematic inductive method of analysis, to complement other approaches, especially deductive ones, such as FTA. It is found in nuclear, chemical[21] and other applications

In all the following phases the risks were classified in two categories: physical and procedural. The former were those risks associated with the condition and operation of mechanical components, instrumentation, computers and software modules. The latter were the risks associated with the planning, the equipment acquisition and the software development.

2.3 Performance Monitoring

The performance of the gas turbine during remote operation is constantly monitored by a dedicated module. This module observes the critical performance parameters (currently EGT and RPM) and generates warning messages or executes automatic safety actions if necessary. The assessment of the critical parameters is based on engine control systems theory.

Control systems apply alarm rules in order to detect values above certain limits during the engine operation. There are four unusual patterns of data points:

- The instability pattern, where points are present outside the control limits,
- The mixture pattern, characterised by the absence of points near the centreline,
- The stratification pattern, with absence of points near the control limits and
- The trend pattern, where an upward or downward trend is present.

The most noticeable for the present study is the instability pattern, which indicates the existence of data points outside the control limits. The rules which are most widely applied for the classification are the Western Electric rules (Figure 5). The control band is divided into three zones where zone A is

the area enclosed within 2 and 3 standard deviations (σ) of the parameter, zone B, which stays between 1σ and 2σ and C, between ±1σ. A guideline for the alarm generation is: any point beyond zone A, 2 out of 3 consecutive points in zone A, 3 out of 5 consecutive points in zone B and 8 consecutive points on the same side of the centreline[25].

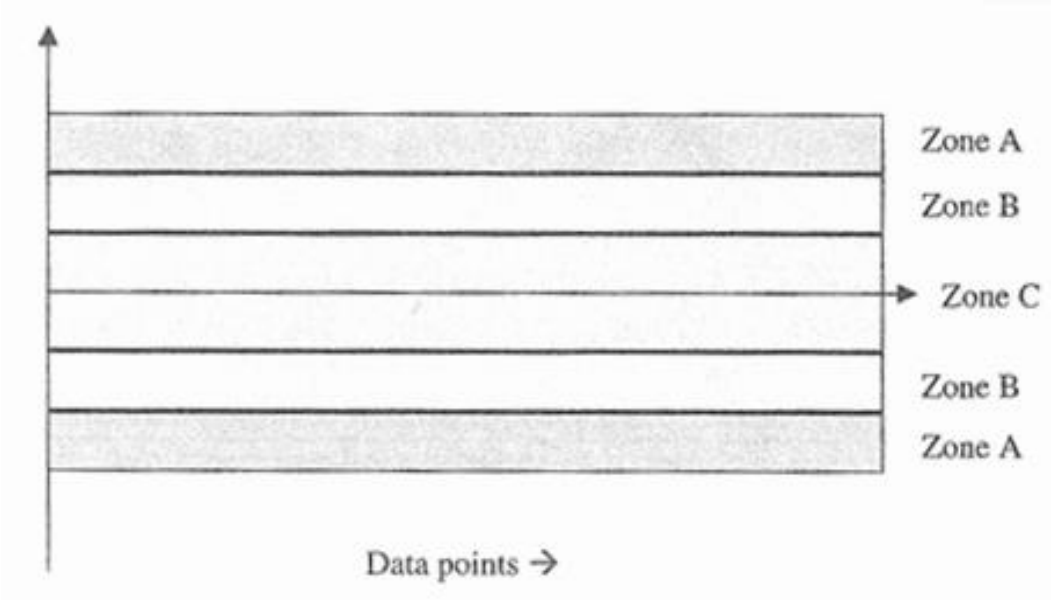


Figure 5. Western electric rules[25]

2.4 Security over the Internet

The web application was configured in a modular manner. The lowest tiers of the architecture are not exposed directly to the Internet, as they remain situated behind the firewall of the University. An Apache Hyper Text Transfer Protocol (HTTP) server has been configured as a Reverse Proxy[26] and installed as the front end to the web applications. Virtual hosts have been designed to map the requests and redirect to the web applications. The virtual servers can be installed in the De-Militarised Zone (DMZ) of the IT Department of the University and accessed through the Virtual Private Network (VPN) tunnel. The web applications can be deployed in the local server behind the firewall. The application implements the Secure Socket Layer (SSL) protocol for data exchange. This is a cryptographic protocol that provides encryption of the data flowing between the end parts. Additionally, the back end of the web application (local server), implements a form-based user authentication[27].

3.0 APPLICATION STRUCTURE AND FUNCTIONALITY

The application was designed in a way to satisfy flexibility in the operation of the gas turbine, but optimised for security and safety. The architectural structure and implementation of the software was accomplished in adherence to an appropriate standard for software quality metrics, such as functionality, reliability, usability, efficiency, maintainability and portability[8]. It can be accessed by any standard web browser in various operating systems.

3.1 Architectural Structure

The application is structured as a client – server model. The requested services can clearly be distinguished and associated with a set of servers and clients who use them. Distributed architecture allows effective use of networked systems with many processors. It is also allows addition and

integration of more servers[8]. This architecture easily allows the inclusion of additional gas turbines in future evolution of the project. Also, changes in any sub – system can be isolated, thus preventing any effect on other components, enhancing changeability and maintainability of the designed software. The system comprises a multiple layer structure, which is more scalable than simple 2-tier architecture and also reduces network traffic[8]. The local server that is connected with the engine and the hardware is well hidden behind the reverse proxy and the intermediate firewalls (Figure 6).

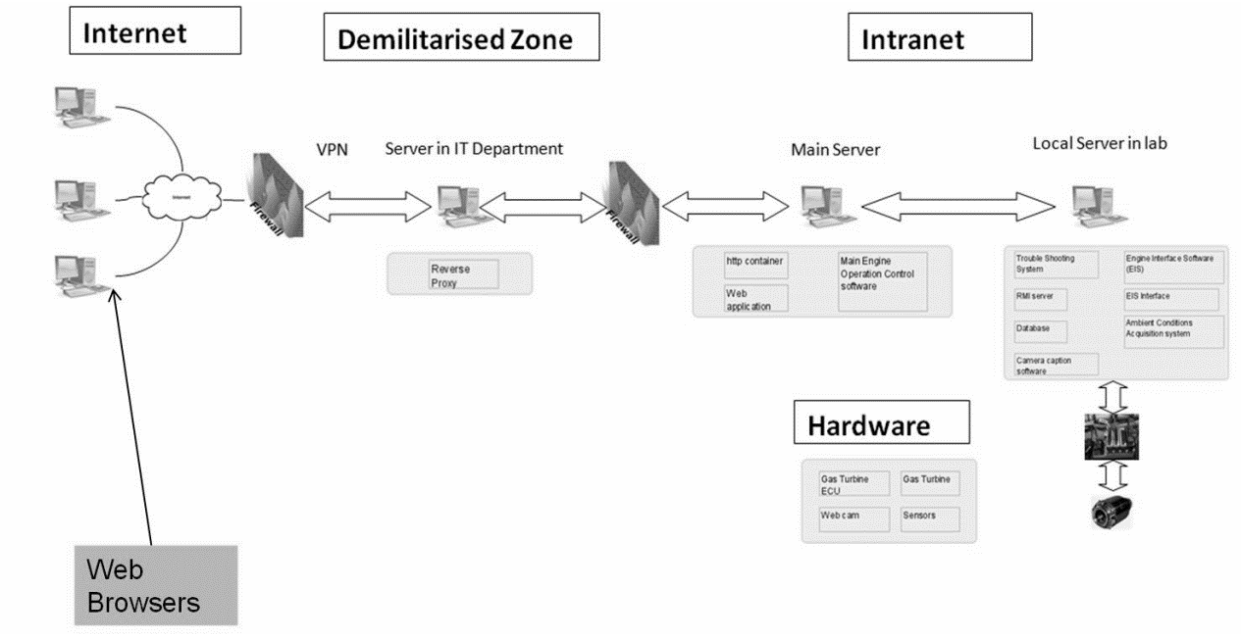


Figure 6. Architectural structure of the Internet Gas Turbine Application and arrangement of front and back end servers

3.2 Functionality

An ergonomic web page has been designed for the remote users (Figure 7). The hosting website also provides access to a training web page that explains the functionality of each item on the operation control web page.

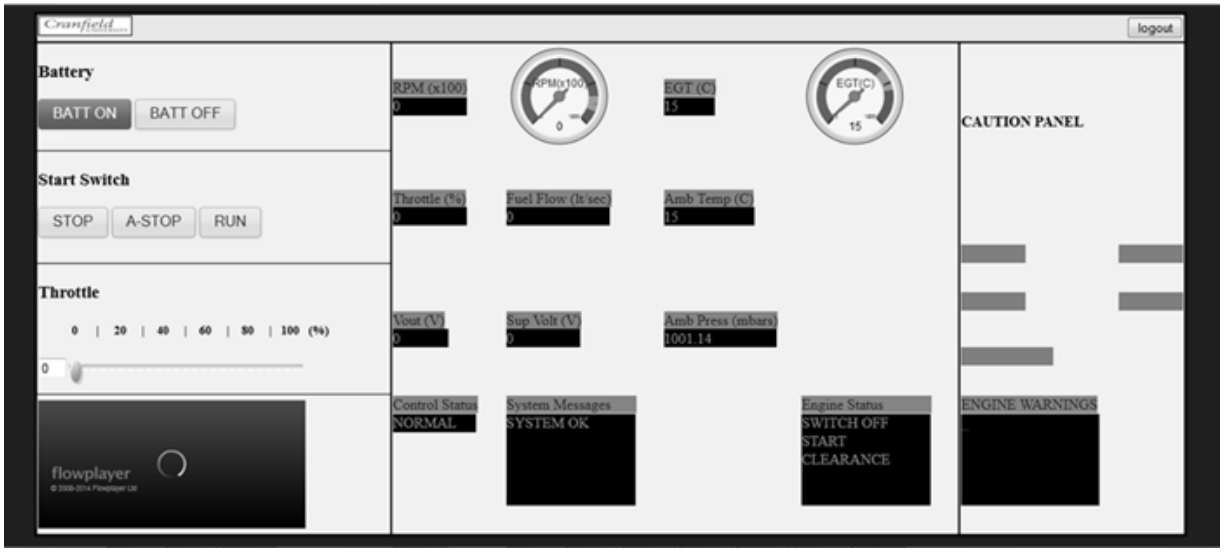


Figure 7. Engine operation panel web page

The operation panel page was designed ergonomically with control, indications and warnings placed in 3 separate columns. The reception of a command on the server side is acknowledged by the illumination of a green indication underneath the corresponding control widget. For the throttle, if a new setting is greater than 20% from the previous is selected, the command will be rejected at the server side and the response will trigger the illumination of a red indication light under the throttle bar. Other main functionality features include the following:

- Only one user at a time is allowed with full privileges to operate the engine. Monitoring can be available for administrators
- Updates asynchronously only the fields necessary and not the whole web page. This is achieved by implementation of asynchronous JavaScript XML (AJAX) technology
- If the user's webpage has been inactive for 3 minutes the system shuts down the gas turbine automatically and resets the application to accept a new user
- If connection is lost, the system shuts down the gas turbine automatically within 7 seconds
- Monitoring of EGT and RPM. If a relative limit exceedance is observed (value higher than 2 x σ), the user is given a caution message and if no action taken within 3 minutes, the system shuts down the gas turbine automatically. If an absolute limit exceedance is observed (manufacturer upper limits), the user receives a warning message and the system shuts down the gas turbine automatically within 5 seconds
- Monitoring of the mains power supply at the test house where the engine is installed. If any disruption is noticed, the user receives a message and a 3 minute time space is allowed before the system shuts down the gas turbine automatically

3.3 Results of Validation

The indications of the Olympus gas turbine were captured from the remote operation control software with high accuracy. The RPM indication was totally matched (100%) throughout the whole throttle range. The EGT was nearly 100% matched, with deviation less 1% (Figure 8). This deviation was due to higher precision of representation of the numerical values at the remote operation software. The same levels of accuracy were obtained for all other secondary indications of the gas turbine.

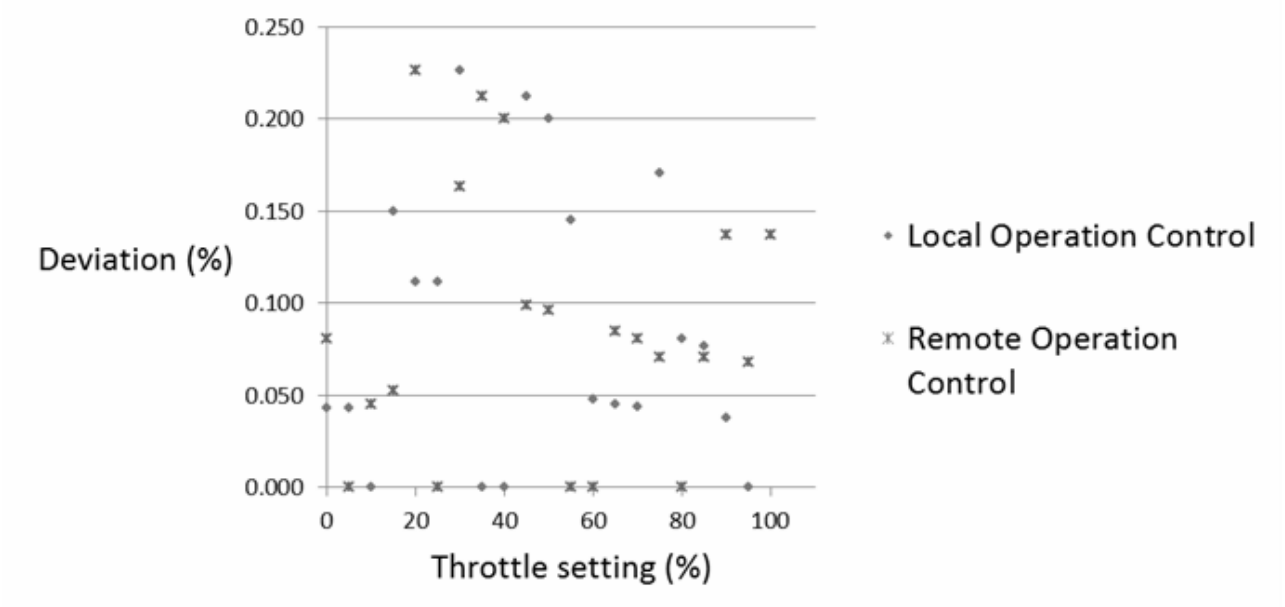
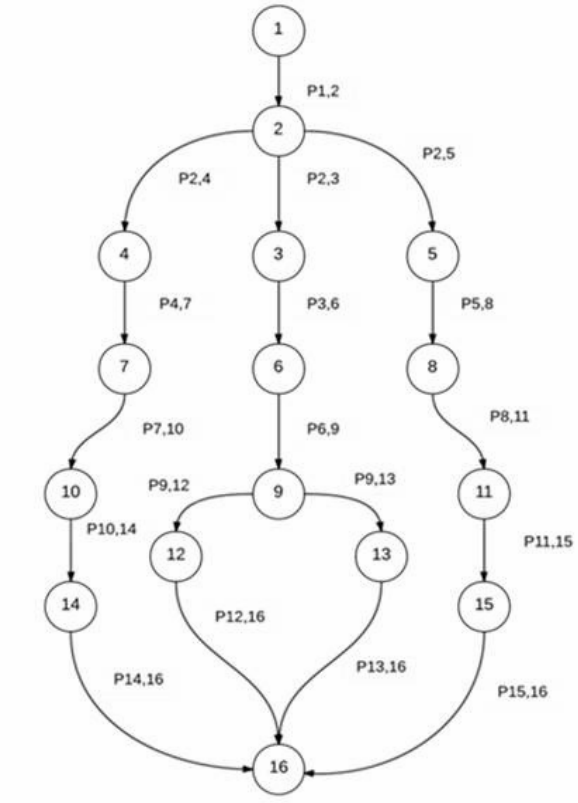


Figure 8. Accuracy of capturing EGT value with remote operation control software

There were a total of 168 test cases derived from the Model Checking tool (NuSMV) that was applied for the final validation process. These cases not only covered a full engine test sequence, but tested every possible function that could be done from the application remotely, as well as all the automated safety features. A dedicated software platform was also created in order to imitate the functionality of the gas turbine and transmit all the possible fault codes in the same form as they are transmitted from the gas turbine Engine Control Unit (ECU). The test platform simulated the performance of the actual engine through field data obtained at various operating conditions, referred to International Standard Atmosphere Sea Level (ISASL) conditions.

Accomplishment of all the defined test cases confirmed 100% statement coverage of the designed software, which is the minimum requirement for criticality level C according to DO-178[(12)]. Preventively, the test cases were designed to also ensure 100% decision coverage, although this is a requirement for higher criticality levels.

A preliminary estimation of reliability was accomplished by implementation of Equations 1 through 4. The system was represented by a finite absorbing Discrete Time Markov Chain (28), constructed by the software and hardware components that form the application (Figure 9). Initial results showed an expected reliability of around 0.5, due to low reliability of components s14 and s15 during starting up of the application. Introduction of time delay (500ms) in the starting up sequenced, reducing the number of experienced failures and improved reliability. There were no other significant failures observed during the validation testing period.



Component	Description	Reliability
s1	Reverse Proxy	0.99
s2	Glassfish Web Server	0.99
s3	Command REST service	0.99
s4	Engine data REST service	0.99
s5	Peripheral data REST service	0.99
s6	MEOCS command thread	0.99
s7	MEOCS engine data thread	0.99
s8	MEOCS peripheral data thread	0.99
s9	EISI command thread	0.99
s10	EISI engine data thread	0.99
s11	EISI peripheral data thread	0.99
s12	EIS switch signal generation	0.99
s13	EIS throttle signal generation	0.99
s14	EIS data reception	0.37
s15	ACAS data reception	0.37
s16	Integrated hardware	0.99

Figure 9. Representation of the Internet Gas Turbine application as a Discrete Time Markov Chain (DTMC)

4.0 CONCLUSIONS – FUTURE PERSPECTIVE

The design took into account the main features of operation and safety of a gas turbine, and combined them with Internet security and performance elements to eventually produce a system that operates successfully and reliably. The final result of this work was an application for real-time remote operation and monitoring of aero gas turbines through the Internet, with standard browsers, without prerequisites of additional installed software components at the user's end.

The new application has provisions and capabilities of expansion and collaboration with other online applications, in order to form powerful experimental and operation facilities at low cost, as the majority of the software tools used was open source. It also forms a basis that can be used to develop real – time online diagnostic applications that implement different health monitoring techniques. It currently monitors gas path parameters but integration of appropriate instrumentation will allow inclusion of additional diagnostic methods by monitoring vibration or shaft dynamics. An integrated health monitoring solution combined with the relatively low cost of development can have a direct impact on gas turbine reliability and availability, accompanied by a positive effect on the overall economic performance of the aircraft on which the monitored gas turbines may be installed.

The system includes features to ensure physical safety and mitigation of Internet-based risks. The underlying risks were identified throughout the software process and all the major risks were mitigated accordingly. The application has adhered to various quality and safety guidance standards and was validated in accordance to DO-178, as per FAA requirements. The reliability was examined and potential bottlenecks were identified, leading to corrective actions that alleviated observed weaknesses.

The indications of the gas turbine indication system were accurately captured and transferred into the software modules. Finally, the developed software modules have demonstrated sufficient abidance to software quality metrics, as defined by relevant standards.

4.1 Future perspective

The Internet Gas Turbine application includes provision for several expansion and evolution capabilities. This has been obtained by the modular architecture of the software design, which allows modifications of certain modules only in order to accommodate new features. The use of software patterns and interface classes enhances the flexibility of the program in terms of inserting additional functionality. Analytic plans for each expected expansion capability have been developed thus providing guidelines for future researchers to easily adapt the existing software. The application is also accompanied by full developer's documentation according to relevant standards. The main future capabilities are listed below:

- Installation of a variety of instrumentation to enable real time monitoring of more engine parameters on the existing micro gas turbine and allow full automation of operation
- Addition of multiple gas turbines of different types that can be operated independently
- Integration with the TURBOMATCH WebEngine online performance simulation. This will enable instantaneous validation of performance simulation models online
- Collaboration with software modules for real time diagnostic or trend analysis of the engine
- Operation of UAV's with the use of standard Internet technology and Wi-Fi transmission

ACKNOWLEDGMENTS

The completion of this project was accomplished with the support of a team that provided essential guidance throughout the whole duration of the development. Dr Stuart Barnes set the basis for the software engineering methodology and Dr Mark Stilwell supported the internet related part of the work. However, this project could not have been accomplished without the inspiration and motivation from the Head of Propulsion Engineering Centre in Cranfield University, Pr Perikles Pilides.

REFERENCES

1. LUFTHANSA TECHNIK (2016), *manage/m Technical Operations Suite*, available at: www.manage-m.com/managem_page_about (accessed 21/09).
2. SIEMENS (2016), *The Benchmark in Controls – Technical Highlights Siemens Power Plant Automation™ – SPPA-T3000*, available at: www.energy.siemens.com/hq/pool/hq/automation/automation-control-pg/sppa-t3000/T3-B-ContrSys-us-V11.pdf (accessed 21/09).
3. KO, C. C., CHEN, B. M., JIANPING CHEN, ZHUANG, Y. AND CHEN TAN, K. (2001), "Development of a web-based laboratory for control experiments on a coupled tank apparatus", Education, IEEE Transactions on, vol. 44, no. 1, pp. 76-86.
4. OKAJIMA, H., S., S., LEDEL, L., C., FRAGNITO, H., L. AND ROCHA, H., V. (2006), "WebLab Development Using a Java and LabView Integrated Solution for Kyatera Network", *3rd TIDIA Fapesp Workshop,* 15 -17 November, Sao Paolo, Brasil, pp. 204.
5. APOSTOLIDIS, A., SAMPATH, S., LASKARIDIS, P. AND SINGH, R. (2013), "WebEngine - A Web-Based Gas Turbine Performance Simulation Tool", Proceedings of ASME Turbo Expo 2013 GT2013, Vol. 4, June 3–7, 2013, San Antonio, Texas, USA, ASME, New York, NY, pp. V004T08A007.
6. AMT NETHERLANDS (2014), Olympus AMT Netherlands gas turbine, available at: http://www.amtjets.com/index.php (accessed 04/10).
7. AMT NETHERLANDS, (2001), Olympus Manual, Geldrop, Netherlands.
8. SOMMERVILLE, I. (2006), *Software engineering,* 8th edition, Pearson Education, GB.
9. BOEHM, B. W. (1988), "A spiral model of software development and enhancement", Computer, vol. 21, no. 5, pp. 61-72.
10. GALLINA, B. AND GUELFI, N. (2007), "A Template for Requirement Elicitation of Dependable Product Lines", in Sawyer, P., Paech, B. and Heymans, P. (eds.) Requirements Engineering: Foundation for Software Quality, Springer Berlin / Heidelberg, , pp. 63-77.
11. BROY, M. (2009), Seamless Model Driven Systems Engineering Based on Formal Models. In *Proceedings of the 11th International Conference on Formal Engineering Methods: Formal Methods and Software Engineering* (ICFEM '09), Karin Breitman and Ana Cavalcanti (Eds.). Springer-Verlag, Berlin, Heidelberg, pp 1-19.
12. HAYHURST, K., J., VEERHUSEN, D., S., CHILENSKI, J., J. AND RIERSON, L., K., (2001), NASA / TM-2001-210 - A Practical Tutorial on Modified Condition/Decision Coverage, NASA, Hampton, Virginia, USA.
13. HILDERMAN, V. AND BAGHI, T. (2007), Avionics certification: a complete guide to DO-178 (software), DO-254 (hardware), Avionics Communications, Leesburg, VA.
14. AMEUR, Y., A., BONIOL, F. AND WIELS, V. (2010), "Toward a wider use of formal methods for aerospace systems design and verification", International Journal on Software Tools for Technology Transfer, vol. 12, no. 1, pp. 1-7.
15. BERARD, B., BIDOIT, M., FINKEL, A., LAROUSSINIE, F., PETIT, A., PETRUCCI, L. AND SCHNOEBELEN, P. (2010), Systems and Software Verification: Model-Checking Techniques and Tools, 1st ed, Springer Publishing Company, Incorporated.
16. LI, W. (2005), Risk assessment of power systems: models, methods, and applications, Wiley, Hoboken, N.J. , Great Britain.
17. GOKHALE, S. S. AND TRIVEDI, K. S. (2002), "Reliability prediction and sensitivity analysis based on software architecture", Software Reliability Engineering, 2002. ISSRE 2003. Proceedings. 13th International Symposium on, pp. 64.
18. CLARKE, P., J., POWER, J., F., BABICH, D. AND KING, T., M. (2012), "A testing strategy for abstract classes", Software Testing, Verification and Reliability, vol. 22, no. 3, pp. 147-169.
19. BRITISH STANDARDS INSTITUTION, (2007), BS EN 61025:2007 - Fault Tree Analysis (FTA), British Standards Online, available at

https://bsol.bsigroup.com/Bibliographic/BibliographicInfoData/000000000030101041 (accessed 05/17).

20. BRITISH STANDARDS INSTITUTION, (2001), BS IEC 61882:2001 - Hazard and operability studies (HAZOP studies). Application guide, British Standards Online, available at https://bsol.bsigroup.com/Bibliographic/BibliographicInfoData/000000000030031511 (accessed 05/17) .
21. FULLWOOD, R. (1999), Probabilistic Safety Assessment in the Chemical and Nuclear Industries, Butterworth-Heinemann, UK.
22. HIGNETT, K. C. (1996), Practical safety and reliability assessment, E & FN Spon, London.
23. Stamatelatos, M., (2002), Fault Tree Handbook with Aerospace Applications, http://www.hq.nasa.gov/office/codeq/doctree/fthb.pdf ed., NASA, Washington DC, USA.
24. BRITISH STANDARDS INSTITUTION, (2006), BS EN 60812:2006 - Analysis techniques for system reliability. Procedure for failure mode and effects analysis (FMEA), British Standards Online, available at https://bsol.bsigroup.com/Bibliographic/BibliographicInfoData/000000000030101028 (accessed 05/17).
25. JAW, L., C. AND MATTINGLY, J., D. (2009), Aircraft engine controls: design, system analysis, and health monitoring, American Institute of Aeronautics and Astronautics, Reston, USA
26. THE APACHE SOFTWARE FOUNDATION (2016), mod_proxy - Apache HTTP Server, available at: http://httpd.apache.org/docs/2.2/mod/mod_proxy.html (accessed 21/09).
27. HEFFELFINGER, D. (2011), Java EE 6 Development with NetBeans 7, Packt Publishing, Great Britain.
28. TRIVEDI, K. S. (1982), Probability and statistics with reliability, queuing, and computer science applications, Prentice-Hall, Englewood Cliffs, NJ.