Project Information

Title

Visual SparkS

Target

Public [X] Restricted [ ]

Proposer Information

Name(s)	Halit Oğuztüzün & Spark Kalibrasyon Hizmetleri Ltd. Şti. (METU Technopolis)
E-Mail(s)	oguztuzn@ceng.metu.edu.tr mahdi_saeedi@sparkkalibrasyon.com.tr

IP (Intellectual Property) Information

All outputs of the project shall be intellectual property of Spark Kalibrasyon Hizmetleri Ltd. Şti. [Spark]

Project Description and Background Information

Description

The purpose of the project is to design block-based graphical syntax and develop an editing environment for SparkS, a domain-specific scripting language. SparkS is being developed at Spark Kalibrasyon Hizmetleri [Spark], the sponsor for this project, for efficient scripting and execution of calibration processes for electronic equipment by metrology engineers and technicians. The SparkS editor, which will be a part of an integrated development environment (IDE) for SparkS, shall support two modes of operation concurrently: text and graphics modes. In the text mode the user composes or modifies a script using a) rudimentary features such as highlighting, indentation, suggestion, auto completion, bracket matching and so on, b) source code browsing, c) macro expansion, d) diagnostics reporting, and, generally speaking, e) facilitation and enforcement of the grammar rules and lexical conventions. As SparkS concrete syntax is textual, this is the fundamental mode of operation, and it is presently available in prototype form. The block-based graphical syntax will be new. In the graphical mode the user can enter or modify the constituent parts of a script through a graphical user interface. The two modes will be supported with two panes so that the user can have both views of the script concurrently and can smoothly switch back and forth between them in the course of editing. The text mode should also be supported stand alone for experienced users.

Similar Products/Projects

National Instruments [LabVIEW] and Keysight [VEE] are widely known visual languages with IDEs for developing, simulating and executing calibration and other procedures in both virtual and real lab

environments. However, these languages have much wider scope than SparkS. The [ArduBlock] project, which provides a graphical environment for Arduino programming, is similar in sprit to the proposed project.

Justification of the proposal

The purpose of the project is to provide a visual alternative for SparkS, a scripting language for calibration processes. The output of the project will be an editor supporting both block-based and text-based syntax for SparkS.

Workers in the field of calibration need a very intuitive and easy-to-use editing environment for calibration scripts. Most of them are not trained in programming; they have difficulty dealing with textual programming notations.

The basic problems are a) block-based graphical syntax design for an existing textual language, and b) development of an editing environment supporting both graphical and textual syntax of the language in harmony. Generally speaking, visual languages and environments are of great interest to computer scientists. There are journals, such as Journal of Visual Languages and Computing [JVisual], and conferences, such as IEEE Symposium on Visual Languages and Human-Centric Computing [VLHCC], dedicated to research on technological spaces involving visual languages. Use of block-based graphical syntax to initiate novices in programming became popular with the MIT [Scratch] project. The proposed project addresses the needs of grown-ups rather than pupils in K-12 education, yet the overall approach seems relevant. The recent Google [Blockly] project is expected to give impetus for the block-based approach.

Contributions, Innovation and Originality Aspects of the Project

The block-based graphical syntax for SparkS will be new. Its design will require a good understanding of calibration workflow, needs of users, good taste in GUI design and creativity.

National Instruments LabVIEW and Keysight VEE are industrial-strength visual languages with versatile graphical editing environments for test and measurement activities in a lab. SparkS, however, takes a more abstract view and relies on its underlying Metrology.Net calibration engine for device-level issues [CalLab]. More to the point, ArduBlock is a block-based graphical language for Arduino programming. A text-based programming language for Arduino programming, called Sketch, is already available. The ArduBlock environment keeps both kinds of syntax in agreement. Although the domain of application is different, a similar approach can be effective for our project.

The targeted product will be specific to the SparkS language. It will support efficient, intuitive and syntax-error-free editing of SparkS scripts.

A versatile editing environment will help the SparkS language be accepted by the calibration community at the national and international levels.

A systematic user study will be needed as a follow-up project. On the tool front, other front-end components of the SparkS IDE, such as a test point editor, and back-end components, such as a debugger, are envisaged for subsequent projects.

Technical Aspects of the Project

A language-based editor for a programming language allows the programmer to edit a program efficiently in an environment specialized for that language. SparkS is a domain-specific language to express and run calibration processes for electronics equipment. (Documentation for the initial version of the language is available at [SparkS].) Thus, the SparkS editor is expected to facilitate the scripting activities of users in an editing environment explicitly based on the language specification. Two most important quality attributes of the editor are usability and modifiability. 1) Usability: The expected benefits of the editor for the users are efficient and comfortable editing, and no grammatical errors. Further, the graphical mode should make scripting easy for beginners. 2) Modifiability: The SparkS language is in the stage of fast evolution. Therefore, changes to the language specification are expected, even in the duration of this project. To counter the modifiability challenge the editor must be explicitly based on the grammar of the language, expressed in [ANTLR], so that changes to the context-free part of the syntax should be semi-automatically reflected to the editor, and changes that are non-context-free in nature should be handled with minimal impact. The editor must be using the same ANTLR

grammar as the present parser is using. Further, the editor should be extensible in terms of the features it provides.

Targeted Output, Targeted User/Domain Profile

The end product will be a language-based editor for SparkS, supporting both graphical and textual syntax. It will be used by the workers in the field of calibration to read, write and modify scripts in SparkS.

Functionality tests should show that user errors pertaining to the context-free part of the syntax are totally eliminated in the resulting scripts. This can be checked by feeding the resulting script to the parser. Functionality tests must be undertaken by the project team, to be reviewed by the sponsor. There will be user tests, organized and run by the project team, addressing the usability factor. Every class of users will be represented in the user tests. Modifiability factor will be tested with the participation of the appointed engineers of Spark under different change scenarios.

The primary users of SparkS are calibration engineers and technicians. Calibration engineers design test processes for equipment (electronics devices in our case), and technicians execute them. Technicians should be able to understand, modify and run SparkS scripts. Quality auditors and administrators are also users who need to read and understand SparkS scripts. The mentioned users are not necessarily experienced in software development although they are experts in their field.

Project Development Environment

The SparkS software runs as a client on a Windows PC. The programming environment will be MS Visual Studio, and C# (or other .NET languages as necessary) will be the language of use. Visual Studio Community is available through CENG MSDN account.

Iterative and incremental development will be adopted due to the nature of the challenge. The project team should have background in grammars and parsing, and gain familiarity with ANTLR parser generator as it is the tool used in the construction of the SparkS parser. Use of free and open software (FOS) should be evaluated. Google [Blockly] can be particularly useful for implementing the editor for graphical syntax, and [ScintillaNET] for textual syntax. The produced technical documentation will be incorporated into the existing SparkS documentation. Specifically, an accurate architecture description for the editing environment is essential. Documentation must be in MS Word.

External Support

The SparkS language definition, sample scripts, and the front-end (parser) of the language processor will be provided to the project team. Related back-end components will be provided as needed. The existing editor, developed using an FOS tool [Geany], will also be provided. (It supports only the text mode with rudimentary features.) Domain users to participate in user testing, to take place at sponsor's site, will be identified by the sponsor. Required domain background for the project team will be provided by the sponsor. This includes the basic terminology and overall workflow for calibration. In particular, information about specific equipment and technical standards will not be necessary.

All support will be provided by the sponsor.

References

(Please provide references / links (URLs) for your answers in above sections.)

[ANTLR] http://www.antlr.org/
[ArduBlock] http://blog.ardublock.com/
[Blockly] https://developers.google.com/blockly/
[CalLab] http://www.callabsolutions.com/
[Geany] https://www.geany.org/
[JVisual] http://www.journals.elsevier.com/journal-of-visual-languages-and-computing
[LabVIEW] http://www.ni.com/labview/

[ScintillaNET] http://scintillanet.codeplex.com/
[Scratch] https://scratch.mit.edu/
[Spark] http://www.sparkkalibrasyon.com.tr/anasayfa
[SparkS] http://etd.lib.metu.edu.tr/upload/12618735/index.pdf
[VEE] www.keysight.com/find/vee
[VLHCC] https://sites.google.com/site/vlhcc2017/