The Eleventh 2016 International Conference on Knowledge, Information and Creativity Support Systems (KICSS 2016)

Pangsit: spreadsheet-like domain-specific programming language for data entry application development

Authors:
Noprianto
Benfano Soewito, Ph.D

Presented by:
Benfano Soewito, Ph.D

Introduction

- In imperative general-purpose programming language, such as C, Java, or Python, creating data entry form means writing step by step codes, to build the form, along with creation and position of form controls, before showing them on screen.
- After that, code for data entry validation and processing should also be written.
- Depending on programming language, graphical toolkit (or library) used, and target storage or service, all of these may take more than 100 lines of code, with possibilities of bugs, only at the data entry step.
- Therefore, an easy to use and specific programming language for data entry creation and processing may be useful.

Challenges

- How to design the programming language so it can be even more useful, for more programmers?
- There are a lot of programming languages available today.
- Based on existing research, we then model this language after spreadsheet programs.

Spreadsheet programs

- Spreadsheet programs are ubiquitous. They can be found installed on virtually any computer that is capable of running spreadsheet program.
- Spreadsheets are used in almost all businesses.
- Based on research, spreadsheets are probably the most widely used programming systems, and also used in businesses, from simple calculations to complex financial models.
- They are the most popular end-user systems.
- Many companies even use spreadsheets in serious financial applications.

Domain-specific programming language

- Programming language or executable specification language that offers, through appropriate notations and abstractions, expressive power focused on, and usually restricted to, a particular problem domain.
  - Research domain: data entry application

Domain-specific programming languages: advantages

- Allow solutions to be expressed in the idiom and at the level of abstraction of the problem domain.
- Concise, self-documenting to a large extent, and can be reused for different purposes.
- May enhance reliability, maintainability, and portability.
- Can embody domain knowledge, enable conservation and reuse.
- Allow validation and optimization, and improve testability.

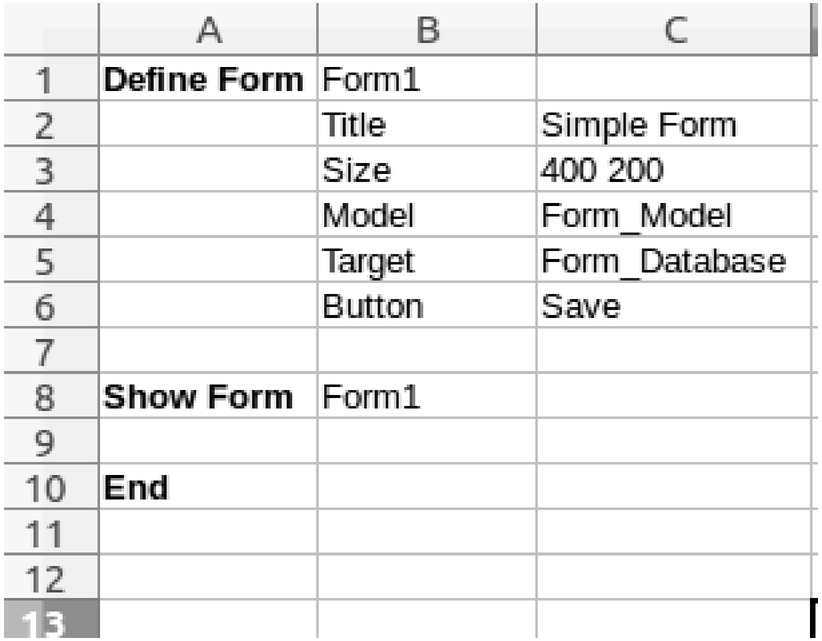
Domain-specific programming languages: disadvantages

- Need cost of design, implementation, and maintenance.
- Education to their users is also needed.
- Finding the proper scope for a domain-specific language is also not easy.
- Difficulty in balancing between domain-specificity and general-purpose programming language.
- Software developed with a domain-specific language may not as efficient as hand-coded software.

Introducing Pangsit Programming Language

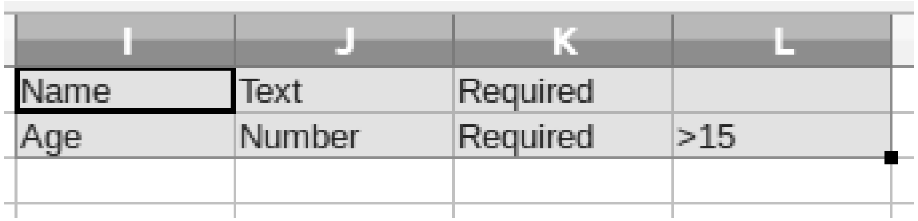
- How to make programmers more productive, by reducing the steps and complexities needed, in order to create a data entry application.
- Using spreadsheet style and some constructs from another programming languages (Basic and Python).
- Spreadsheet program will also be used as development tool, as long as capable of save or export to Office Open XML format (ISO/IEC 29500), with .xlsx file name extension.

Data entry form



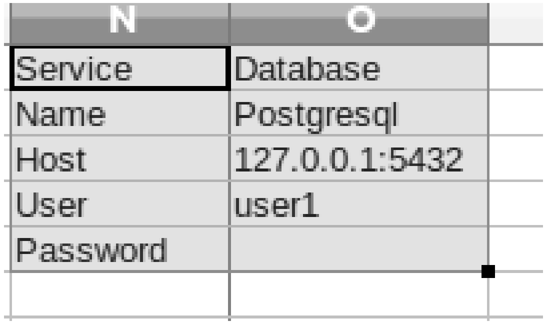
	A	B	C
1	**Define Form**	Form1	
2		Title	Simple Form
3		Size	400 200
4		Model	Form_Model
5		Target	Form_Database
6		Button	Save
7			
8	**Show Form**	Form1	
9			
10	**End**		
11			
12			
13			

1. Form definition



I	J	K	L
Name	Text	Required	
Age	Number	Required	>15

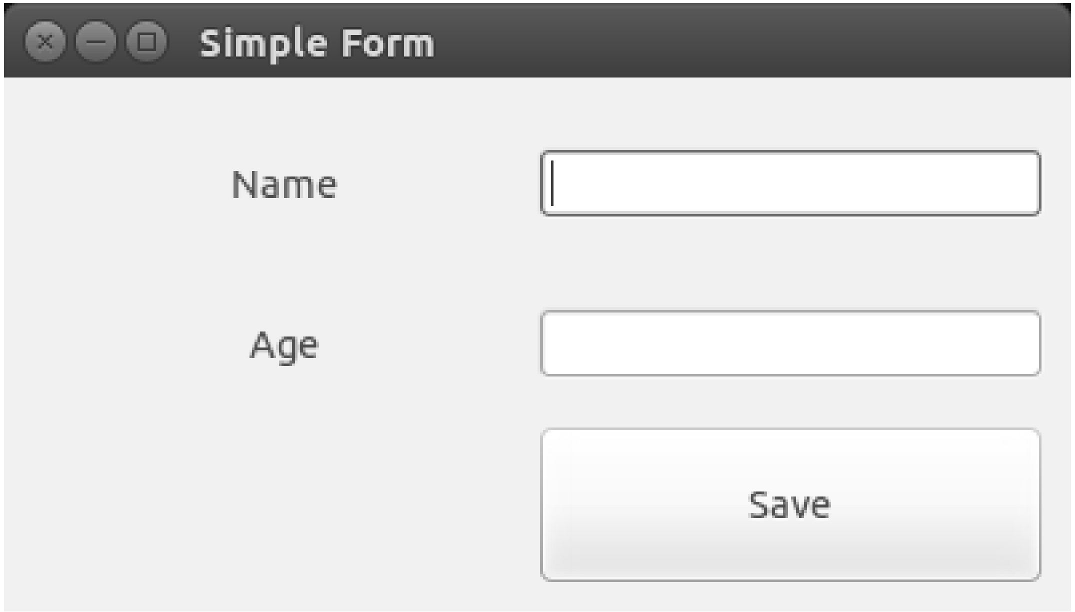
2. Form model definition



N	O
Service	Database
Name	Postgresql
Host	127.0.0.1:5432
User	user1
Password	

3. Form save target

Output application



Simple Form

Name

Age

- If valid values are given, based on Target property in form definition, form data will be saved into database system.
- Form target storage or service may be ranged from local or remote file system (including FTP), local or remote database system (file-based, RDBMS, or No-SQL databases), or HTTP-based services (including a few social media).

Current progress

- Language grammar is being stabilized (some of them are described in the paper).
- Parser is being developed, initial version should be available before 2017.
- Pangsit programming language will be released as free/open source software, under business-friendly license.
- Main internet resource for Pangsit programming language will be hosted at pangsit.org.

Thank you

References (in the paper)

1) R. Abraham, M. Burnett, and M. Erwig. Spreadsheet programming. Wiley Encyclopedia of Computer Science and Engineering, 2008.

2) Y. Ayalew, M. Clermont, and R.T. Mittermeir. Detecting errors in spreadsheets. arXiv preprint arXiv:0805.1740, 2008.

3) M. Burnett. What is end-user software engineering and why does it matter?. In End-User Development (pp. 15-28), Springer Berlin Heidelberg, 2009.

4) M. Burnett, C. Cook, O. Pendse, G. Rothermel, J. Summet, and C. Wallace. End-user software engineering with assertions in the spreadsheet paradigm. In Proceedings of the 25th international conference on Software engineering (pp. 93-103), IEEE Computer Society, May 2003.

5) M. Hanus. Multi-paradigm declarative languages. In Logic Programming (pp. 45-75), Springer Berlin Heidelberg, 2007.

6) A.J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, H. Lieberman, and B. Myers. The state of the art in end-user software engineering, ACM Computing Surveys (CSUR), 43(3), 21, 2011.

7) J.W. Lloyd. Declarative programming in Escher. Technical Report CSTR-95-013, Department of Computer Science, University of Bristol, 1995.

8) B.A. Nardi and J.R. Miller. The spreadsheet interface: A basis for end user programming. Hewlett-Packard Laboratories, 1990.

9) R.R. Panko. What we know about spreadsheet errors. Journal of Organizational and End User Computing (JOEUC), 10(2), 15-21, 2005.

10) S.G. Powell, K.R. Baker, and B. Lawson. Errors in operational spreadsheets. Journal of Organizational and End User Computing (JOEUC), 21(3), 24-36, 2009.

11) R.W. Sebesta. Concepts of programming languages 10 th ed, Pearson Education Inc, 2012.

12) J. Tyszkiewicz. Spreadsheet as a relational database engine. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (pp. 195-206), ACM, June 2010.

13) A. Van Deursen, P. Klint, and J. Visser. Domain-Specific Languages: An Annotated Bibliography, Sigplan Notices, 35(6), 26-36, 2000.

14) E. Visser. WebDSL: A case study in domain-specific language engineering (pp. 291-373), Springer Berlin Heidelberg, 2008.