

DOMAIN SPECIFIC LANGUAGE FOR EMAIL AUTOMATION

Nadejda BARBAROV, Nelli GARBUZ*, Cătălin LATCOVSCHI, Daniel ISTRATI

Department of Software Engineering and Automation, gr. FAF-221, Faculty of Computers, Informatics, and Microelectronics, Technical University of Moldova, Chisinau, Republic of Moldova.

*Corresponding author: Nelli Garbuz, nelly.garbuz@isa.utm.md

Tutor/coordinator: Leon BRÎNZAN, university assistant

Abstract. *The purpose of this article is to explain how a domain-specific language may be useful in marketing. The benefits of using a domain-specific language in email automation are examined in the article, including increased precision, quicker processing times, reduced error rates, etc. DSL-based automation improves the client experience and makes the workflow more seamless. The DSL's functionality, grammatical characteristics, and other details are covered in the article along with the processes that were used to design it. By focusing on key components such as campaign management, personalization, automation, integration, and analytics, this new language equips users with the tools and capabilities they need to elevate their email marketing efforts in an increasingly digital world. The abstract syntax tree was created using a formal grammar as its foundation. The ANTLR 4 parser generator is used as the front-end for the DSL built as explained in this article. Python is used to implement the back end.*

Keywords: ANTLR, formal grammar, marketing, federated networks.

Introduction

A domain-specific language (DSL) is a 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 [1].

According to statistics, in the year 2023, 374.3 billion emails were sent and received per day and by the end of the year 2025, it is predicted that the number of emails sent per day will be 376.4 billion [2]. Without a doubt, email still remains a crucial tool for connecting and engaging in today's fast-paced digital world. With so many communication channels available, it can be overwhelming for marketers to break through the noise and stand out.

Email marketing is a very efficient way to expand the brand, interact with existing consumers, acquire new ones, and increase sales. Furthermore, according to at least 80% of professionals, email marketing promotes client acquisition and retention. As a result of marketing mail, 60% of consumers have made a purchase [3]. However, sending emails manually takes a lot of time and effort, and is not sustainable in the long term.

Email automation refers to the process of using technology to execute automated, personalized, and contextual email campaigns to targeted recipients based on predefined triggers, conditions, or schedules, in order to improve accuracy and save time [3]. Email automation is a powerful marketing automation tool that allows delivery of the correct message to the right people at the right time. This is particularly useful for lead nurturing and, eventually, to increase sales from both current and new customers. Research from the Annuitas Group reveals that marketing automation users experience a staggering 451% increase in qualified leads, with nurtured leads making purchases 47% larger than non-nurtured leads [4].

Existing email automation softwares

Since email marketing has been among the best ways for businesses to get the word out, there are already some softwares specified for email automation on the market, such as Mailchimp,

Brevo and Campaigner. Even though all of them have the same goal, each of them focus on specific features.

Mailchimp is a widely recognized and popular email marketing platform known for its user-friendly interface and comprehensive features. It is particularly favored by beginners and small businesses due to its ease of use and accessibility. Mailchimp offers a range of features including email campaign creation, customizable templates, audience segmentation, A/B testing, and analytics. Mailchimp is ideal for users who are new to email marketing and prefer an all-in-one marketing solution [5].

Brevo is another email automation platform that focuses on providing unlimited email sending capabilities for businesses. Unlike some other email marketing platforms that charge based on the number of contacts or emails sent, Brevo offers flat-rate pricing plans that allow users to send emails to an unlimited number of contacts, which make it suitable for businesses with large contact lists or those that send high volumes of emails regularly. The platform is suitable for businesses looking for a scalable email marketing solution that can accommodate their growing subscriber base without incurring additional costs based on usage [5].

Campaigner is a comprehensive email marketing platform designed for advanced marketers who require sophisticated features and functionality. The platform offers a wide range of advanced features including advanced segmentation, dynamic content personalization, A/B testing, automated workflows, and integrations with third-party tools. It also offers dedicated support and training resources for users, making it well-suited for businesses that require assistance in maximizing the effectiveness of their email marketing campaigns. In addition to this, Campaigner may have a steeper learning curve compared to some other platforms [5].

Consequently, choosing an email automation software becomes difficult, since having so many platforms with their advantages and disadvantages is hard to find a software that will cover all needs of an organization.

Benefits of email automation DSL

Considering the advantages and the limitations of the existing email automation softwares, the DSL has the following benefits:

More Seamless Workflow

Along with optimization, email automation also offers efficiency, since it allows specialists to send emails quickly and keeps customers satisfied. According to statistics, 49% of sales and marketing professionals consider that time saving of repetitive tasks is one of the most important aspects of sending emails [6].

Customization and Flexibility

Users can easily modify and expand their automation logic to accommodate new campaigns, audience segments, or marketing objectives, without the need for extensive reconfiguration or redevelopment. It allows organizations to adapt and evolve their automation workflows as their business grows or their marketing needs change. Moreover, DSLs can be designed to handle large volumes of data, including contact lists with an unlimited number of entries.

Collaboration and Knowledge Sharing

By using a shared vocabulary and syntax, team members can more effectively collaborate on designing, reviewing, and refining automation strategies. This cooperative method supports communication and coordination among various departments or teams engaged in email marketing, resulting in more unified and successful campaigns.

Ease of Maintenance

With a DSL, email automation workflows and configurations are expressed in a structured and semantically meaningful way. This clarity makes it easier to understand, maintain, and modify automation logic over time.

Testing and Quality Assurance

Email automation DSL enables the expression of test scenarios, including edge cases and boundary conditions, directly within the DSL itself. This allows automated testing frameworks to validate the correctness of automation logic, ensuring that emails are sent accurately and reliably to recipients, and minimizing the risk of errors or inconsistencies in campaign execution.

Grammar

A formal grammar is a set of formation rules for strings in a formal language. The rules represent how the strings will be formed from the language's alphabet that have to be valid according to the language's syntax [7]. A formal grammar is defined as $G = (V_n, V_t, P, S)$, where:.

- V_n is a finite set of nonterminal symbols
- V_t is a finite set of terminal symbols
- P is a finite set of production rules
- S is the start symbol.

To specify the grammar representation for the email automation language we need to use the Extended Backus–Naur form. In table 1, there are the metanotations.

Table 1

Metanotations	
Symbol	Definition
< >	Non-terminal symbol
*	Zero or more occurrences
+	One or more occurrences
	Separate the alternative symbols
→	Derivation
(...)	Grouping elements
[...]	Optional element

$S = \{ \text{lower_case} \}$
 $V_t = \{ "a", "b", \dots, "z", "A", "B", \dots, "Z", "0", "1", \dots, "9", "/", ":", "!", "-", "@", "!", "01", "02", "12", "31", "\n", "=", "!", "<", ">", "<=", ">=", "send_email", "loop", "end", "if", "else", "def", "and", "or", "CC:", "BCC:" \}$

$V_n = \{ \text{lower_case, upper_case, letter, digit, integer, character, string, name, file_path, directory, file_name, email_address, string_with_dots, domain, date, year, month, day, time, hour, minute, second, text, symbol, special_character, subject, attachment, recipient, cc_recipient, bbc_recipient, signature, header, function_definition, parameter, parameters, data_type, email_script, email_statement, email_body, body, statement, send_email_statement, email_details, wait_statement, time_value, time_unit, if_statement, condition, comparison, comparison_operator, logical_operation, logical_operator, loop_statement, assignment_statement, variable_name, expression, value, function_call, arguments} \}$

$P = \{$
 $\text{<lower_case> ::= "a" | "b" | ... | "z"}$
 $\text{<upper_case> ::= "A" | "B" | ... | "Z"}$
 $\text{<letter> ::= <lower_case> | <upper_case>}$
 $\text{<digit> ::= "0" | "1" | ... | "9"}$
 $\text{<integer> ::= <digit> | <digit> <integer>}$
 $\text{<character> ::= <letter> | <digit> | "/" | ":" | "."}$
 $\left. \right\}$

```
<string> ::= <character> | <character> <string> | <string> <character>
<name> ::= <string> | <string> <name>
<file_path> ::= "/" <directory> "/" <file_name>
<directory> ::= <string> | <string> "/" <directory>
<file_name> ::= <string>
<email_address> ::= <string_with_dots> "@" <domain>
<string_with_dots> ::= <string> | <string> "." <string_with_dots>
<domain> ::= <lower_case> "." <lower_case> | <lower_case> <domain> | <domain>
<lower_case>
<date> ::= <year> "-" <month> "-" <day>
<year> ::= <digit> <digit> <digit> <digit>
<month> ::= "01" | "02" | ... | "12"
<day> ::= "01" | "02" | ... | "31"
<time> ::= <hour> ":" <minute> ":" <second>
<hour> ::= <digit> <digit>
<minute> ::= <digit> <digit>
<second> ::= <digit> <digit>
<text> ::= <symbol> | <symbol> <text>
<symbol> ::= <letter> | <digit> | <special_character>
<special_character> ::= "!" | "@" | ... | "\n"
<subject> ::= <text>
<attachment> ::= "Attachment:" <file_path> "\n"
<recipient> ::= <email_address> | <email_address> <recipient>
<cc_recipient> ::= "CC:" <recipient>
<bcc_recipient> ::= "BCC:" <recipient>
<signature> ::= <text>
<header> ::= "From:" <email_address> "\n" "Date:" <date> "\n" "Subject:" <subject> "\n"
    "To:" <recipient>
    "\n" <cc_recipient>? <bcc_recipient>? "\n"
<function_definition> ::= "def" <function_name> "(" <parameters> ")" ":" <body> "\n"
<function_name> ::= <string>
<parameters> ::= <parameter> | <parameter> "," <parameters>
<parameter> ::= <name> ":" <data_type>
<data_type> ::= "string" | "integer" | "email_address" | "file_path" | "date" | "time"
<email_script> ::= <function_definition> | <email_statement>
<email_statement> ::= <header> <email_body> <attachment>?
<email_body> ::= <text> | <statement> | <statement> <body>
<body> ::= <text> | <statement> | <statement> <body>
<statement> ::= <send_email_statement> | <wait_statement> | <if_statement> |
    <loop_statement> |
    <assignment_statement>
<send_email_statement> ::= "send_email(" <email_details> ")"
<email_details> ::= <header> <body> <attachment>?
<wait_statement> ::= "wait(" <time_value> ")"
<time_value> ::= <integer> <time_unit>
<time_unit> ::= "seconds" | "minutes" | "hours" | "days"
<if_statement> ::= "if" <condition> ":" <body> "else:" <body> "end"
<condition> ::= <comparison> | <logical_operation>
<comparison> ::= <expression> <comparison_operator> <expression>
<comparison_operator> ::= "==" | "!=" | "<" | ">" | "<=" | ">="
<logical_operation> ::= <condition> <logical_operator> <condition>
```

```

<logical_operator> ::= "and" | "or"
<loop_statement> ::= "loop" ":" <body> "end"
<assignment_statement> ::= <variable_name> "=" <expression>
<variable_name> ::= <string>
<expression> ::= <value> | <variable_name> | <function_call>
<value> ::= <string> | <integer> | <email_address> | <file_path> | <date> | <time>
<function_call> ::= <function_name> "(" <arguments> ")"
<arguments> ::= <expression> | <expression> "," <arguments>

```

Parse tree

ANTLR (ANother Tool for Language Recognition) is a parser generator for reading, processing, executing, or translating structured text or binary files. From a given grammar, ANTLR generates a parser that can build and walk parse trees and also generates a listener interface (or visitor) that makes it easy to respond to the recognition of phrases of interest [8].

Parse tree (also known as derivation tree or concrete syntax tree) is a hierarchical structure that represents the syntactic structure based on a given grammar [9].

```

def func(to:email_address,subject:string,content:string):
  send_email(
    From:send@gmail.com
    Date:2023-03-15
    Subject:subject
    To:to
    content
  )
end

func(test@gmail.com,"Test subject","This is a test body")

```

Figure 1. Email automation code example

According to the defined grammar of the email automation DSL and the code snippet in fig. 1, a parse tree for the email automation DSL is represented in figure below.

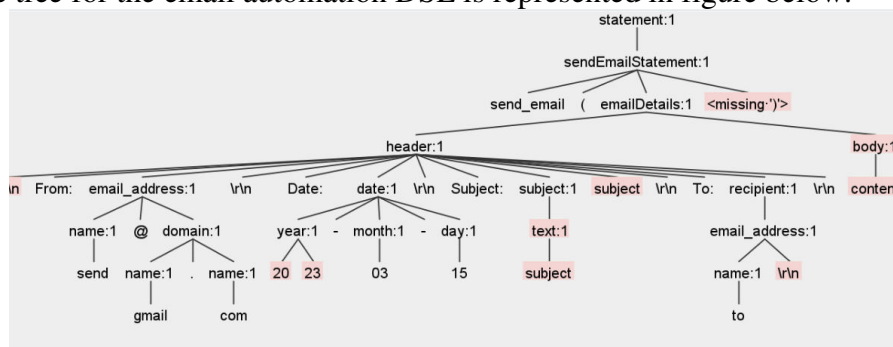


Figure 2. Parse tree for email sending

Conclusions

In summary, the need for a new language in the domain of email automation arises from the limitations of existing solutions and the growing complexity of modern email marketing. Platforms like Mailchimp, Brevo, and Campaigner have useful functions and tools, but in many situations, they fall short of giving a really smooth and customized email automation experience.

With the flexibility to build domain-specific constructions, rules, and conventions, users may precisely customize their email automation processes to meet the demands of their organizations. DSLs also provide a mechanism that may support teamwork and shared work efficiently by their enhanced communication and the sharing of the knowledge.

Scalability and ease of maintenance are further characteristics of email automation DSL. With such an organized and language-oriented syntax, maintenance and modification of automation logic over time are much more manageable, hence allowing organizations to adjust their workflow according to the growth of the business or the changes required for their marketing campaign.

Furthermore, the email automation DSL aids in the testing and quality assurance of complete workflows. This is because, in this language, one may express test scenarios directly, making sure that the logic of automation is correct and reliable to minimize chances for errors and inconsistencies.

In essence, the introduction of a domain-specific language for email automation represents a step forward towards empowering businesses to navigate in competitive markets with ease by using a user-friendly platform designed specifically for their needs, making email marketing challenges easier to navigate with precision.

References

- [1] *Domain Specific Language*, March 19 2024.[Online] [Accessed: 03.04.2024] Available:<https://itexus.com/glossary/domain-specific-language/#gref>
- [2] A. Mosby, *How many emails are sent per day in 2024*, December 25, 2023. [Online] [Accessed: 06.04.2024] Available:<https://www.yaguara.co/how-many-emails-are-sent-per-day/>
- [3] K. George, *Everything You Ever Wanted To Know About Email Automation* [Online] [Accessed: 06.04.2024] Available:<https://email.uplers.com/blog/email-automation-guide/?amp=1>
- [4] J.Desyllas, *34 Marketing Automation Statistics You Need To Know*, December 8, 2023. [Online] [Accessed: 08.04.2024]. Available: <https://moosend.com/blog/10-marketing-automation-statistics-need-know/>
- [5] A. N. Smith, K. Main, *10 Best Email Marketing Software Of 2024* [Online] [Accessed: 08.04.2024]. Available:https://www.forbes.com/advisor/business/software/best-email-marketing-software/#mailchimp_crm_section
- [6] P.Vaughan, *Email Automation: How to Set Up Automated Workflows [Data]*. April 18 2023. [Online] [Accessed: 04.04.2024]. Available: <https://blog.hubspot.com/marketing/email-marketing-automation-examples>
- [7] *Formal Grammar*. [Online] [Accessed: 03.04.2024] Available:<https://brasil.cel.agh.edu.pl/~11sustrojny/en/formal-grammar/index.html>
- [8] T.Parr, *What is ANTLR?* [Online] [Accessed: 07.04.2024] Available: <https://www.antlr.org/>
- [9] *What is Parse Tree*. [Online] [Accessed: 04.04.2024] Available:<https://synergycodes.com/glossary/what-is-parse-tree/>