





Evaluating the semantic transparency of Guaraná: A domain-specific language for enterprise application integration

Jose Bocanegra¹  | Rafael Z. Frantz²  | Fabricia Roos-Frantz²  | Fabio P. Basso³ 

¹School of Engineering, Department of Systems and Computing Engineering, Universidad de los Andes, Bogotá, Colombia

²Department of Exact Sciences and Engineering, Unijuí University, Ijuí, Brazil

³Campus Alegrete, Federal University of Pampa, Alegrete, Brazil

Correspondence

Jose Bocanegra, School of Engineering, Department of Systems and Computing Engineering, Universidad de los Andes, Bogotá, Colombia.
Email: j.bocanegra@uniandes.edu.co

Funding information

Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul, Grant/Award Number: 19/2551-0001268-3; National Council for Scientific and Technological Development (CNPq), Grant/Award Number: 309315/2020-4; Research Support Foundation of Rio Grande do Sul (FAPERGS), Grant/Award Number: 17/2551-0001206-2

Abstract

Guaraná domain-specific language is aimed to design enterprise application integration models. Although the language has been broadly used in the industry and academy, the design of its graphical notation was based in the common sense and social opinion of its developers rather than theoretical principles and empirical evidence. Thus, Guaraná becomes as an option for studying, modifying, and improving its concrete syntax. In this article we conducted an experiment with 85 participants in order to evaluate the semantic transparency of the concrete syntax of Guaraná. The study concludes that the language is not found to be semantically immediate because most of the language constructs used in the graphical notation of Guaraná are not semantically immediate, that is, a novice reader is not able to infer their meaning from their appearance alone.

KEYWORDS

DSL, enterprise application integration, graphical notation, integration patterns, integration process, integration solution, semantic transparency

1 | INTRODUCTION

The semantic transparency of a language can be located in a continuum state which ranges from semantic immediacy to semantic perversity.¹ Thus, a symbol is *semantically immediate* if a novice reader would be able to infer its meaning from its appearance alone; a symbol is *semantically translucent* if its appearance provides a clue to meaning but require some initial explanation; a symbol is *semantically opaque* if there is a purely arbitrary relationship between its appearance and its meaning; and a symbol is *semantically perverse* if a novice reader would be likely to infer a different meaning from its appearance.

In this article we conducted an experiment to evaluate the semantic transparency of Guaraná, a domain-specific language (DSL) to design enterprise application integration (EAI) models. EAI is the research field devoted to the development of methodologies, techniques, and tools to build integration solutions.² An integration solution is the piece of software that can orchestrate a set of software systems so that they can collaborate to exchange data and reuse

Abbreviations: CD, cognitive dimensions; DSL, domain-specific language; EAI, enterprise application integration; UML, Unified Modeling Language

functionality. Integration platforms are software tools to support the design, implementation, and execution of integration solutions.³ The design of an integration solution is carried out with a DSL and results in a conceptual model to overcome an integration problem in a high-level of abstraction; the implementation is assisted with a software development kit, which allows for the transformation of the conceptual model into executable code; and the execution is actually made by a run-time system, which mostly include monitors to check and follow the execution of integration solutions and make sure they meet the quality of service expected. There are several integration platforms in the market, and amongst them, open-source message-based integration platforms, such as Mule,⁴ Apache Camel,⁵ Spring Integration,⁶ Fuse,⁷ ServiceMix,⁸ Petals,⁹ Jitterbit,¹⁰ WSO2 ESB,¹¹ and Guaraná¹² have gained attention from the EAI community.

Amongst these open-source tools, Guaraná is the single integration platform built within the paradigm of model-driven engineering,¹³ that is, it promotes models as first-class citizens in every stage of the integration solution process development, whereas other platforms are code-centric. The integration community has been working hard on shifting tools from a code-centric to a model-centric development approach.^{14,15} Guaraná has been used in several projects in both public and private sector in Spain, such as at Andalusia Development and Innovation Agency, Huelva County Council, Huelva Science and Technology Park, Huelva Association of Strawberry Producers and Exporters, University of Seville, University of Huelva, Andalusian Foundation for Aerospace Development, Advanced Center for Aerospace Technologies, Sadiel Enterprise, GNERA Energy S.L, USISA S.A, Isotrol Company, and Cibernos S.A.

Guaraná provides a DSL which enables the development of platform-independent models, which is a key feature in this context, since the resulting models are not bound with any particular technology, even with the software development kit provided with it. A DSL increases the level of abstraction since it provides language constructs that are close to the problem domain. These constructs are usually smaller and easier for software engineers to learn and use, they are more expressive and usually increase productivity and quality, besides they reduce maintenance efforts of applications.¹⁶⁻²¹

According to Fowler²⁰ DSLs can be classified as internal and external DSLs. An internal DSL is created from an existing general-purpose language, hosted on that language and so share its syntax and tools for model checking and verification. An external DSL is represented in a separate and new language, which provides its own syntax and so requires the development of supporting tools for model checking, verification, syntax highlighting, parsers, editing environment for that language. Having this classification in mind, the authors of Guaraná state that it is an external DSL.

As is stated by Moody,¹ graphical notations play a pivotal role in the design and construction of languages in software engineering; however, these notations, usually, are not properly addressed by researchers and software engineers.²² In the case of Guaraná, the design of its concrete syntax has not involved a scientific approach. As an attempt to improve the concrete syntax of this language, this article describes an experiment, of type “descriptive study,”²³ to evaluate the semantic transparency of Guaraná, where semantic transparency is defined as “the extent to which the meaning of a symbol can be inferred from its appearance.”¹ A semantically transparent DSL helps in reducing user’s cognitive load because the meaning of its language constructs can be perceived directly and easily learned.

A total of 85 novice participants performed a task which involves understanding EAI models described in Guaraná. As a result, most of the language constructs used in the concrete syntax of Guaraná are semantically translucent.

The remainder of this article provides a brief description of Guaraná (Section 2), an analysis of the related work (Section 3), the presentation of the experimental design (Section 4), the results and discussion (Section 5), the threats to validity (Section 6), and the conclusions and future work (Section 7).

2 | THE GUARANÁ DOMAIN-SPECIFIC LANGUAGE

Guaraná provides a graphical notation to design platform-independent models for EAI solutions. In the following, we provide an overview of the main language constructs of Guaraná. A detailed discussion of this DSL is presented by Frantz.²⁴

Message: Wraps the information that is exchanged and transformed within the workflow of an integration solution. Messages are composed of a header, a body, and may also include attachments. The

header contains custom and some pre-defined properties, such as message identifier, correlation identifier, sequence size, sequence number, return address, expiration date, and message priority. The body contains the payload data. Attachments enable messages to carry extra pieces of data associated with the payload, such as images or any other binary raw data.

- Task:** Implements an enterprise integration pattern,²⁵ such as split, aggregate, translate, chop, filter, correlate, merge, re-sequence, replicate, dispatch, enrich, slim, promote, demote, or delay. Generally speaking, a task may have one or more inputs by means of it receives messages, and one or more outputs to deliver messages. Tasks are classified as stateless or stateful depending on whether the work that they carry out on a particular message is independent from the previous and/or future messages or not.
- Slot:** A buffer that connects the output of a task or a port with the input of another task or a port. They support several policies to serve messages, including priority-based and first-come, first-served. They are the key to allowing tasks to process messages as asynchronously as possible.
- Port:** Abstracts away from the details required to interact with resources within a software ecosystem. There are four types of ports: entry, exit, solicitor, and responder.
- Integration solution:** Aggregates a number of ports, tasks, and slots. Conceptually, it is a workflow that routes messages that are read from entry ports through a process that transforms them and writes the results to exit ports; through the process, additional information may be gathered using solicitor ports or delivered on-demand using responder ports.
- Resource:** Represents an information source or sink that usually belongs to an application, such as data files, databases, APIs, or even user interfaces. Resources exist prior to integration solutions and are not changed at all when they are integrated.

Figure 1 illustrates the main constructors and their concrete syntax in the Guaraná language by means of an abstraction of a typical integration solution. In every task, the small rounded connectors located on the sides of a task icon indicate the inputs and the outputs of that task. Slots are connected to tasks by means of these connectors. The figure also brings the notation we use to represent the resources integrated. Note that this notation does not specify which layer of the resource (database, channel, file, API, user interface, and so on) is being used as communication channel by the integration solution to transfer data from/to the resource. Note that messages are not part of the conceptual model of the language, that is, they only exist and flow at run-time. However, we show them in the figure only for didactic reasons.

Tasks are classified according to the semantics they represent. Router tasks do not change the state of the messages they process, they only route a message through a process. Modifiers help to add or remove data carried by messages, they do not change the schemata of the data. Transformers help to transform one or more messages into a new message and so with a different schema. Router tasks are presented in Table 1, Modifiers tasks are presented in Table 2, and Transformer tasks are presented in Table 3.

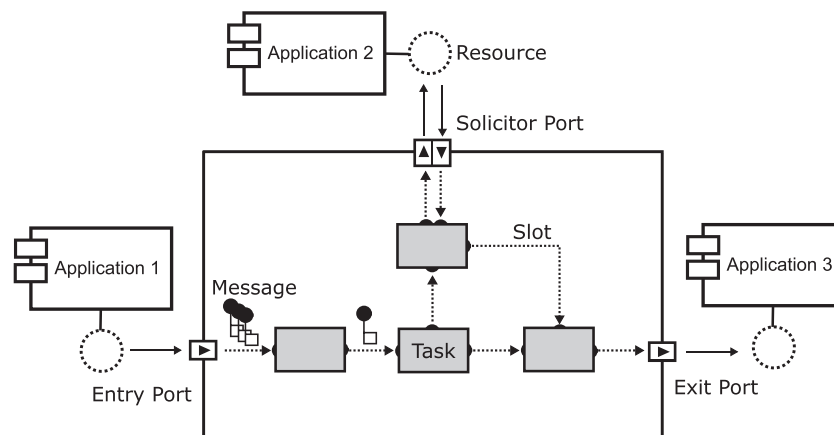


FIGURE 1 Abstract view of a typical integration solution

TABLE 1 Router tasks












Icon	Task	Description
	Correlator	Analyses inbound messages and outputs sets of correlated ones.
	Merger	Merges messages from different input slots into one output slot.
	Resequencer	Reorders messages into sequences with a pre-established order.
	Filter	Filters out unwanted messages.
	Idempotent transfer	Removes duplicated messages.
	Dispatcher	Dispatches a message to exactly one slot.
	Distributor	Distributes messages to one or more slots.
	Replicator	Replicates a message to all of the output slots.
	Semantic validator	Validates the semantics of a message.
	Threader	Increases the number of threads to run tasks in the model.
	Custom router	Allows for routing a message according to custom semantics.

TABLE 2 Modifier tasks



















Icon	Task	Description
	Slimmer	Removes contents from the body of a message according to a static policy.
	Context-based slimmer	Removes contents from the body of a base message according to a dynamic policy that is provided by a context message.
	Content enricher	Adds static contents to the body of a message.
	Context-based content enricher	Adds dynamic contents from a context message to the body of a base message.
	Header Enricher	Adds static contents to the header of a message.
	Context-based Header Enricher	Adds dynamic contents from a context message to the header of a base message.
	Header promoter	Promotes a part of the body of a message to its header.
	Header demoter	Demotes a part of the header of a message to its body.
	Set correlation ID	Sets a correlation ID value to the corresponding property in the header of a message.
	Set return address	Sets a return address value to the corresponding property in the header of a message.
	Custom modifier	Allows to modify the header and body of a message according to custom semantics.

TABLE 3 Transformer tasks

Icon	Task	Description
	Translator	Transforms the body of a message from one schema into another.
	Splitter	Splits a message that contains repeating elements into several messages.
	Aggregator	Constructs a new message from several messages produced previously by a Splitter.
	Chopper	Breaks a message into two or more messages.
	Assembler	Constructs a new message from two or more messages.
	Cross builder	Constructs a new message that contains the cartesian product of all inbound messages.
	Custom transformer	Allows for transformation of a message according to custom semantics.

3 | RELATED WORK

The evaluation of a language notation is a topic that has been extensively worked on the academic community. In this section we present several evaluation studies on DSLs.

Genon et al.²⁶ performed an evaluation of some of the Moody's principles for the Use Case Map (UCM) Visual Notation. This is a scenario modeling language which is intended for the elicitation, analysis, specification, and validation of requirements. In a second work,²⁷ the same authors also evaluated the BPMN 2.0 Visual Notation. In both works, the evaluation of languages was done by comparing the visual elements with the individual perception of the authors. Unfortunately, end users of the language are not involved in determining the degree of semantic transparency of the language.

Genon et al.²⁸ proposed a set of experiments designed to identify a new symbol set for i^{3em} (a language for requirements engineering) and to evaluate its semantic transparency. The experiments comprise several tasks. For instance, in the first experiment, the authors obtained drawings hand-sketched by participants to represent i^{3em} concepts. In the second experiment authors focused on identifying the stereotypical drawings out of the results of first experiment. Finally, in the third experiment the participants had to elect the drawing that is the most semantically transparent for a given referent concept.

The work by Caire et al.²⁹ analysed the semantic transparency of a new concrete syntax of i^* modeling language. This study differs from ours in that the semantics of the notation are not being evaluated, but rather that users are requested to propose a notation for language constructs. Boone et al.³⁰ evaluated the visualization of CHOOSE, an Enterprise Architecture Approach for small and medium-sized enterprises. In the article, regarding the semantic transparency of the language, the authors claim that “there is clearly a lot of room for improvement regarding this principle. Only four symbols show a certain presence of semantic transparency, which are the symbols of goal, conflict, human actor, and device. This means 28 symbols do not suggest the meaning of their construct at all.” However, there is not a proof or an experiment that involves users to determine that degree of semantic transparency.

Saed et al.³¹ evaluated the cognitive effectiveness of the visual syntax of feature diagrams. One of the principles evaluated is semantic transparency. The authors concluded that “the visual syntax of feature diagrams cannot be considered as semantically transparent. Users of feature diagrams are required to memorize the semantics of the symbols prior to reading or creating feature diagrams as they cannot infer the meanings of symbols simply by viewing them.”

Albuquerque et al.³² presented an evaluation methodology for quantitatively analyzing the cognitive dimensions (CD)³³ of textual DSLs for detecting architectural problems. Some of these CD are viscosity (resistance to change), visibility (ability to view entities easily), and role-expressiveness (the purpose of an entity is readily inferred).

El Kouhen et al.³⁴ reported on a set of experiments over the Unified Modeling Language (UML). The results confirm the lack of semantic transparency of the language. The conclusion provided by the authors is that radical improvement

is required to enforce the cognitive effectiveness of UML. The proposed solution is “to involve end-users as co-designers of these languages rather than as passive consumers as it has been so far.”

The work by Granada et al.³⁵ reported on an experiment to validate the semantic transparency of WebML icons. WebML is a DSL applied for designing complex data-intensive web applications at a conceptual level. The participants of the experiment were provided with two questionnaires. The first comprises the current set of symbols while the second contains the new proposal. At the end, the participants choose the best symbol between the two options.

Mafalda et al.³⁶ proposed an alternative concrete syntax for the KAOS requirements modeling language. The authors reported a set of experiments with language users in order to determine the semantic transparency of novice-designed symbols and expert-designed symbols (standard KAOS notation). As a result, the semantic transparency of the novice-designed symbols was significantly higher than the one in the standard KAOS concrete syntax.

As a conclusion, the work reported in this article is one of the first attempts to evaluate the semantic transparency of a DSL oriented to represent EAI models. An important aspect to be highlighted is that in the experiment only participated novice users of the language.

4 | EXPERIMENTAL DESIGN

This section reports on the experiment we have conducted to evaluate the semantic transparency of Guaraná. The experiment falls under category one-group posttest only design,³⁷⁻³⁹ because there is no comparison group, and all the subjects fall into the same participant category. The protocol we have applied to conduct this experiment has its foundations in the works of Jedlitschka and Pfahl,⁴⁰ Perry et al.,⁴¹ and Wohlin et al.²³ The following sections detail this protocol to present the experimentation we have carried out.

4.1 | Goal

Following the Goal/Question/Metric template proposed by Jedlitschka and Pfahl⁴⁰ we state the overall experimental goal thus: *Analyze the semantic transparency of Guaraná for the purpose of identifying the extent to which the meaning of a symbol can be inferred with respect to its appearance from the viewpoint of several participants in the context of a software engineering course at three universities.*

4.2 | Research question

The following research question was formulated to be answered with the evaluation:

RQ: Are all the seven language constructs selected for testing in the graphical notation of Guaraná semantically immediate?

The possible answers to this research question are *yes*, when all the seven symbols are evaluated as semantically immediate, or *no*, when none of the seven symbols are evaluated as semantically immediate.

We measured statistical data over each parameter, deriving results that classify each symbol according to one of four values shown in Figure 2.

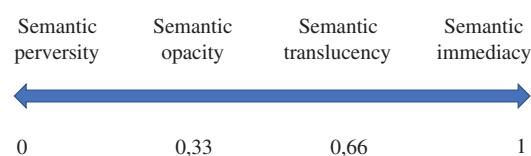
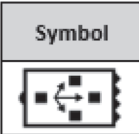

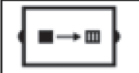



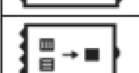


FIGURE 2 Measure scale

TABLE 4 Language constructs to be evaluated in the task

Symbol	Concept represented by the symbol
	Header Enricher
	Correlator
	Filter
	Replicator
	Assembler
	Translator
	Splitter

4.3 | Variables

An independent variable is a variable that can be controlled by the researcher, and can be chosen and manipulated. In the experiment, the independent variable is the set of language constructs of the notation provided in Table 4. A dependent variable is what needs to be measured in the experiment and what is affected during its execution. The dependent variable responds to the independent variable. In this experiment, the dependent variable is the meaning of each language construct in the notation, which is inferred by the user.

Guaraná has several constructors and our experience of designing and implementing integration solutions in practice has shown that the majority of the solutions designed make use of no more than ten different types of tasks. Other tasks are usually for unusual operations. So, we have chosen an integration solution that makes use of the most common of these tasks, it is not surprising that this solution is often used by the integration community to illustrate a relatively complete scenario in terms of operations (tasks) necessary to implement the integration workflow. Please, note that our study focuses on the semantic transparency of tasks and other constructors such as ports, applications and resources, were not considered because these constructors are exclusive of the Guaraná DSL and other integration platforms do not share them as they do with tasks, because the concrete syntax of tasks is inspired on the graphical representation of the integration patterns introduced by Hohpe and Woolf.²⁵

4.4 | Subjects

A total of 85 subjects participated in the experiment. The participants were recruited from classes in software engineering in three different universities: two in Brazil (65 subjects) and one in Colombia (20 subjects). Participants in Brazil were second year undergraduate students while participants in Colombia were third year undergraduate students.

All participants have previously taken in their curricula a software engineering course that covers the topic of application integration. However, none of the participants had prior contact to the notation under test. They were invited to participate in the study voluntarily by announcements in the lectures as well as by personal invitation.

4.5 | Metrics and data analysis techniques

We consider that the number of participants is statistically significant because it is close to the result of applying the Equation 1 (taking as reference the work of Yamane⁴²),

$$n = N * X / (X + N - 1) \quad (1)$$

where, $X = Z^2 * p * (1 - p) / MOE^2$. Thus, Z is the critical value of the normal distribution (e.g., for a confidence level of 95%, the critical value is 1.96), p is the sample proportion, MOE is the margin of error, and N is the population size. In this experiment the population size is 110 because it is the number of available undergraduate students of second and third year in both universities.

In this work, we seek to determine the number of correct answers given by the participants on the meaning of a language construct on the total number of responses. In other words, **sensitivity** is equivalent to $TP / (TP + FN)$, where TP are the true positives and FN are false negatives.

Taking as reference the work of El Kouhen et al.,³⁴ in this study, we used positive values as measurements for the semantic transparency (see Figure 2) which ranges in a scale from 0 to 1. Thus, approximately, values between 0 and 0.33 represent opacity, values between 0.33 and 0.66 indicate translucency, and values above 0.66 suggest immediacy.

The analysis of the results is based on the sensitivity and specificity test proposed by Altman and Bland.⁴³ This test was originally applied to quantify the diagnosing ability of a medical test. Specifically, the test measures the proportion of true positives that are correctly identified as such (sensitivity); and the proportion of true negatives (specificity).

4.6 | Objects

The objects in this experiment comprises several items: the case study and the task that the participants have to develop.

4.6.1 | Case study

The case study describes a typical EAI scenario which involves a travel agency and a set of support systems. Usually, a travel agency not only requires an integration solution that eases the process of searching for flights and hotels, but they also need to automate the booking process. This case study introduces an integration solution that takes a travel booking request as input and registers the booking of the flights and hotel selected. Figure 3 illustrates the integration solution of this case study. Note that this solution involves five applications, which are a Travel System, an Invoice System, a Mail Server, a Flights Façade, and a Hotels Façade. Travel System is an off-the-shelf application that the travel agency uses to register information regarding their customers and booking requests. The invoice service runs on the Invoice System that is a separate application to allow customers to pay their travels using a credit card. The Mail Server runs the electronic mail service and this application is used for providing customers with information about their bookings. The Façades represent interfaces for booking flights and hotels. Conversely the Travel System and Invoice System are applications that were not designed with integration in mind, so, the integration solution must interact with them by means of their data layer. We make the assumption that every booking registered in the Travel System contains all of the necessary information about the payment, flight and hotel, and a record locator which uniquely identifies the booking. The Travel System is periodically polled by the integration solution for new travel bookings, so that flights and hotels can be booked, the customer can be invoiced and provided with an email with the information about his/her travels.

4.6.2 | Task

Once the case study has been presented, participants performed a task which is described in the following. The task contains the following statement: “A visual integration model, described in Guaraná, can represent seven concepts: header enrichment, filtering, correlation, replication, assembly, translation, and division. Draw a straight line to match each

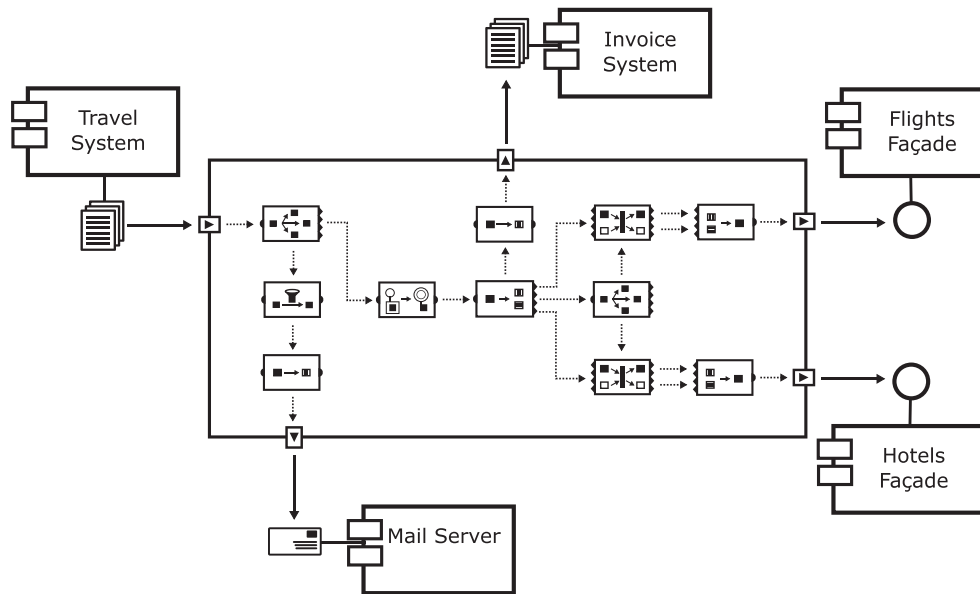


FIGURE 3 Integration solution example

language construct on the left with the concept that best represents it.” The table to provide the answer is depicted in Table 4.

4.7 | Instrumentation

Each participant received a document (comprised by four sheets) that contains (i) the case study; (ii) the diagram that represents the case study; (iii) the description of the task; and (iv) the table to answer the task. In the task, the user is presented with the visual language constructs used to specify an integration solution in Guaraná and with a set of concepts (represented as words). The participant shall connect each language construct with its correspondent concept.

4.8 | Data collection procedure

The experiment was conducted in a single session. Participants were requested to express their willingness by signing an informed consent, which ensures that the participants have voluntarily expressed their intention to participate in the research. In that form, the participants were informed about the objectives behind the test, their responsibilities, the amount of time required to complete the test, the conditions under which the experiment is carried out, and the destination of the collected data over the course of the experiment.

Each of these tasks was presented in an introductory section of the study theme, followed by the activities of the experiment. In order to ensure that there were no problems with execution, a trial was conducted. Once a threat was detected in a trial related to the participants’ anxiety about the design of the experiment, that is, they were anxious to know what this type of study would generate, we tried to resolve this bias for real executions. Thus, it was necessary to resolve the following bias of the study regarding the task “objectives behind the test”: to mitigate expectations regarding the lack of knowledge about semantic transparency assessment, which could affect the study in the sense that the participant does not trust the treatment given to the data.

It was then decided not to make the assessment a mystery, explaining the respondents’ intuition would contribute to finding semantic transparency data, but not disclosing the protocol in details, for example, the metrics. It was concluded that these initial explanations of the study’s intentions would not affect the responses, since in fact it is recommended as a characterization of the study in experimental evaluations.

Each participant in the experiment received two identical copies of the informed consent. One copy was kept by the participant and the second was signed and was kept by the observer as a supporting document for the research. In the

next activity, each participant received the instrument. The instrument comprises four pages, which should be printed on one side only.

The first page contains the case study description. The second page includes the diagram that describes the case study. That diagram is depicted using Guaraná. The third page provides the statement for the task. The fourth page contains the table to provide the answers to the task.

The instrument is provided in three stages. In the first stage the participant receives pages 1 and 2. The participant will keep those two pages throughout the experiment. The participant will have 10 min to read the statement and to understand the diagram. In the second state the participant receives pages 3 and 4. The participant will have 10 min to read the statement and to answer the task. Then, the observer receives the statement, the answer, and pages 1 and 2. At the end, answers are grouped and scanned for their subsequent analysis. The above times were established since the designers of the experiment developed the task and those numbers were sufficient to complete it.

4.9 | Post-experiment

Once all participants completed the proposed tasks, they were asked to complete a post-experiment survey. That survey comprises three questions:

PEQ1: Do you consider that the time for the execution of the experiment was adequate?

PEQ2: Do you think that the case study was clear and easy to understand?

PEQ3: Do you consider that the two tasks you had to carry out were clear?

These question have to be answered by using a Likert scale.⁴⁴ This scale comprises the following values: strongly disagree, disagree, undecided, agree, and strongly agree.

5 | EXECUTION, RESULTS, AND DISCUSSION

This section provides information about the execution of the experiment, its results, and the discussion.

5.1 | Execution

The experiment was executed in three different universities: two in Brazil and one in Colombia. The participants were asked to perform the tasks detailed in Section 4.6. The execution of the experiment was conducted in classroom settings under the supervision of professors that play the role of observers. All participants were informed that their participation was entirely optional and they could leave the experiment at any moment. Then, they read and signed the consent inform to approve their participation in the study. Subsequently, the participants were required to fill a post-experiment questionnaire, to obtain qualitative information of the results.

Once all the responses were collected, the data were summarized (see Table 5). In the table, rows represent the expected concepts while the columns represent the selected concept by the participant. The cells contains the number of answers that correlate the expected and the selected concept. Each column is summarized, and all totals must have the same value. The last row depicts the sensitivity, or the number of correct answers divide by the number of total answers. Values with tendency to semantic immediacy are highlighted in green while values that tend to semantic translucency are highlighted in yellow.

5.2 | Results and discussion

In order to evaluate the semantic transparency of Guaraná, seven symbols where selected as parameters. Table 5 shows the data collected. We found three symbols with a tendency to meet semantic immediacy: *Assembler* (0.776), *Chopper* (0.741), and *Filter* (0.706).

In contrast, the symbols which tends to semantic translucency are *Replicator* (0.576) *Header Enricher* (0.482), *Translator* (0.459), and *Correlator* (0.424).

TABLE 5 Experiment results

Expected	Selected						
	Replicator	Filter	Translator	Header Enricher	Chopper	Correlator	Assembler
Replicator	49	1	3	7	16	9	0
Filter	3	60	9	12	0	1	0
Translator	5	15	39	11	1	7	7
Header Enricher	4	1	14	41	0	19	6
Chopper	5	0	4	1	63	10	2
Correlator	15	6	13	10	1	36	4
Assembler	4	2	3	3	4	3	66
Total	85	85	85	85	85	85	85
Sensitivity	0.576	0.706	0.459	0.482	0.741	0.424	0.776

TABLE 6 Post-experiment results

Question	Strongly disagree	Disagree	Undecided	Agree	Strongly agree
PEQ1	2	2	10	39	20
PEQ2	2	8	17	33	13
PEQ3	2	11	10	33	17

As a result, 42.86% of the analyzed symbols are semantically immediate. However, the results are negative in regards to the expected semantic transparency motivated by our research question.

In a detailed analysis, we consider that *Filter* is a known symbol because is used in several contexts. As a result, its meaning is easy to understand for most participants. *Chopper* is a task that divides a message into parts while *Assembler* is a task that composes one message from others. The icons used by Guaraná for these processes are suggestive. *Replicator*, *Correlator*, *Header Enricher*, and *Translator* are concepts used in EAI domain. Thus, we consider that is reason why these symbols are not semantically immediate.

The difficulty in understating the semantics of Guaraná may also be related with the fact that its language is an external DSL. External DSLs use a custom syntax and are represented in a separate and new language, that is, it does not use a syntax of a hosting language such as happens in internal DSLs.⁴⁵ The semantics of Guaraná was strongly influenced by the integration patterns documented by Hohpe and Woolf,²⁵ so knowing these patterns in advance may help to capture the semantic layer of its concrete syntax. These patterns were not subject of explanation in the lectures given to the participants, which suggests an overall picture may be crucial to understanding the language.

As conclusion, software engineering researchers have ignored or undervalued the role of visual syntax,¹ and the case of Guaraná is not the exception. When the authors created the DSL there was not a design rationale process for documenting design decisions. On the contrary, the design of the language was based on common sense. As a result, graphic design choices were often counter intuitive, and the majority of DSL constructs are not semantically immediate.

5.3 | Results post-experiment

Of the 85 participants in the experiment, only 73 filled out the post-experiment survey. The other 12 participants withdraw of this part of the experiment. As depicted in Table 6, 79.2% of the participants (39 agree and 20 strongly agree) considered that the time for the execution of the experiment was adequate. 63% (33 agree and 13 strongly agree) said the case study was clear and easy to understand. Finally, 68.3% (33 agree and 17 strongly agree) indicated that the tasks to perform were clear.

As a conclusion we need to improve the clarity of the tasks to be performed by the participants. Thus, we consider that it is necessary to use a tool that guides the process and reduces observer interference. The tool could help us to randomize

the symbols to be evaluated, to use different models (not only expressed in Guaraná but other languages), and to limit the execution time of the experiment.

5.4 | Limitations and lessons learnt

The first limitation of our descriptive study is the absence of an Hypothesis. Along the experimental design we proposed an Hypothesis reflecting the research question, leading us to a weak evidence. So, we opted to remove the initial research Hypothesis.

The experimental design considers a unique data group. This decision implied in impossibility to compare data that could be derived from the same data-set. After an exhaustive process carried out to align the initial Hypothesis, we reasoned that the adopted experimental design was not prepared to regroup the data into at least two specific categories. Finally, changes in this sense would characterize a different experimental design and study.

Another limitation, derived from the experimental design, is a absence of a significance evaluation performed through *t*-value and *p*-value statistical methods. This limitation is also derived from the lack of knowledge from the researcher perspective about whether the sample of 85 participants are good enough to represent the real population. This is due to the fact that the current experimental studies in ESE and IHC, devoted to evaluate DSLs, do not provide parameters about minimum samples that reflect a population. Current experimental studies are rather computational than built on user experience, which makes significance evaluation regarding user experience an open topic in the area.

Hypothesis reformulation were affecting the experimental design. Smaller modifications in the Hypothesis were making us implying in errors of type 1 and 2 from the IHC empirical point of view.⁴⁶ Then, we noticed that hypothesis adjustments were leading us into a threat to validity: Fishing and the error rate. For these reasons, we decided to remove the hypothesis from the experimental design.

We also learned from the reviewers feedback, which provided us with new experimental insights for further research. Hence, we will revisit the same amount of data using different experimental designs. For the aforementioned reasons, this experiment must be considered as a starting point for the next studies that will analyze the same data from different research hypothesis and perspectives. For instance, new hypothesis would explore whether the results are the same considering groups of answers by University, by undergraduate courses, by each of Guaraná's symbols, and others. Unfortunately, due to initial characteristic of the experiment, randomization is an experimental feature that can not be adopted.

6 | THREATS TO VALIDITY

A first challenge from our study is that we planned a controlled experiment research protocol, which is derived from Empirical Software Engineering (ESE) literature,²³ while adopting artifacts for evaluation derived from Human-Computer Interaction (HCI) literature.⁴⁷ These two worlds do not always converge in terms of evaluation, once ESE motivates experimenters to collect quantitative data rather than qualitative, as those observed from HCI experiments.⁴⁶ In order to understand whether this experiment would be valid from these two perspectives of evaluation, we discussed about the protocol with three experts from ESE and also three experts from HCI. They all agreed that we could proceed with the planned controlled experiment, but that all the possible limitations should be clearly debated. In this sense, this section presents the conditions for the validity of our study, discussing threats to validity that may constraint our findings.

6.1 | Constructor validity

This threat can be related to social elements from participants and also to the experimental design.

Experimental design threats: according to Wohlin et al.,²³ a first threat from this type of study is the “Inadequate preoperational explication of constructs,” which means that a good research protocol must

anticipate what the researchers intent to observe. Due to the characteristic of the experimental design, which was designed to a non-comparative descriptive experiment, we considered that this threat was partially mitigated, that is, we could have drawn other research questions and Hypothesis. However, the experimental design is correct and focused to answer the unique proposed research question. In order to mitigate the experimental design bias, the researchers wrote an initial research protocol, performing a trial to identify possible obstacles and whether the theory was sufficient clear. We agreed that the protocol should be improved, so it was refined with possible threats, which could be associated with the metrics and research goals. So, we ensured that the theory of measuring semantic transparency of Guaraná was aligned with our next observations. Another possible threat is the “Mono-operation bias,” which may affect studies that cannot give a full picture of the theory. In our case, a mono-operation bias could happen when a sample is characterized by homogeneous opinions, that is, with students with the same culture about the usage and development of DSLs. In order to mitigate this bias, we looked for experimental execution considering students from six classes from three Universities. This allowed to collect a diversified set of samples. We also ensured that the same students were not respondent twice in this experiment, avoid the threat “Interaction of different treatments.” To avoid threat from “Confounding constructs and levels of constructs,” we draw our experiment to ignore the previous know-how of the students in application integration, presenting the same introduction to the topic before applying the questionnaire.

Experimental social threats: a threat that may affect the answer of samples is the “Hypothesis guessing,” where participants try to suppose what results will be derived. To detect eventual threats in this regard, we executed a trial before executing the reported evaluations. This trial detected this bias, generating expectations that could affect the results. This bias was mitigated by making clear our evaluation goals and also discussing how the semantic transparency would be evaluated. In addition, we discussed that the students were not under evaluation, avoiding the “Evaluation apprehension” threat. Finally, the “Experimenter expectancies” can occur when the experimenter can affect the results of what is observed, a threat mitigated with the usage of four experimenters.

6.2 | Internal validity

In our experiment, we agreed that the student know-how about EAI would not affect our analysis. In this sense, we analyzed all the subjects from the perspective of a unique group, that is, without a control group. For this reason, our study can be affected by “single group threats”²³ such as:

- A) *Instrumentation*, which is related with low quality of artifacts used for experiment execution. This is a threat when considering that the students are fluent in only two languages (Spanish and Portuguese). In order to mitigate this bias, we developed an experimental package with artifacts written in Spanish and in Portuguese. In this sense, the presented case study was applied in Spanish or in Portuguese in the classrooms, including the initial presentation, the experiment questions, and the post-experiment questions.
- B) *Selection*, which relates to the need for volunteers. Once we used students from classrooms, we mitigate this bias by using interval between disciplines and telling the students that their participation would be totally volunteer, not counting points in the classroom nor the presence.
- C) *Mortality*, when participants leave the experiment, which is avoided by removing incomplete answers. In this sense, we managed the mortality following two groups of discarded answers: a) For the participants who left the incomplete answers for instruments (Stage 1 and Stage 2), removing them from the analyzed data set; b) For participants who correctly performed the instruments, but who did not respond to the post-experiment form. Thus, 12 participants were included in (a) but not in (b). It is important to highlight that, since the post-experiment form in (b) was used to collect information for future improvement of this type of study, the observed mortality does not affect our conclusions about Guaraná’s semantic transparency.

6.3 | External validity

These threats affect how generalizable our results are, and include: *A) Interaction of setting and treatment*, which may occur when using an artificial case study for evaluation by participants. In order to avoid this threat, we adopted a real case study. *B) Interaction of selection and treatment*, applied when the population is not representative for generalization of results. Since our participants are not experts in EAI, this threat makes our results non generalizable. In other words, experts could provide different opinions than the ones collected from students.

Different integration solutions may use a similar number of task types, however these tasks may be chained differently in the integration workflow to solve different integration problems. Thus, more than the example (the integration problem approached), what can influence the generalization of the result is the set of task types used to implement the integration workflow. We advocate the result would be quite similar for different integration solutions if a similar set of task types are used, but we understand specific integration problems may add a degree of complexity to understand the purpose of the integration solution and so influence the comprehension of integration workflow and the individual tasks in the model.

6.4 | Conclusion validity

A possible threat to conclusion validity is the low statistical power provided by low number of samples. In order to eliminate this bias from our study, we executed the experiment with six different undergraduate classes, from three different universities. This allowed to collect 85 samples, thus configuring an adequate statistical power to draw our conclusions. Another threat for conclusion is the “Fishing and the error rate,” when researchers try to catch some finding that is not sustained by the research questions. This threat is partially resolved, once we focused in the research question and, thus, avoiding fishing rate. The error rate could be mitigated with an analysis of significance level, which is a limitation in our study. The threat “Reliability of measures” is mitigated by the adoption of metrics also adopted by related works, thus providing a reliability of what is measured. The threat “Reliability of treatment implementation” can occur when different persons try to apply the same experiment, a threat mitigated by the execution of a trial and a consensus achieved by the experimenters through five meetings and a research protocol following recommendations from Wohlin et al.²³

7 | CONCLUSIONS AND FUTURE WORK

This article presented an experimental approach as a descriptive research to evaluate the semantic transparency of Guaraná, a DSL developed for the field of EAI. In the experiment participated 85 subjects from three universities, two in Brazil and one in Colombia. The experiment analyzed a subset of seven symbols of the DSL: Header Enricher, Correlator, Filter, Replicator, Assembler, Translator, and Chopper. In a scale between 0 and 1 (where 0 means a semantically perverse language constructor and 1 a semantically immediate language constructor) three symbols are semantically immediate and four are semantically translucent. The semantically immediate language constructs (Assembler, Chopper, and Filter) are those present in other domains, in which they have an universal meaning. In contrast, the semantically translucent language constructs (Replicator, Translator, Header Enricher, and Correlator) are those present in the EAI domain. In summary, we can conclude that we should improve the semantic of the Guaraná DSL by proposing some changes in four symbols. Hence, we obtained a negative result in answering our research question since most of the language constructs used in the graphical notation of Guaraná are not semantically immediate: only 42.86% of the analyzed symbols are semantically immediate. It is important to mention that in this work we only analyze a subset of symbols of the DSL. Thus, we propose, as future work, to extend the experiment to all the 17 symbols of the DSL.

The concrete syntax of Guaraná DSL has been designed on the basis of the graphical proposal introduced by Hoppe and Woolf²⁵ in their book about integration patterns. These patterns have influenced the development of several open-source message-based integration platforms for the design and implementation of integration solutions, such as Mule,⁴ Apache Camel,⁵ Spring Integration,⁶ Fuse,⁷ ServiceMix,⁸ Petals,⁹ Jitterbit,¹⁰ and WSO2 ESB,¹¹ and are widely used in the documentation and examples present in those platforms. Since the research results have determined that most of the Guaraná constructors are not semantically immediate, it can be inferred that the constructors of these integration platforms are not either. In this way, we believe that these results can help the integration community to improve the design of the DSL

of their platforms and make the documentation and examples provided more understandable. Additionally, our protocol can be used to evaluate the semantic transparency of other integration patterns of the Guaraná DSL and even of other DSLs from other platforms which are also based on the Hohpe and Woolf²⁵ integration patterns.

Despite the negative result obtained in the experiment in face of our first expectations, we consider the experiment very positive for new improvements in the studied DSL. These types of studies that look for issues on cognitive effectiveness about the DSL construction are important when developing a graphical DSL, particularly in the early stages of DSL design. We conclude that the proposed experiment format provides evidence associated with graphic elements that can help software engineers to promote greater acceptance of a graphical DSL. Its execution time is low, on average 10 min, and therefore suitable for collecting data on the semantic transparency of model elements.

The experiment is interesting both from the point of view of ESE and HCI, as it brings these two worlds together to identify points of improvement needed in DSLs. Once that this was our first an effort to evaluate the semantic transparency of a DSL, we have some lessons learnt that can direct improvements in our future study plannings, as follows: 1) research questions could explore individual notations instead of a general analysis as the one adopted in this descriptive study; 2) hypothesis could be formulated also by each notation of a DSL and also the data-set could be re-grouped according to criteria, which could lead to comparative studies. In this sense, an explanatory study could be planned to understand why four of seven notations from Guaraná DSL are not semantically immediate; 3) the application of significance tests to descriptive studies in Software Engineering and user experience contexts, including *t*-value and *p*-value analysis. In this sense, it is important to identify whether a sample is representative from a population, a data that which is not acknowledged in regards to experiments published in the area scoping ESE and IHC; and 4) draw a comparative experiment of type exploratory instead of a descriptive one, which would demand an experimental design with statistical tests.

Finally, the next task derived of this study is to update the graphical notation of the symbols and involve end users in the DSL design process.

ACKNOWLEDGMENTS

This work was supported by the National Council for Scientific and Technological Development (CNPq) under Grant 309315/2020-4; by the Research Support Foundation of Rio Grande do Sul (FAPERGS) under Grant 17/2551-0001206-2; and by FAPERGS under Grant 19/2551-0001268-3.

AUTHOR CONTRIBUTION

Jose Bocanegra Study conception. Rafael Zancan Case study preparation. Fabio Basso Analysis and interpretation of results. Fabricia Roos-Frantz Draft manuscript preparation. All authors reviewed the results and approved the final version of the manuscript.


DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

ORCID

Jose Bocanegra  <https://orcid.org/0000-0002-8342-7346>

Rafael Z. Frantz  <https://orcid.org/0000-0003-3740-7560>

Fabricia Roos-Frantz  <https://orcid.org/0000-0001-9514-6560>

Fabio P. Basso  <https://orcid.org/0000-0003-4275-0638>

REFERENCES

1. Moody DL. The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Trans Softw Eng.* 2009;35(6):756-779.
2. He W, Xu LD. Integration of distributed enterprise applications: a survey. *IEEE Trans Ind Inform.* 2014;10(1):35-42.
3. Freire DL, Frantz RZ, Roos-Frantz F, Sawicki S. Survey on the run-time systems of enterprise application integration platforms focusing on performance. *Softw Pract Exp.* 2019;49(3):341-360.
4. Dossot D, D’Emic J, Romero V. *Mule in Action.* Manning; 2014.
5. Ibsen C, Anstey J. *Camel in Action.* Manning Publications Co; 2010.

6. Fisher M, Partner J, Bogoevice M, Fuld I. *Spring Integration in Action*. Manning Publications Co; 2012.
7. Russell J, Cohn R. *Fuse ESB*; Book on Demand; 2012.
8. Konsek H. *Instant Apache ServiceMix How-to*. Packt Publishing; 2013.
9. Russell J, Cohn R. *Petals ESB*; Book on Demand; 2012.
10. Russell J, Cohn R. *Jitterbit Integration Server*. Book on Demand; 2012.
11. Indrasiri K. *Introduction to WSO2 ESB*. Springer; 2016.
12. Frantz RZ, Corchuelo R, Roos-Frantz F. On the design of a maintainable software development kit to implement integration solutions. *J Syst Softw*. 2016;111:89-104.
13. Schmidt DC. Guest editor's introduction: model-driven engineering. *IEEE Comput*. 2006;39(2):25-31.
14. Thoo E, Pezzini M, Guttridge K, Bhullar B. Magic quadrant for enterprise integration platform as a service. Technical report. Gartner; 2019.
15. Li Y. *Architecting Model Driven System Integration in Production Engineering*. PhD thesis. KTH Royal Institute of Technology, Department of Production Engineering, Stockholm, Sweden; 2017.
16. Estublier J, Vega G, Ionita AD. Composing domain-specific languages for wide-scope software engineering applications. Proceedings of the International Conference on Model Driven Engineering Languages and Systems (MODELS). MODELS; 2005:69-83.
17. van Deursen A, Klint P. Little languages: little maintenance? *J Softw Maint*. 1998;10(2):75-92.
18. Baker P, Loh S, Weil F. Model-driven engineering in a large industrial context: motorola case study. Proceedings of the International Conference on Model Driven Engineering Languages and Systems. MODELS; 2005:476-491.
19. Tolvanen JP, Kelly S. Defining domain-specific modeling languages to automate product derivation: collected experiences. Proceedings of the Software Product Line Conference (SPLC). SPLC; 2005:198-209.
20. Fowler M. *Domain-Specific Languages*. Addison-Wesley; 2010.
21. Ghosh D. *DSLs in Action*. Manning Publications Co.; 2011.
22. Van Der Linden D, Hadar I. A systematic literature review of applications of the physics of notations. *IEEE Trans Softw Eng*. 2018;45(8):736-759.
23. Wohlin C, Runeson P, Hst M, Ohlsson MC, Regnell B, Wessln A. *Experimentation in Software Engineering*. Springer; 2012.
24. Frantz RZ, Reina-Quintero AM, Corchuelo R. A domain-specific language to design enterprise application integration solutions. *Int J Cooperat Inf Syst*. 2011;20(2):143-176.
25. Hohpe G, Woolf B. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley; 2003.
26. Genon N, Amyot D, Heymans P. Analysing the cognitive effectiveness of the UCM visual notation. Proceedings of the International Workshop on System Analysis and Modeling; 2010:221-240; Springer, New York, NY.
27. Genon N, Heymans P, Amyot D. Analysing the cognitive effectiveness of the BPMN 2.0 visual notation. Proceedings of the International Conference on Software Language Engineering; 2010:377-396; Springer, New York, NY.
28. Genon N, Caire P, Toussaint H, Heymans P, Moody D. Towards a more semantically transparent i* visual syntax. Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality; 2012:140-146; Springer, New York, NY.
29. Caire P, Genon N, Heymans P, Moody DL. Visual notation design 2.0: towards user comprehensible requirements engineering notations. Proceedings of the 21st IEEE International Requirements Engineering Conference (RE); 2013:115-124; IEEE.
30. Boone S, Bernaert M, Roelens B, Mertens S, Poels G. Evaluating and improving the visualisation of CHOOSE, an enterprise architecture approach for SMEs. Proceedings of the IFIP Working Conference on the Practice of Enterprise Modeling; 2014:87-102; Springer, New York, NY.
31. Saeed M, Saleh F, Al-Insaif S, El-Attar M. Evaluating the cognitive effectiveness of the visual syntax of feature diagrams. *Requirements Engineering*. Springer; 2014:180-194.
32. Albuquerque D, Cafeo B, Garcia A, Barbosa S, Abrahão S, Ribeiro A. Quantifying usability of domain-specific languages: an empirical study on software maintenance. *J Syst Softw*. 2015;101:245-259.
33. Green TR. Cognitive dimensions of notations. *People and Computers V*; Cambridge University Press; 1989:443-460.
34. El Kouhen A, Gherbi A, Dumoulin C, Khendek F. On the semantic transparency of visual notations: experiments with UML. *International SDL Forum*. Springer; 2015:122-137.
35. Granada D, Vara JM, Brambilla M, Bollati V, Marcos E. Analysing the cognitive effectiveness of the WebML visual notation. *Softw Syst Model*. 2017;16(1):195-227.
36. Santos M, Gralha C, Goulão M, Araújo J. Increasing the semantic transparency of the KAOS goal model concrete syntax. Proceedings of the International Conference on Conceptual Modeling; 2018:424-439; Springer, New York, NY.
37. Creswell JW, Creswell JD. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.; 2017.
38. Spector PE, Spector PF. *Research Designs*. Vol 23. Sage; 1981.
39. Patten ML, Newhart M. *Understanding Research Methods: An Overview of the Essentials*. Taylor & Francis; 2017.
40. Jedlitschka A, Pfahl D. Reporting guidelines for controlled experiments in software engineering. Proceedings of the International Symposium on Empirical Software Engineering. ISESE; 2005:95-104.
41. Perry DE, Porter AA, Votta LG. Empirical studies of software engineering: a roadmap. Proceedings of the Conference on the Future of Software Engineering. CFSE; 2000:345-355.
42. Yamane T. *Statistics: An Introductory Analysis-3*. Harper and Row; 1973.
43. Altman DG, Bland JM. Diagnostic tests. 1: sensitivity and specificity. *Br Med J*. 1994;308(6943):1552.
44. Albaum G. The Likert scale revisited. *J Market Res Soc*. 1997;39(2):1-21.

45. Riti P. External DSL. *Practical Scala DSLs*. Springer; 2018:59-69.
46. Lazar J, Feng J, Hochheiser H. Chapter 8 - Interviews and focus groups. *Research Methods in Human Computer Interaction*. 2nd ed. Morgan Kaufmann; 2017:187-228.
47. MacKenzie IS. *Human-Computer Interaction: An Empirical Research Perspective*. Morgan Kaufmann; 2013.

How to cite this article: Bocanegra J, Frantz RZ, Roos-Frantz F, Basso FP. Evaluating the semantic transparency of Guaraná: A domain-specific language for enterprise application integration. *Softw Pract Exper*. 2022;52(4):967-983. doi: 10.1002/spe.3045