Speak Well or Be Still

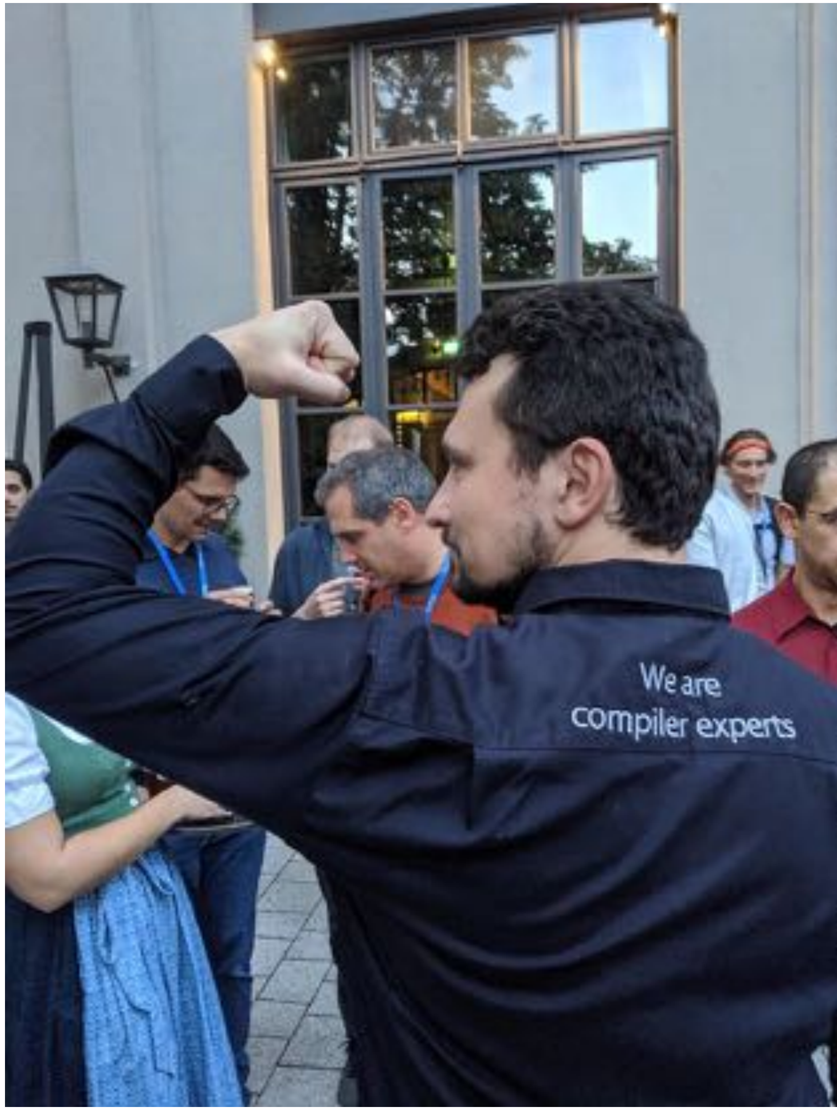
Solving Conversational AI with Weighted Attribute Grammars

MeSS: MDE for Smart IoT Systems

dr. Vadim Zaytsev, 5 July 2022

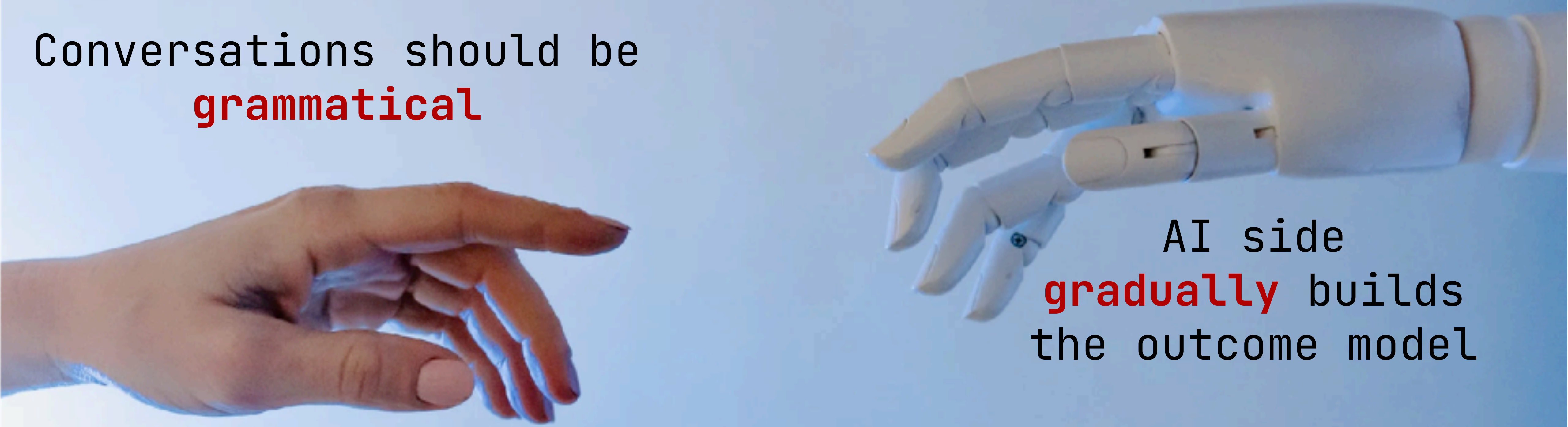
Introduction

- Vadim Zaytsev aka @grammarware ()
  - research (, , CWI)
  - teaching ()
  - industry (RAINCODE mainframe to .NET, raincode LABS)
- Relevant details:
  - ELIZA clone [1999]
  - grammars [2004..]
  - WAG [2021..]



Vision

Conversations should be **grammatical**



AI side **gradually** builds the outcome model

Each side grows a model of the **other** side

State of the Art: RiveScript

```
+ greetings
- Hi there!{weight=20}
- Hello!{weight=25}
- nuqneH

+ say something random
- This {random}message|sentence{/random} has a random word.
```

https://www.rivescript.com/docs/tutorial

Grammars in CC Books

<N T P S>

- Nonterminals
- Terminals
- Production rules
- Starting symbol

```
foo ::= b 'a' r.
b ::= 'b'+.
r ::= '---'.
```

Grammars in WAGIoT

<N A T R P C S>

- Names
- Attributes
- Triggers
- Reactions
- Processes
- Computations
- Start

```
time ← 'what time is it?'.
time → [1] 'it is' dt.
time → [w] 'no idea, ' ^n.
```

WAGIoT Example

<N,A,T,R,P,C,S>

$$S \leftarrow \mathtt{setup}\ \mathtt{activity}^*\ \mathtt{stop}. \tag{1}$$

$$\mathtt{setup} \leftarrow \mathtt{greet}?\ \mathtt{getname}. \tag{2}$$

$$\mathtt{greet} \leftarrow \texttt{'hello'}. \tag{3}$$

$$\mathtt{greet} \leftarrow \texttt{'good morning'}. \tag{4}$$

$$\mathtt{greet} \rightarrow \texttt{'greetings, human!'}\ \texttt{'what is your name?'}. \tag{5}$$

$$\mathtt{getname} \leftarrow \texttt{'I am'}\ \underset{\cdot}{n} := \mathtt{Id}. \tag{6}$$

$$\mathtt{getname} \rightarrow \texttt{'nice to meet you,'}\ \underset{\cdot}{n}. \tag{7}$$

$$\mathtt{activity} \leftarrow \mathtt{time} \mid \cdots. \tag{8}$$

$$\mathtt{time} \leftarrow \texttt{'what time is it?'}. \tag{9}$$

$$\mathtt{time} \rightarrow [1]\texttt{'it is'}\ [\![DateTime.Now]\!]\ \texttt{','}\ \dot{n} \blacktriangleright w := 5. \tag{10}$$

$$\mathtt{time} \rightarrow [w]\texttt{'it is'}\ [\![DateTime.Now.Hour]\!]\ \texttt{'o'clock'} \blacktriangleright w := w - 1. \tag{11}$$

$$\mathtt{stop} \leftarrow \texttt{'stop'} \mid \texttt{'off'}. \tag{12}$$

UNIVERSITY OF TWENTE. | Formal Methods & Tools

WAGIoT Example

<N,A,T,R,P,C,S>

$$
\begin{array}{rcll}
S & \leftarrow & [1:1]\texttt{setup activity}^{*}\ \texttt{stop}. & (13) \\
\texttt{setup} & \leftarrow & [1:1]\texttt{greet}?\ \texttt{getname}. & (14) \\
\texttt{greet} & \leftarrow & [1:2]\texttt{'hello'}. & (15) \\
\texttt{greet} & \leftarrow & [1:2]\texttt{'good morning'}. & (16) \\
\texttt{greet} & \rightarrow & [1:1]\texttt{'greetings, human!'}\ \texttt{'what is your name?'}. & (17) \\
\texttt{getname} & \leftarrow & [1:1]\texttt{'I am'}\ \underset{.}{n} := \texttt{Id}. & (18) \\
\texttt{getname} & \rightarrow & [1:1]\texttt{'nice to meet you,'}\ \underset{.}{n}. & (19) \\
\texttt{activity} & \leftarrow & [1:x]\texttt{time} \mid \cdots. & (20) \\
\texttt{time} & \leftarrow & [1:1]\texttt{'what time is it?'}. & (21) \\
\texttt{time} & \rightarrow & [1:1+w]\texttt{'it is'}\ [\![DateTime.Now]\!]\ \texttt{','}\ \dot{n} \blacktriangleright w := 5. & (22) \\
\texttt{time} & \rightarrow & [w:1+w]\texttt{'it is'}\ [\![DateTime.Now.Hour]\!]\ \texttt{'o'clock'} \blacktriangleright w := w-1. & (23) \\
\texttt{stop} & \leftarrow & [1:1]\texttt{'stop'} \mid \texttt{'off'}. & (24)
\end{array}
$$

WAGIoT Example

<N,A,T,R,P,C,S>

$$
\begin{aligned}
S &\leftarrow [1:1]\texttt{setup activity}^{*}\ \texttt{stop}. && (25)\\
\texttt{setup} &\leftarrow [1:1]\texttt{greet}?\ \texttt{getname}. && (26)\\
\texttt{greet} &\leftarrow [c_1 : c_1 + c_2]\texttt{'hello'} \blacktriangleright c_1 := c_1 + 1. && (27)\\
\texttt{greet} &\leftarrow [c_1 : c_1 + c_2]\texttt{'good morning'} \blacktriangleright c_2 := c_2 + 1. && (28)\\
\texttt{greet} &\rightarrow [1:1]\texttt{'greetings, human!'}\ \texttt{'what is your name?'}. && (29)\\
\texttt{getname} &\leftarrow [1:1]\texttt{'I am'}\ \underset{\cdot}{n} := \texttt{Id}. && (30)\\
\texttt{getname} &\rightarrow [1:1]\texttt{'nice to meet you,'}\ \underset{\cdot}{n}. && (31)\\
\texttt{activity} &\leftarrow [1:x]\texttt{time} \mid \cdots. && (32)\\
\texttt{time} &\leftarrow [1:1]\texttt{'what time is it?'}. && (33)\\
\texttt{time} &\rightarrow [1:1+w]\texttt{'it is'}\ [\![DateTime.Now]\!]\ \texttt{','}\ \dot{n} \blacktriangleright w := 5. && (34)\\
\texttt{time} &\rightarrow [w:1+w]\texttt{'it is'}\ [\![DateTime.Now.Hour]\!]\ \texttt{'o'clock'} \blacktriangleright w := w - 1. && (35)\\
\texttt{stop} &\leftarrow [1:1]\texttt{'stop'} \mid \texttt{'off'}. && (36)
\end{aligned}
$$

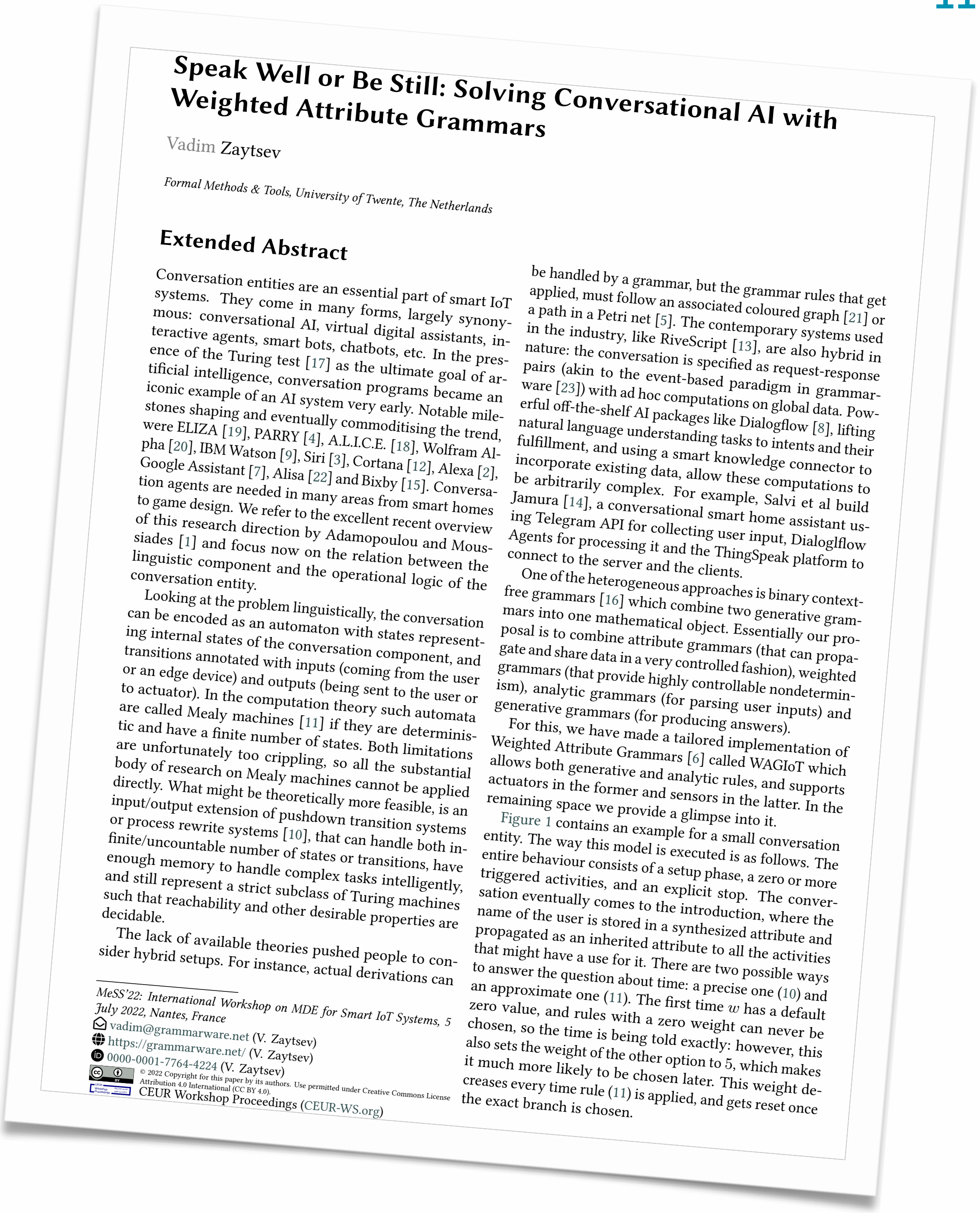
WAGIoT Example

<N,A,T,R,P,C,S>

$$
\begin{aligned}
S\,\{\dot{n}, \underset{\cdot}{n}\} &\leftarrow [1:1]\texttt{setup activity}^{*}\ \texttt{stop} \blacktriangleright \dot{n} := \underset{\cdot}{n}. && (37)\\
\texttt{setup}\,\{\underset{\cdot}{n}\} &\leftarrow [1:1]\texttt{greet}?\ \texttt{getname}. && (38)\\
\texttt{greet}\,\{c_1, c_2\} &\leftarrow [c_1 : c_1 + c_2]\texttt{'hello'} \blacktriangleright c_1 := c_1 + 1. && (39)\\
\texttt{greet}\,\{c_1, c_2\} &\leftarrow [c_1 : c_1 + c_2]\texttt{'good morning'} \blacktriangleright c_2 := c_2 + 1. && (40)\\
\texttt{greet}\,\{\} &\rightarrow [1:1]\texttt{'greetings, human!'}\ \texttt{'what is your name?'}. && (41)\\
\texttt{getname}\,\{\underset{\cdot}{n}\} &\leftarrow [1:1]\texttt{'I am'}\ \underset{\cdot}{n} := \texttt{Id}. && (42)\\
\texttt{getname}\,\{\underset{\cdot}{n}\} &\rightarrow [1:1]\texttt{'nice to meet you,'}\ \underset{\cdot}{n}. && (43)\\
\texttt{activity}\,\{\} &\leftarrow [1:x]\texttt{time} \mid \cdots. && (44)\\
\texttt{time}\,\{w, \dot{n}\} &\leftarrow [1:1]\texttt{'what time is it?'}. && (45)\\
\texttt{time}\,\{w, \dot{n}\} &\rightarrow [1:1+w]\texttt{'it is'}\ [\![DateTime.Now]\!]\ \texttt{','}\ \dot{n} \blacktriangleright w := 5. && (46)\\
\texttt{time}\,\{w, \dot{n}\} &\rightarrow [w:1+w]\texttt{'it is'}\ [\![DateTime.Now.Hour]\!]\ \texttt{'o'clock'} \blacktriangleright w := w - 1. && (47)\\
\texttt{stop}\,\{\} &\leftarrow [1:1]\texttt{'stop'} \mid \texttt{'off'}. && (48)
\end{aligned}
$$

Conclusion

- Grammars, unite!
  - analytic
  - generative
  - weighted
  - attribute
- MDE, MT, …, slicing?
- cf. event-based [doi]
- Questions?



Speak Well or Be Still: Solving Conversational AI with Weighted Attribute Grammars

Vadim Zaytsev

Formal Methods & Tools, University of Twente, The Netherlands

Extended Abstract

Conversation entities are an essential part of smart IoT systems. They come in many forms, largely synonymous: conversational AI, virtual digital assistants, interactive agents, smart bots, chatbots, etc. In the presence of the Turing test [17] as the ultimate goal of artificial intelligence, conversation programs became an iconic example of an AI system very early. Notable milestones shaping and eventually commoditising the trend, were ELIZA [19], PARRY [4], A.L.I.C.E. [18], Wolfram Alpha [20], IBM Watson [9], Siri [3], Cortana [12], Alexa [2], Google Assistant [7], Alisa [22] and Bixby [15]. Conversation agents are needed in many areas from smart homes to game design. We refer to the excellent recent overview of this research direction by Adamopoulou and Moussiades [1] and focus now on the relation between the linguistic component and the operational logic of the conversation entity.

Looking at the problem linguistically, the conversation can be encoded as an automaton with states representing internal states of the conversation component, and transitions annotated with inputs (coming from the user or an edge device) and outputs (being sent to the user or to actuator). In the computation theory such automata are called Mealy machines [11] if they are deterministic and have a finite number of states. Both limitations are unfortunately too crippling, so all the substantial body of research on Mealy machines cannot be applied directly. What might be theoretically more feasible, is an input/output extension of pushdown transition systems or process rewrite systems [10], that can handle both infinite/uncountable number of states or transitions, have enough memory to handle complex tasks intelligently, and still represent a strict subclass of Turing machines such that reachability and other desirable properties are decidable.

The lack of available theories pushed people to consider hybrid setups. For instance, actual derivations can be handled by a grammar, but the grammar rules that get applied, must follow an associated coloured graph [21] or a path in a Petri net [5]. The contemporary systems used in the industry, like RiveScript [13], are also hybrid in nature: the conversation is specified as request-response pairs (akin to the event-based paradigm in grammarware [23]) with ad hoc computations on global data. Powerful off-the-shelf AI packages like Dialogflow [8], lifting natural language understanding tasks to intents and their fulfillment, and using a smart knowledge connector to incorporate existing data, allow these computations to be arbitrarily complex. For example, Salvi et al build Jamura [14], a conversational smart home assistant using Telegram API for collecting user input, Dialogflow Agents for processing it and the ThingSpeak platform to connect to the server and the clients.

One of the heterogeneous approaches is binary context-free grammars [16] which combine two generative grammars into one mathematical object. Essentially our proposal is to combine attribute grammars (that can propagate and share data in a very controlled fashion), weighted grammars (that provide highly controllable nondeterminism), analytic grammars (for parsing user inputs) and generative grammars (for producing answers).

For this, we have made a tailored implementation of Weighted Attribute Grammars [6] called WAGIoT which allows both generative and analytic rules, and supports actuators in the former and sensors in the latter. In the remaining space we provide a glimpse into it.

Figure 1 contains an example for a small conversation entity. The way this model is executed is as follows. The entire behaviour consists of a setup phase, a zero or more triggered activities, and an explicit stop. The conversation eventually comes to the introduction, where the name of the user is stored in a synthesized attribute and propagated as an inherited attribute to all the activities that might have a use for it. There are two possible ways to answer the question about time: a precise one (10) and an approximate one (11). The first time w has a default zero value, and rules with a zero weight can never be chosen, so the time is being told exactly: however, this also sets the weight of the other option to 5, which makes it much more likely to be chosen later. This weight decreases every time rule (11) is applied, and gets reset once the exact branch is chosen.

MeSS'22: International Workshop on MDE for Smart IoT Systems, 5 July 2022, Nantes, France

vadim@grammarware.net (V. Zaytsev)
https://grammarware.net/ (V. Zaytsev)
0000-0001-7764-4224 (V. Zaytsev)