

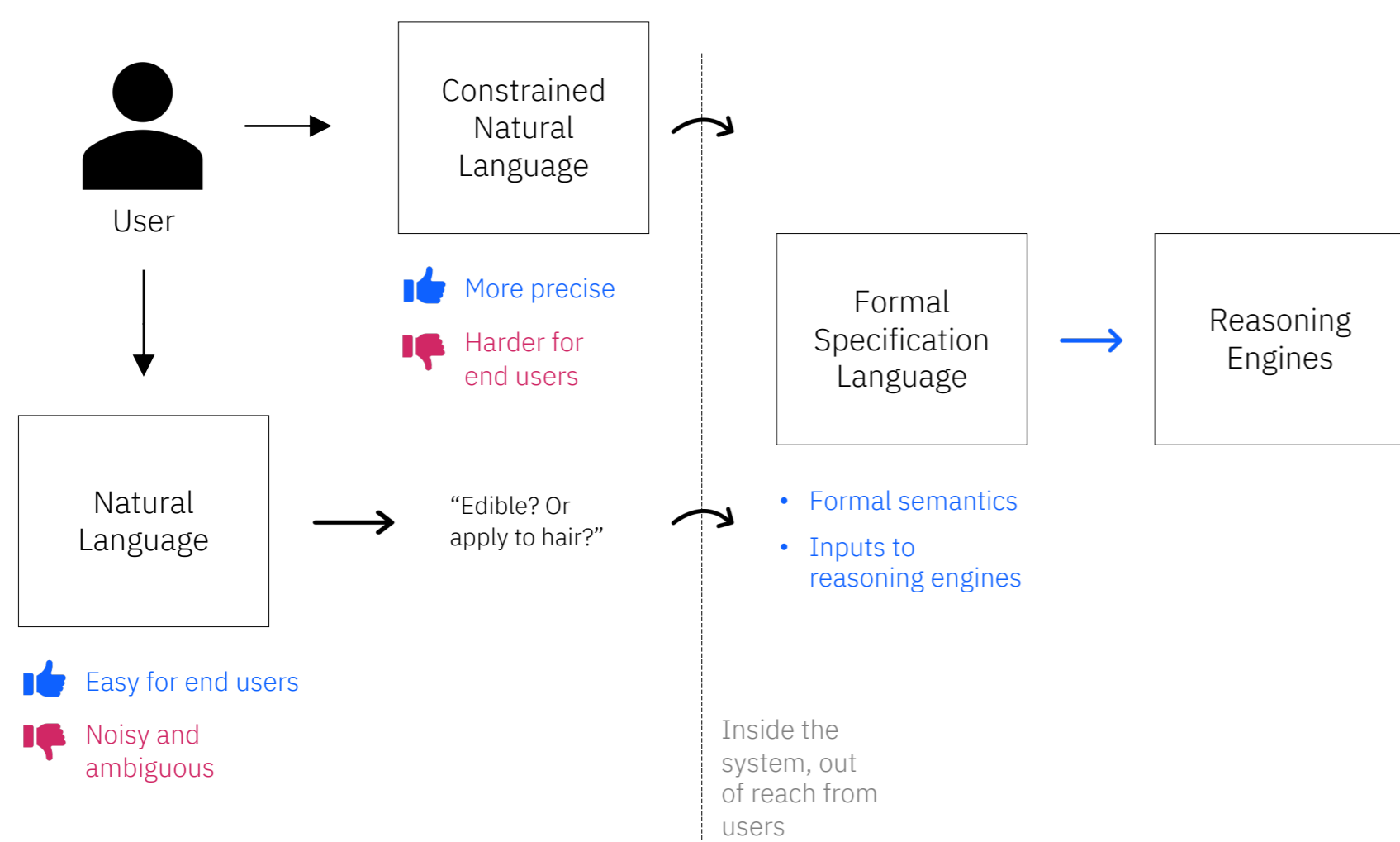
Natural Language is the Future, the Future is the Past(a)!

Francesco Fuggitti, Tathagata Chakraborti

IBM Research

Natural Language is the Future!

Reasoning with user inputs

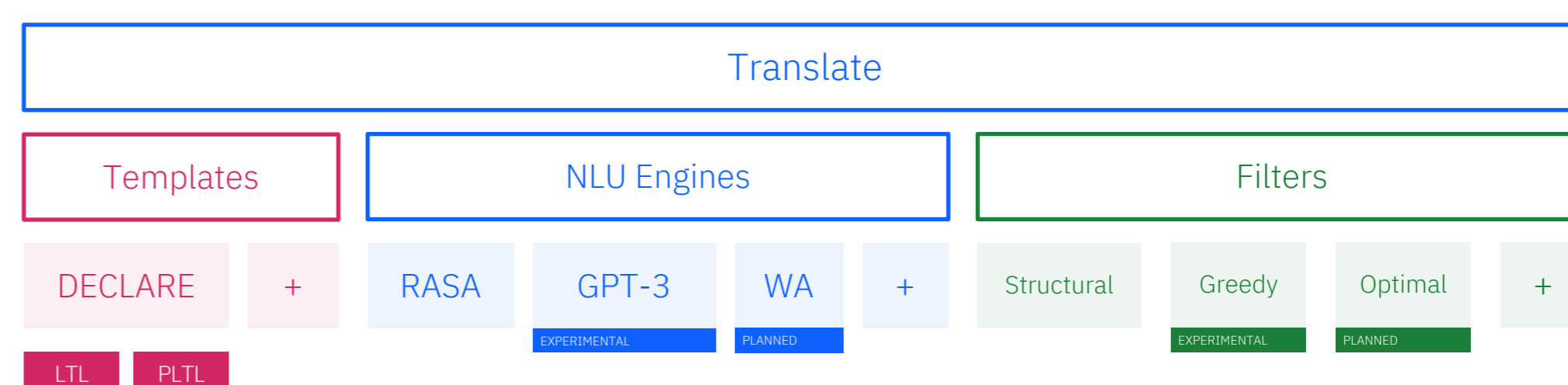


Linear-time Temporal Logic

A wonderful formal language to describe control rules.

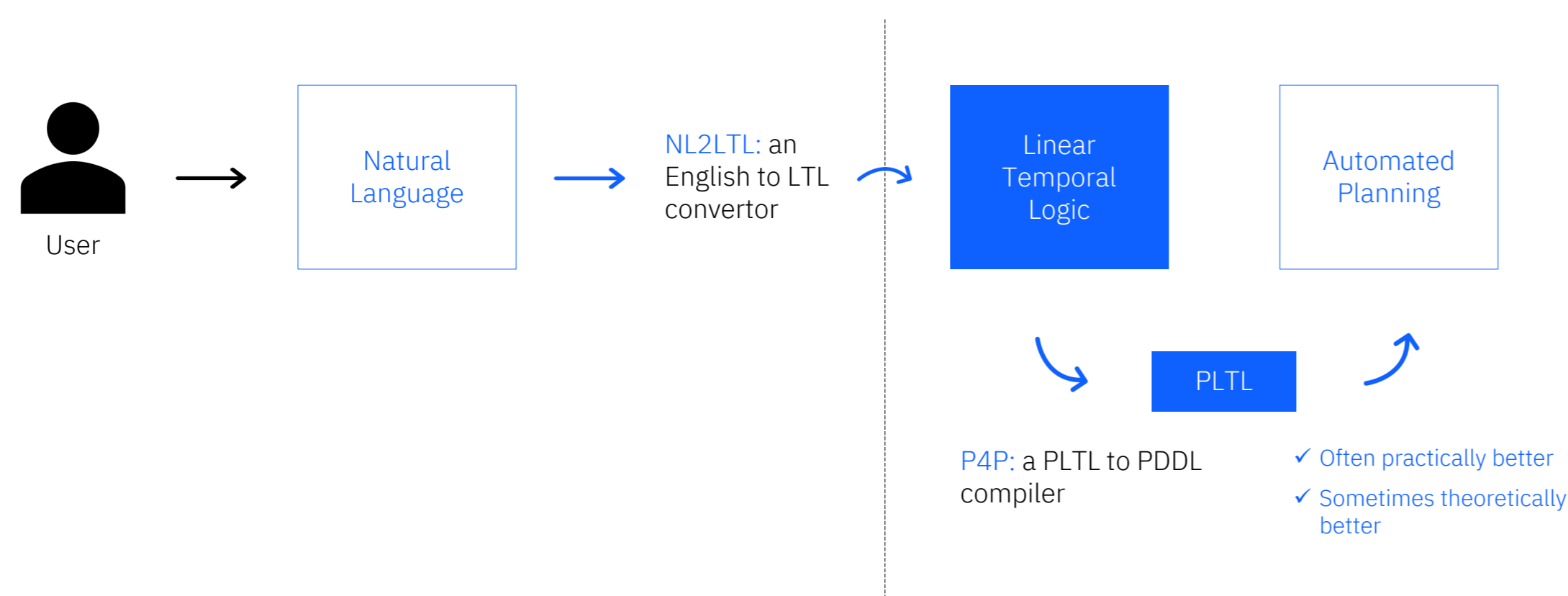
$\text{button} \rightarrow X(\text{red} \cup (\text{elevator} \wedge \text{doors}))$	After the <u>button</u> is pressed, the light will turn <u>red</u> until the <u>elevator</u> arrives at the floor and the <u>doors</u> open.
$G(\text{weekdays} \wedge 7\text{am}) \rightarrow \text{coffee}$	Always make <u>coffee</u> at <u>7am</u> on <u>weekdays</u> .
$G(\text{red} \rightarrow \neg X \text{green})$	Once the traffic light is <u>red</u> , the light cannot become <u>green</u> immediately after.
$G(\text{request} \rightarrow F \text{acknowledgement})$	Every <u>request</u> will eventually receive an <u>acknowledgement</u> .
$G(\text{message} \rightarrow (\neg \text{sent} \cup \text{acknowledgement}))$	When a <u>message</u> is sent, an <u>acknowledgment</u> will eventually be returned, and the message will not be marked as <u>sent</u> before an <u>acknowledgment</u> is returned.

NL2LTL - API overview



The Future is the Past!

If you can, go with the Past¹



¹Planning for Temporally Extended Goals in Pure-Past Linear Temporal Logic. Bonassi, L.; De Giacomo, G.; Favorito, M.; Fuggitti, F.; Gerevini, A.E.; Scala, E. In ICAPS, 2023.

Pure-Past Linear-time Temporal Logic

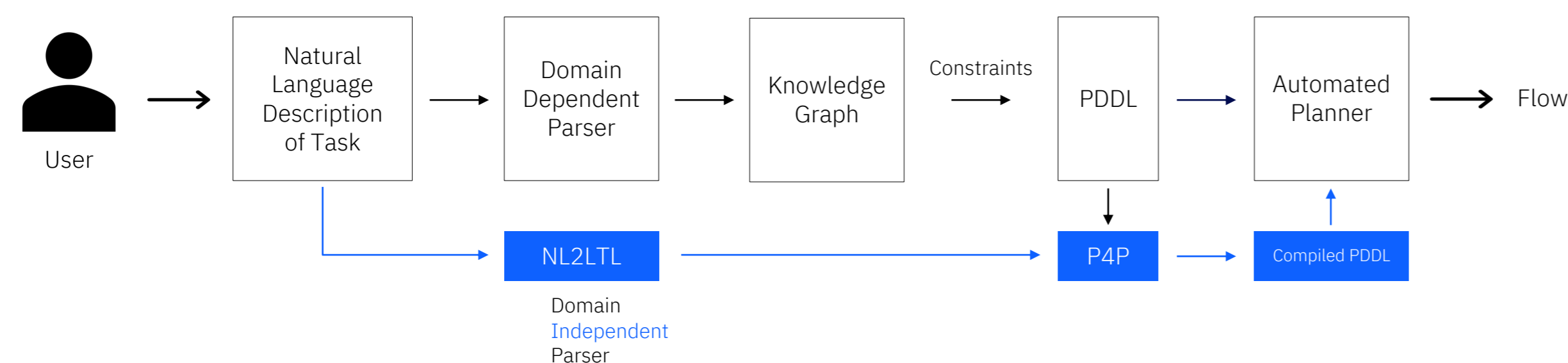
As wonderful as LTL, less natural but better in practice!

$\text{task} \wedge (\neg \text{room S sanitized})$	Before achieving the <u>task</u> , the agent was not in the <u>room</u> anymore since the room was <u>sanitized</u> .
$\text{goal} \wedge Y(\text{condition})$	Enforce the agent to achieve the <u>goal</u> after some <u>condition</u> has been met
$H(\text{green} \rightarrow Y \text{yellow})$	Every time the traffic light is green, it has always been preceded by the <u>yellow</u> light.
$H(\text{ticket} \rightarrow Y(\neg \text{bus S ticket}))$	The agent has always paid the <u>ticket</u> before getting the <u>bus</u>

Envisioned Product Impact

Alternative NLP processing pipeline: Stacked parsers

A Goal-driven Natural Language Interface for Creating Application Integration Workflows. Michelle Brachman, Christopher Bygrave, Tathagata Chakraborti, Arunima Chaudhary, Zhining Ding, Casey Dugan, David Gros, Thomas Gschwind, J Johnson, Jim Laredo, Christoph Miksovic Czasch, Qian Pan, Priyanshu Rai, Ramkumar Ramalingam, Paolo Scotton, Nagarjuna Surabathina, and Kartik Talamadupula. AAAI 2022 Demonstration.



Describe your flow in a sentence in plain english
Example: I want to be notified via my IBM email when my colleagues schedule vacation on their Workdays

BEFORE

- Developer must manually inspect AMR trees and write code for a few limited patterns.
- We also lose all such patterns when we move to a new parser e.g., UA.

AFTER

- Domain dependent parser + knowledge graph get us what things are in the flow.
- Domain independent parser gives us the control rules.