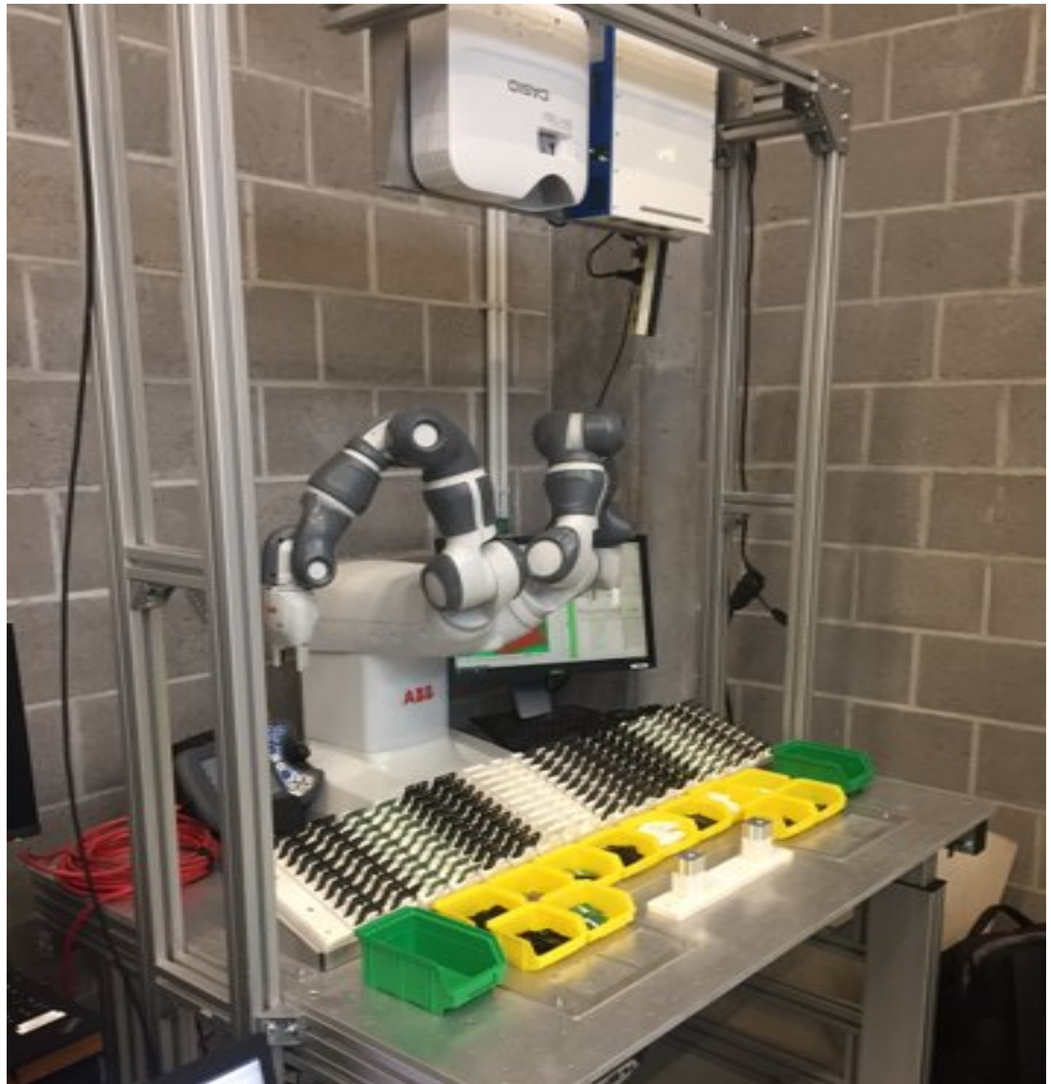
Knowledge Representation using Dependent Type Theory

Nick Harley
njharley@ai.vub.ac.be
June 21, 2022


ARTIFICIAL
INTELLIGENCE
RESEARCH GROUP


Application: AI Assisted Operator

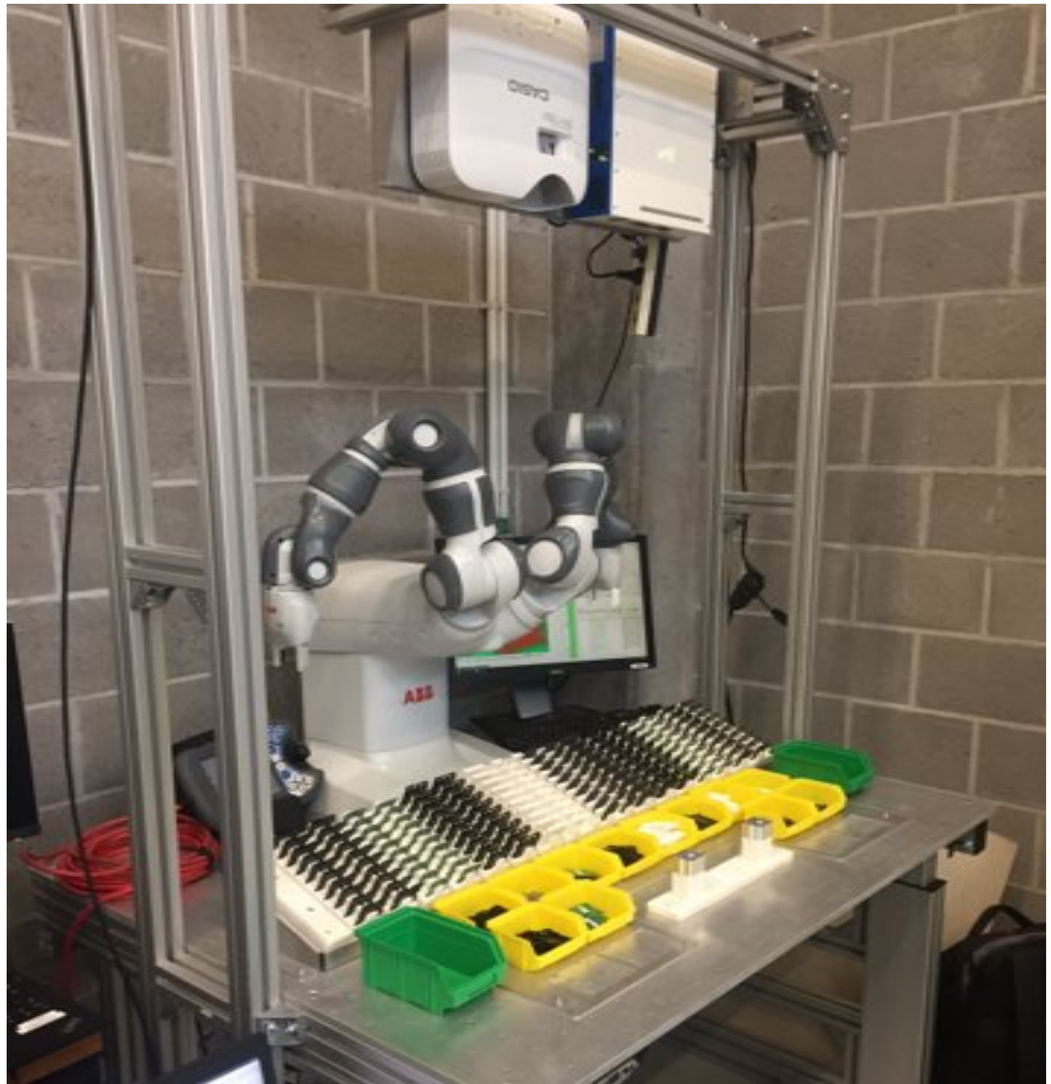


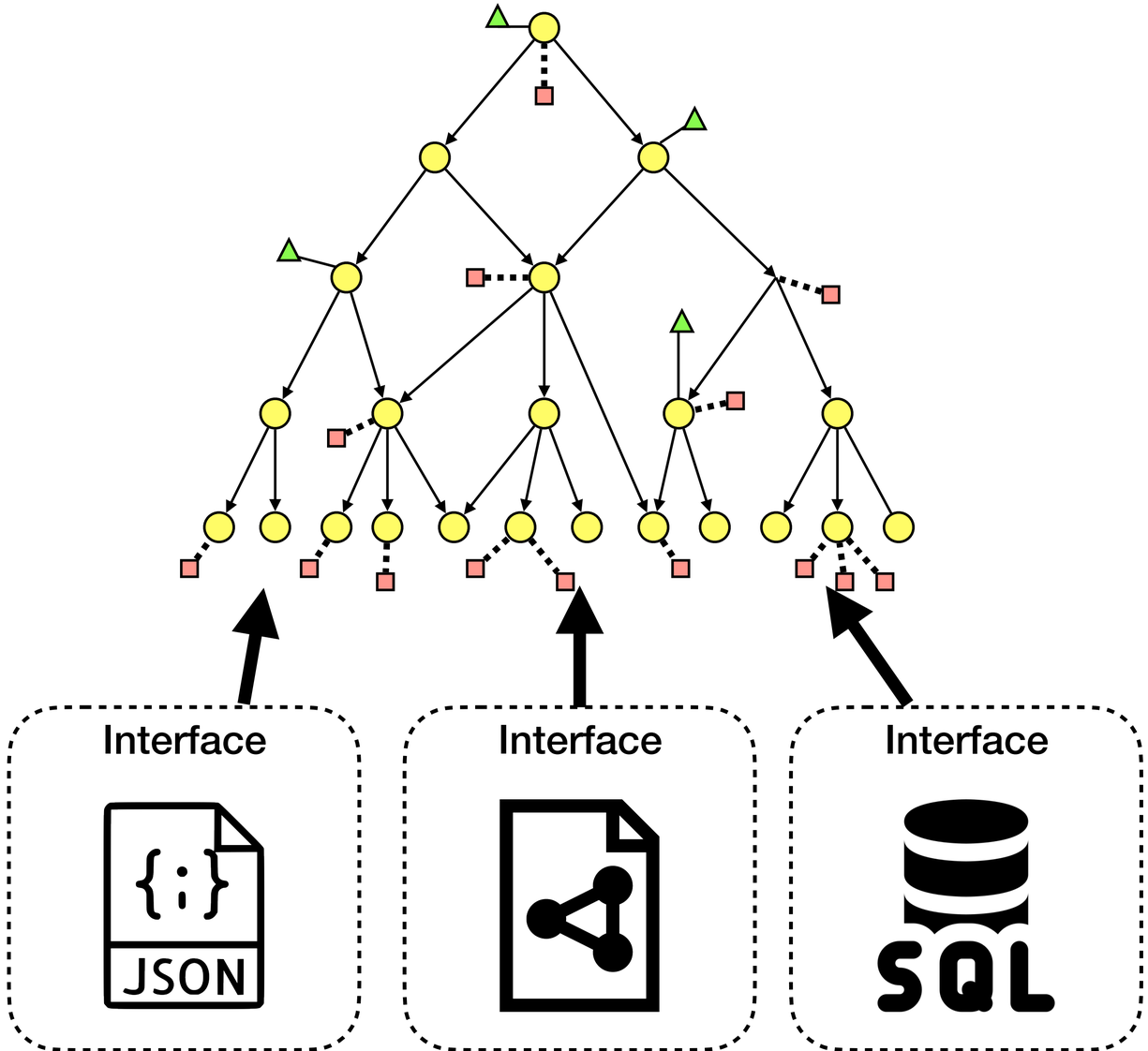
- Historical execution data (SQL)
- Robot world model (RDF)
- Execution plans (PDDL)
- Assembly instructions (JSON)

Knowledge Representation: CHAKRA

FLANDERS MAKE
DRIVING INNOVATION IN MANUFACTURING

Common **H**ierarchical **A**bstract **K**nowledge **R**epresentation for **A**nything

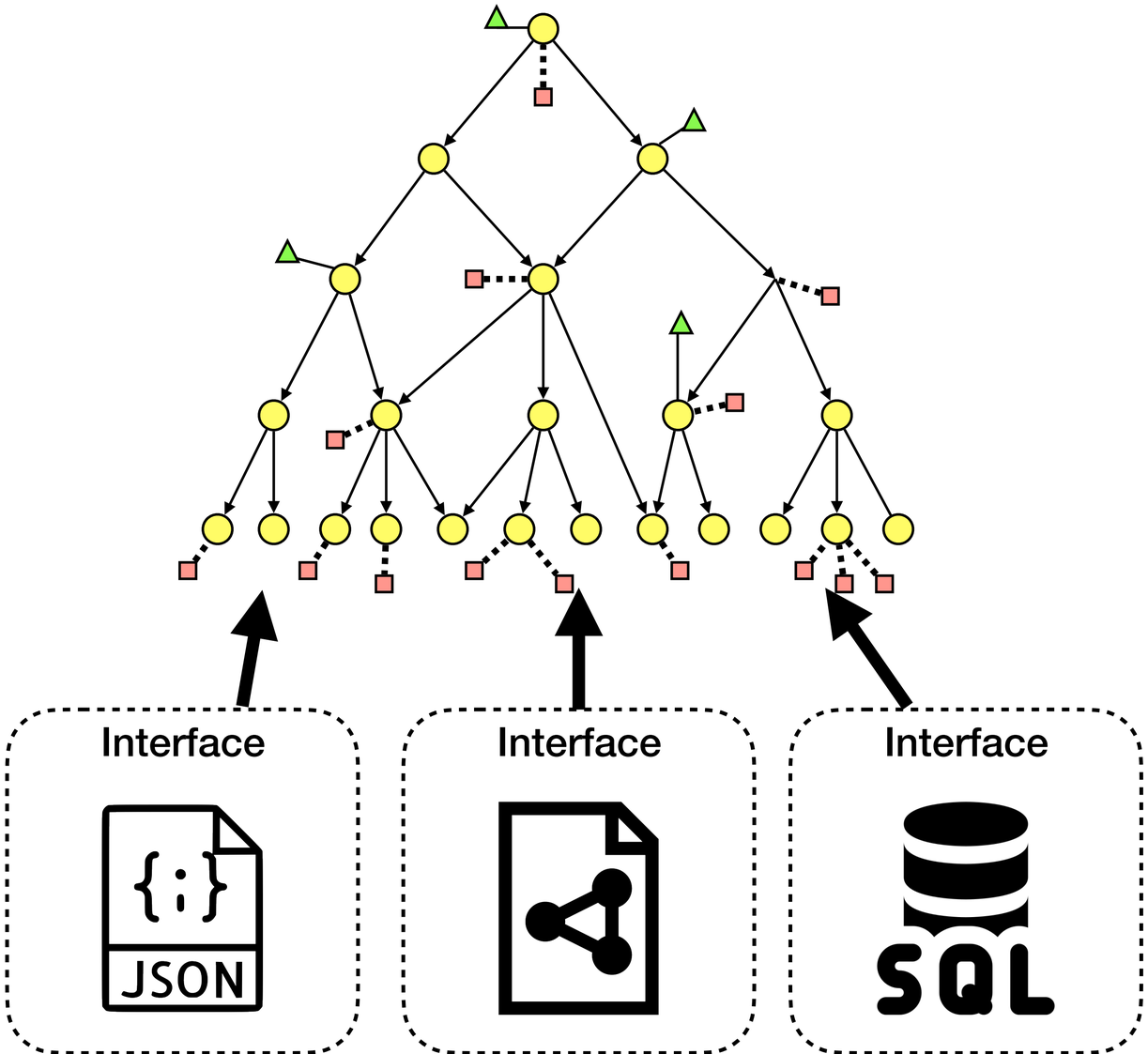




Knowledge Representation: CHAKRA

Example: A specification for the execution x of an assembly step y in which an operator error z occurred:

```
Exists x,y,R.
  IsExecutionOf(x,y,R) /\
  Exists z.
    HasPart(x,z) /\
    LogType(z,Error)
```



Logical Specification: Dependent Type Theory

The calculus of inductive constructions:

1. A functional programming language
2. A constructive higher-order logic

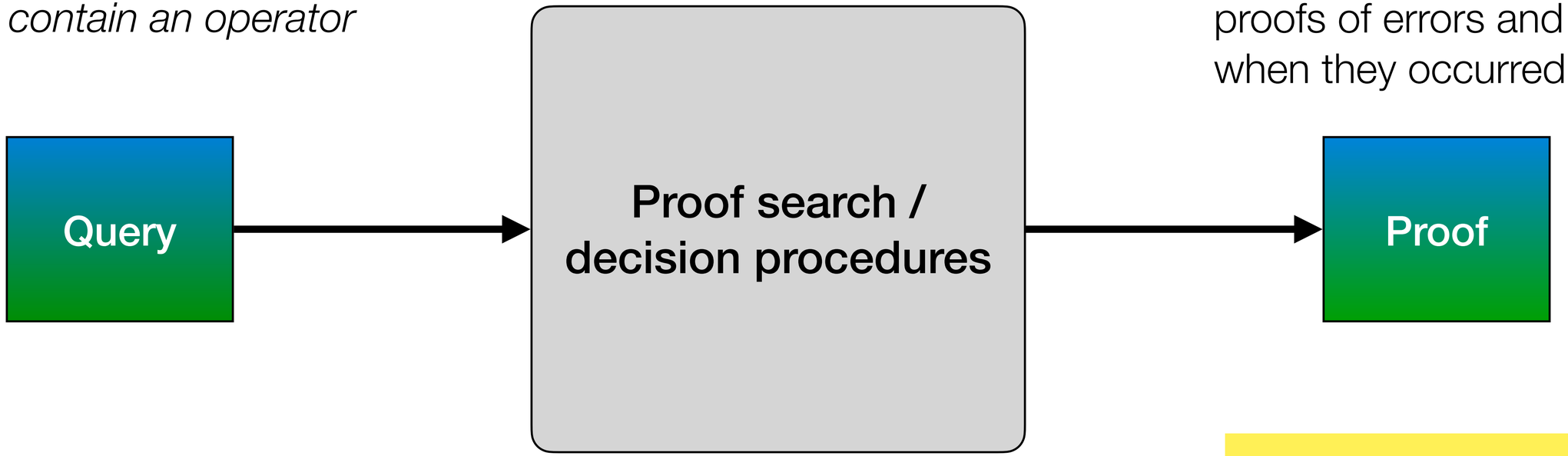
Data, programs, propositions, predicates, theorems, proofs and decision procedures are all expressible as terms in the language

```
Nat : Type

0 : Nat

[1, 2, 3] : List(Nat)

λ(x:Nat).x + 1 : Nat -> Nat

Prop : Type

Even : Nat -> Prop

even0 : Even(0)

odd_s(0,even0) : Odd(1)

even_s(1,odd_s(0,even0)) : Even(2)

decideifeven : forall x:Nat,
     Even(x) \/ ~ Even(x)
```

Reasoning: Proof Not Truth

“Find all assembly steps which contain an operator error”

Data structure containing proofs of errors and where / when they occurred

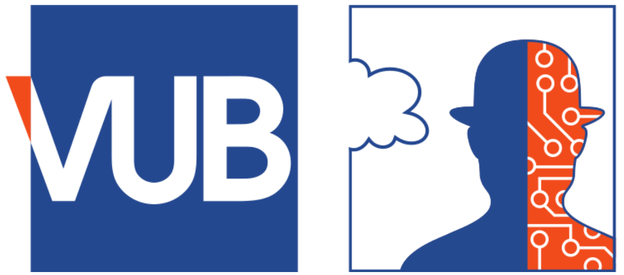


Advantages:

1. Better integration with programming languages
2. Compromise between open and closed world

Proof objects are data which can be computed with

Thanks!



ARTIFICIAL
INTELLIGENCE
RESEARCH GROUP

AI FLANDERS
BUILDING OUR DIGITAL FUTURE