

First steps along the road to Digital Engineering: The promises (and pitfalls) of SysML v2

4th Model Based Space Systems and Software Engineering

Based on materials from
Ed Seidewitz and Sandy Friedenthal



Ákos Horváth PhD, CTO

INC**QUERY**
GROUP

Denver, CO, USA
Vienna, Austria
Budapest, Hungary

The Future of System Engineering

From Document-Centric to Model-based
and to *Digital Engineering*

Part of the digital transformation

Full life cycle from SoS to component level

Agile system development including automated workflow and CM of the
digital thread

Model patterns and reuse

Aim is to

managing complexity & risk

more rapidly respond to change

reuse and design evolution

reasoning about & analyzing systems

shared stakeholder understanding

automated documentation & reporting



Source: INCOSE SE Vision 2035



SysML language

Origins



SysML has evolved to address user and vendor needs

- *v1.0 adopted in 2006; v1.7 adopted 2022*
- Tailored to the needs of System Engineering
- Based on the UML language
- SysMLv1 has facilitated awareness and adoption of MBSE

In numbers

> 1000
Projects

17+
Years

70+
Countries

200 >
Tool Vendors

5000 >
papers



Growth



Much has been learned from using SysML v1 for MBSE

- *Became the standard of SE*
- *Strong eco -system and governmental backup*
- Tailored to the needs of System Engineering
- Based on the UML language
- SysMLv1 has facilitated awareness and adoption of MBSE

Future



Next Generation Systems Modeling Language

SysML v2 is the next generation systems modeling language intended to address some of the limitations of SysML v1

- *Beta is out as of June '23*
- Goals:**
- *Better precision and expressiveness*
- *Consistency and integration* to other languages
- *Extensibility* within the language
- *Interoperability* with other *engineering tools*

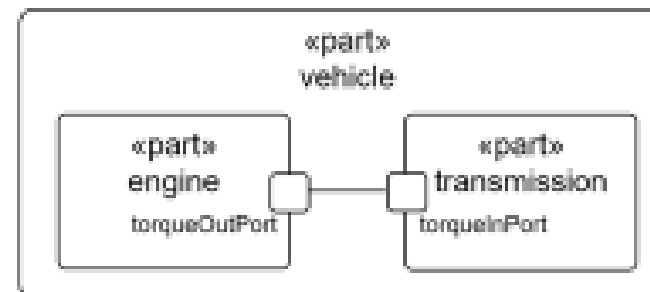
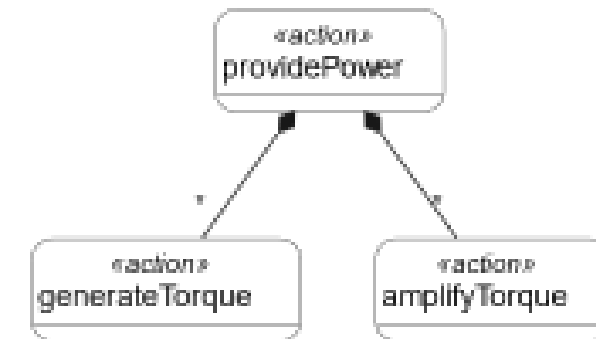
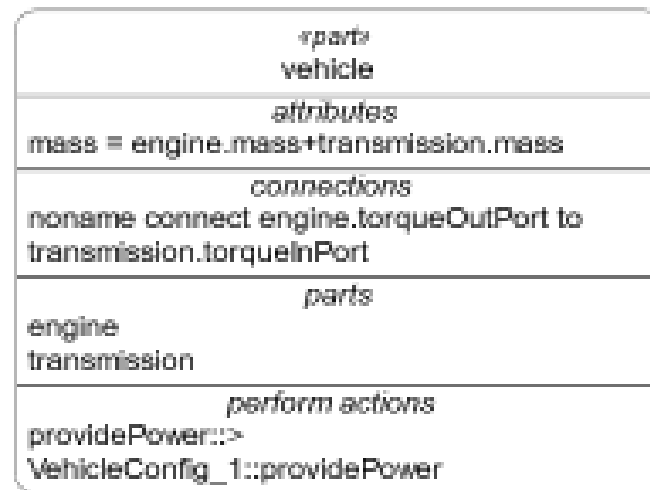
SysML v2 – Brief Introduction

Simple Vehicle Model SysML v2 Textual and Graphical Syntax

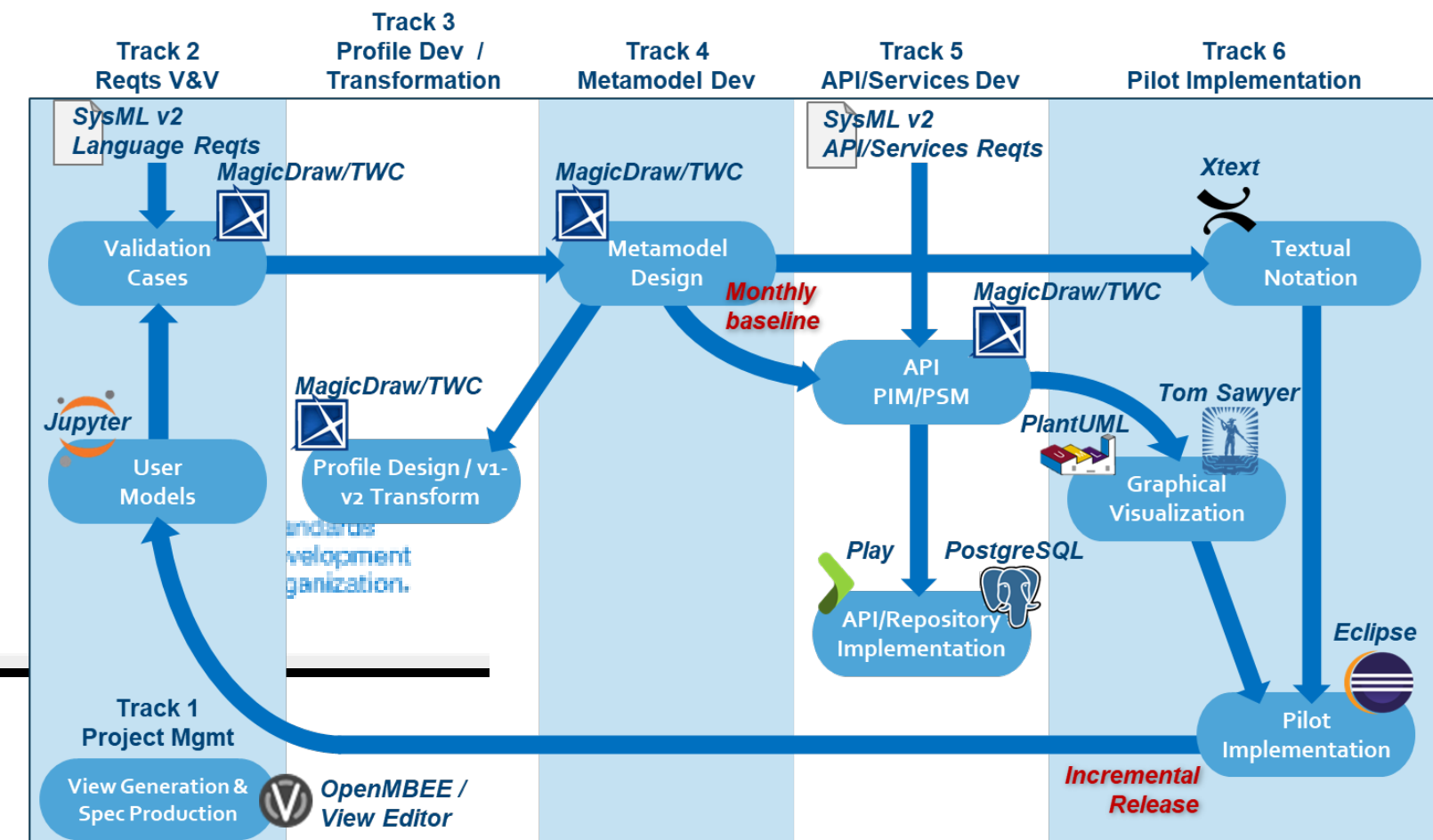
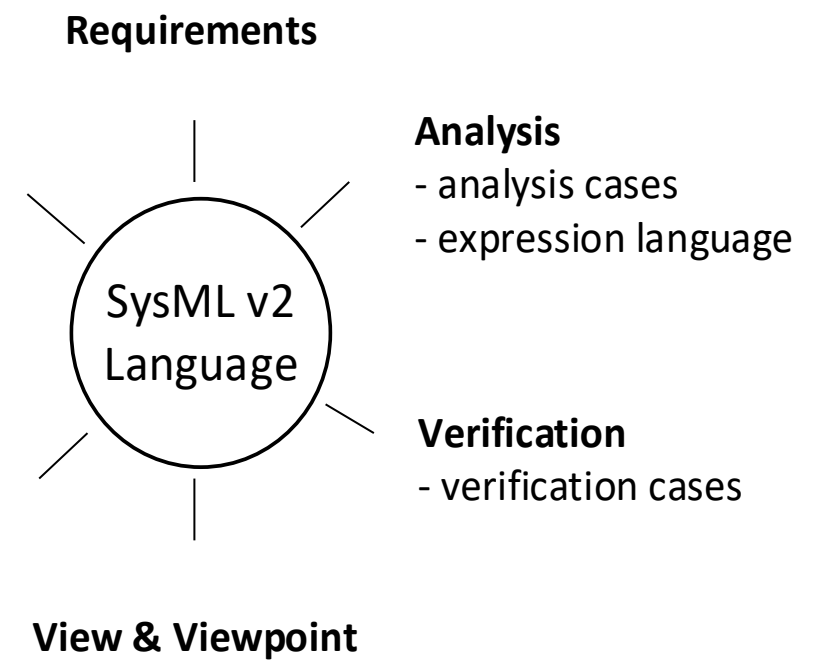
```

part vehicle{
  attribute mass = engine.mass+transmission.mass;
  perform providePower;
  part engine{
    attribute mass;
    port torqueOutPort;
    perform providePower.generateTorque;
  }
  part transmission{
    attribute mass;
    port torqueInPort;
    perform providePower.amplifyTorque;
  }
  connect engine.torqueOutPort to transmission.torqueInPort;
}

action providePower{
  action generateTorque;
  action amplifyTorque;
}
    
```



- Behavior**
- function-based
 - state-based
 - sequence-based
 - use cases
- Structure**
- decomposition
 - interconnection
 - classification



SysML v2 – How it will support me?

More powerful System modelling language

- *Learning curve of the language?*
- *Available methodologies?*
- *Why and How move from v1?*

Extensibility both on language and modelling level

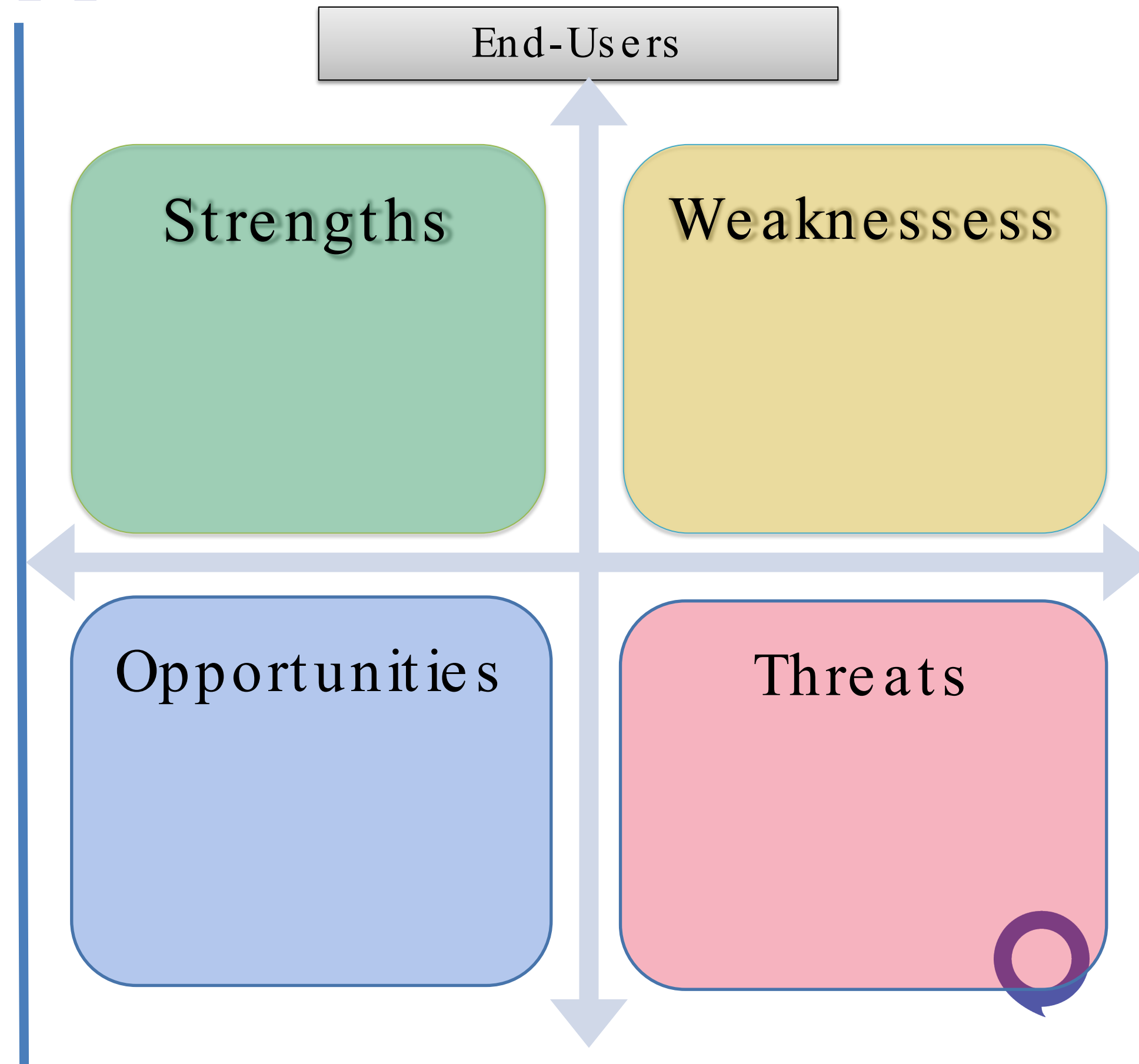
- *Adapting my profile to KerML?*
- *Compatibility of libraries?*

Both graphical and textual modelling front ends

- *Integrated tool support?*
- *Synchronization between the formats?*
- *Model versioning and diffing?*

Easy adaptation from SysML v1

- *Adapting legacy scripts/tooling ?*
- *Model update from v1 to v2?*



SysML v2 – How I will support it?

Standardised , modern API and services for interoperability and fine -grained access

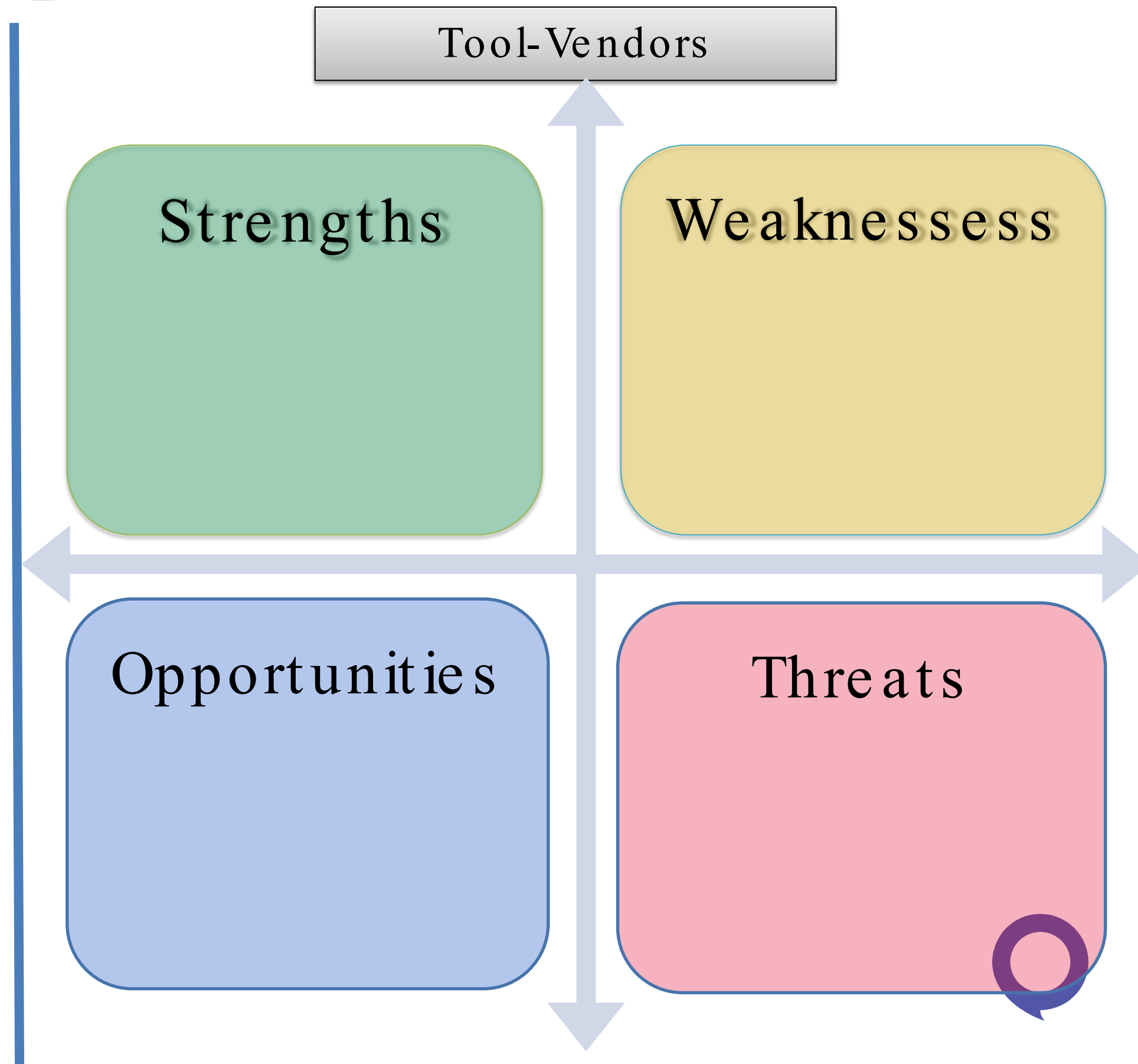
- *True interoperability (not like XMI!)?*
- *Too low-level API?*

Development hurdles/cost of new tooling

- *Synchronization mapping between textual and graphical front -ends?*
- *Standardized ID generation mechanism?*
- *Level of legacy code reuse?*
- *Cross v1-v2 usage of language elements?*
- *Pilot implementation is ready → reference implementation?*
- *Model sizes are growing 3 -5 times → scalability problems?*

Business aspects

- *More complex language → pricing?*
- *Dual support of legacy and new tooling*



SysML v2 – How I will support it and it will support me?

Standardised mapping to other languages?

- *Who will drive the mapping?*
- *AADL, Matlab, formal languages?*

SysML v2 will have a formal semantics

- *When will it be complete?*
- *Will we have a model-checking “capable” subset?*

Regularised specification of analysis or simulation cases, verification cases, and use cases

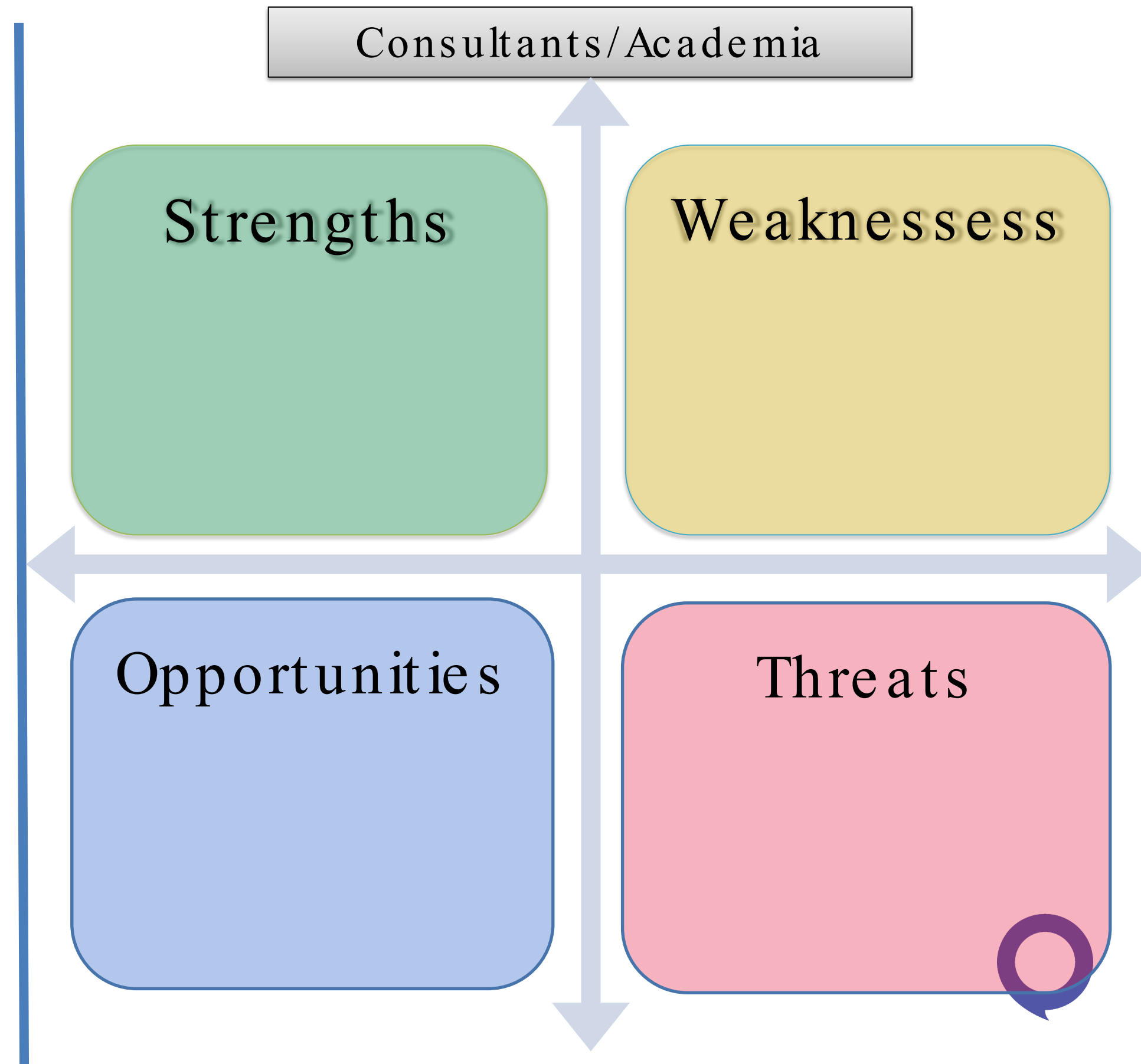
- *Connection to already available approaches (e.g., B-method)*

Extensible domain libraries Mathematical, logical, utility functions,

- *How to evade customer -specific redefinitions?*

New methodologies are on the horizon

- *Learning curve? Past experience?*



Thank
YOU



Ákos Horváth PhD, CTO



akos.horvath@incquerylabs.com



+36 20 415 7393

<https://incquery.io>

