sybila

Executable Biochemical Space for Specification and Analysis of Biochemical Systems

Matej Troják, David Šafránek, Luboš Brim, Jakub Šalagovič, Jan Červený

Systems Biology Laboratory @ Masaryk University
Global Change Research Centre AS CR, v. v. i.

Specification & annotation

The goal of the presentation

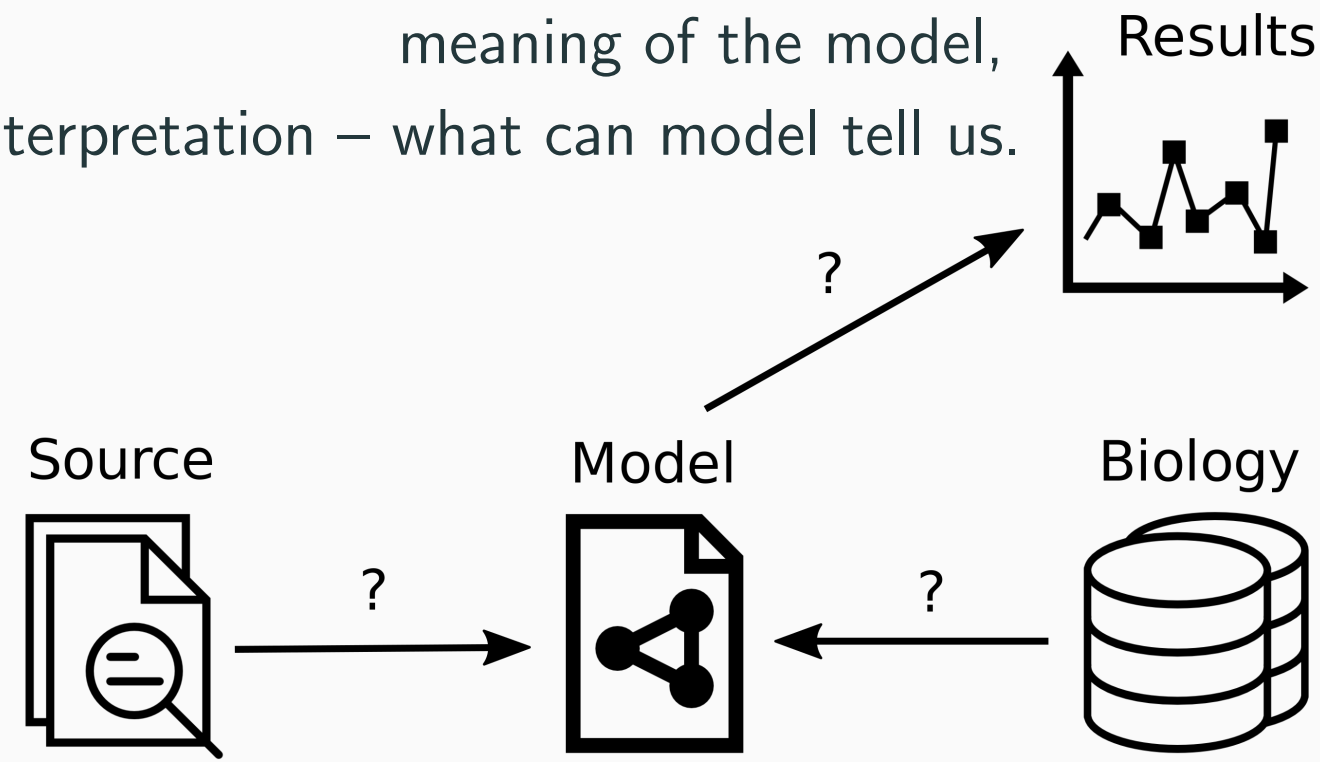
Explain advantages of a BioChemical Space Language

- what is it
- what is it good for
- how it can be used

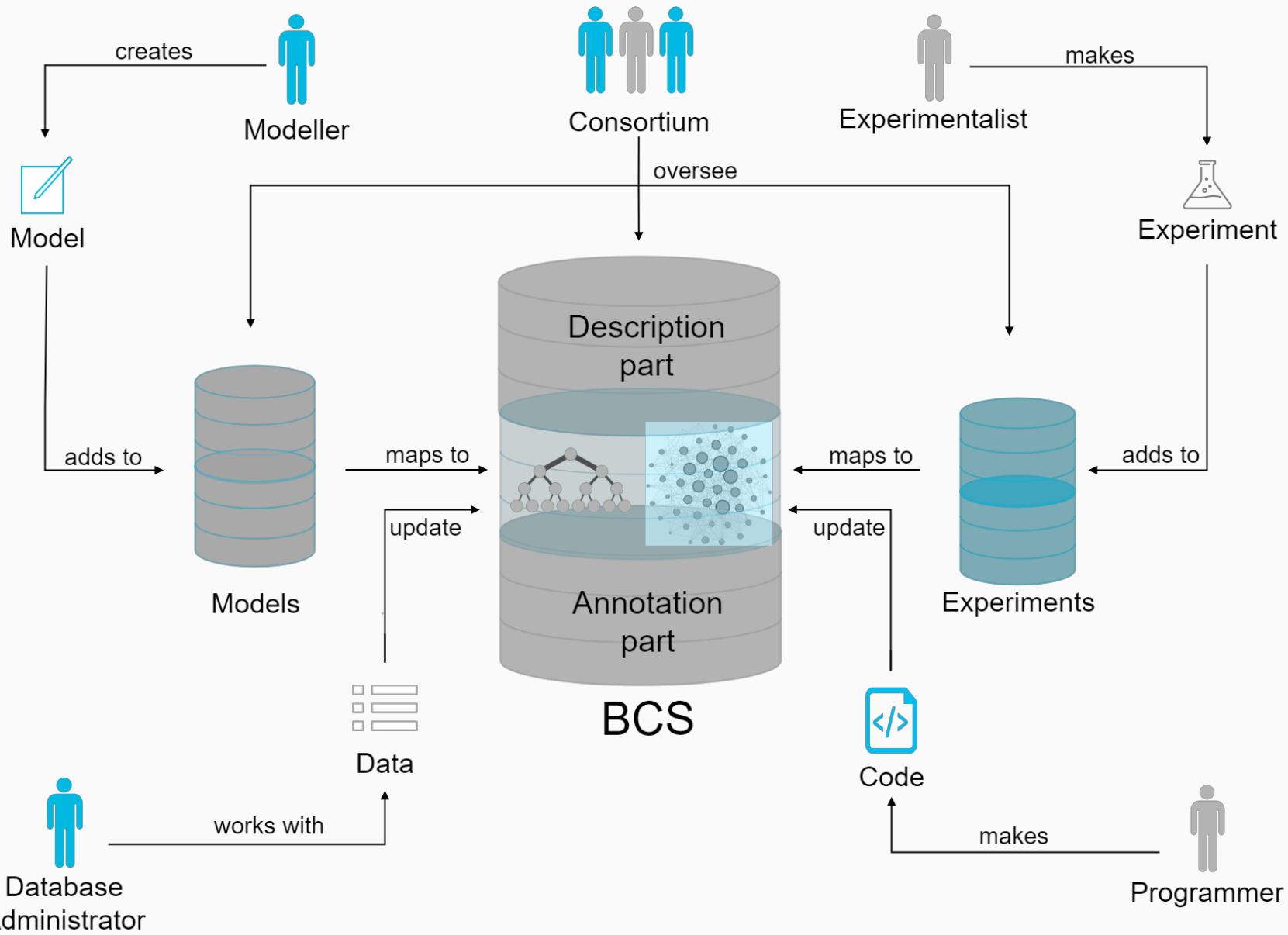
Motivation

Frequent issues with **mathematical models** in systems/synthetic biology:

- reconstruction – from data and previous models,
- understanding – what is the biological meaning of the model,
- interpretation – what can model tell us.



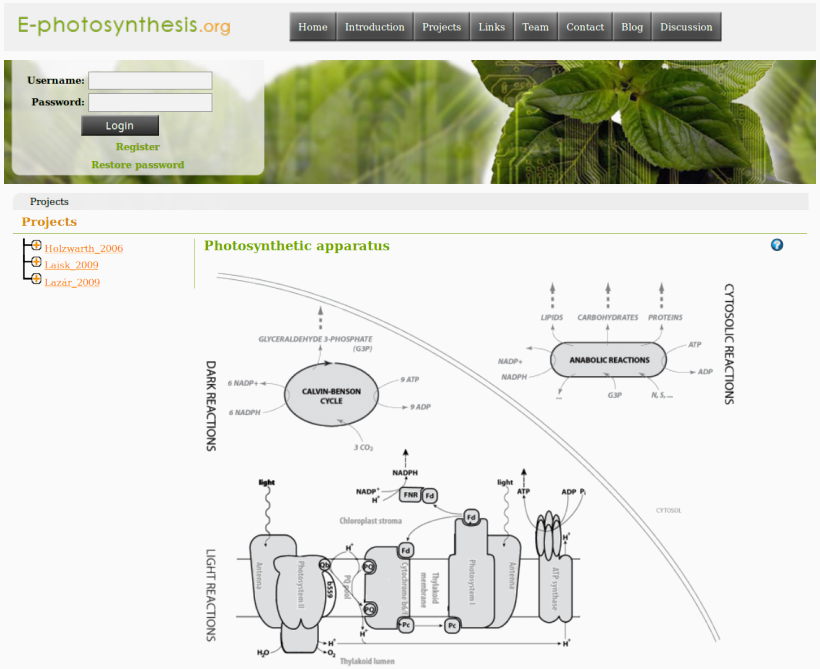
Comprehensive Modelling Platform

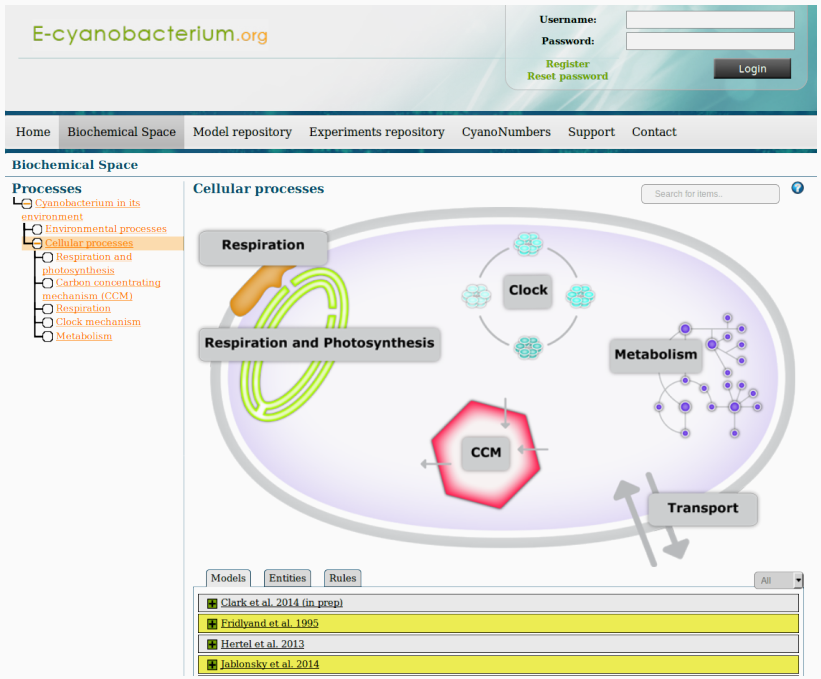


Comprehensive Modelling Platform

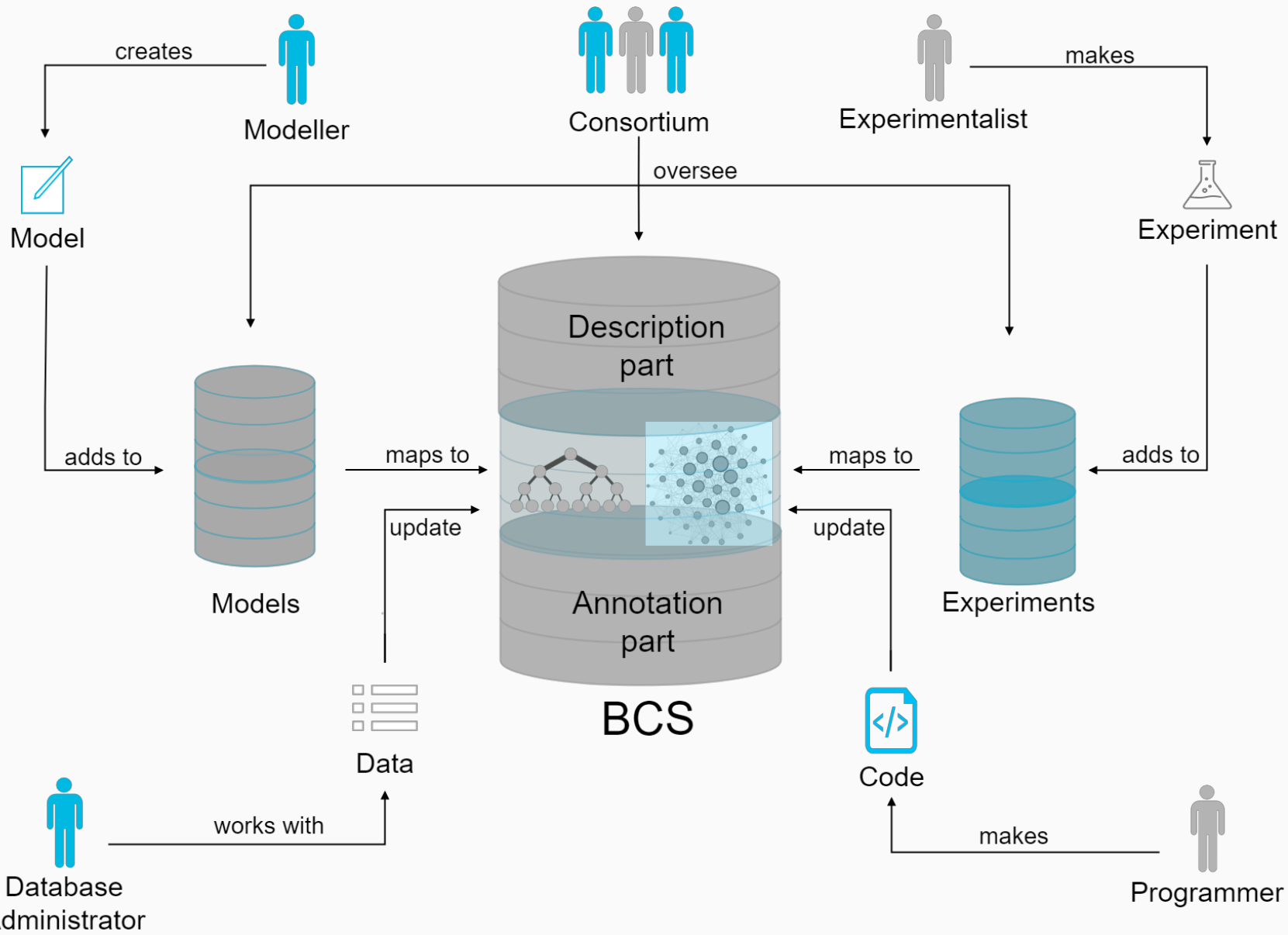
Web-based framework for integration of biological knowledge with computational models and wet-lab experiments.

- `e-photosynthesis.org`
- `e-cyanobacterium.org`





Comprehensive Modelling Platform



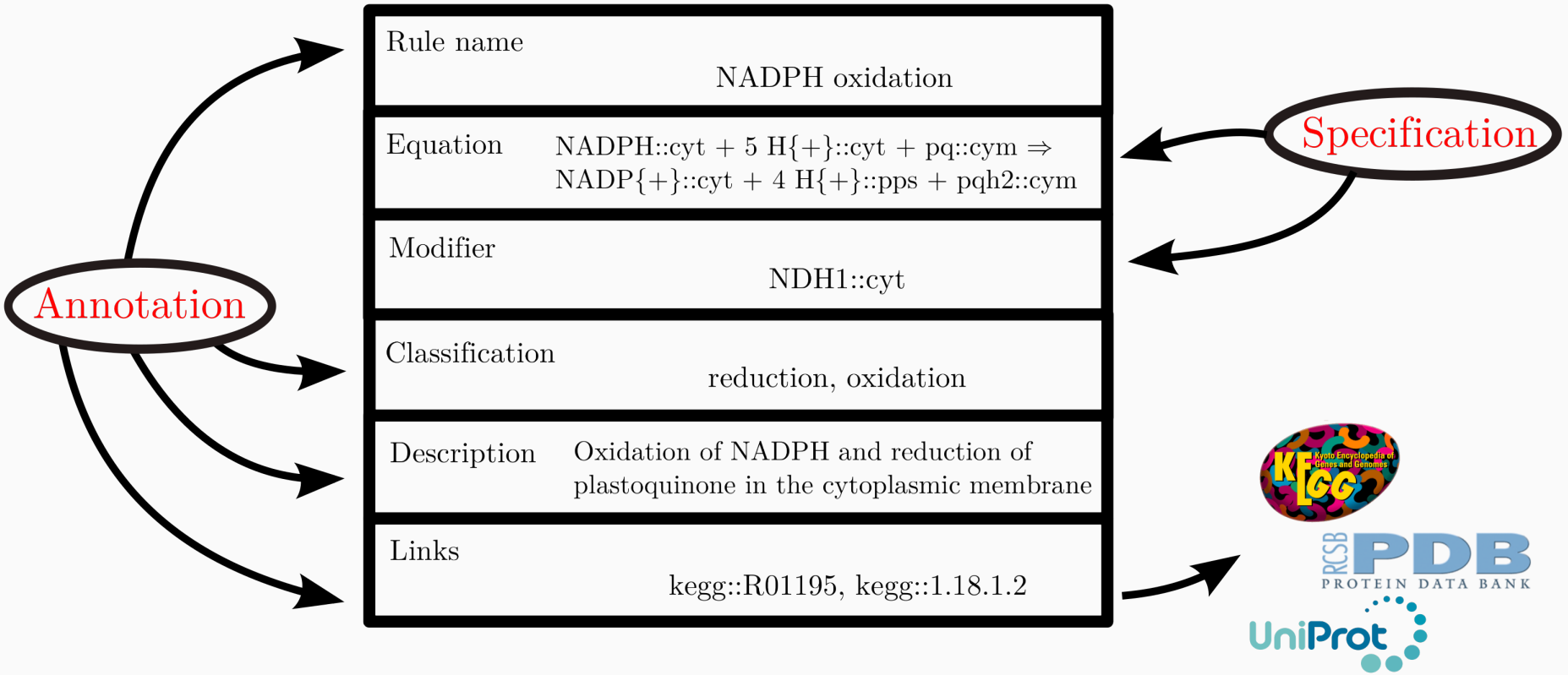
BioChemical Space

BioChemical Space (BCS) – a formal knowledge-base providing

- *specification* – we need to specify objects and relationships among them;
- *annotation* – determine meaning of the objects and relationships in a particular context

of domain-specific biological systems.

Example


Rule name
NADPH oxidation
Equation
NADPH::cyt + 5 H{+}::cyt + pq::cym ⇒
NADP{+}::cyt + 4 H{+}::pps + pqh2::cym
Modifier
NDH1::cyt
Classification
reduction, oxidation
Description
Oxidation of NADPH and reduction of
plastoquinone in the cytoplasmic membrane
Links
kegg::R01195, kegg::1.18.1.2
Annotation
Specification
KEGG
Kyoto Encyclopedia of Genes and Genomes
RCSB PDB
PROTEIN DATA BANK
UniProt

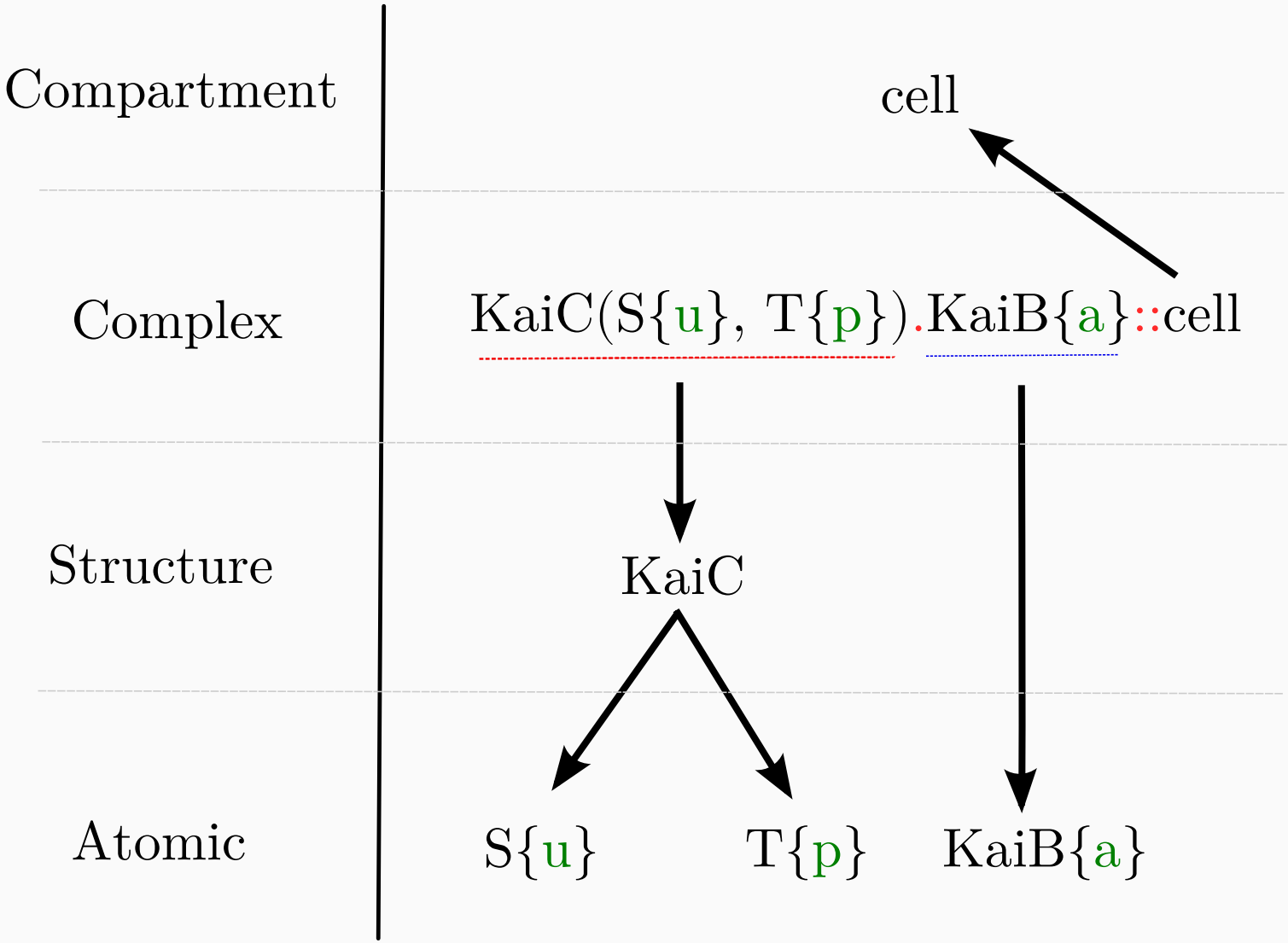
BioChemical Space Language (BCSL)

Important: the target users are outside of computer science (biology, mathematics, chemistry, . . .)

BCSL combines the following aspects:

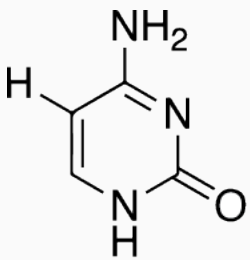
- **human-readability** (easy to read, write, and maintain),
- **rule-based** description – avoiding combinatorial explosion,
- unique level of **abstraction**,
- **hierarchy** – compositional assembly from simpler structures,
- **executability** – operational semantics allowing analysis.

The objects

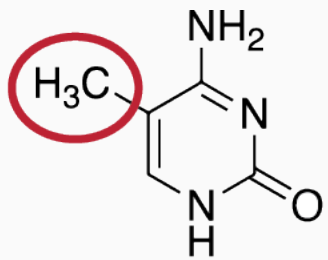

Compartment
Complex
Structure
Atomic
cell
KaiC(S{u}, T{p}).KaiB{a}::cell
KaiC
S{u}
T{p}
KaiB{a}

Atomic agents

- describe the most simple (biological) objects
- variable internal state
- level of abstraction



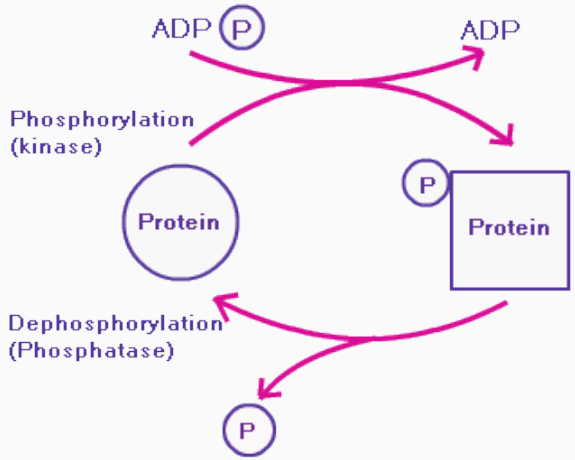
Cytosine



methylated Cytosine

amino acid metylation

Cys{met}, Cys{active}

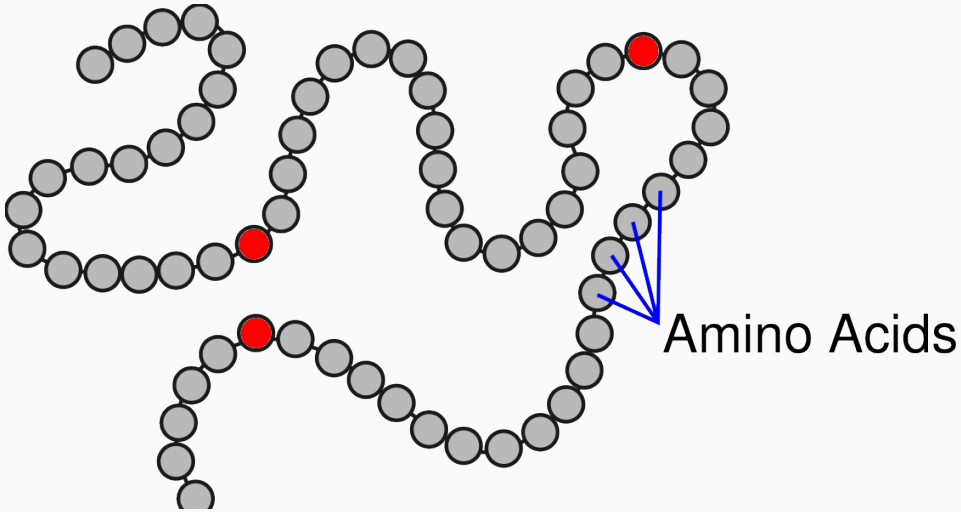


protein phosphorylation

Prot{p}, Prot{u}

Structure agents

- assigned set of unique atomic agents

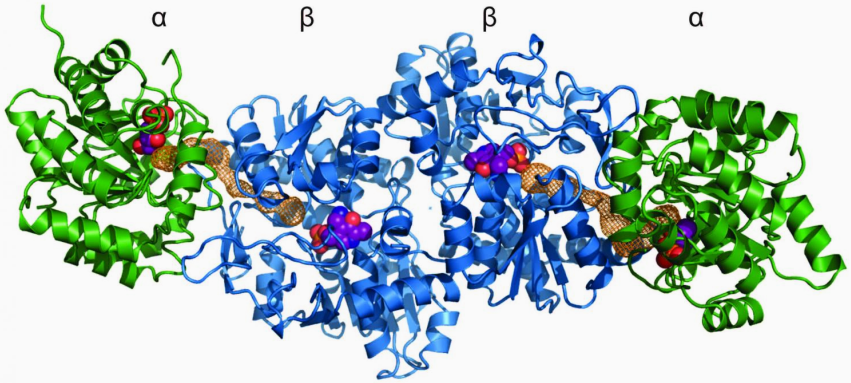


accessible AAs on a protein

Prot(Ser{p}, Cys{u})

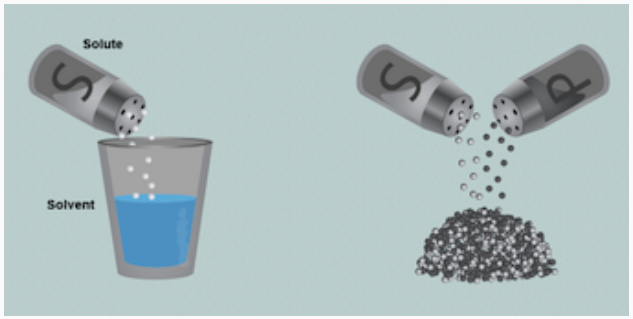
Complex agents

- composed of several atomic and/or structure agents
- abstraction – no particular order, i.e. no bonds
- assigned spatial position – compartment



protein complex

$\alpha.\alpha.\beta.\beta$



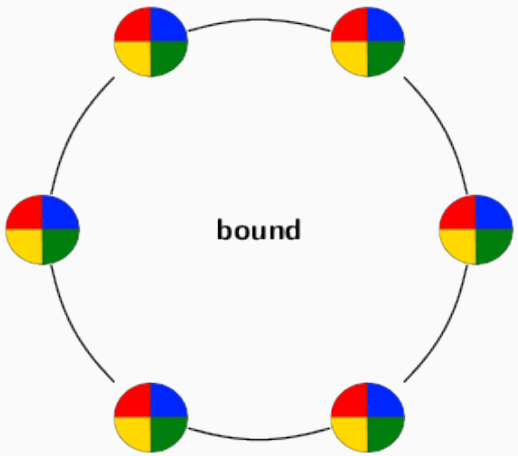
solution/mixture

$H_2O.NaCl$

Abstract Description of Complexes

Biology

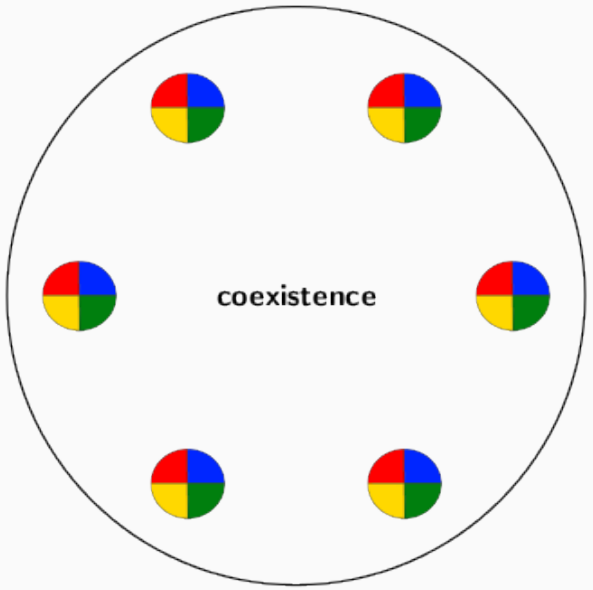
- graph "isomorphism"
- 700 different deviations



- Unphosphorylated protein
- Serine residue phosphorylated protein
- Threonine residue phosphorylated protein
- Both residues phosphorylated protein

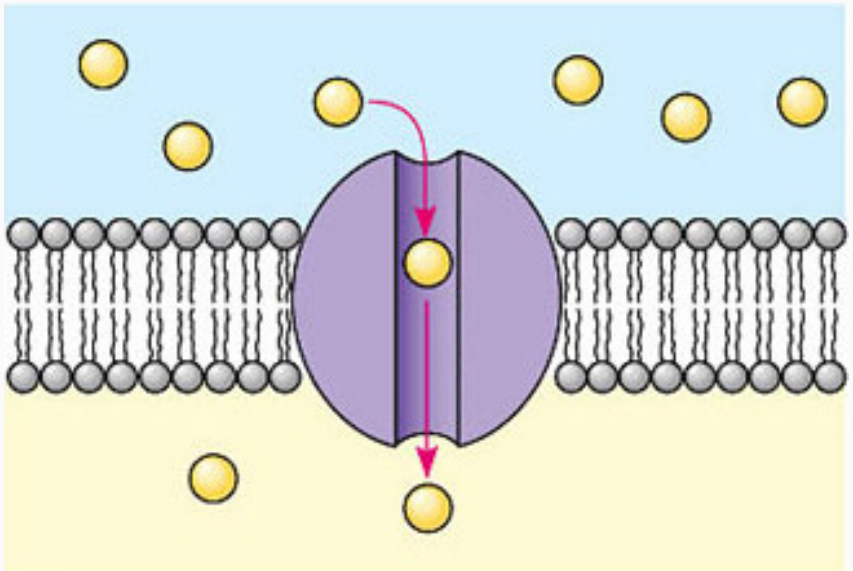
BCS abstraction

- mixture → order not important
- 84 different deviations



Compartments

- determine spatial position of a complex
- other agents indirectly inherit compartment
- particularly useful for modelling of mass transport



Prot1.Prot2::cell, Prot::out

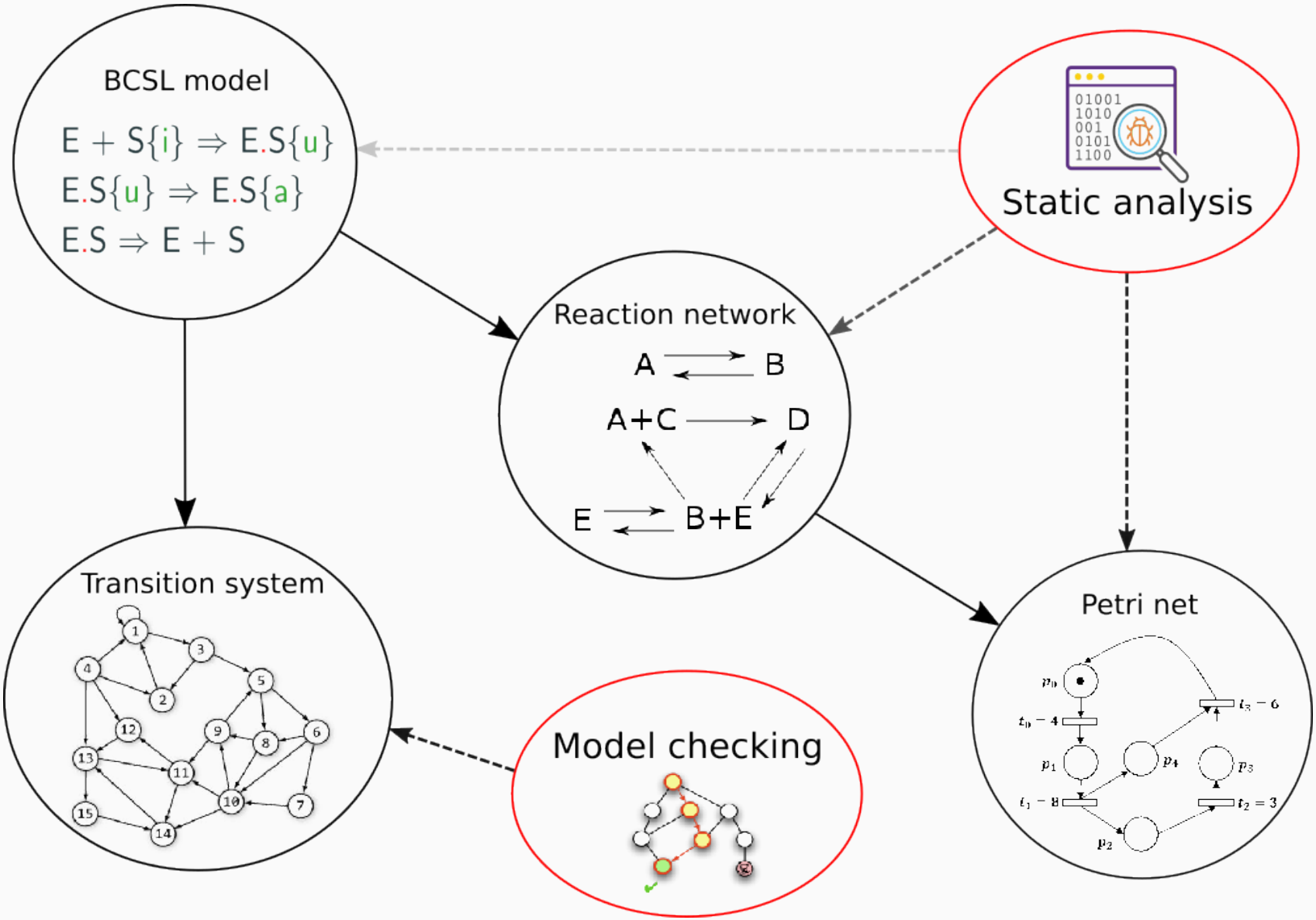
Relationships

- rule-based approach $\Rightarrow$ rules
- generalised version of reactions
- “don’t care, don’t write”

Examples of rules

- **State change**
  1. S{u}::cell ⇒ S{p}::cell
  2. E.S{u}::cell ⇒ E.S{p}::cell
  3. R(active{off})::cyt ⇒ R(active{on})::cyt
- **Complex formation**
  4. E::cell + S{u}::cell ⇒ E.S{u}::cell
  5. $\alpha.\alpha$::out + $\beta.\beta$::out ⇒ $\alpha.\alpha.\beta.\beta$::out
- **Transport**
  6. Prot::cell ⇒ Prot::out
  7. ⇒ mRNA::nuc

Executability and BCSL Models Analysis



Executability and BCSL Models Analysis

Semantics

- match & replace approach
- previously done through Kappa (SASB 2015)

Software support

- BCSgen – maintenance tool
- online eBCSgen – `http://pithya.ics.muni.cz/galaxy`

Static analysis

- rule redundancy elimination
- context-based reduction
- static non-reachability analysis

Compatibility

- degree of specificity
- allows to compare groups of agents

Examples

- $\mathrm{Ser}\{\mathrm{p}\} \lhd \mathrm{Ser}$
- $\mathrm{Prot}(\mathrm{Ser}\{\mathrm{p}\}, \mathrm{Cys}\{\mathrm{u}\}) \lhd \mathrm{Prot}(\mathrm{Cys}\{\mathrm{u}\})$
- $\mathrm{E.S}\{\mathrm{u}\} \lhd \mathrm{E.S}$

Rule redundancy elimination

- identifies potentially useless rules (in the qualitative context)
- these can be removed with no semantic loss

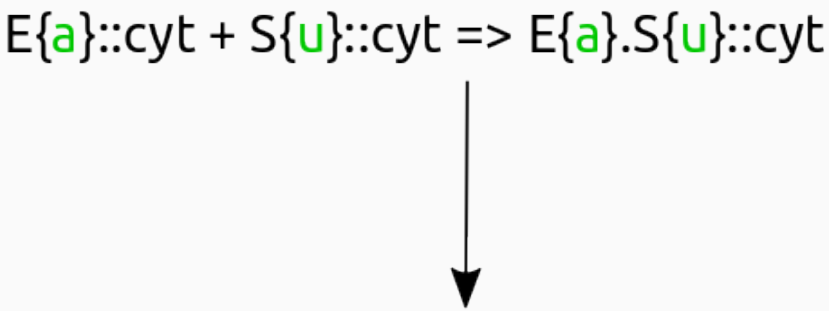
Example

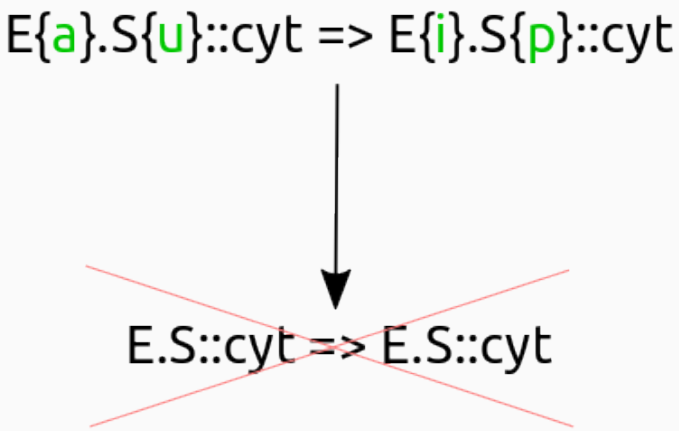
✓ K(S{u}).K::cell ⇒ K(S{p}).K::cell

✗ K(S{u}, T{i}).K::cell ⇒ K(S{p}, T{i}).K::cell

Context-based reduction

- remove all states from the rules
- analyse simplified network
- some rules are incorrect





Static non-reachability analysis

- at some point, we have to create a compatible agent in order to reach desired agent
- (if it is not already in the initial state)
- there has to exist a compatible agent on the right-hand side of a rule

Example

- desired agent: $\mathrm{E\{i\}.S\{p\}::cyt}$
- candidate rules:

  $\mathrm{E\{a\}.S\{u\}::cyt} \Rightarrow \mathrm{E\{i\}.S\{p\}::cyt}$

  $\mathrm{E.S\{u\}::cyt} \Rightarrow \mathrm{E.S\{p\}::cyt}$

  $\vdots$

  $\mathrm{E::cyt} + \mathrm{S::cyt} \Rightarrow \mathrm{E.S::cyt}$

Conclusions

Summary

- BCSL as a language utilising the specific view on the biochemical structures and reactions
- integrating with annotation information

Future work

- more focus on static analysis
- enrich the rules by quantitative aspects

16th International Conference on Computational Methods in Systems Biology

12th-14th September 2018, Faculty of Informatics, Masaryk University, Brno (CZ)

CMSB BRNO 2018

Andrew Turberfield
University of Oxford, UK
Modelling Biomimetic Structures and Machinery Using DNA

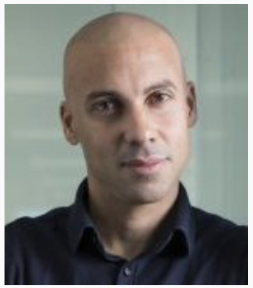
Mustafa Khammash
ETH Zurich, CH
Biomolecular Control Systems

Lecture Notes in Computer Science
LNCS LNAI LNBI

BIOinformatics
ACM Special Interest Group

Ilka Maria Axmann
Heinrich Heine University Düsseldorf, DE
What time is it?

Chris J. Myers
University of Utah, US
A Standard-Enabled Workflow for Synthetic Biology



Andrew Phillips
Microsoft Research, UK
Programming Languages for Molecular and Genetic Devices

BCS Allows Further Abstractions

Entity ID	KaiC
Entity name	KaiC protein
Composition	S, T
Type	structure
...	

Entity ID	KaiC6
Entity name	KaiC complex
Composition	KaiC.KaiC.KaiC.KaiC.KaiC.KaiC
Type	complex
...	

KaiC(S{u}).KaiC.KaiC.KaiC.KaiC.KaiC::cyt $\Rightarrow$
KaiC(S{p}).KaiC.KaiC.KaiC.KaiC.KaiC::cyt

S{u}::KaiC::KaiC6::cyt $\Rightarrow$ S{p}::KaiC::KaiC6::cyt

Drawbacks of the transition to Kappa

- only one type of agent (no hierarchy)
- additional information encoded in different agent types are lost during the conversion
- atomic agents had to be considered as binding sites and structure agents as Kappa agents
  – not always suitable, especially when atomic agents had an independent role in a rule

Drawbacks of the transition to Kappa

- treatment of complexes in Kappa requires explicit bonds between individual agents
  – need to choose one of the many possible isomorphisms (e.g. circular layout, linear layout)
  – not possible to express the fact that the order does not matter
- execution of the models was delayed by the conversion to Kappa and then calling Kappa core to execute its semantic