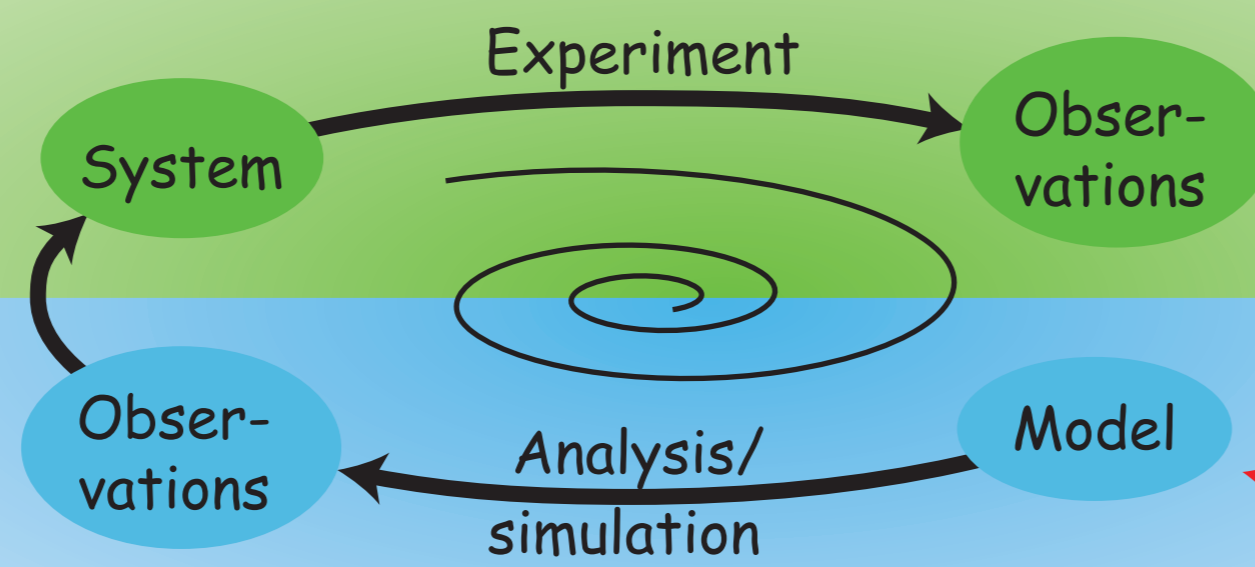


## The setting - systems biology

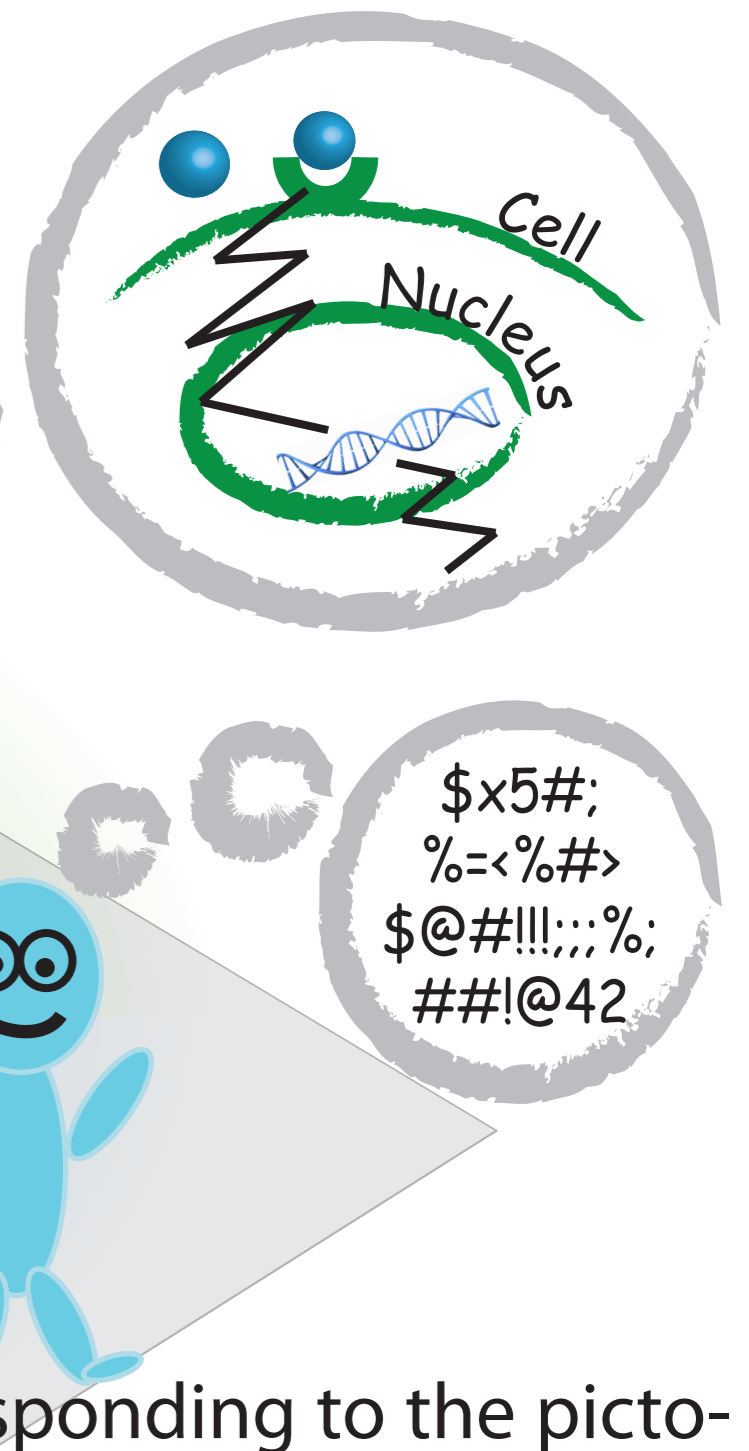
**1) System:** Could e.g. propagate external signals into the nucleus of a cell and activate an appropriate genetic switch ("move", "mate", "die") -- a so called *signal transduction pathway*.

**2) Observations:** E.g. connections between chemical species in a pathway or rates of reactions. These are gathered into informal pictorial diagrams by a biologist.

**4) Observations:** Do they correspond with experimental observations? If no, repeat circle and tweak model. If yes, use model for e.g. medical trials ("does drug X affect part Y of the system?").



**3) Model:** A computer scientist creates a formal model corresponding to the pictorial diagram using low-level formalisms such as the Pi calculus, Petri nets etc.  
**Problems: P1** Constructing big models is difficult due to the gap in abstraction level between biology and the low-level formalisms. **P2** Pictorial diagrams are ambiguous.



## CCS - A Calculus of Chemical Systems

**Aim:** To design a *high-level* language with tools for modeling complex biological systems (cf. problem P1), which is intuitive for biologists and has appropriate graphical *formal semantics* (cf. problem P2).

**Key features of language:** The basic entities are *species*<sup>DC1</sup> with state (e.g. an active/inactive gene). Simple rules are used for defining *reactions*<sup>DC1</sup> which consume some species and produce others.

Multiple reactions take place in parallel, inside a hierarchy of *compartments*<sup>DC1</sup>. General *modules*<sup>DC2</sup> can be defined and used repeatedly with different parameter values.

### Design criteria:

- DC1 Biologically faithful primitives
- DC2 Abstraction and reuse
- DC3 Flexibility
- DC4 Analysis and simulation
- DC5 Visual representation

**Key features (cont.):** Species states are boolean per default, but their data types, and functions on these, can be defined on a *per-application basis*<sup>DC3</sup>. Hence information such as DNA sequences

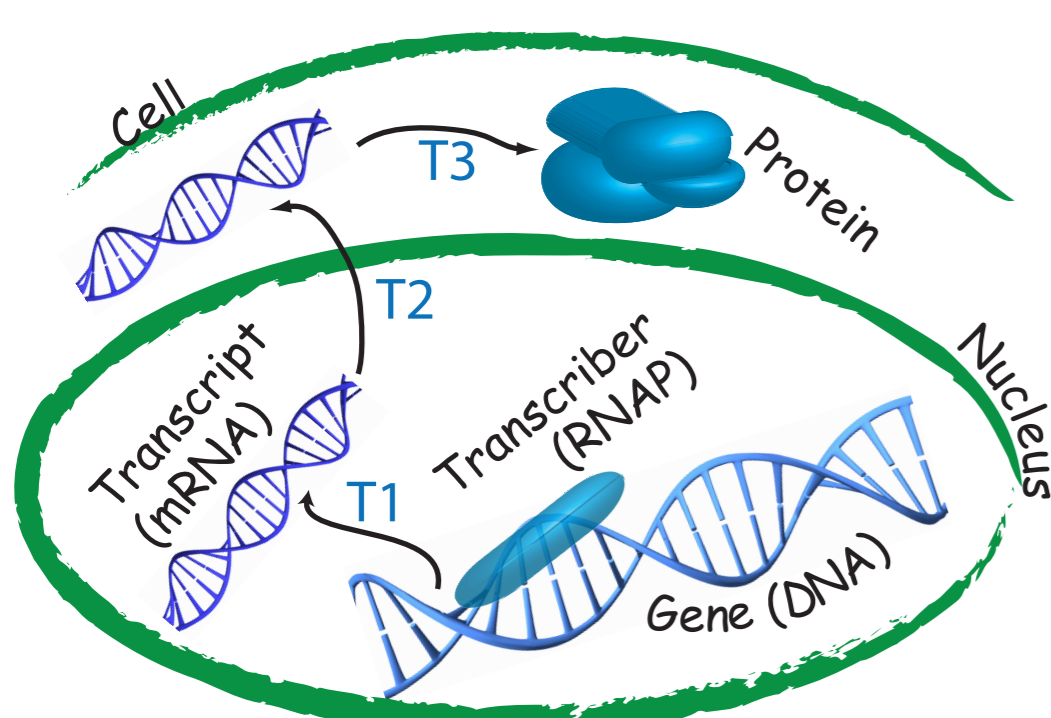
(strings), or position in a compartment (integer pairs), can be represented.

Semantics is given in terms of *coloured Petri nets*, a well established mathematical formalism for which *analysis and simulation*<sup>DC4</sup> methods are known. Petri nets can be intuitively *visualised*<sup>DC5</sup> as a graph of

places (species) and transitions (reactions). However, further work is necessary to find the ideal definition of Petri nets for our purposes.

## Example - the central dogma of molecular biology

**The biology:** Production of proteins from information encoded in genes in three steps: **T1** Transcription in nucleus. **T2** Transport out of nucleus. **T3** Translation to protein.



### CCS model:

```

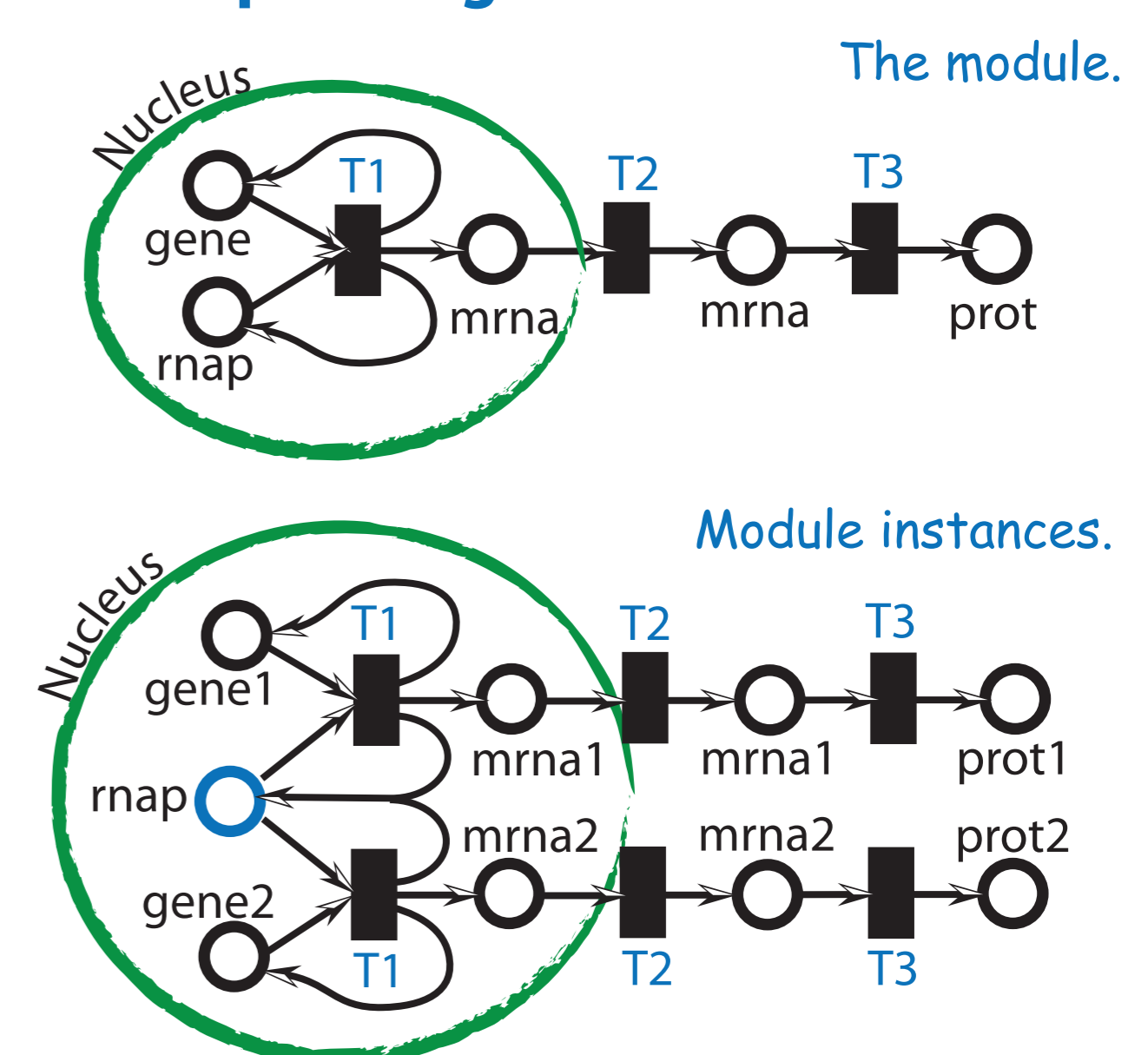
module dogma(comp nucl, spec gene,
spec mrna, spec prot) {
  nucl[gene + rnap ->T1 gene + rnap + mrna] |
  nucl[mrna] ->T2 mrna | mrna ->T3 prot
}

comp cell 1E-15 liters;
comp nuc 1E-16 liters inside cell;
spec gene1, gene2, mrna1, mrna2;
spec prot1, prot2;

cell[
  dogma(nuc, gene1, mrna1, prot1) |
  dogma(nuc, gene2, mrna2, prot2)
]

```

### Corresponding Petri nets:



## Conclusion, status and future work

**Conclusion:** CCS is a high-level language for systems biology which is intuitive for biologists and has formal semantics in terms of Petri nets.

**Status:** We currently have a translator implementation in F# from a subset of CCS to a biology XML language (SBML). This allows the use of third-party tools for experimental simulation of models.

**Future work:** 1) An IDE supporting the development of models in CCS, simulation/analysis of models, and on-the-fly visualisation. 2) Theoretical studies of the Petri net formalism underlying CCS. 3) Application to a nontrivial biological system such as Interferon. 4) A suitable logic for specifying biological properties.