TU/e technische universiteit eindhoven

Modeling the Case Handling principles with Colored Petri Nets

Christian W. Günther

/faculteit technologie management

Outline

- **Case Handling** primer
- *Model:* **Task lifecycle**
- *Model:* **User interaction**
- *Model:* **Data lifecycle**
- **Analysis**
- **Application**
- **Conclusion**

Case Handling Basics

- Technique to implement **PAIS**
- ***Data-driven*** approach
  - No strict separation of control flow and data
- Aims at resolving *fundamental problems* of WFM
  - No ***context tunneling***
  - Far more ***flexible*** control flow
  - Introduces ***skip and redo roles***

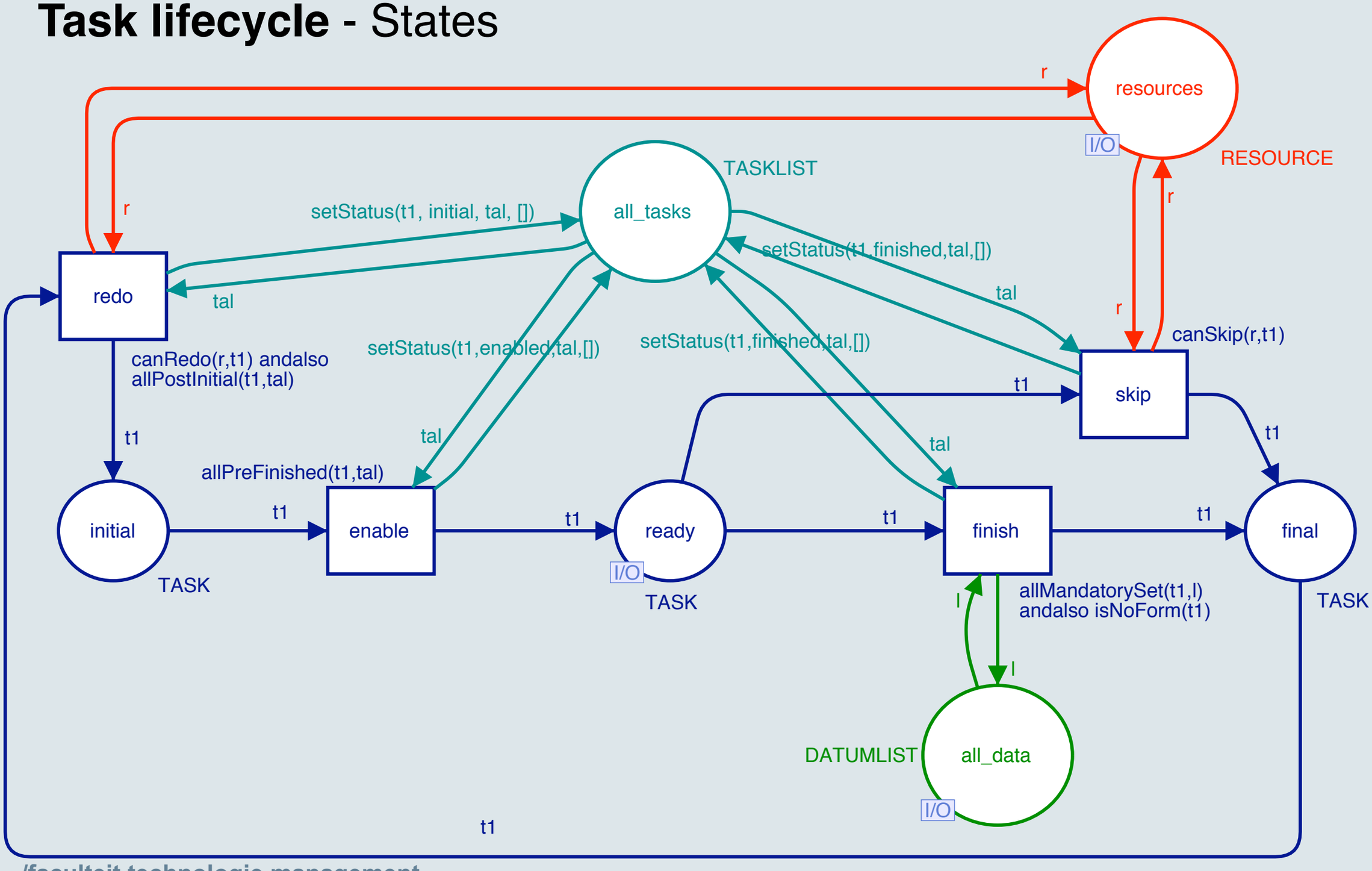
CPN Model

- Tool for research
  - **Understand** and **analyze** Case Handling principles
  - **Formalization** of Case Handling
  - Create ‘clean’ **logs** for Process Mining experiments
- Designed by one person
  - Background in SW Engineering / classic PT nets
  - Design time: 4 (man-) weeks
- Most time spent on:
  - Getting acquainted with CPN Tools and ML
  - Multiple iterations to improve the model

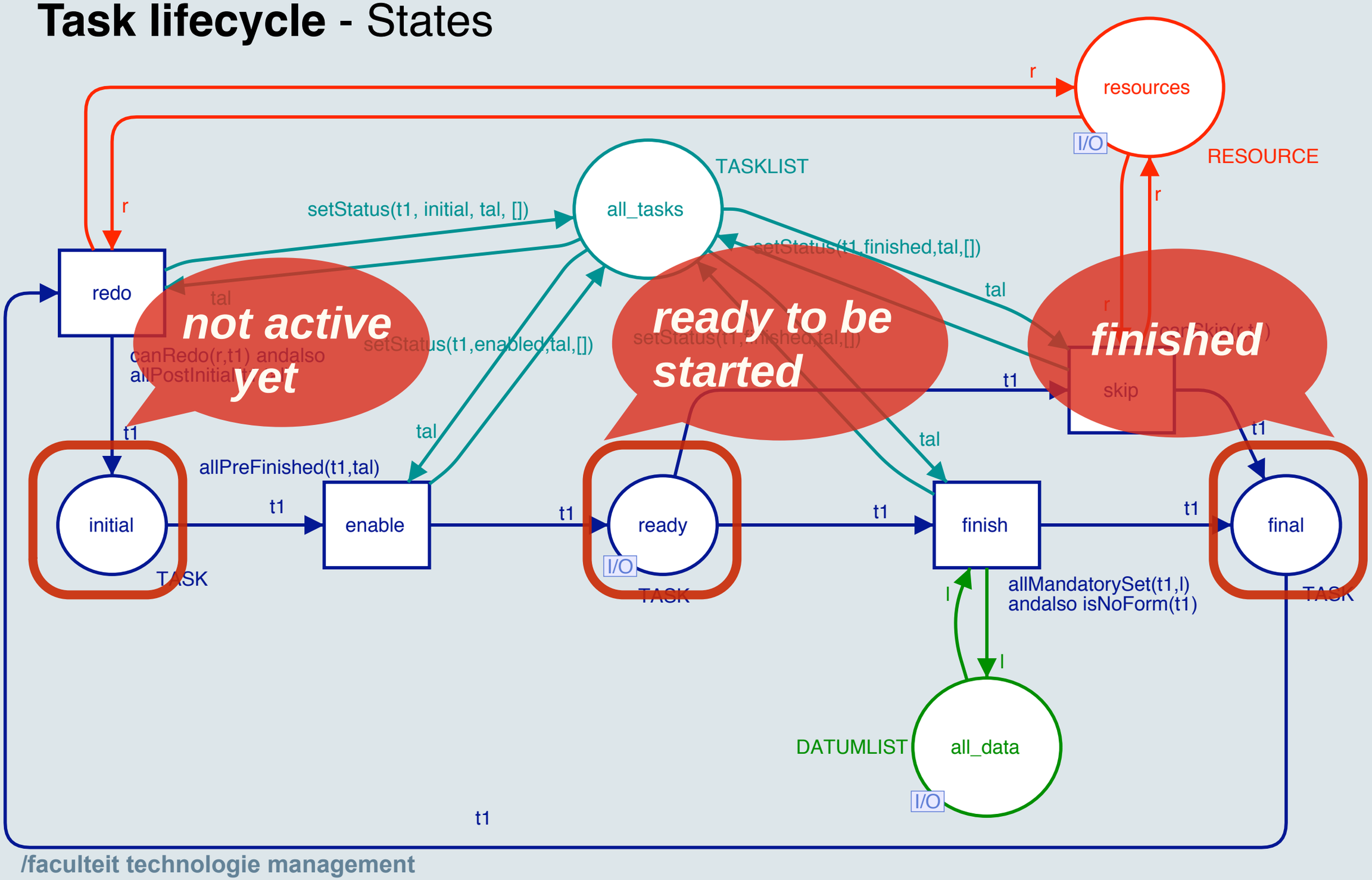
CPN Model

- ***Meta-model***
  - Models aspects of Case Handling systems
  - Process is ***encoded*** in the net's ***marking***
    - TASK tokens contain ***preceding*** and ***following*** tasks
    - Process not reflected in layout
- Divided into ***three parts***
  - ***Task lifecycle*** (mostly system activity)
  - ***User interaction part*** (top-level view)
  - ***Data manipulation*** (refines user interaction)

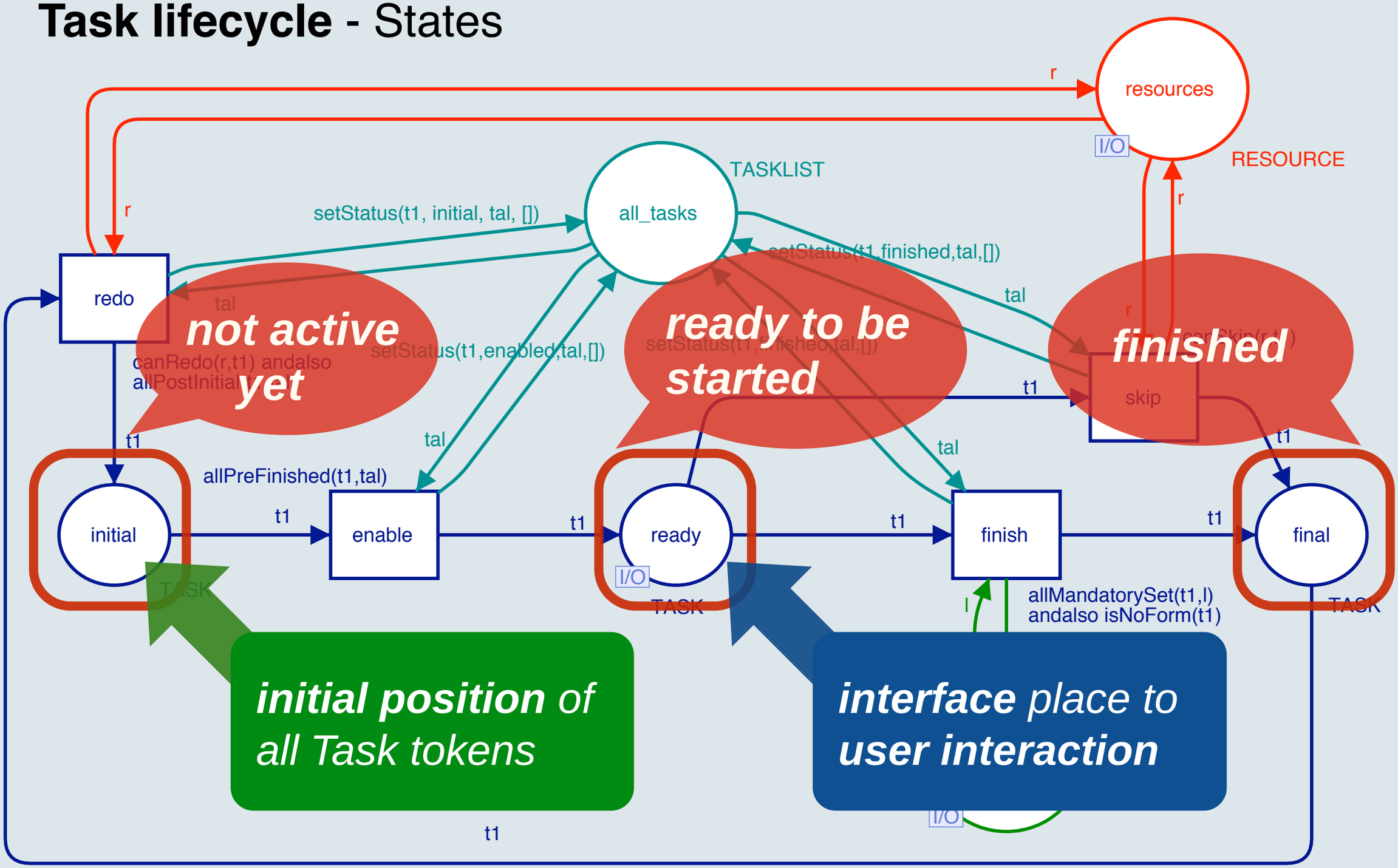
Task lifecycle - States



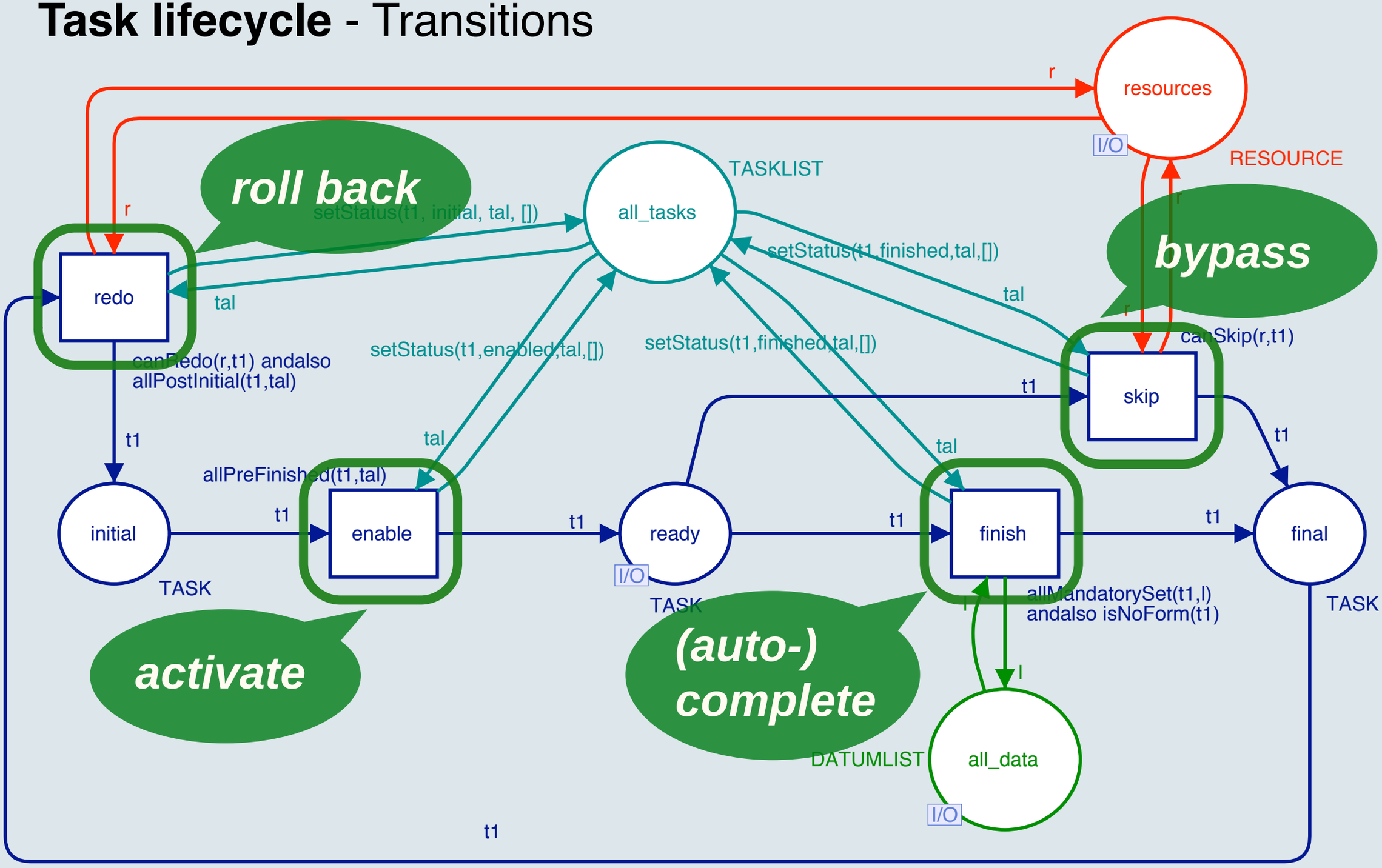
Task lifecycle - States



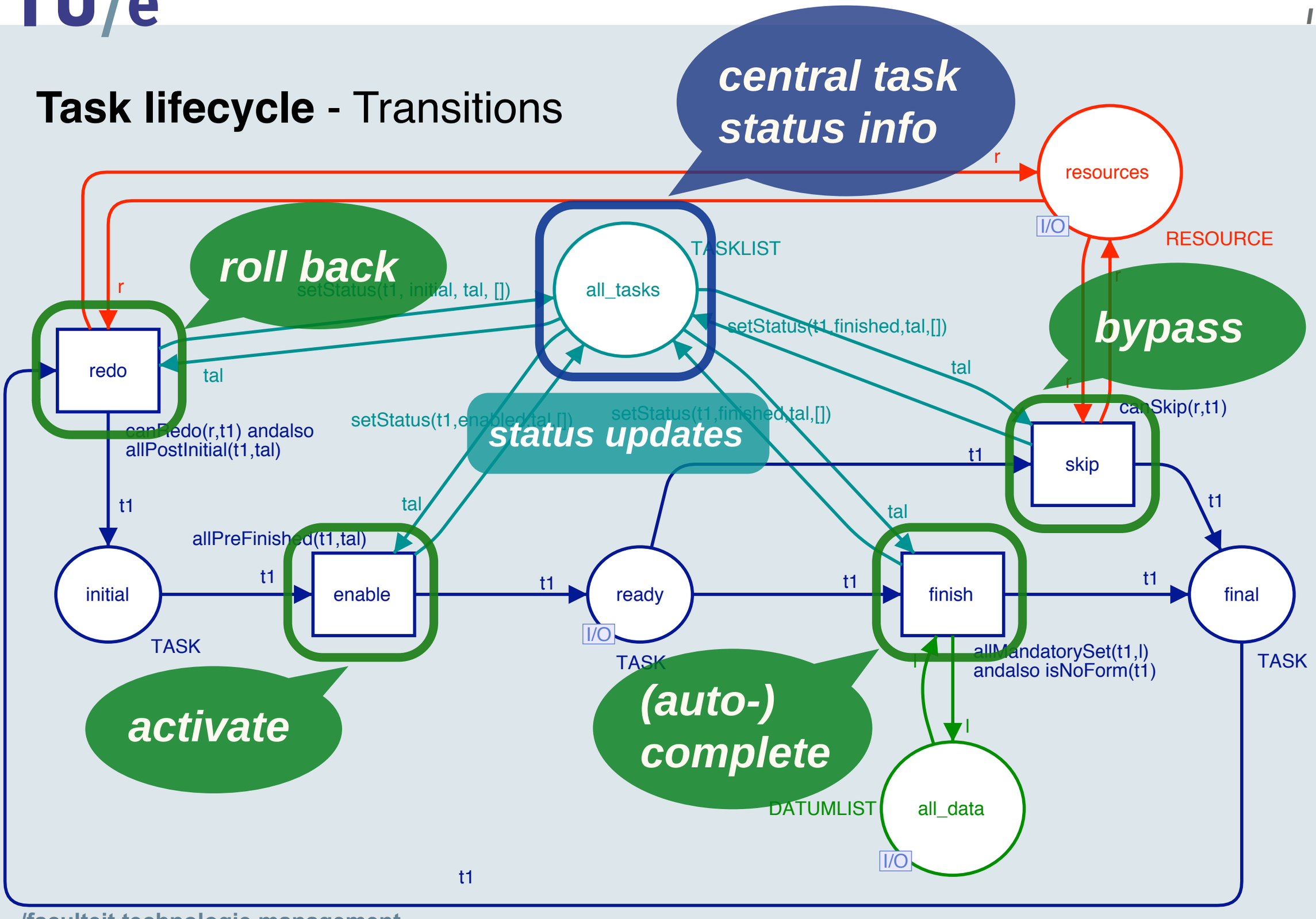
Task lifecycle - States



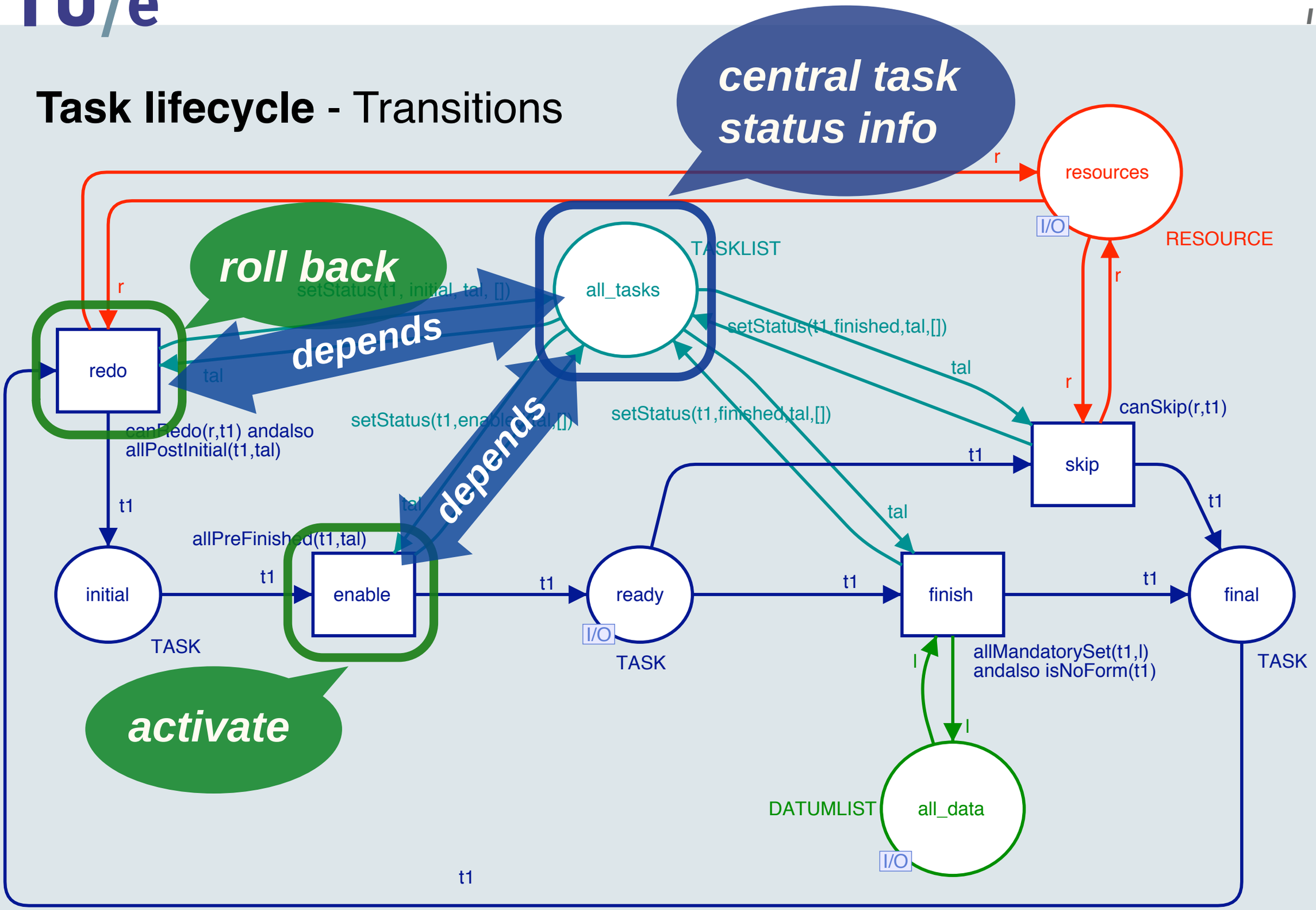
Task lifecycle - Transitions



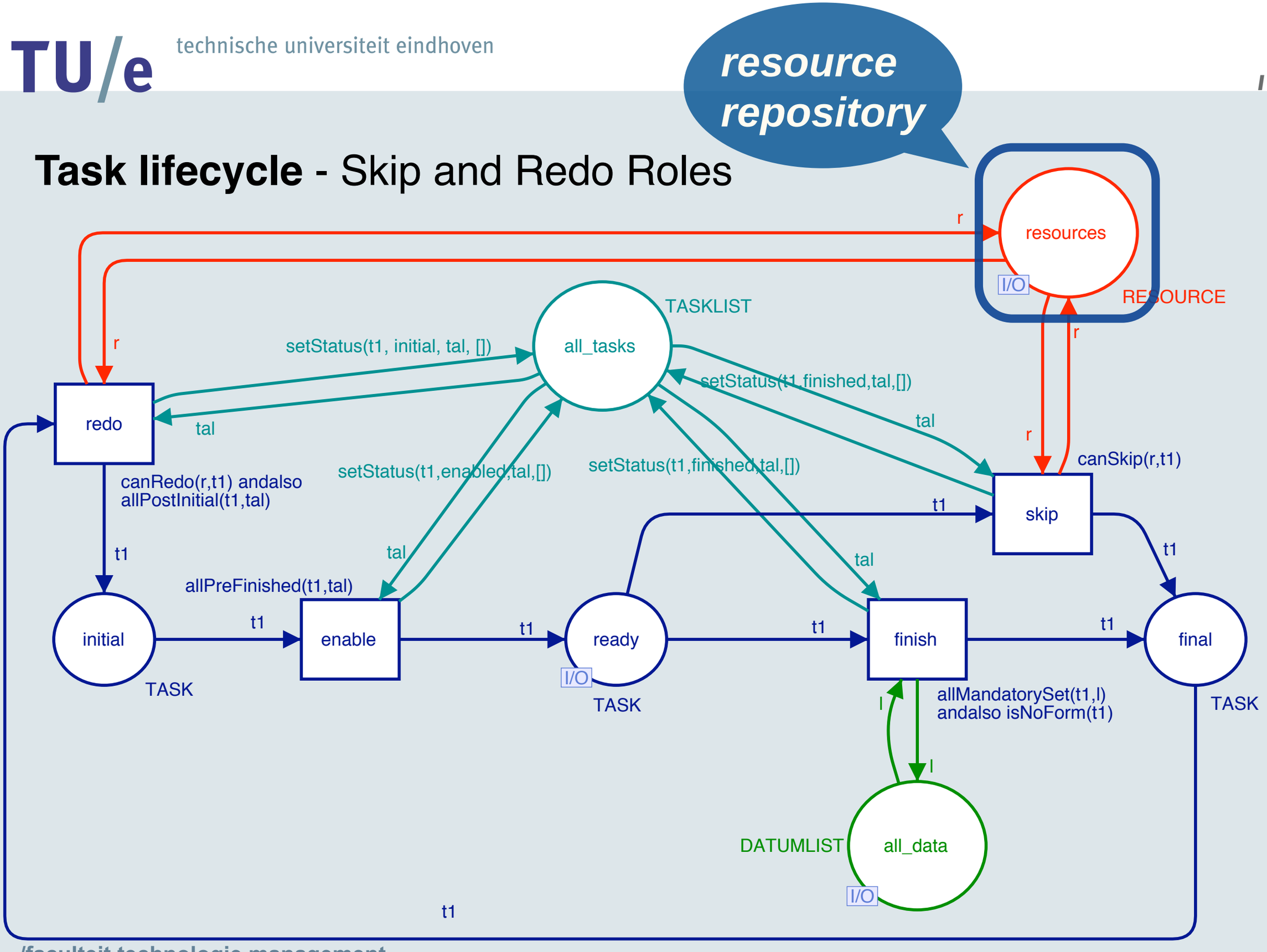
Task lifecycle - Transitions



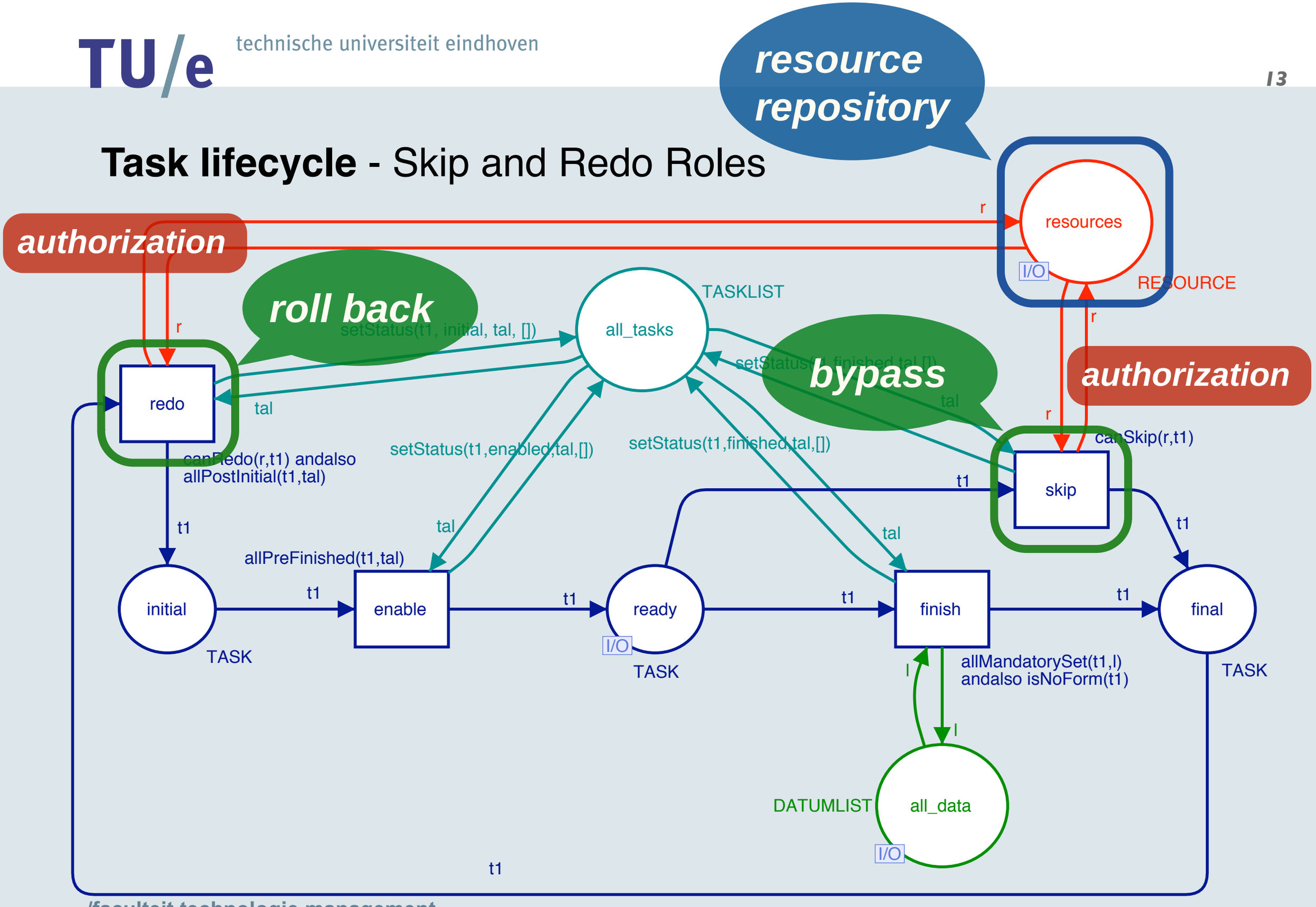
Task lifecycle - Transitions



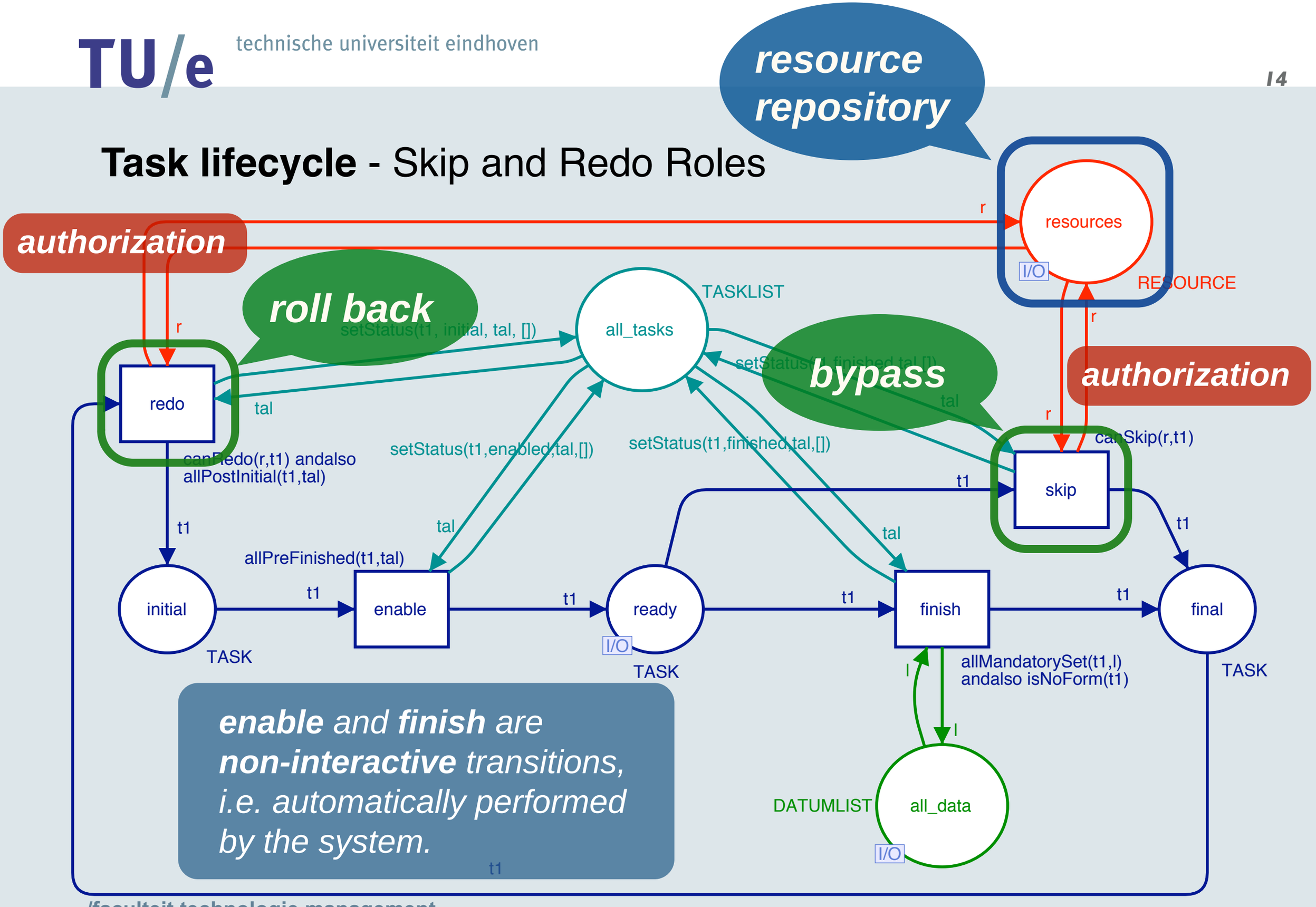
Task lifecycle - Skip and Redo Roles



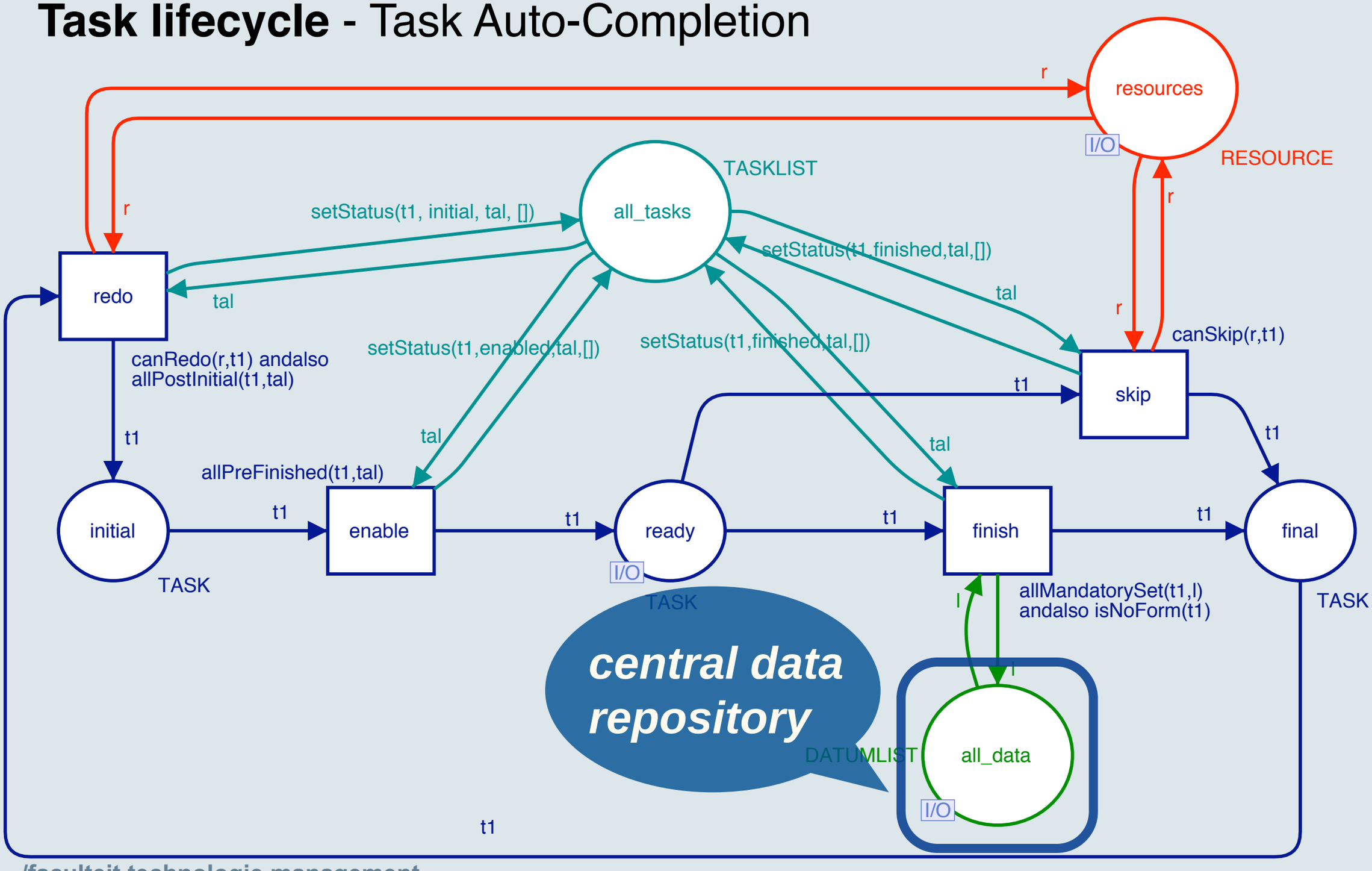
Task lifecycle - Skip and Redo Roles



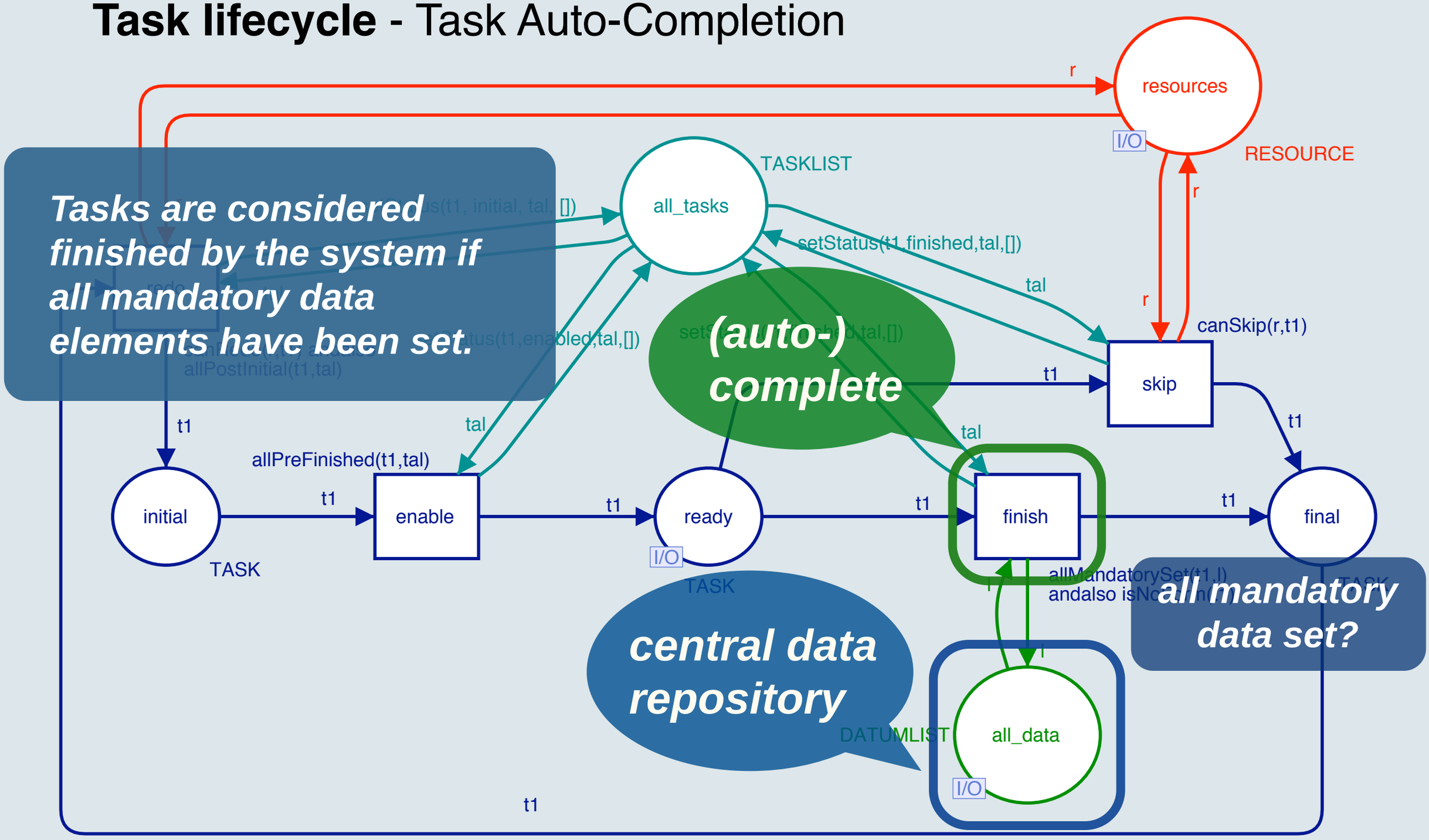
Task lifecycle - Skip and Redo Roles



Task lifecycle - Task Auto-Completion



Task lifecycle - Task Auto-Completion



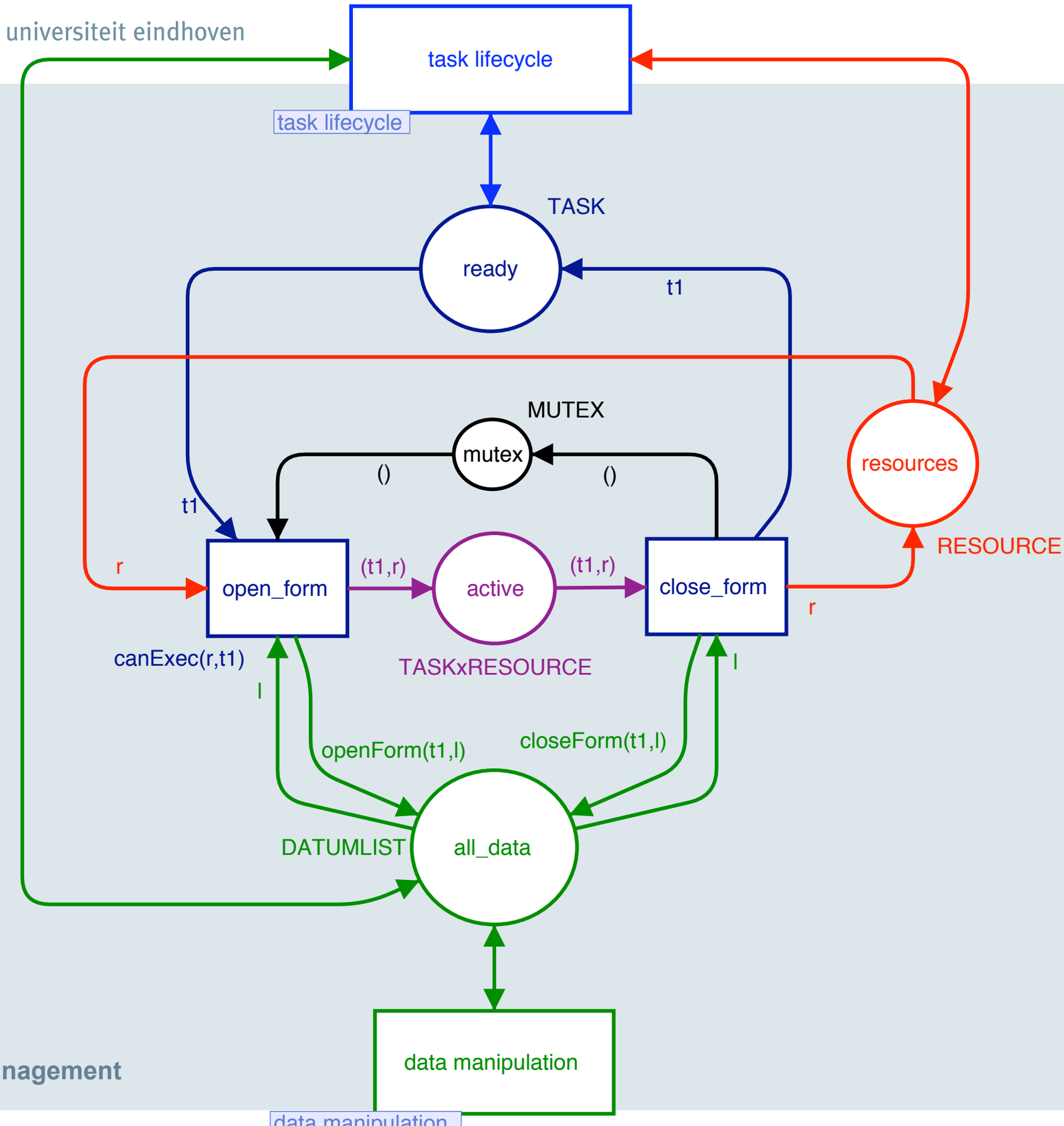
Tasks are considered finished by the system if all mandatory data elements have been set.

(auto-) complete

central data repository

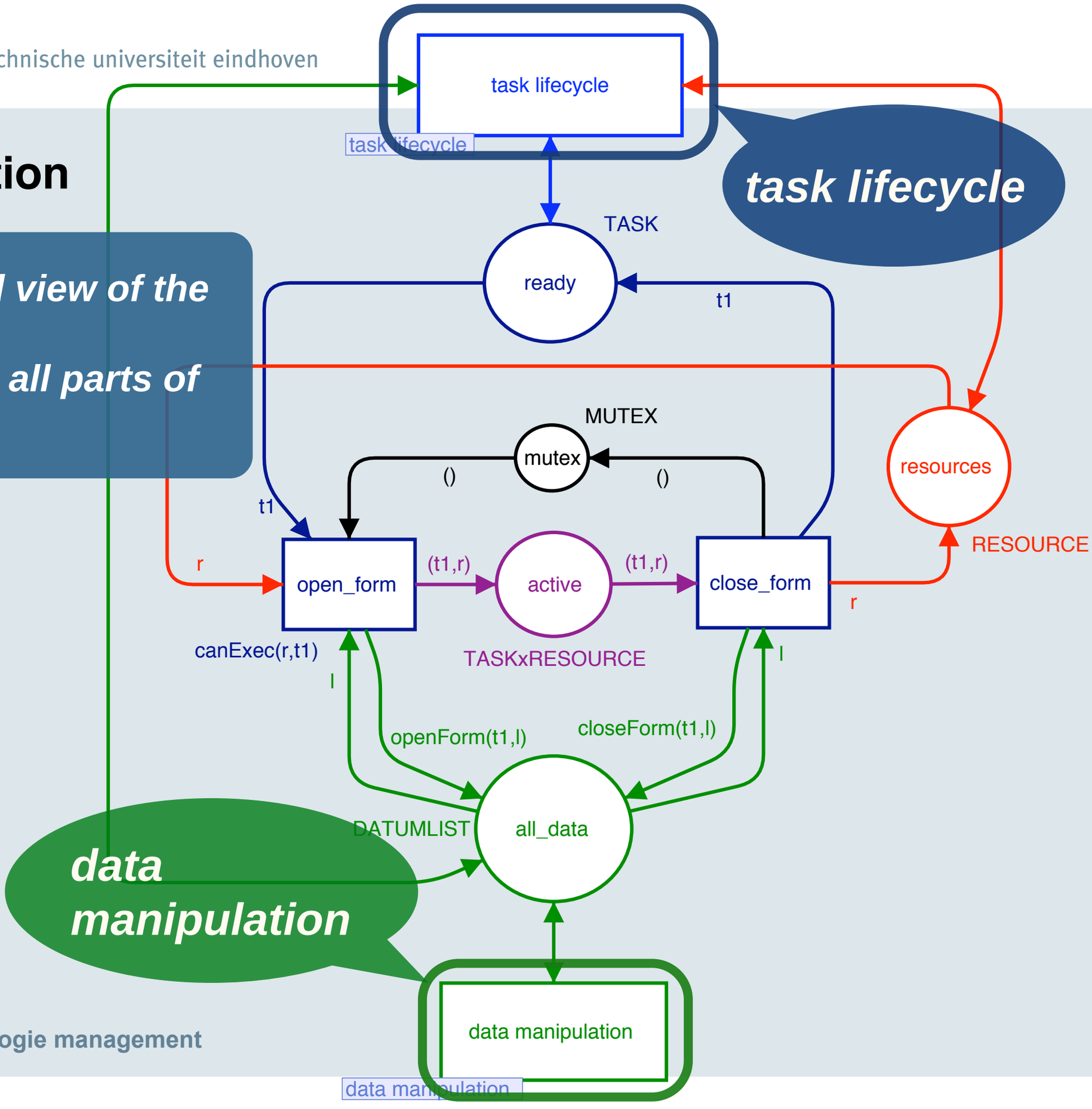
all mandatory data set?

User interaction



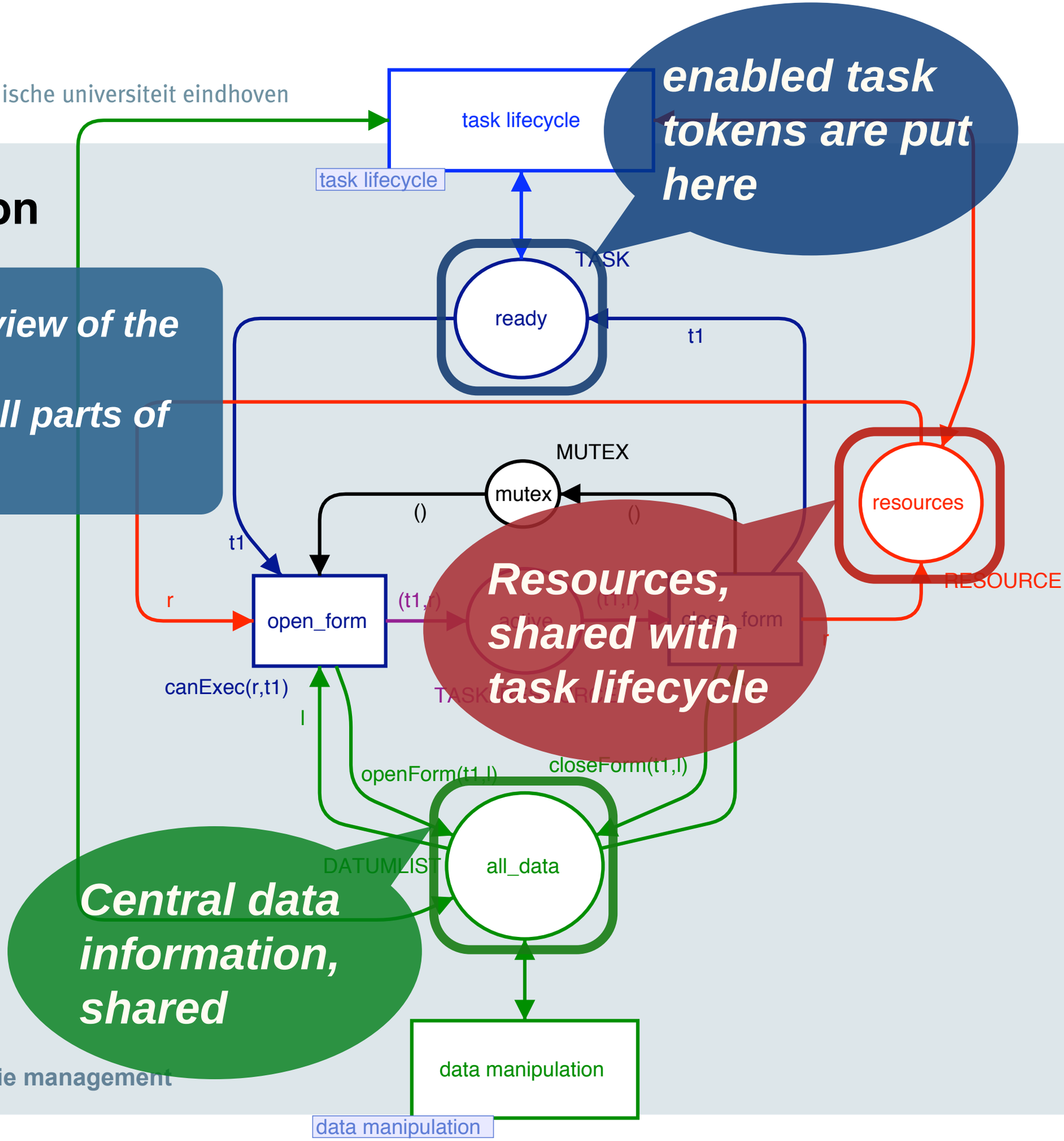
User interaction

User-centered view of the system.
Interconnects all parts of the model.



User interaction

User-centered view of the system.
Interconnects all parts of the model.



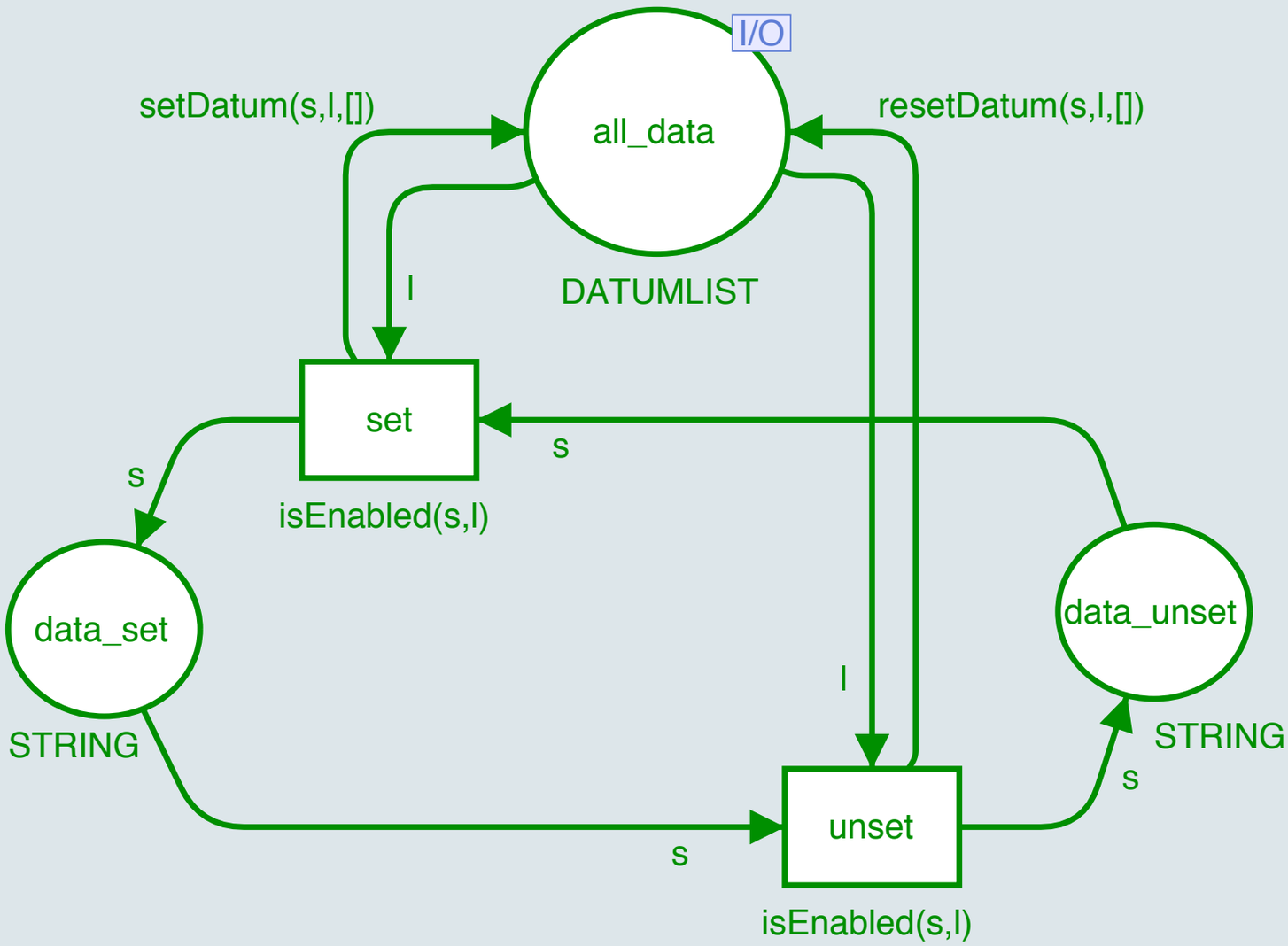
User interaction

- User can ***open enabled tasks to edit form***.
- Needs ***resource*** with ***execute*** rights.
- Enables ***data elements*** on form for ***manipulation***.

- ***Mutex*** place
- Only ***one task*** can be worked on ***at a time***.
- Corresponds to real system behavior.

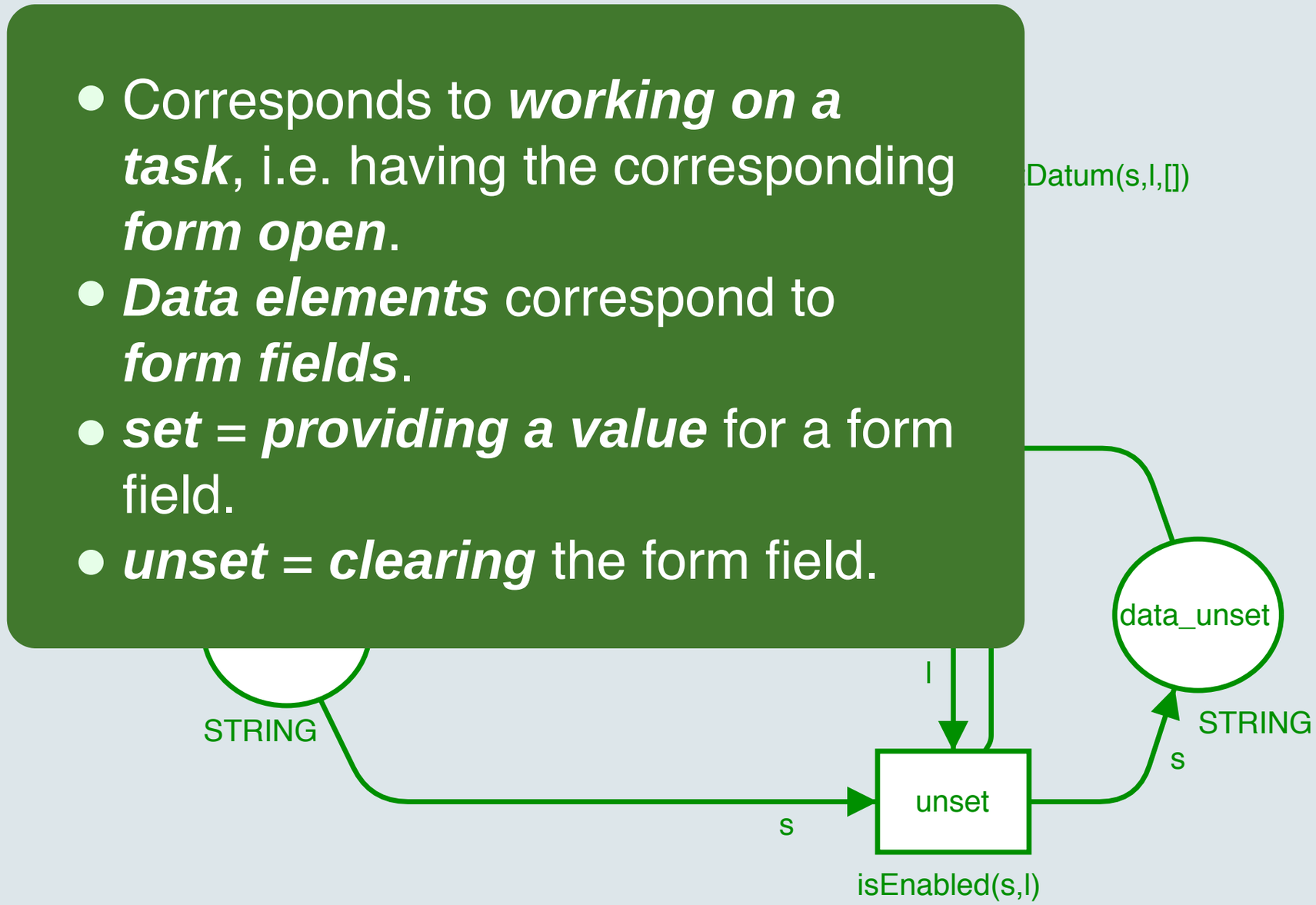
- Open form can be ***closed at any time***.
- ***Frees*** used ***resource***.
- ***Disables data elements***.
- ***Returns task token*** to place '***ready***'.

Data Manipulation

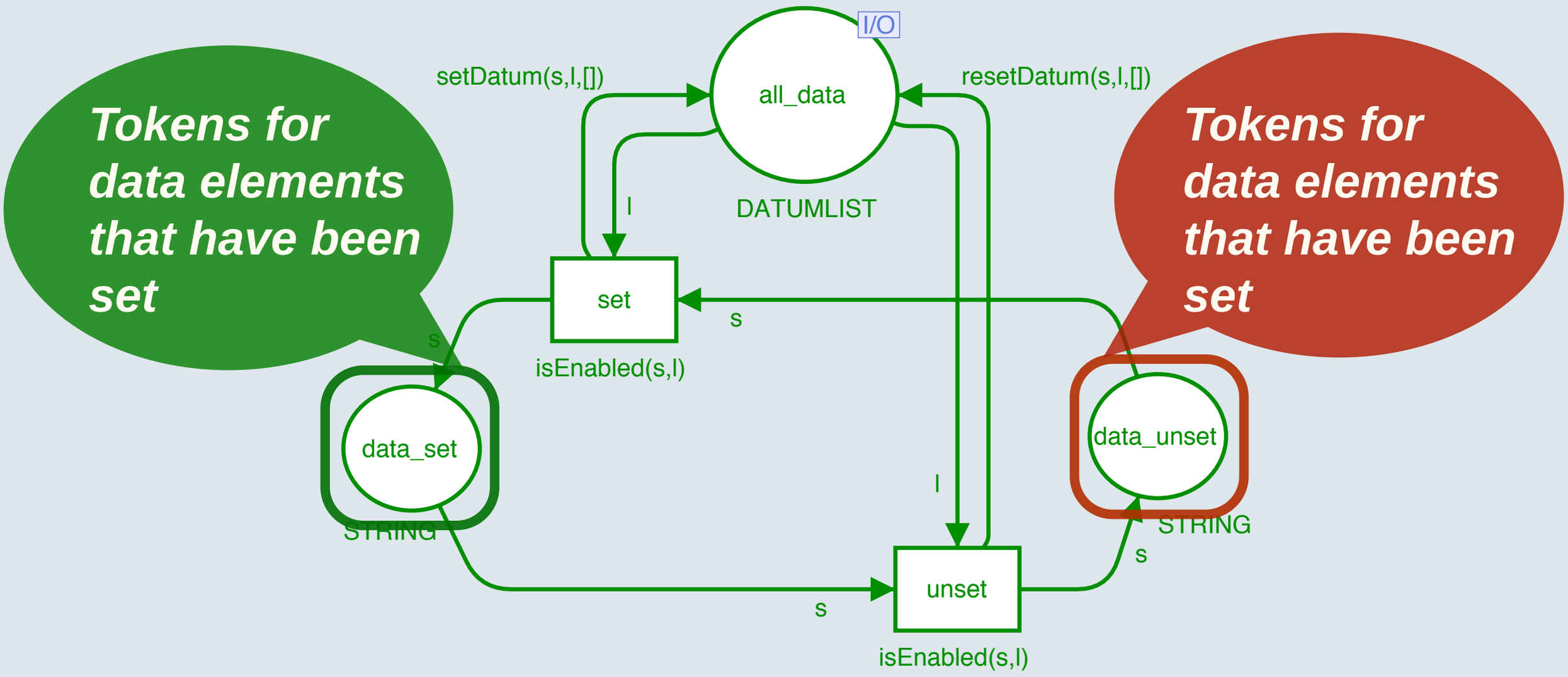


Data Manipulation

- Corresponds to ***working on a task***, i.e. having the corresponding ***form open***.
- ***Data elements*** correspond to ***form fields***.
- ***set = providing a value*** for a form field.
- ***unset = clearing*** the form field.



Data Manipulation

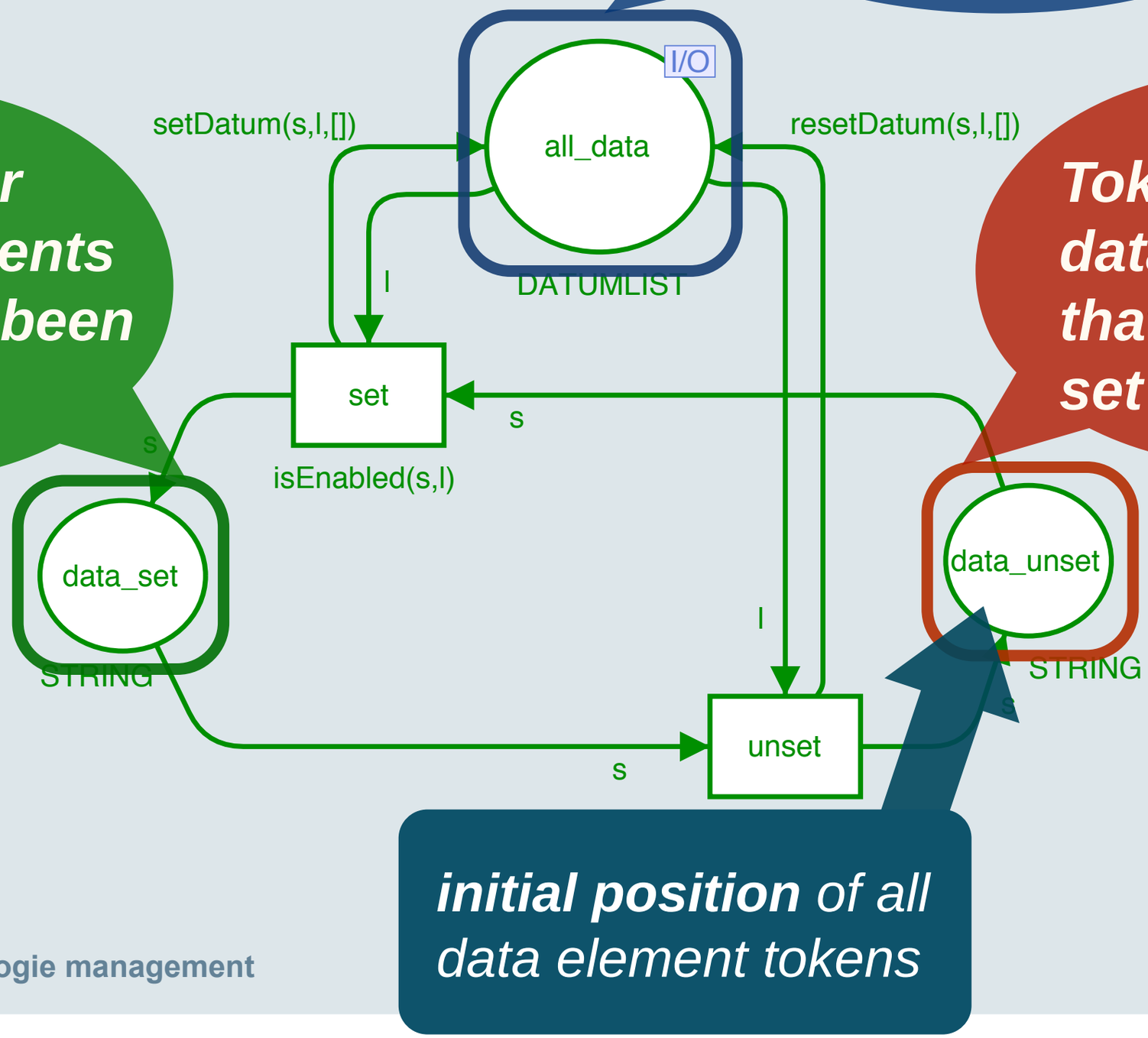


Data Manipulation

Central repository, keeping states of all data elements in the process

Tokens for data elements that have been set

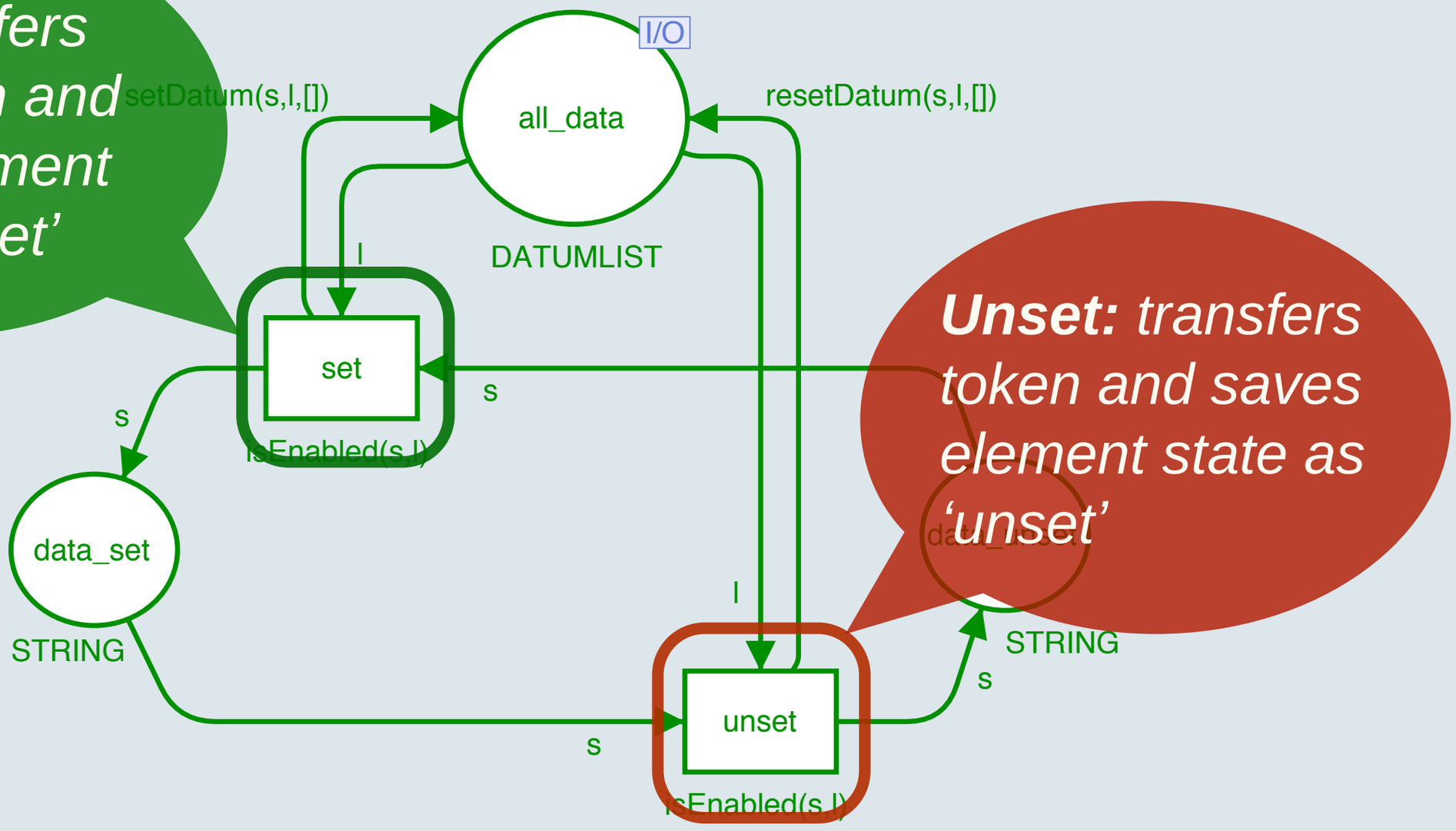
Tokens for data elements that have been set



***initial position** of all data element tokens*

Data Manipulation

Set: *transfers data token and saves element state as 'set'*



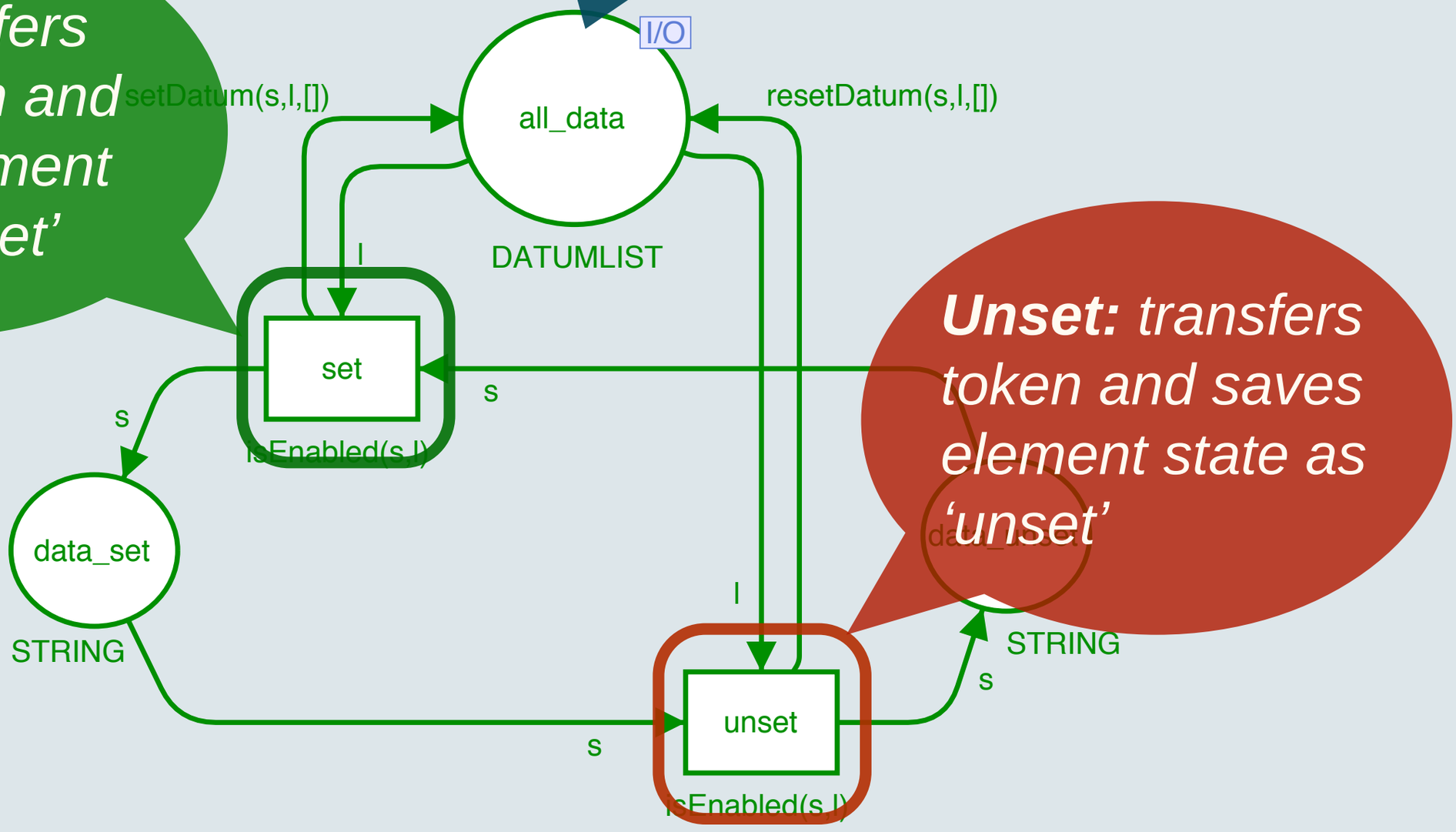
Unset: *transfers token and saves element state as 'unset'*

Data Manipulation

set and ***unset*** can only be fired when the corresponding data element is ***enabled***.

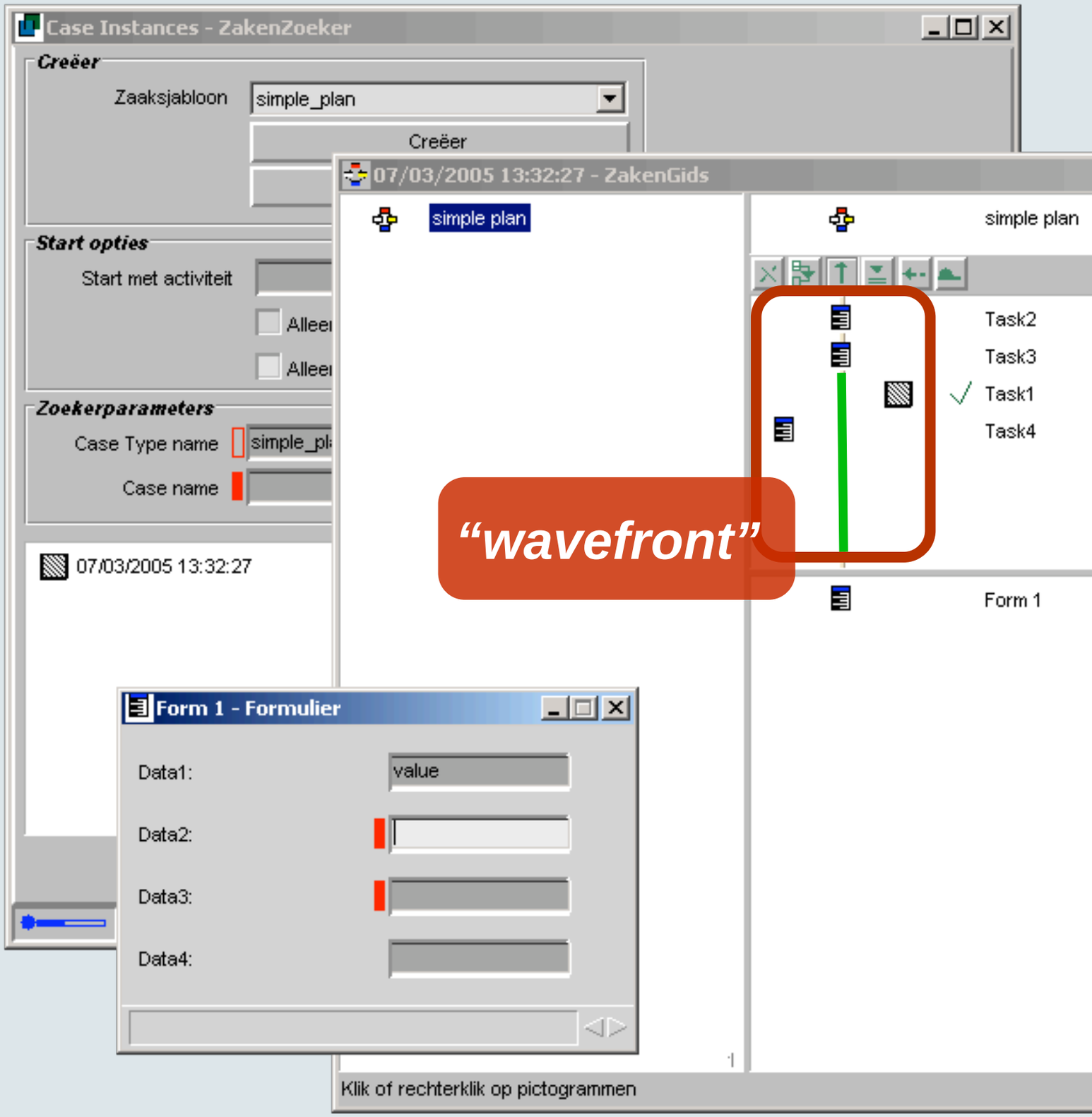
Set: *transfers data token and saves element state as 'set'*

Unset: *transfers token and saves element state as 'unset'*



Real system: FLOW*er*

- ***Wavefront*** represents the state in which tasks are in
  - **left = initial**
  - **center = ready**
  - **right = finished**
- ***Transitions*** of tasks can be triggered here:
  skip, redo, execute
- ***Forms*** show *mandatory data elements* to be set



Analysis

- All places are **bounded**
- **Unlimited navigation** within process
  - All markings are **Home Markings**
  - All transition instances are **live**

- **Discrepancies** Model ↔ Real system (FLOW*er*)
  - No alternative branches / arc conditions
  - Some high-level routing constructs not possible
  - Tight coupling **Task** ↔ **Form**
    - Delayed state transition (auto-completion of tasks)

Application - Understanding Case Handling

- Model ***behaves like real system*** in most aspects
  - Easier to access
  - Control pace by step-wise simulation
  - "***Hidden***" system activities are modeled ***explicitly*** (e.g., auto-completion of tasks)
    - Users get the *"complete picture"*
- Investigation of ***particular properties***
  - Possibility to *tweak system behavior*
  - Research *alternative behavior*

Application - Process Mining research

- Real Case Handling systems' behavior is ***hardly predictable***
- Model can be used to create ***"clean" test logs***
  - Tweak process and model to behave in a certain way
  - Create random amount of logs by simulation
  - Can be used to verify mining algorithms
- Model can be modified to ***isolate certain aspects***
  - e.g., disable skip functionality and see how the mining algorithm copes

Conclusion

- ***CPN*** has proven to be very suitable for modeling
  - Relatively ***easy-to-read*** models
  - ***Formal semantics***
- CPNs are highly leveraged by ***excellent tool support***
  - (Relatively) straightforward to create model
  - ***Simulation*** capabilities
    - Testing the model
    - Create test logs
  - ***State space analysis***
    - ***Objective confidence*** in model (...*where simulation can't go*)