02291 System Integration

Goal-oriented Requirements Engineering

© Giovanni Meroni

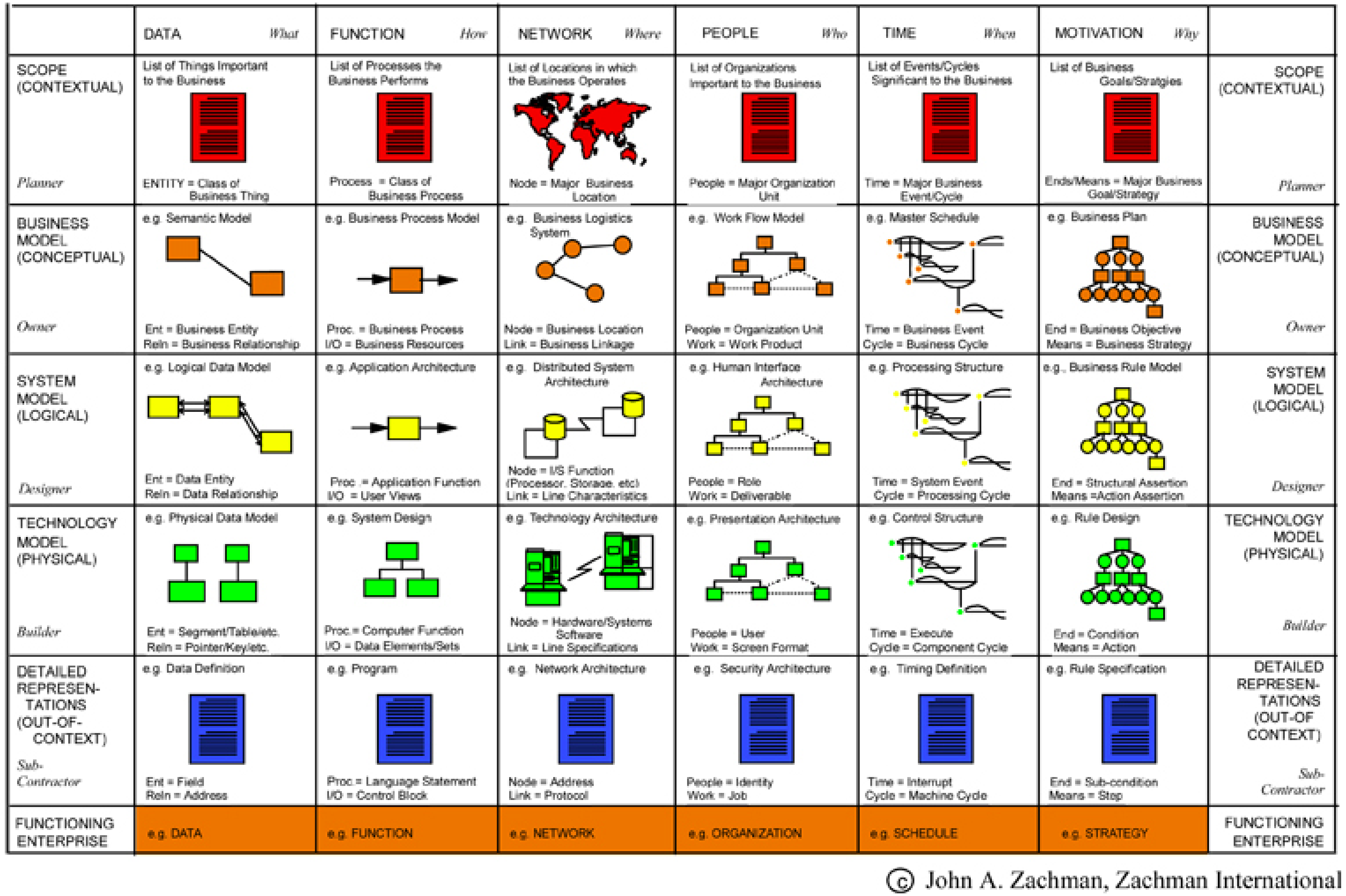
Motivations

- Need to define the context in which an application works
  - Time
  - Place
  - Information
  - Operations
  - Stakeholders
- Need to define the motivations why an application works in a specific way
  - The same outcome could be reached in multiple ways
- Need to consider these aspects together, and not in isolation
- Goal-oriented requirements engineering help us do so

Goal-oriented Requirements Engineering

- Use of goals as a starting point to identify, elicit, document, specify and analyze requirements
- Goals are objectives that the system aims to achieve
- Goals are meant to specify why a requirement is needed, which tasks are required to reach it, and which ones may hamper its achievement

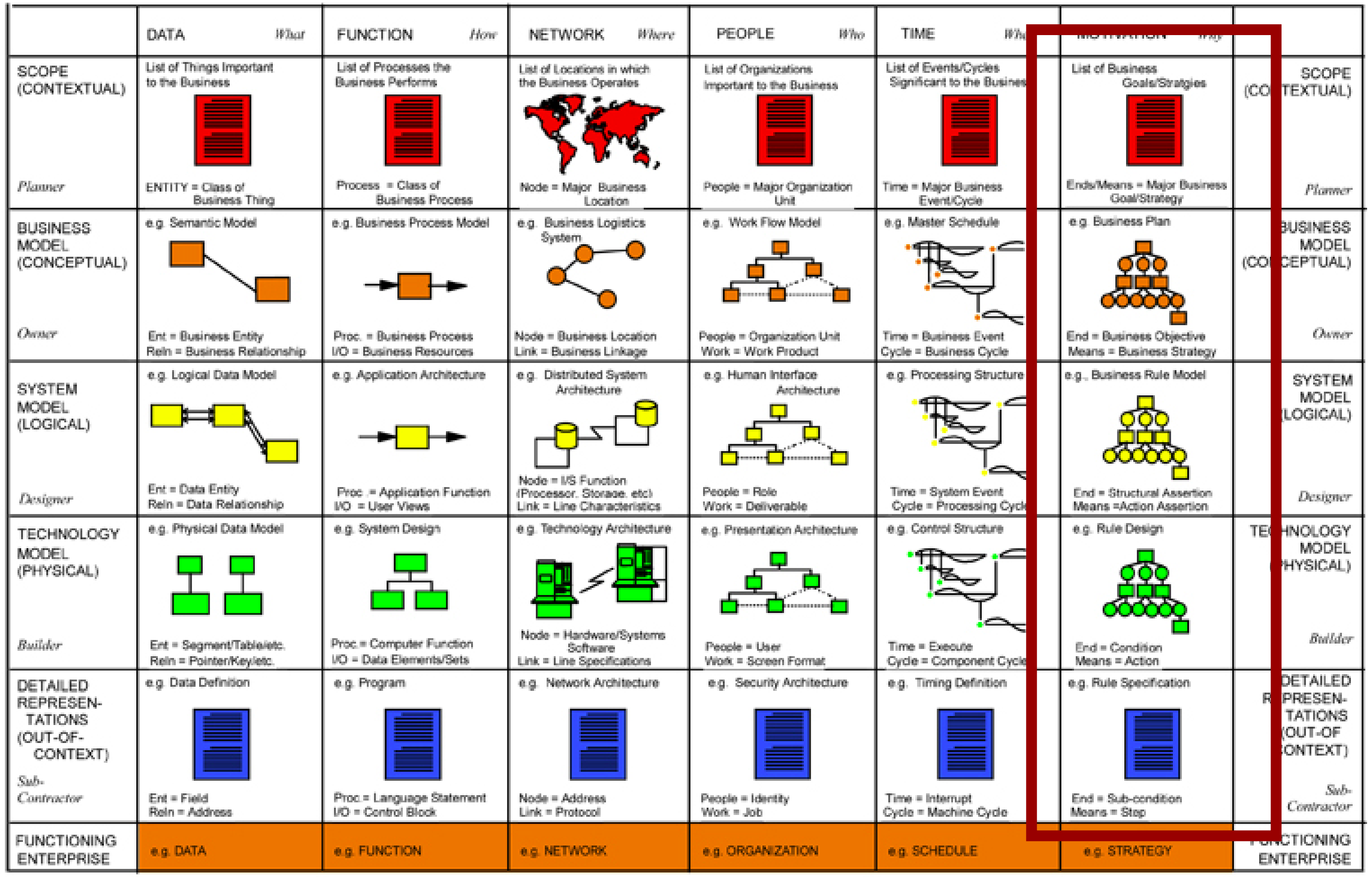
Zachman Framework



	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organization Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	List of Business Goals/Stratgies Ends/Means = Major Business Goal/Strategy	SCOPE (CONTEXTUAL) *Planner*
BUSINESS MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	BUSINESS MODEL (CONCEPTUAL) *Owner*
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. Application Architecture Proc .= Application Function I/O = User Views	e.g. Distributed System Architecture Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means =Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. System Design Proc.= Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture Node = Hardware/Systems Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. Program Proc.= Language Statement I/O = Control Block	e.g. Network Architecture Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

© John A. Zachman, Zachman International

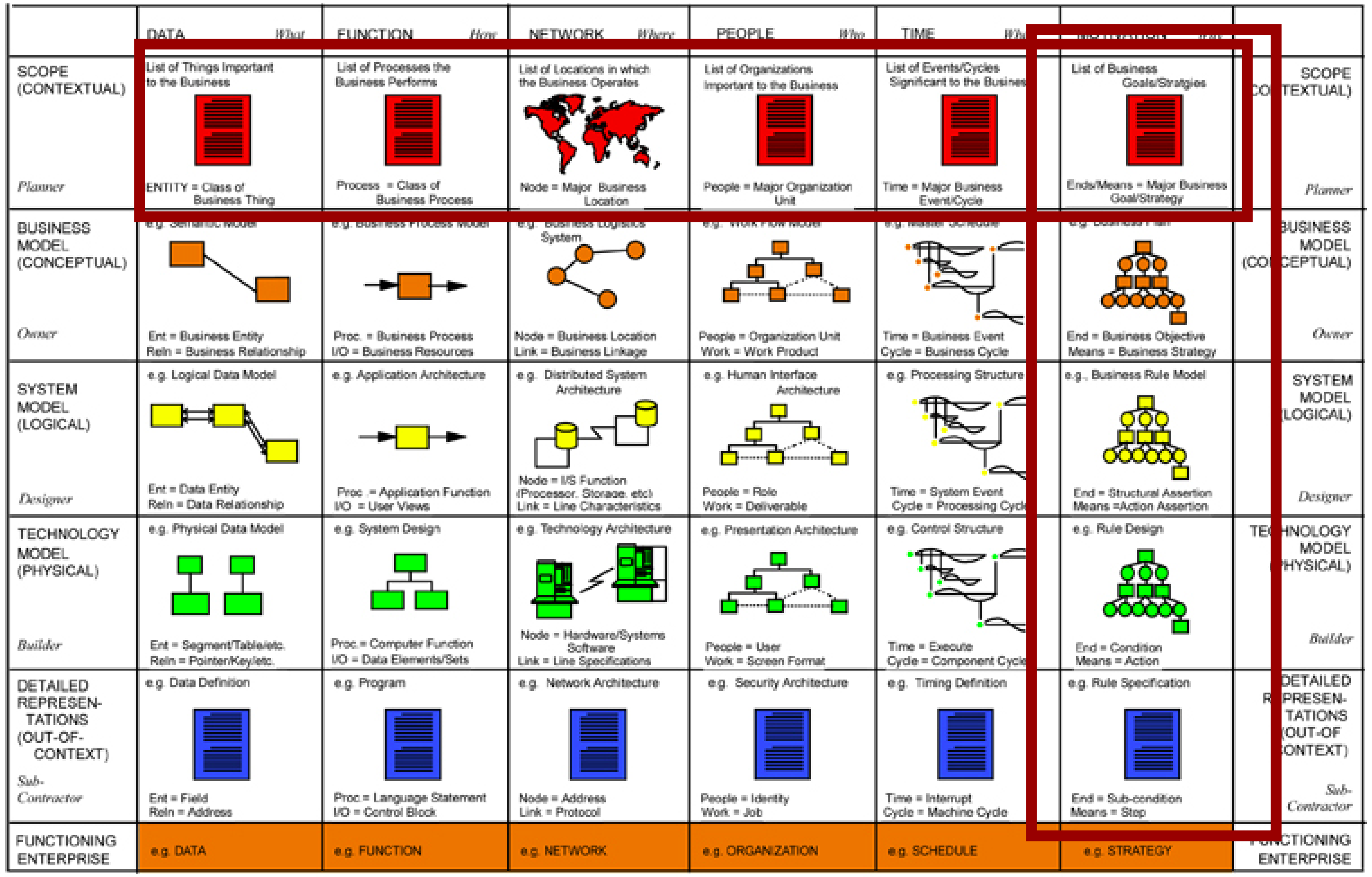
Zachman Framework



	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organization Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	List of Business Goals/Stratgies Ends/Means = Major Business Goal/Strategy	SCOPE (CONTEXTUAL) *Planner*
BUSINESS MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	BUSINESS MODEL (CONCEPTUAL) *Owner*
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. Application Architecture Proc .= Application Function I/O = User Views	e.g. Distributed System Architecture Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means =Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. System Design Proc.= Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture Node = Hardware/Systems Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. Program Proc.= Language Statement I/O = Control Block	e.g. Network Architecture Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

© John A. Zachman, Zachman International

Zachman Framework



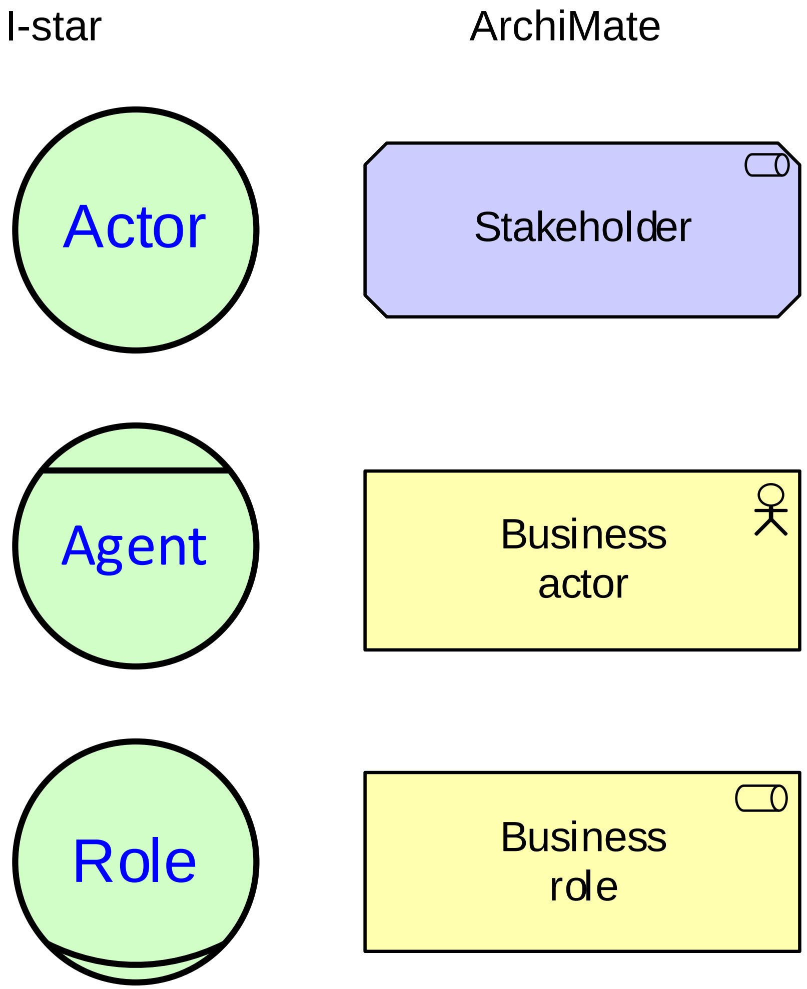
	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organization Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	List of Business Goals/Stratgies Ends/Means = Major Business Goal/Strategy	SCOPE (CONTEXTUAL) *Planner*
BUSINESS MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	BUSINESS MODEL (CONCEPTUAL) *Owner*
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. Application Architecture Proc. = Application Function I/O = User Views	e.g. Distributed System Architecture Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means =Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. System Design Proc.= Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture Node = Hardware/Systems Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. Program Proc.= Language Statement I/O = Control Block	e.g. Network Architecture Node = Address Link = Protocol	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

© John A. Zachman, Zachman International

Goal modelling

- Different languages exist to support goal-oriented requirements engineering:
  - KAOS
  - EEML
  - **I-star**
  - **ArchiMate**

Participants



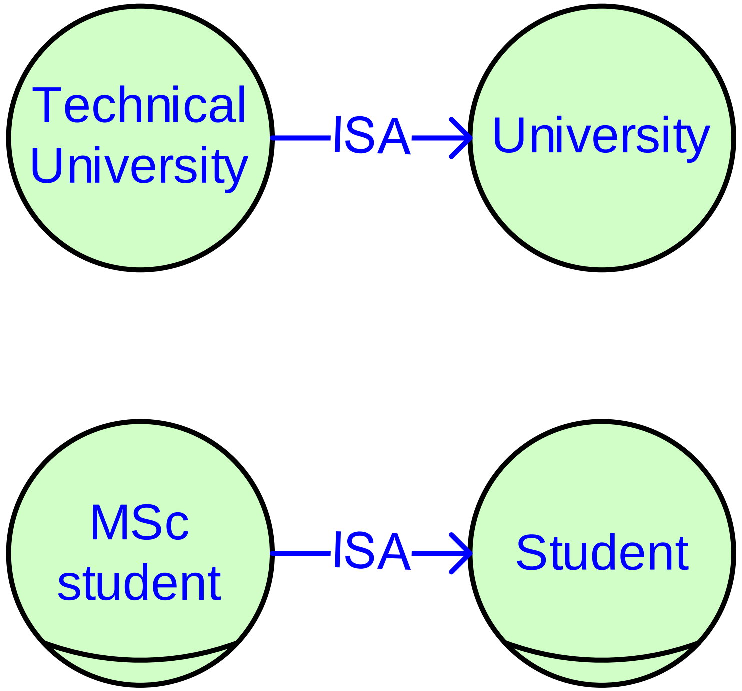
The role of an individual, team or organisation (or classes thereof) that represents their interest in the outcome of the application.

A business entity that is capable of performing behaviour.

the responsibility for performing specific behaviour, or the part one plays in a particular action or event.

Specialization

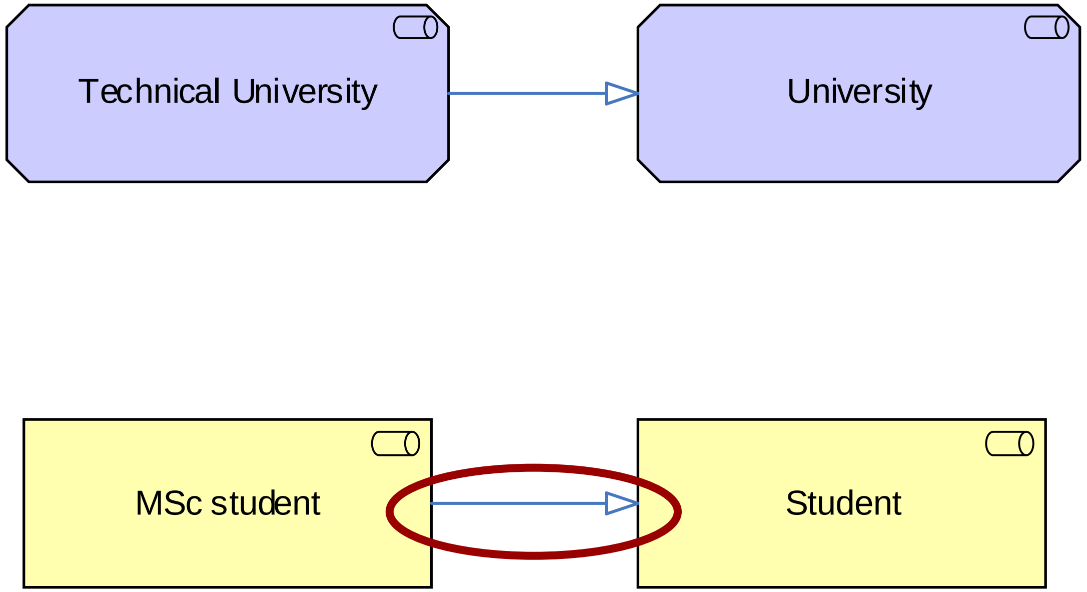
I-star



Technical University —ISA→ University

MSc student —ISA→ Student

ArchiMate



Technical University

University

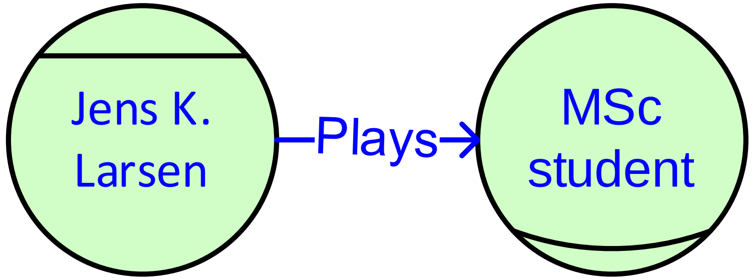
MSc student

Student

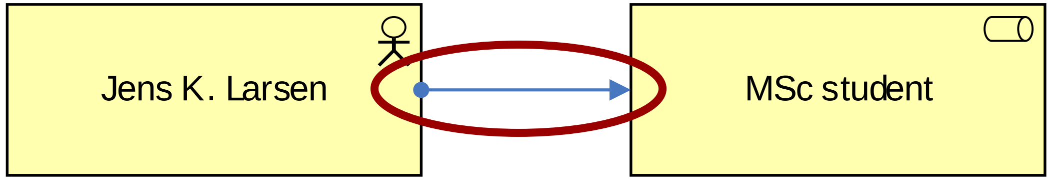
Specialization relation

Playing a role

I-star



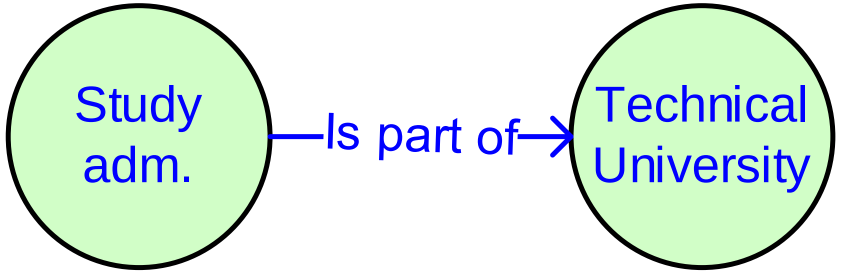
ArchiMate



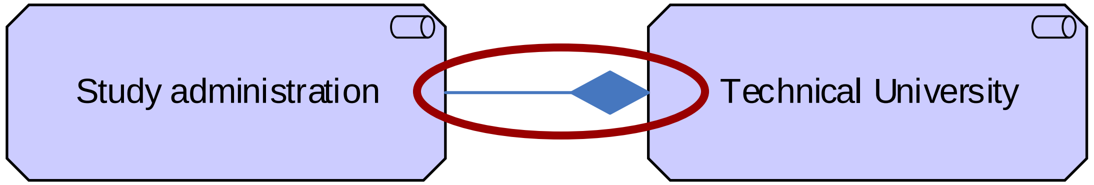
Assignment relation

Composition

I-star

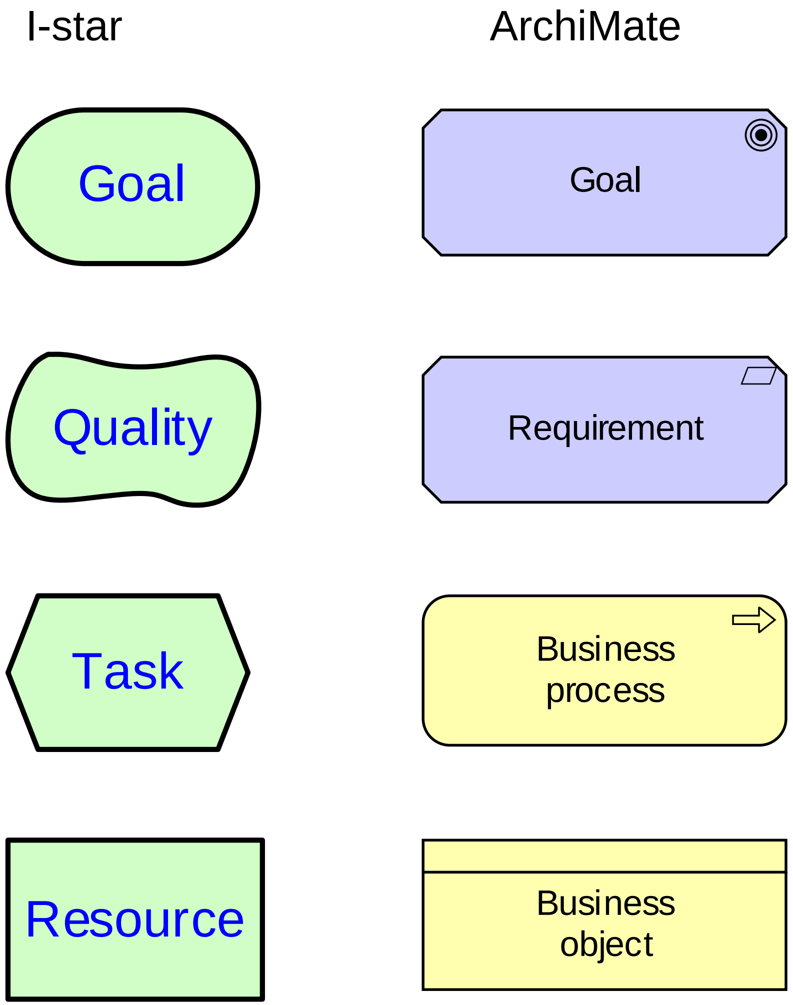


ArchiMate



Composition relation

Intentional elements



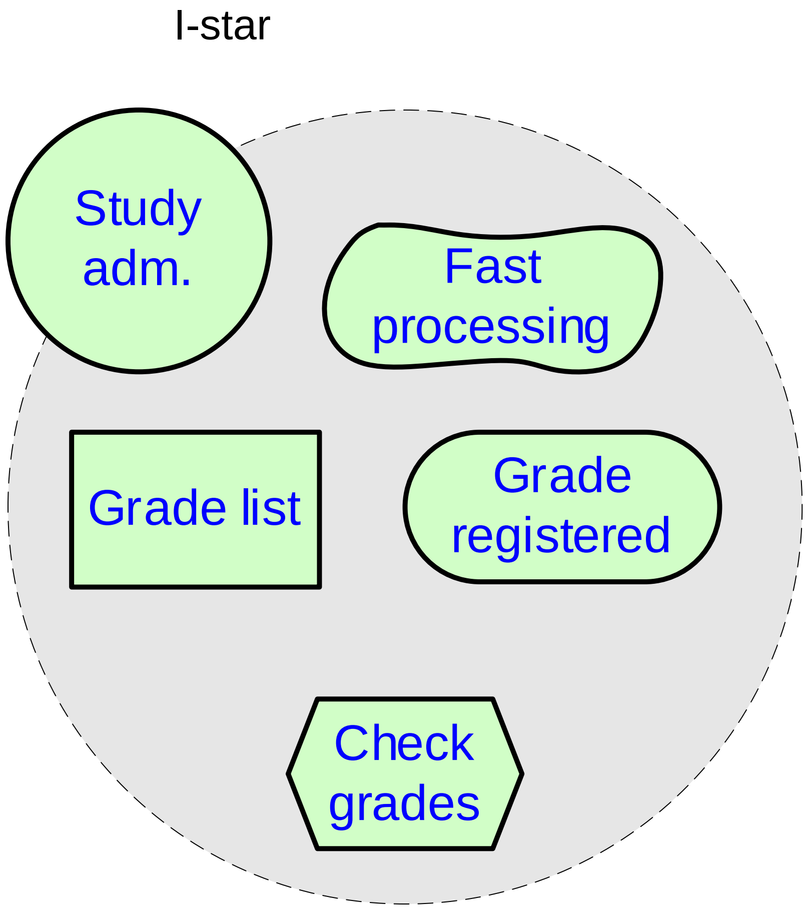
A high-level statement of intent, direction or desired end state for an organisation and its stakeholders.

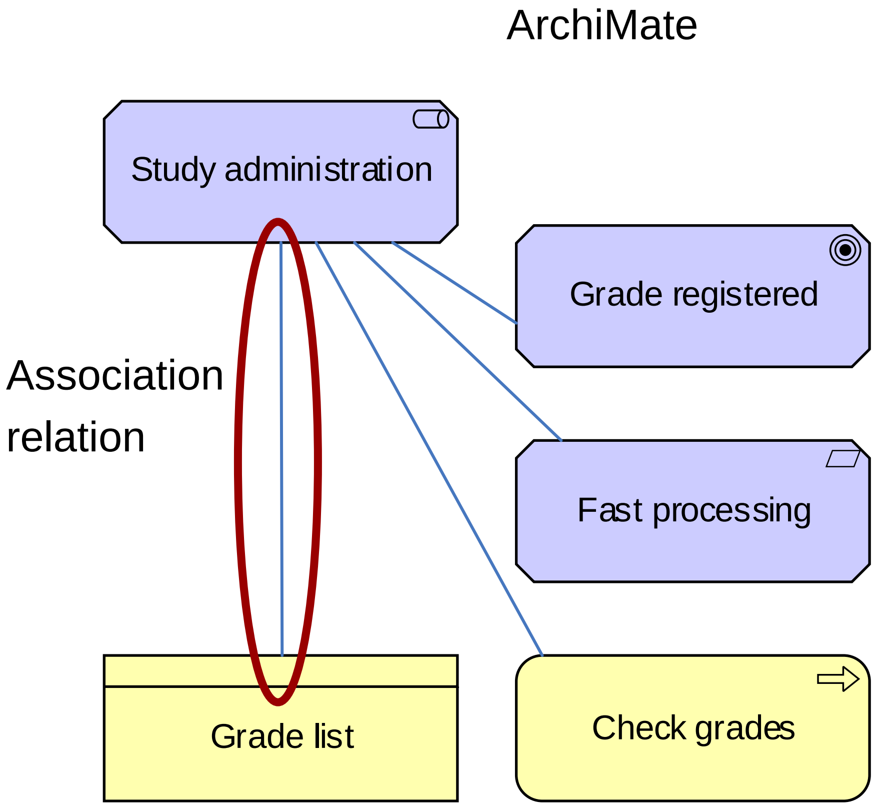
a statement of need that must be met by the architecture.

A set of actions that achieves a specific outcome.

A concept used within a particular business domain

Associating intentional elements to participants

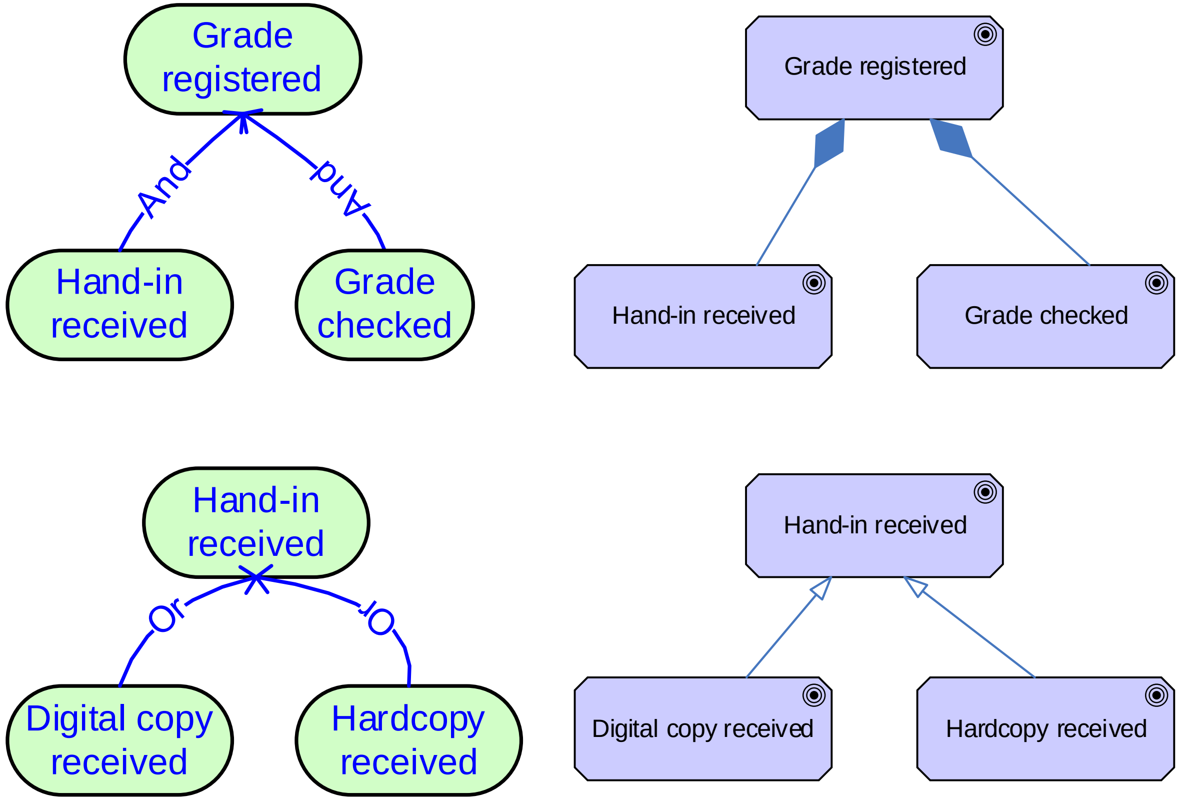




Refining goals

I-star

ArchiMate

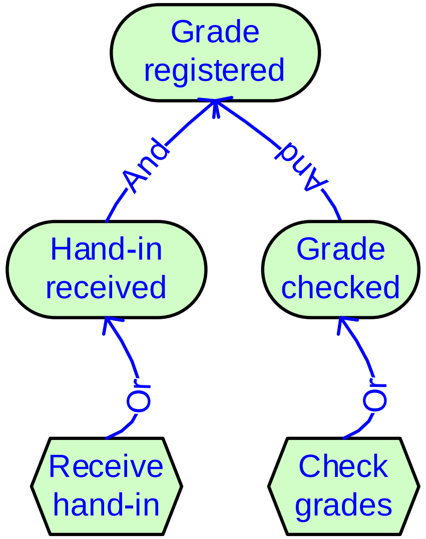


Goal decomposition

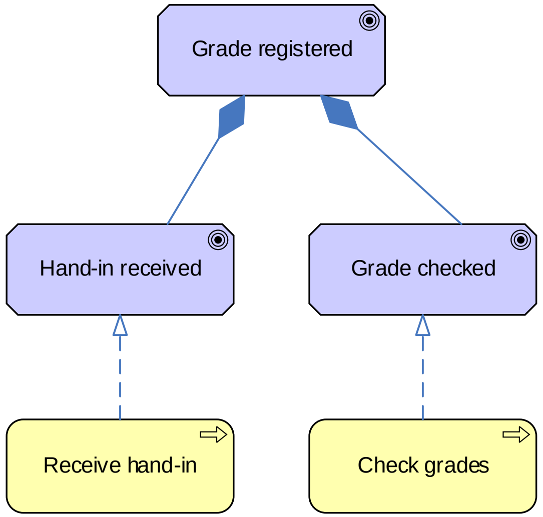
Goal specialization

Refining goals

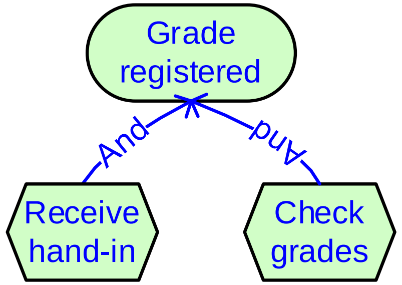
I-star



ArchiMate



Goal realization (inclusive)

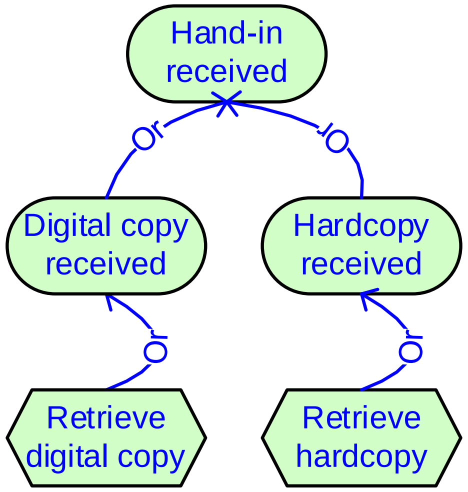


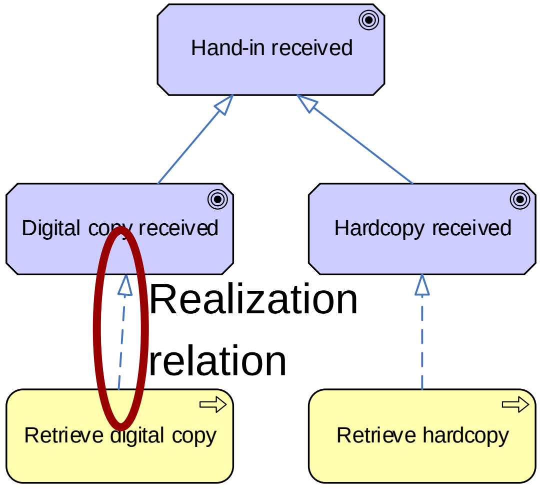
Cannot be modeled

Refining goals

I-star

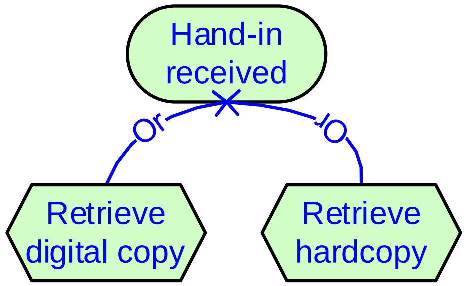
ArchiMate

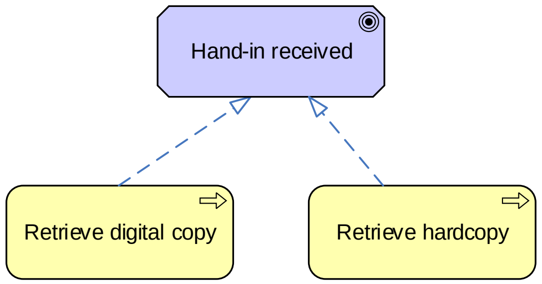




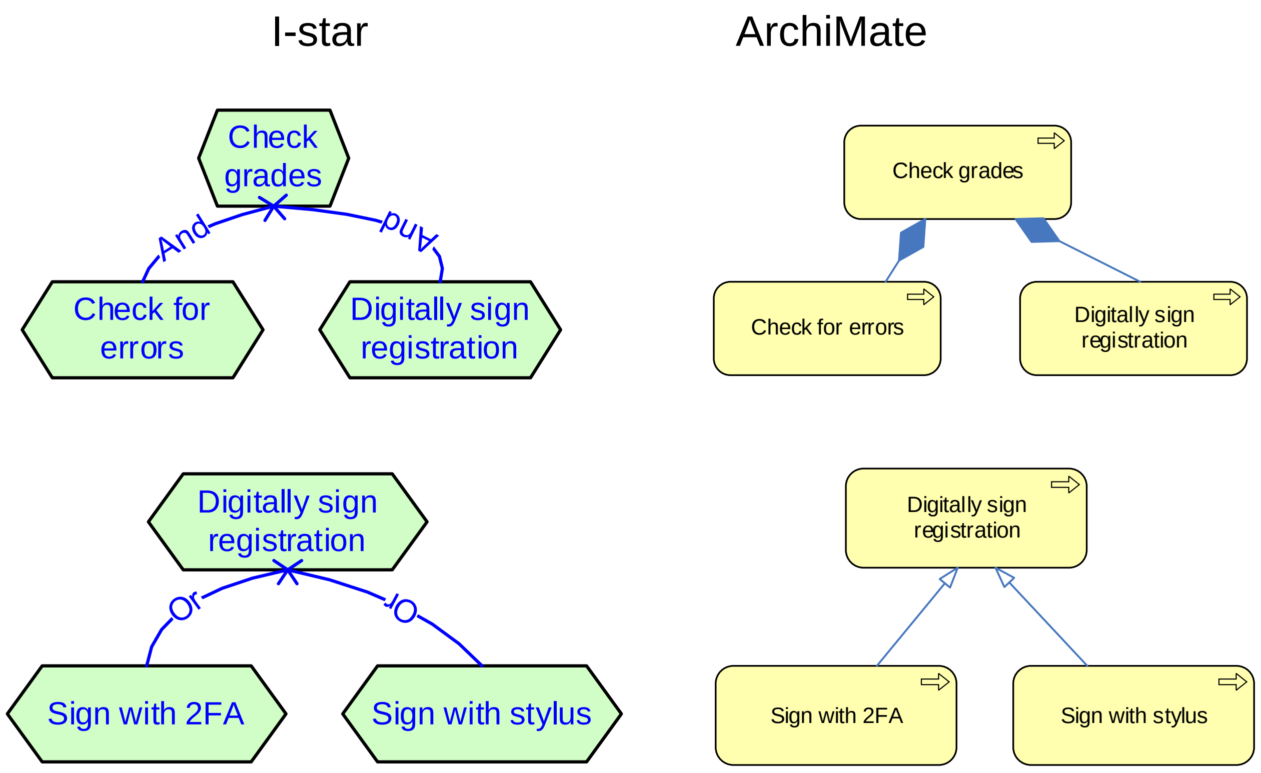
Realization relation

Goal realization (exclusive)





Refining tasks

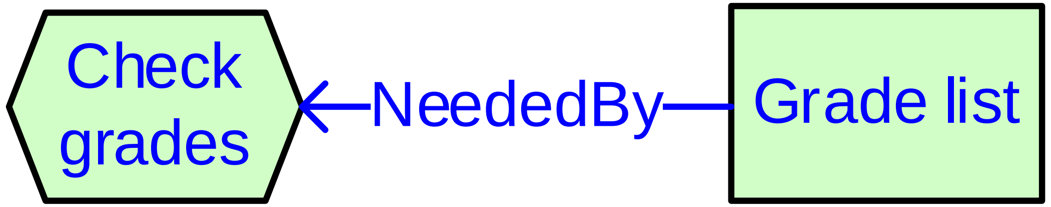


Task decomposition

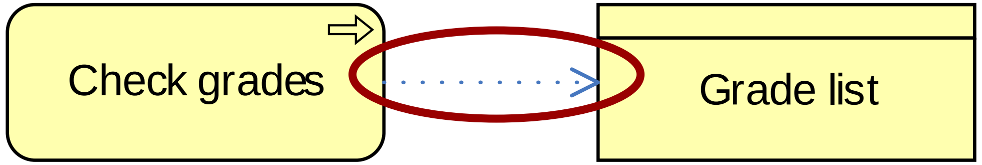
Task specialization

Resources for task execution

I-star



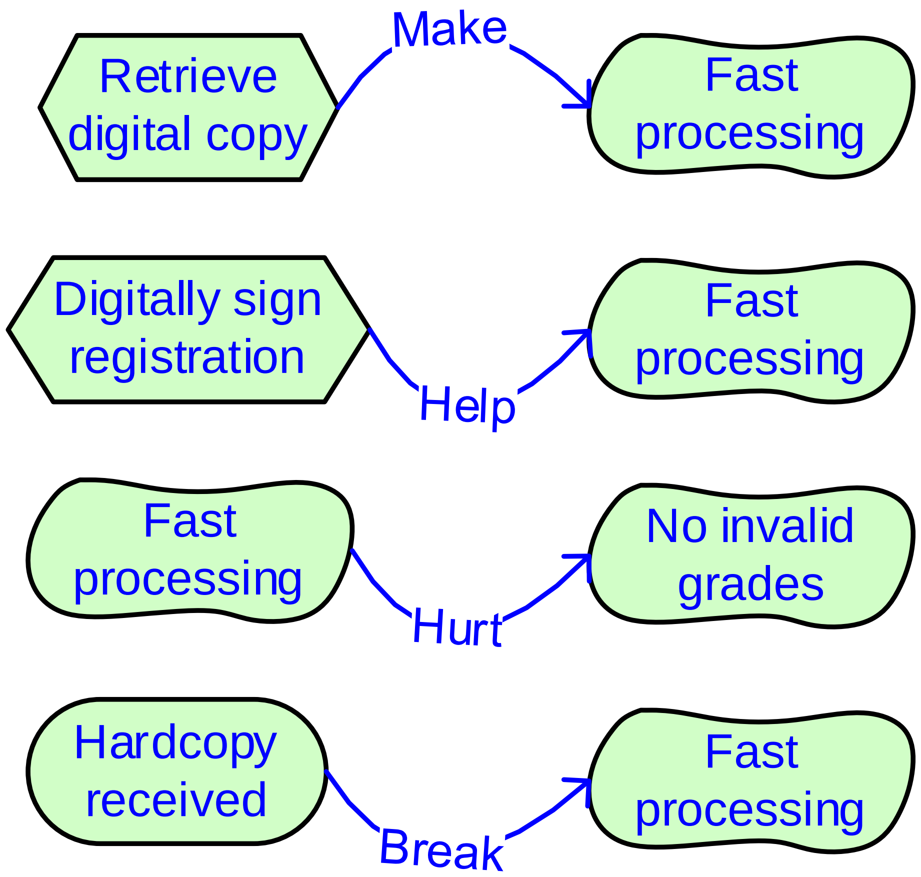
ArchiMate



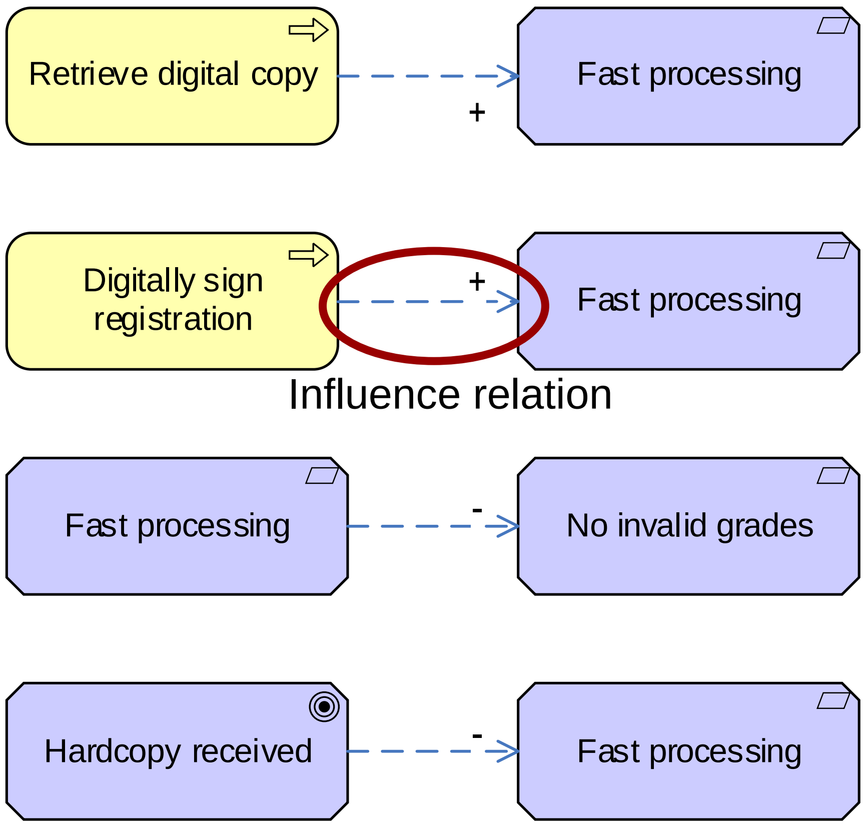
Access relation

Contribution links

I-star



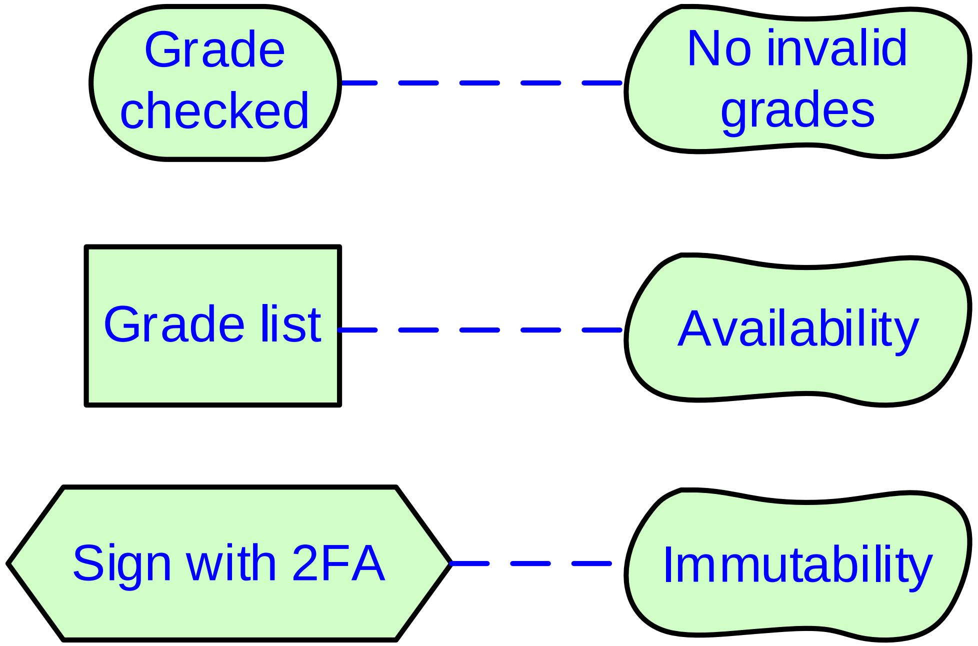
ArchiMate



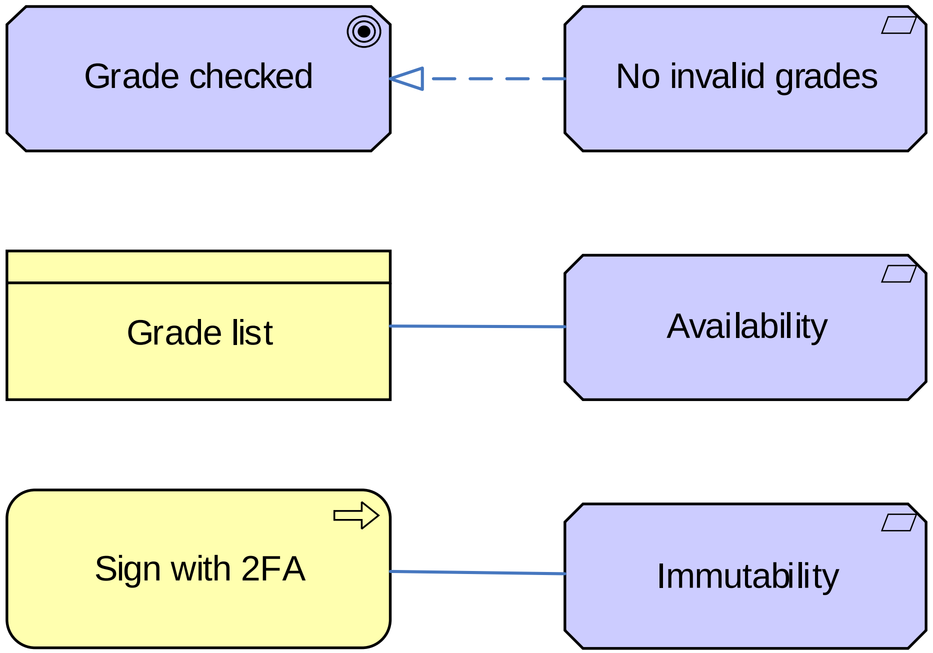
Influence relation

Qualification

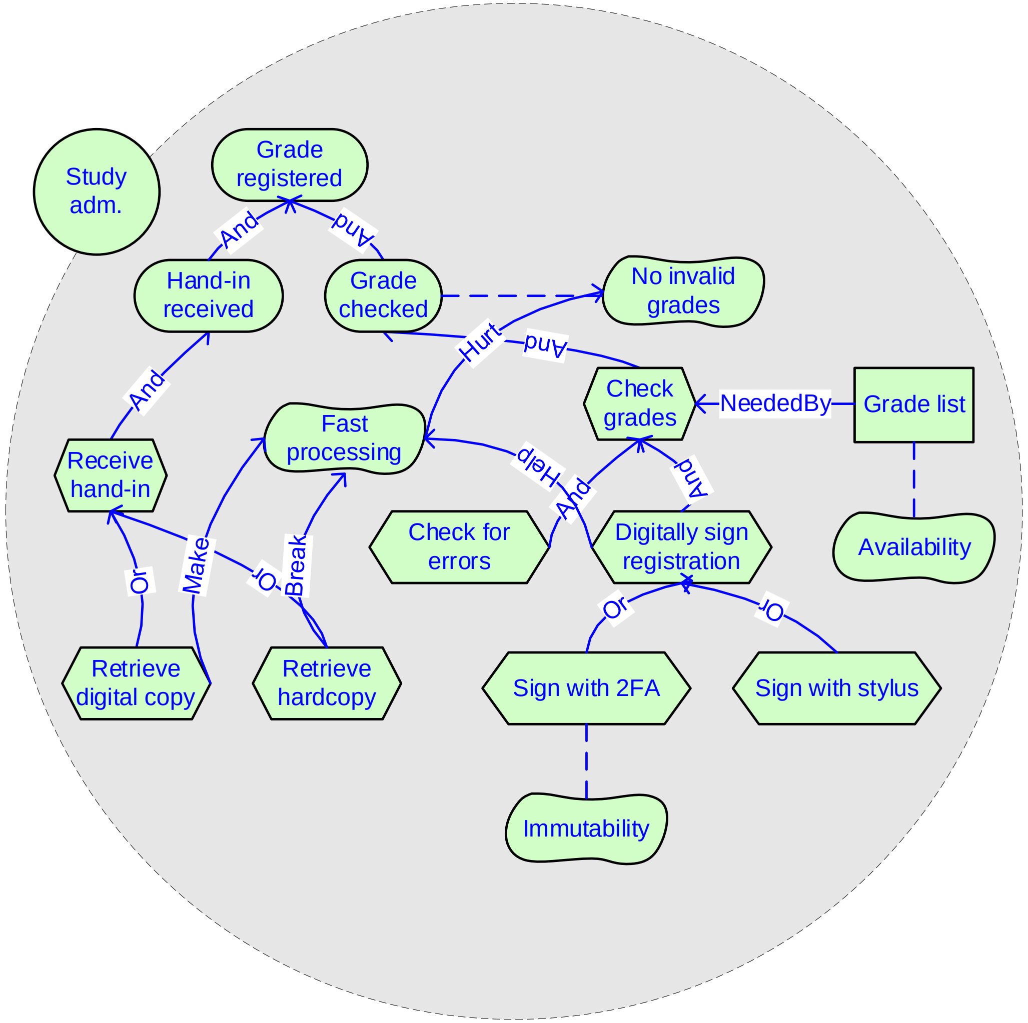
I-star

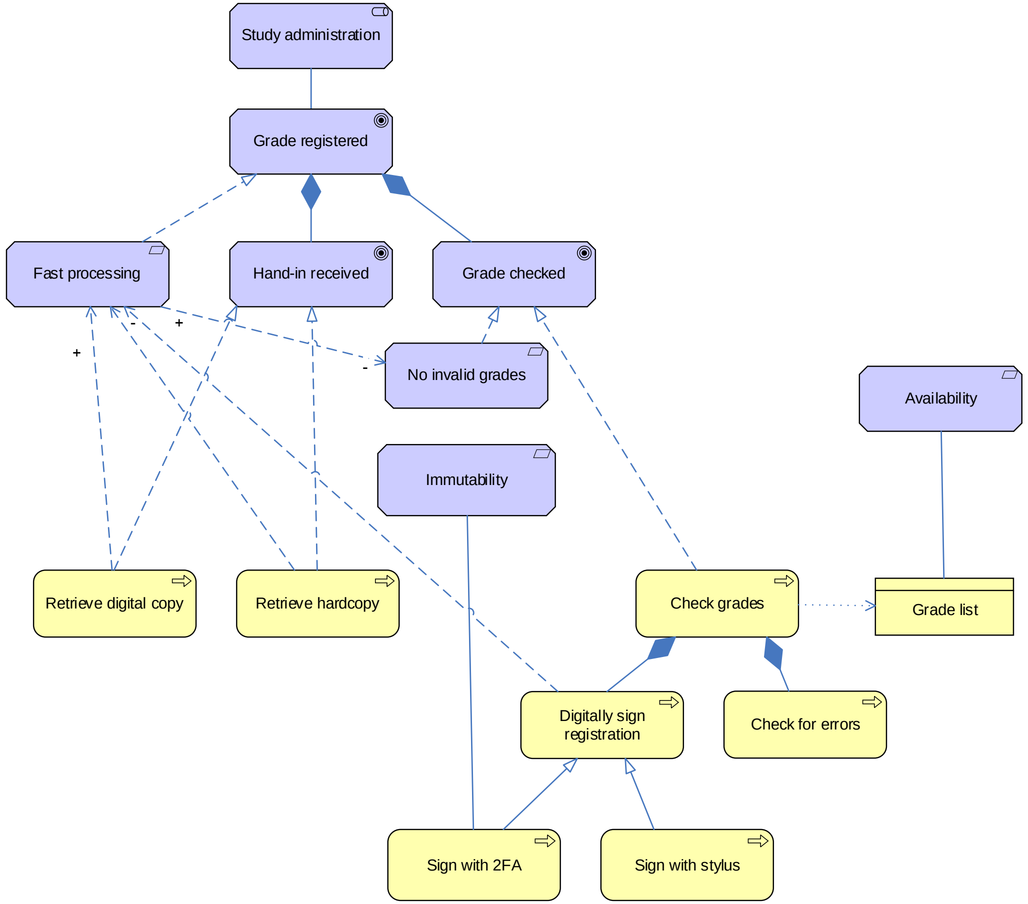


ArchiMate

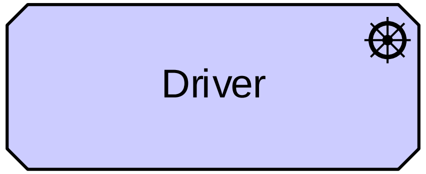


Example

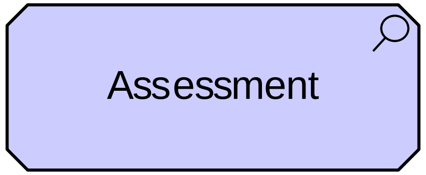

Study adm.
Grade registered
And
And
Hand-in received
Grade checked
No invalid grades
Hurt
And
And
Check grades
NeededBy
Grade list
Receive hand-in
Fast processing
Help
And
And
Check for errors
Digitally sign registration
Availability
Or
Make
Or
Break
Or
Or
Retrieve digital copy
Retrieve hardcopy
Sign with 2FA
Sign with stylus
Immutability


Study administration
Grade registered
Fast processing
Hand-in received
Grade checked
+
+
-
-
No invalid grades
Availability
Immutability
Retrieve digital copy
Retrieve hardcopy
Check grades
Grade list
Digitally sign registration
Check for errors
Sign with 2FA
Sign with stylus

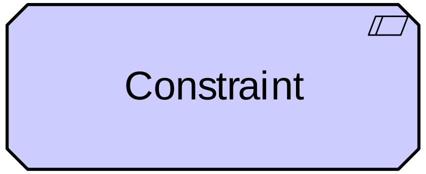
ArchiMate-exclusive intentional elements



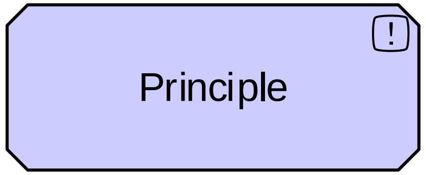
An external or internal condition that motivates an organisation to define its goals and implement the changes necessary to achieve them.



The result of an analysis of the state of affairs of the enterprise with respect to some driver.

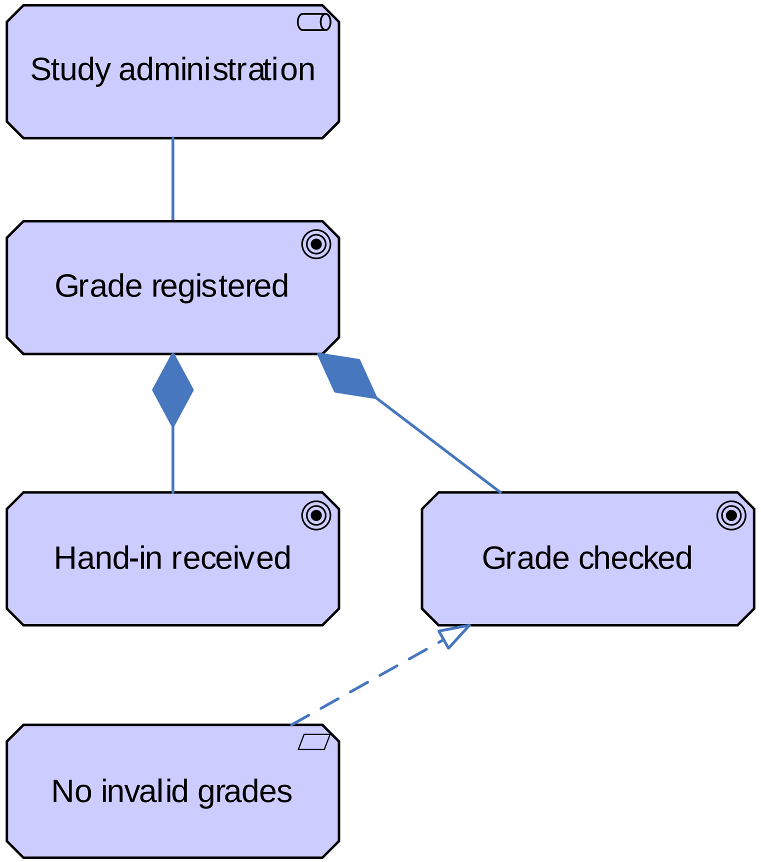


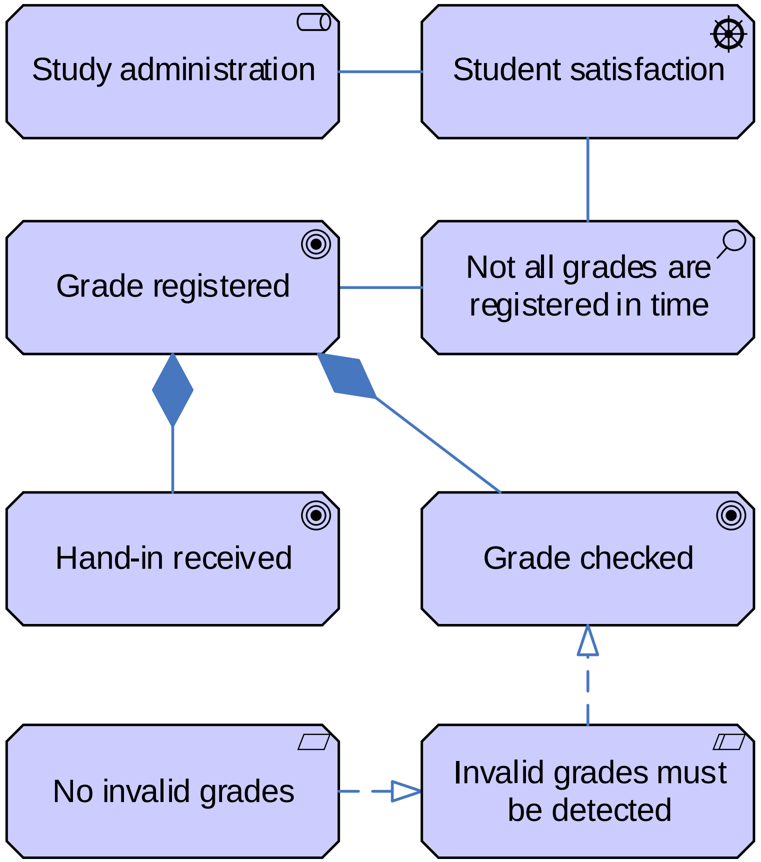
A factor that prevents or obstructs the realisation of goals.



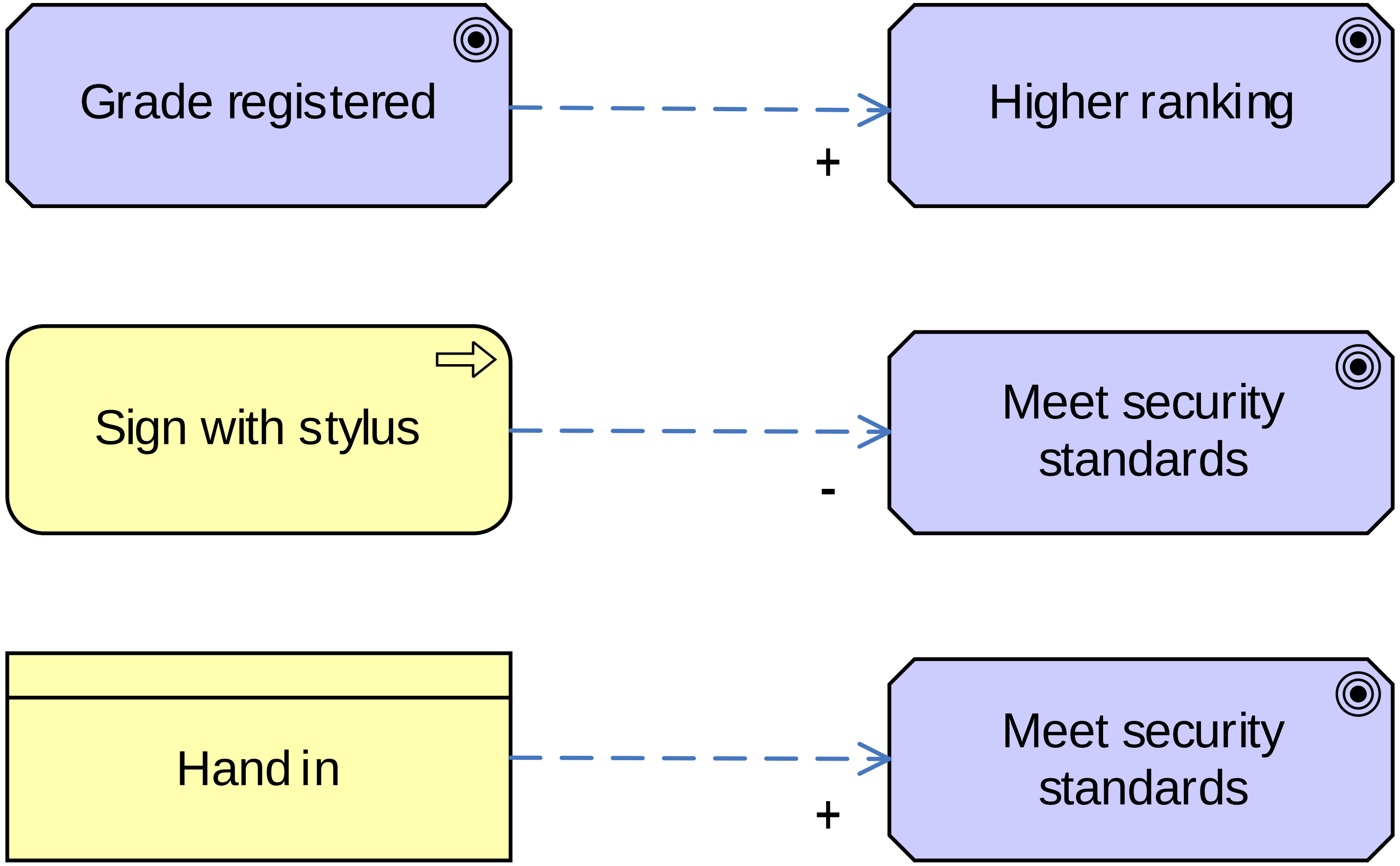
A qualitative statement of intent that should be met by the architecture

ArchiMate-exclusive intentional elements

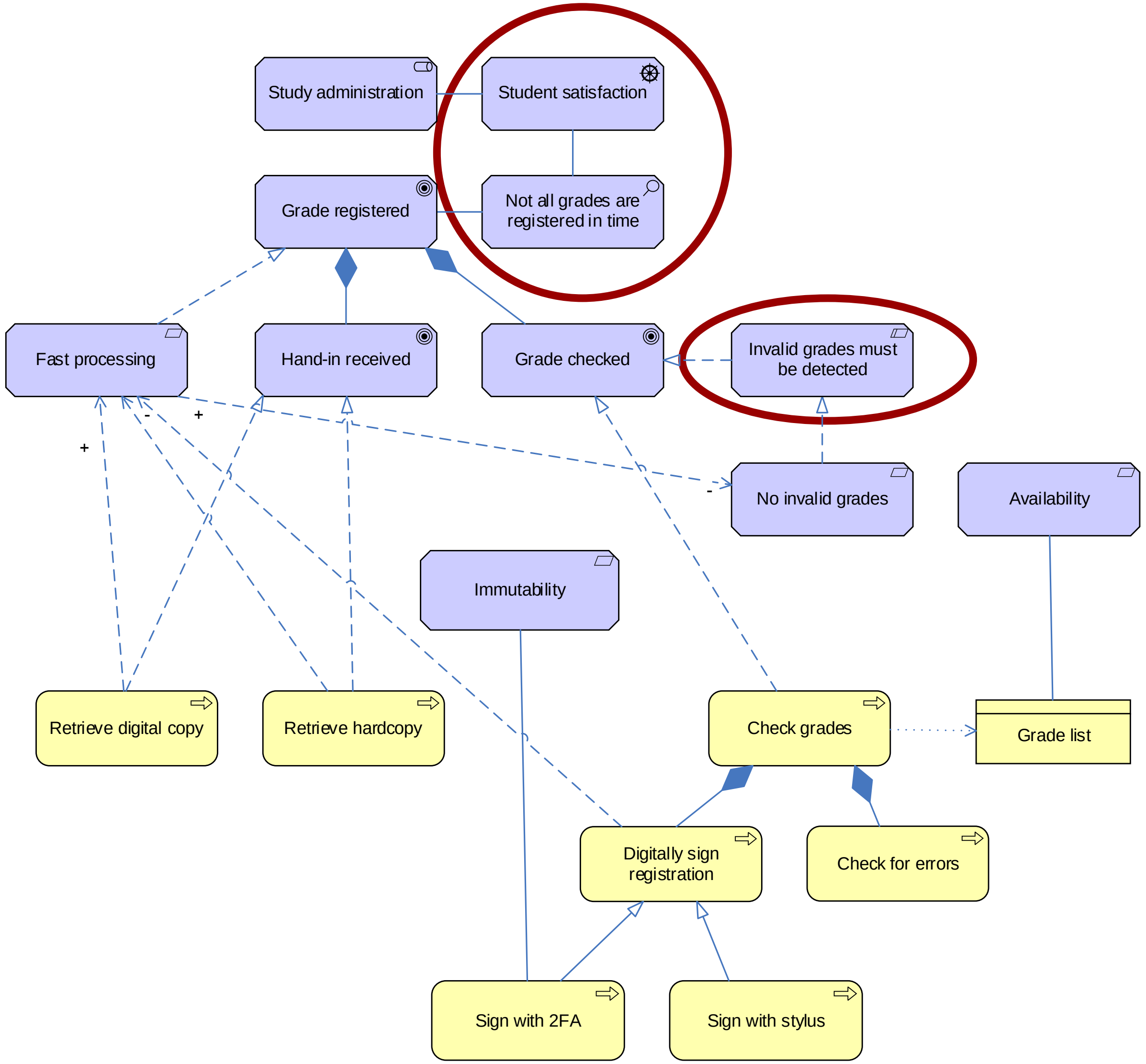




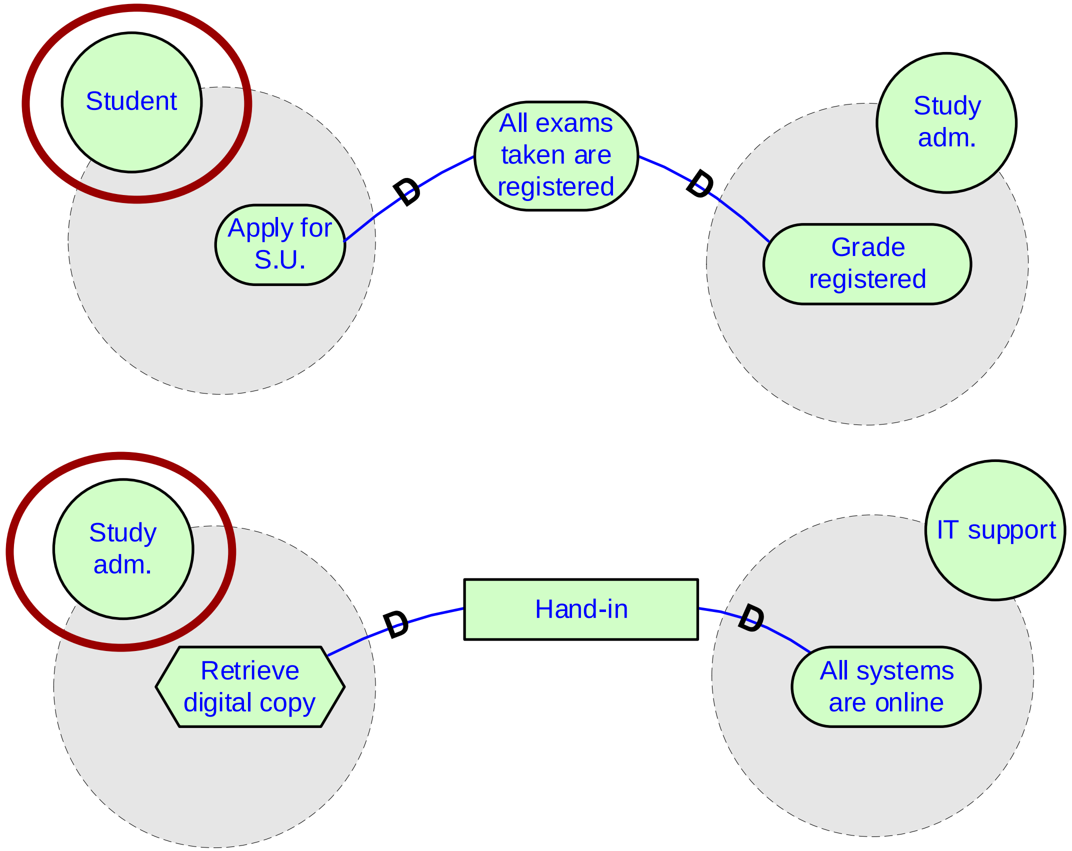
ArchiMate-exclusive contribution links



Extended example in ArchiMate

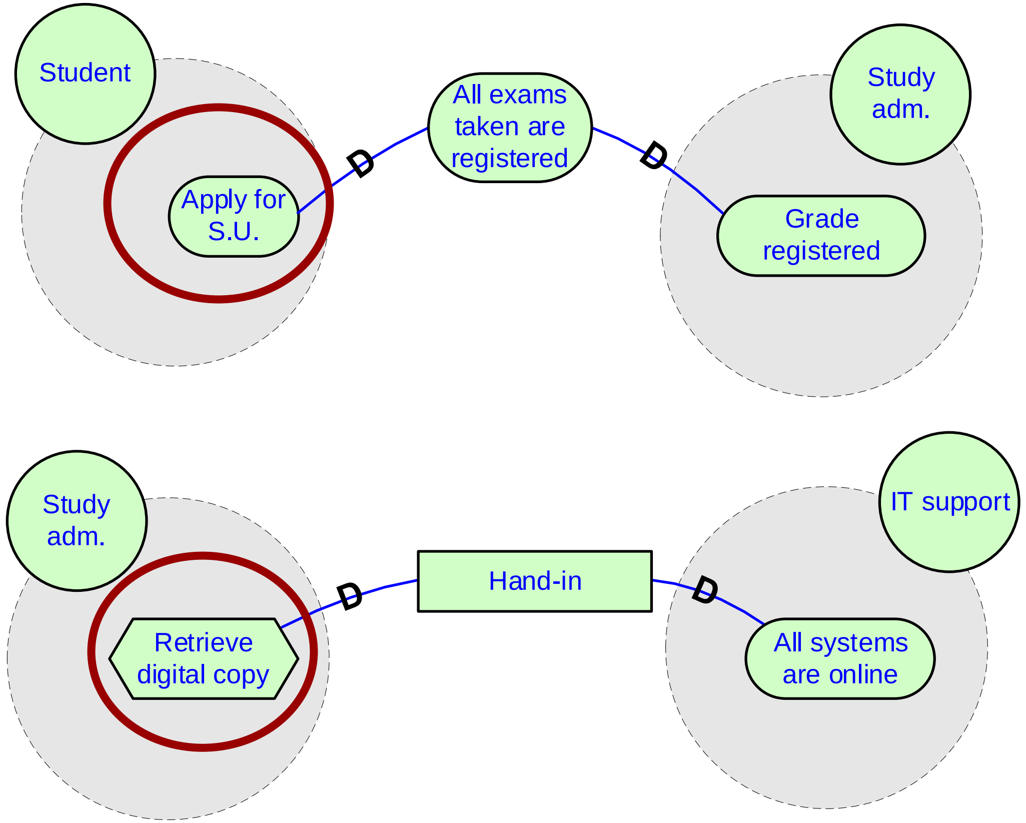


I-star-exclusive social dependencies



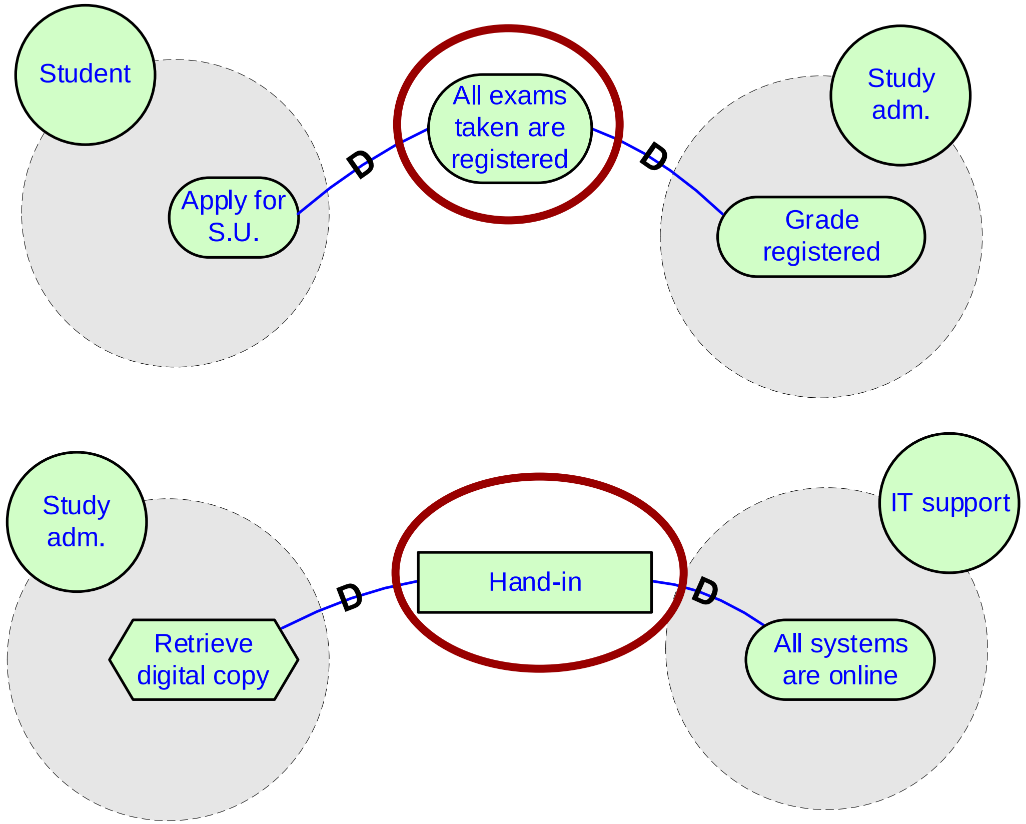
- Represent social relationships
  - **Depender**: an actor that depends for something (the dependum) to be provided

I-star-exclusive social dependencies



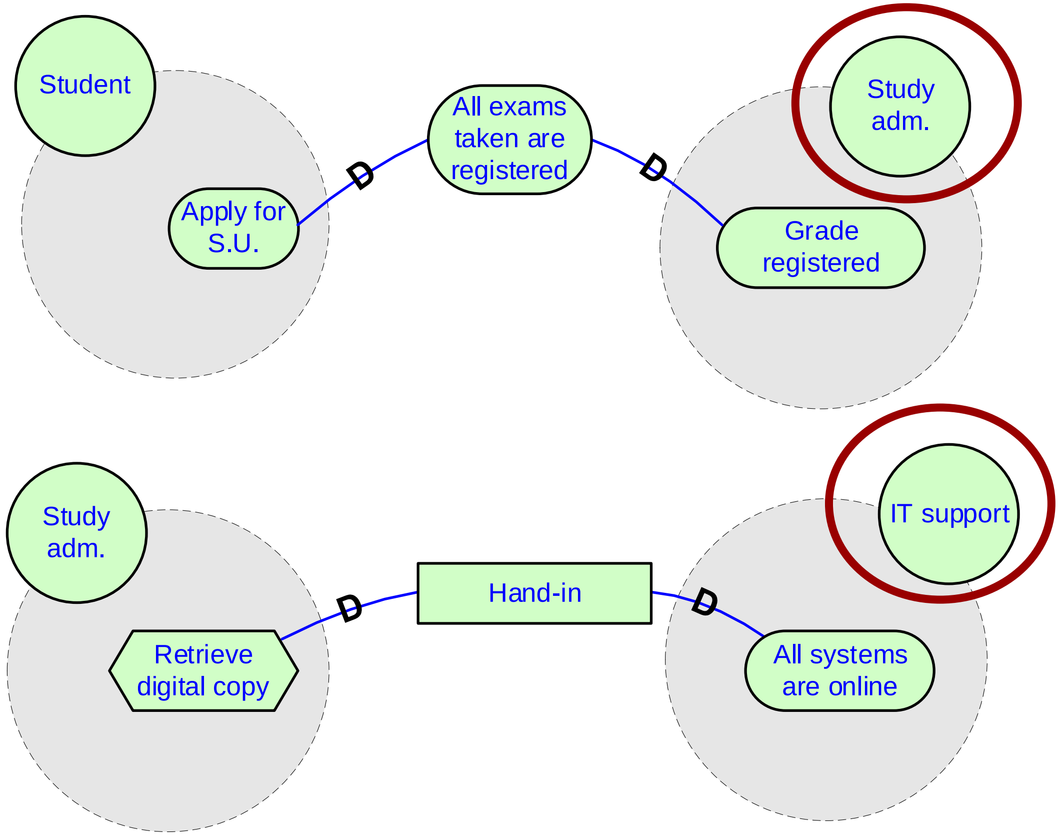
- Represent social relationships
  - Depender: an actor that depends for something (the dependum) to be provided
  - **DependerElement**: an intentional element within the depender’s boundary where the dependency starts from, which explains why the dependency exists

I-star-exclusive social dependencies



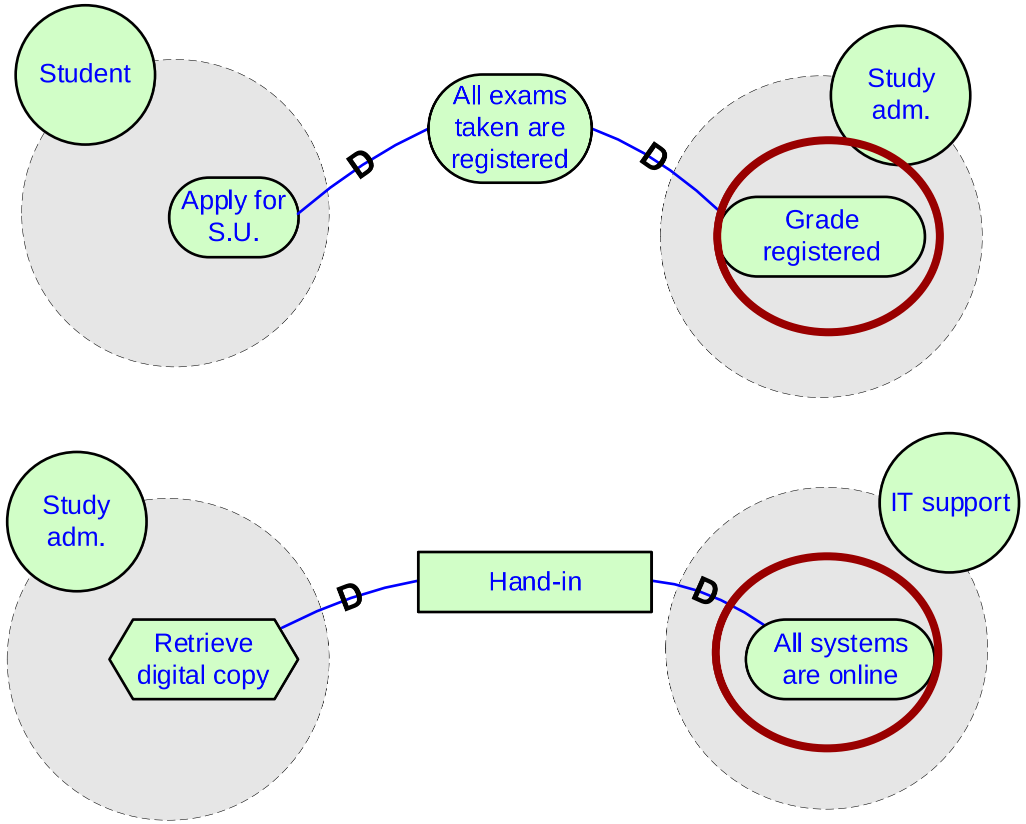
- Represent social relationships
  - Depender: an actor that depends for something (the dependum) to be provided
  - DependerElement: an intentional element within the depender’s boundary where the dependency starts from, which explains why the dependency exists
  - **Dependum**: an intentional element that is the object of the dependency

I-star-exclusive social dependencies



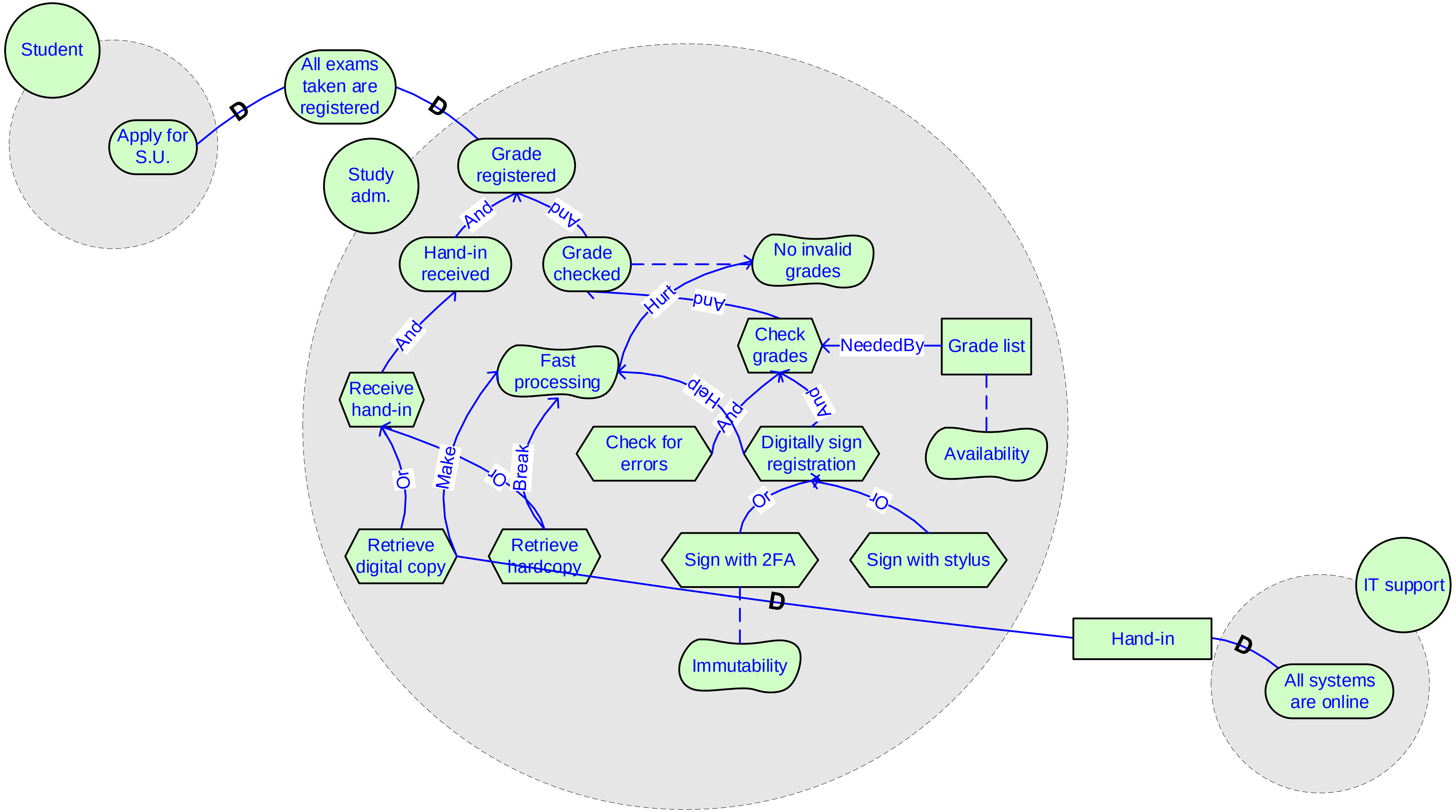
- Represent social relationships
  - Depender: an actor that depends for something (the dependum) to be provided
  - DependerElement: an intentional element within the depender’s boundary where the dependency starts from, which explains why the dependency exists
  - Dependum: an intentional element that is the object of the dependency
  - **Dependee**: the actor that should provide the dependum

I-star-exclusive social dependencies

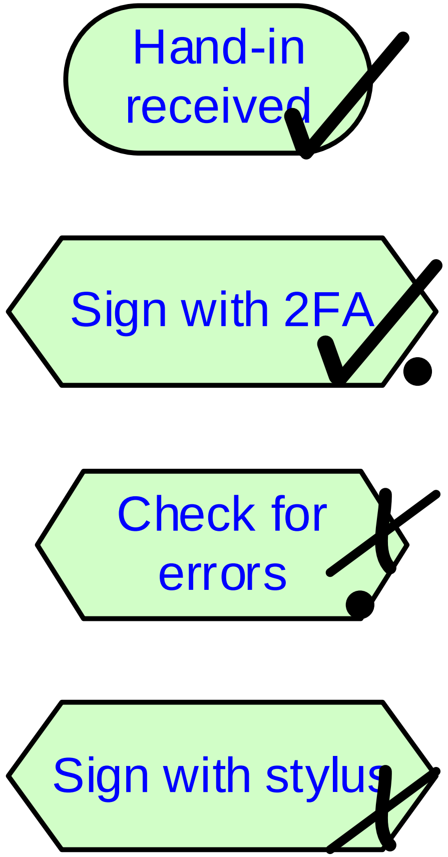


- Represent social relationships
  - Depender: an actor that depends for something (the dependum) to be provided
  - DependerElement: an intentional element within the depender’s boundary where the dependency starts from, which explains why the dependency exists
  - Dependum: an intentional element that is the object of the dependency
  - Dependee: the actor that should provide the dependum
  - **DependeeElement**: the intentional element that explains how the dependee intends to provide the dependum

Extended example in I-star

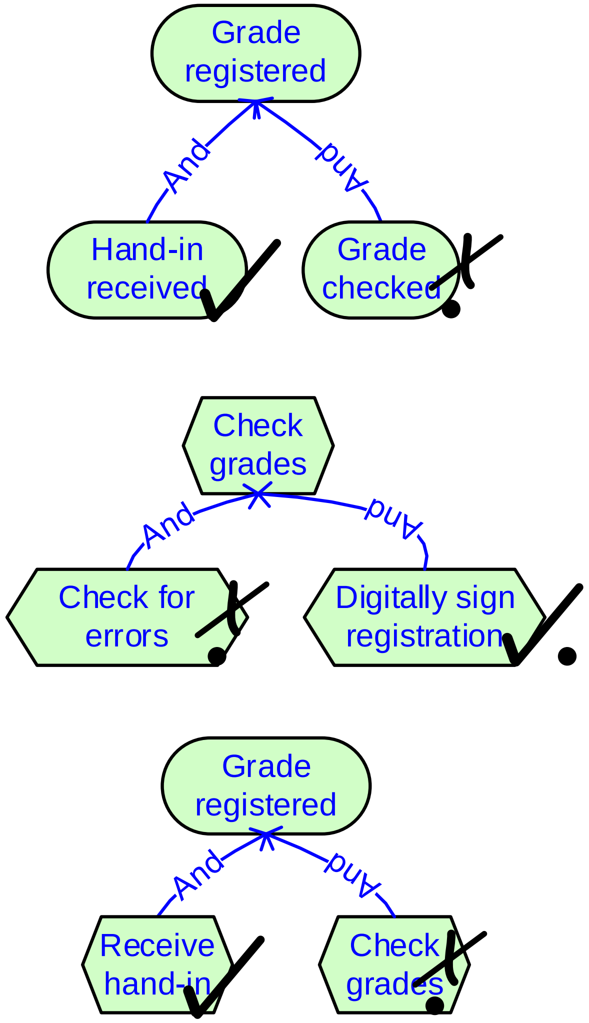


Reasoning



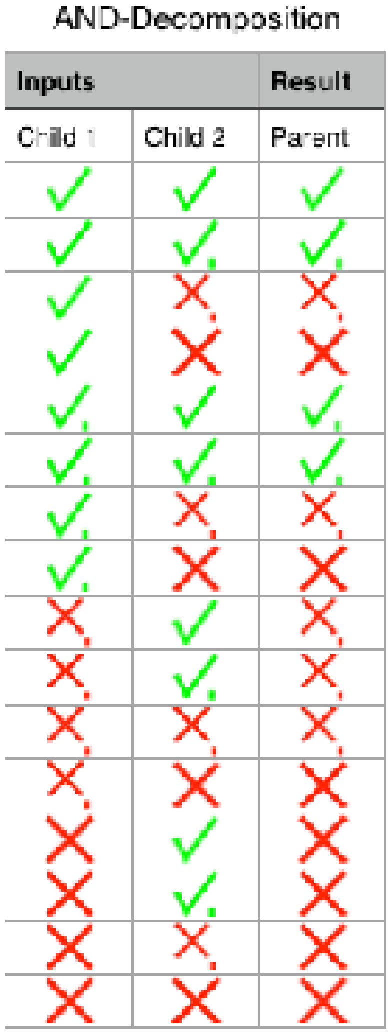
- Goals and Tasks can be characterized with label indicating to which extent they are achieved:
  - Denied
  - Partially denied
  - Partially satisfied
  - Satisfied
- By doing so, it is possible to identify their effect on the model

Forward analysis



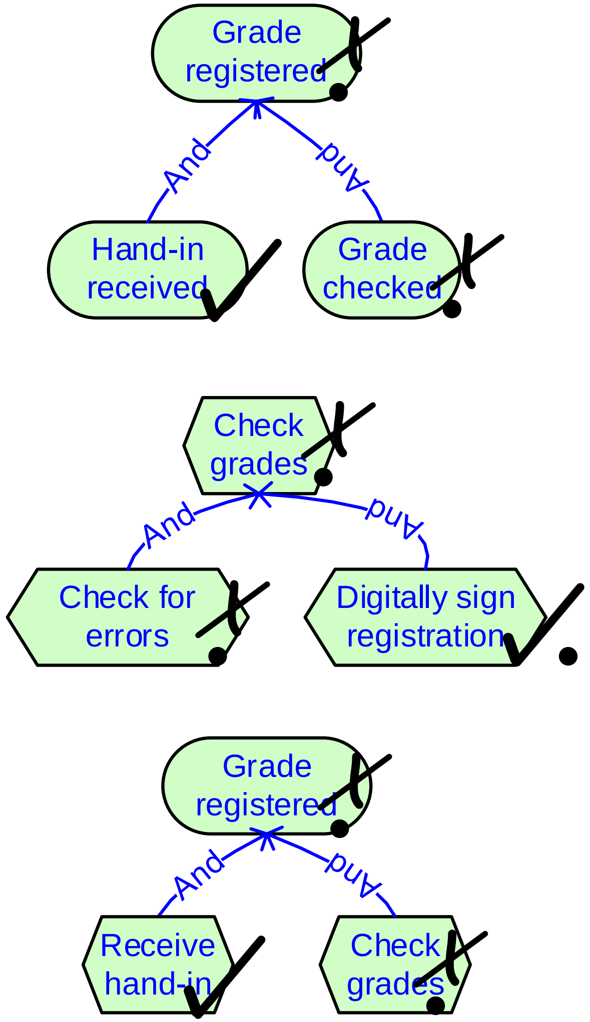
- For goal/task decomposition and inclusive goal realization, the minimum value of all the child nodes is taken.

Forward analysis



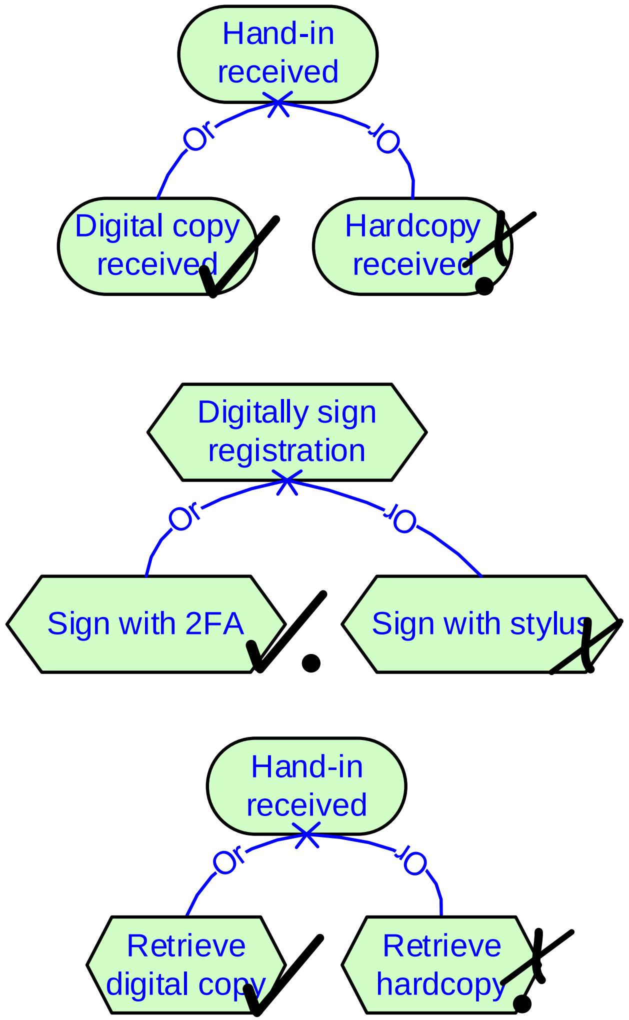
AND-Decomposition

Inputs		Result
Child 1	Child 2	Parent
✓	✓	✓
✓	✓.	✓.
✓	✗.	✗.
✓	✗	✗
✓.	✓	✓.
✓.	✓.	✓.
✓.	✗.	✗.
✓.	✗	✗
✗.	✓	✗.
✗.	✓.	✗.
✗.	✗.	✗.
✗.	✗	✗
✗	✓	✗
✗	✓.	✗
✗	✗.	✗
✗	✗	✗



- For goal/task decomposition and inclusive goal realization, the minimum value of all the child nodes is taken.

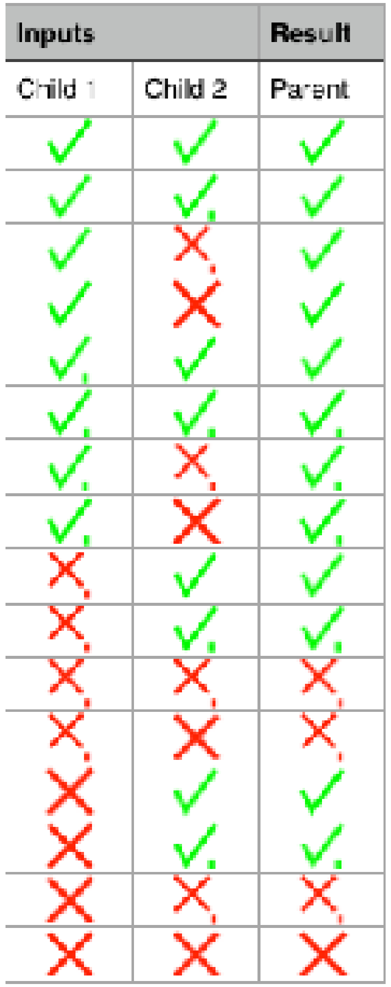
Forward analysis



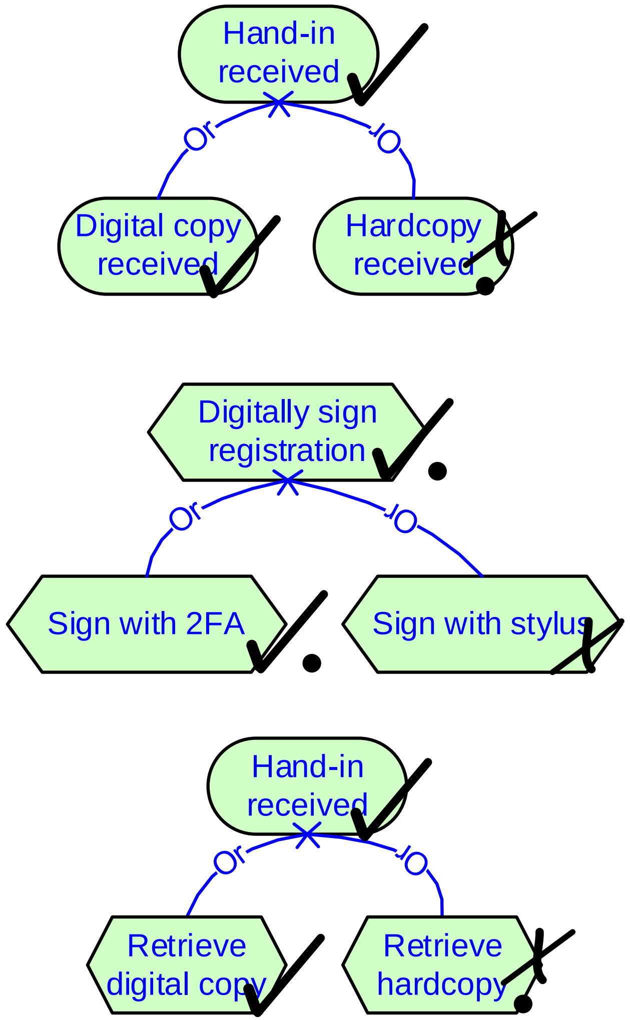
- For goal/task specialization and exclusive goal realization, the maximum value of all the child nodes is taken.

Forward analysis

OR-Decomposition

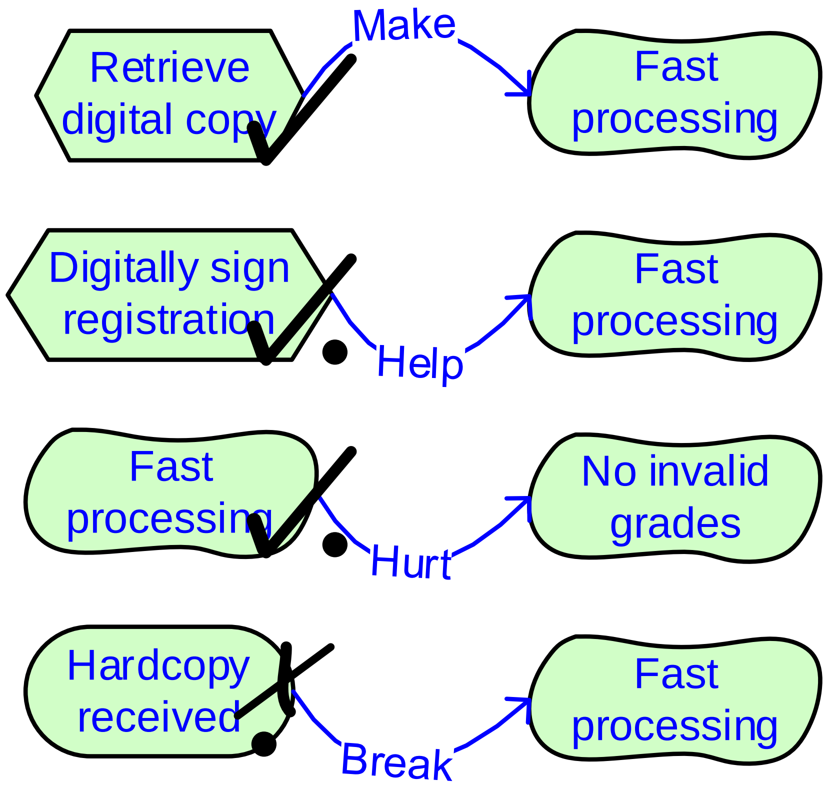


Inputs		Result
Child 1	Child 2	Parent
✓	✓	✓
✓	✓.	✓
✓ ✓ ✓.	✗. ✗ ✓	✓ ✓ ✓
✓.	✓.	✓.
✓.	✗.	✓.
✓.	✗	✓.
✗.	✓	✓
✗.	✓.	✓.
✗.	✗.	✗.
✗. ✗ ✗	✗ ✓ ✓.	✗. ✓ ✓.
✗	✗.	✗.
✗	✗	✗



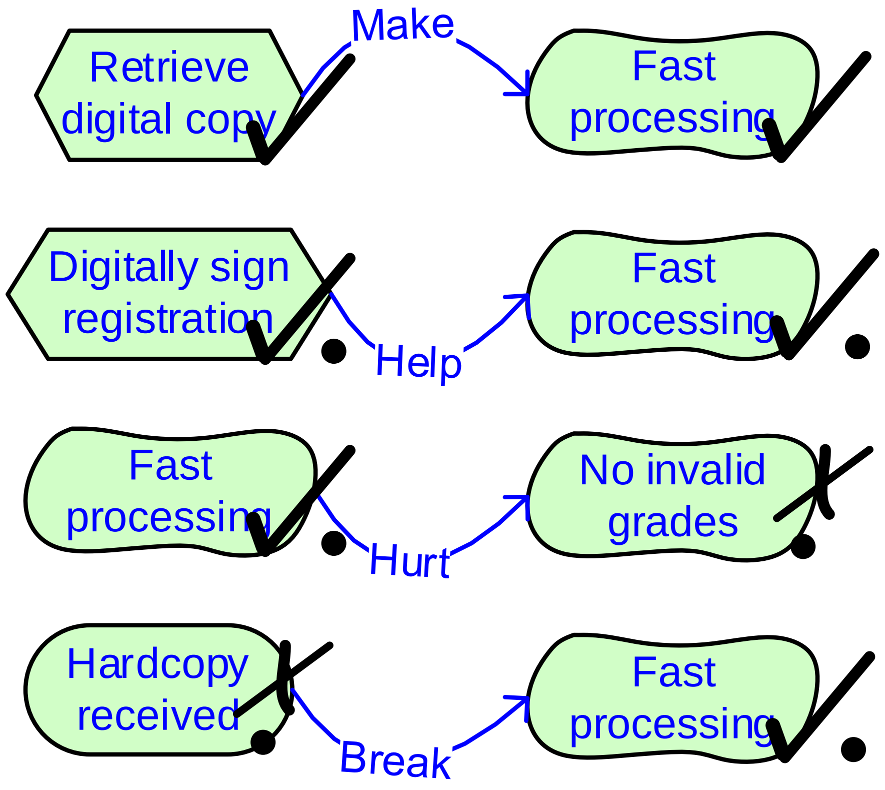
- For goal/task specialization and exclusive goal realization, the maximum value of all the child nodes is taken.

Forward analysis

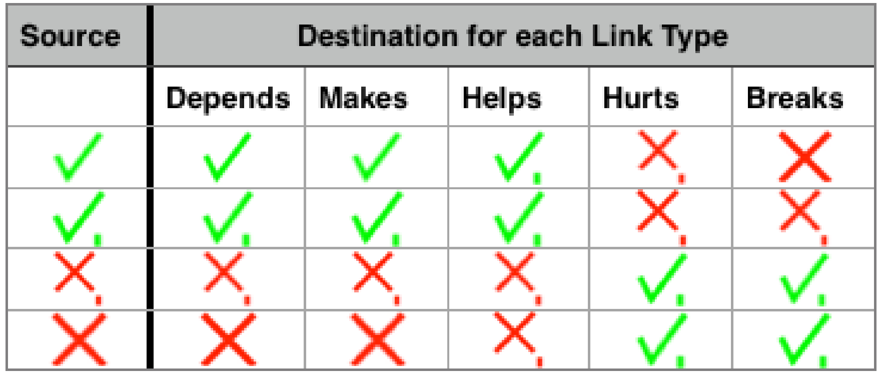


- For contribution links, the value of the goal is propagated to the quality depending on the link type:
  - Make links propagate the value as-is
  - Help links propagate the value as *partial*
  - Hurt links invert and propagate the value as *partial*
  - Break links invert the value

Forward analysis

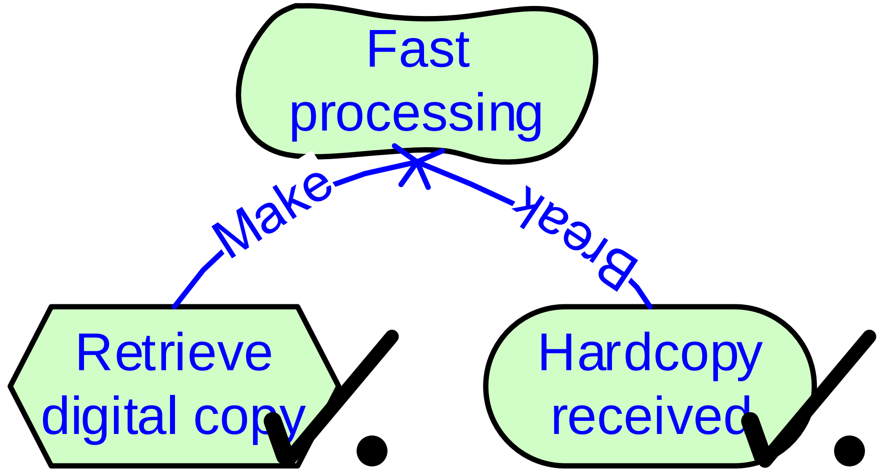


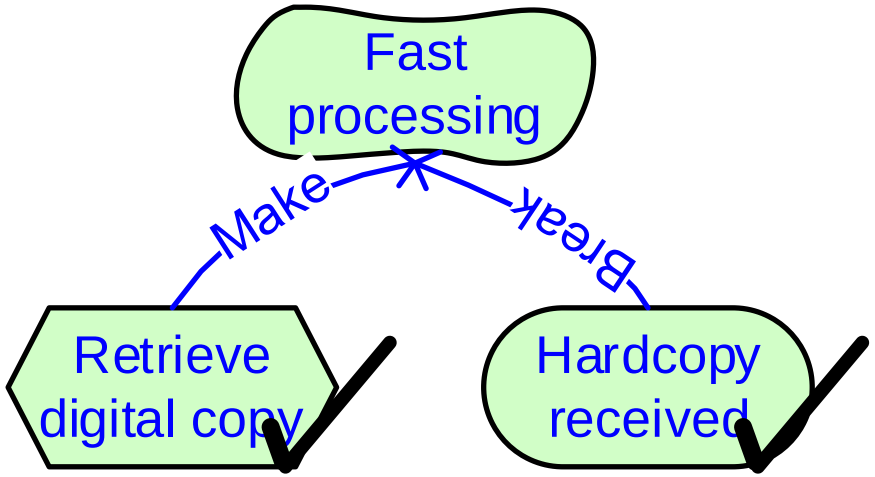
- For contribution links, the value of the goal is propagated to the quality depending on the link type:
  - Make links propagate the value as-is
  - Help links propagate the value as *partial*
  - Hurt links invert and propagate the value as *partial*
  - Break links invert the value



Source	Destination for each Link Type				
	Depends	Makes	Helps	Hurts	Breaks
✓	✓	✓	✓.	✗.	✗
✓.	✓.	✓.	✓.	✗.	✗.
✗.	✗.	✗.	✗.	✓.	✓.
✗	✗	✗	✗.	✓.	✓.

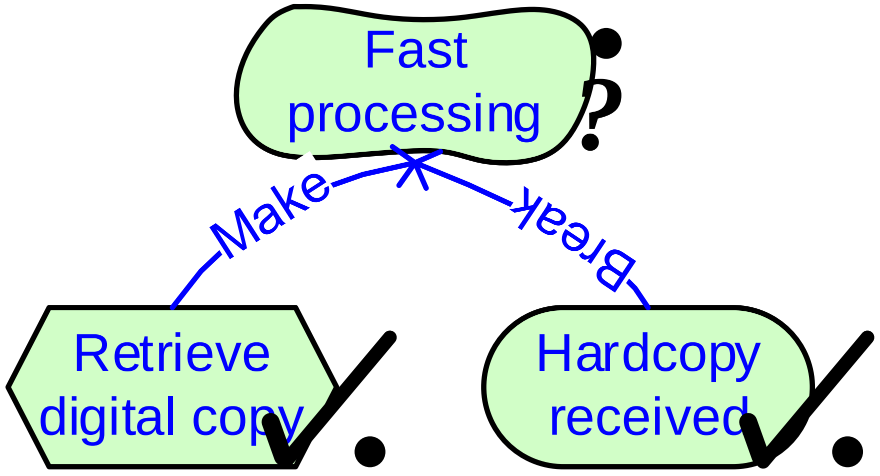
Forward analysis

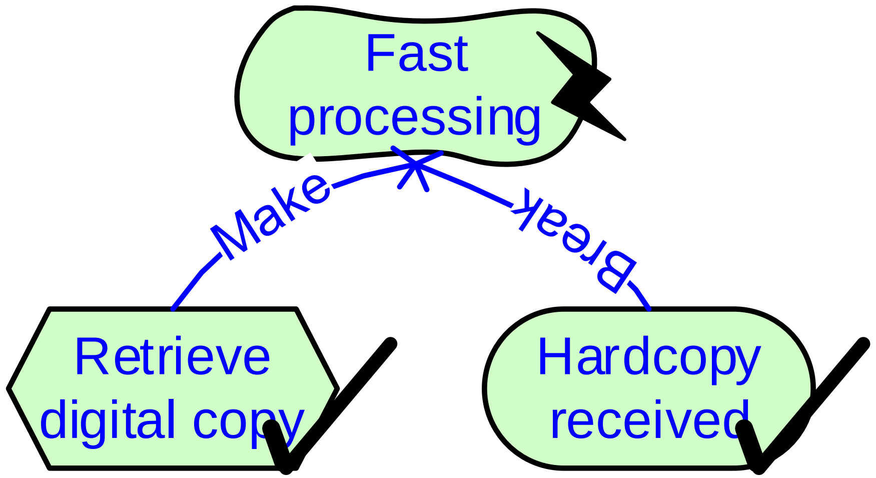




- When different contribution links affect the same quality, we may also get:
  - Unknown: it is unclear if the quality will be satisfied or not
  - Conflict: contributions contradict each other

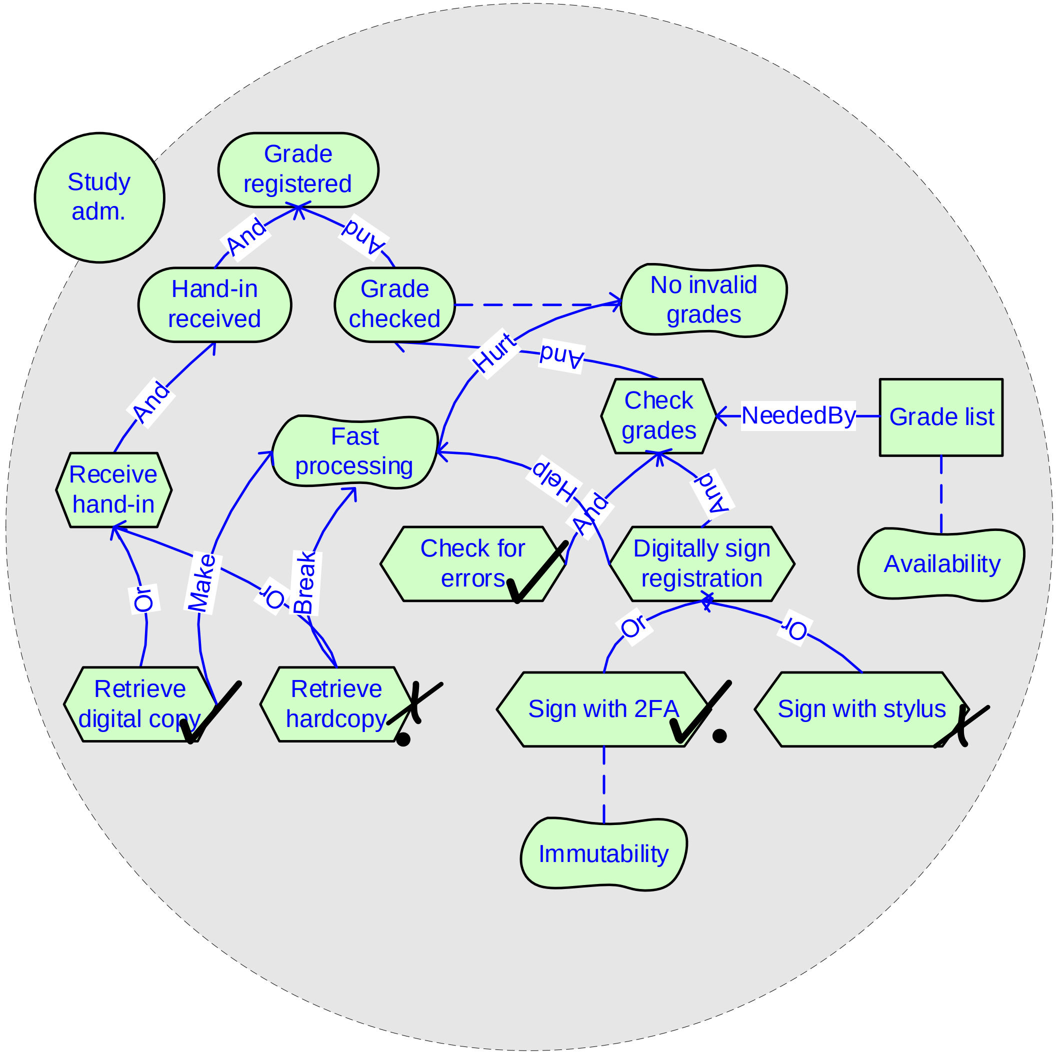
Forward analysis



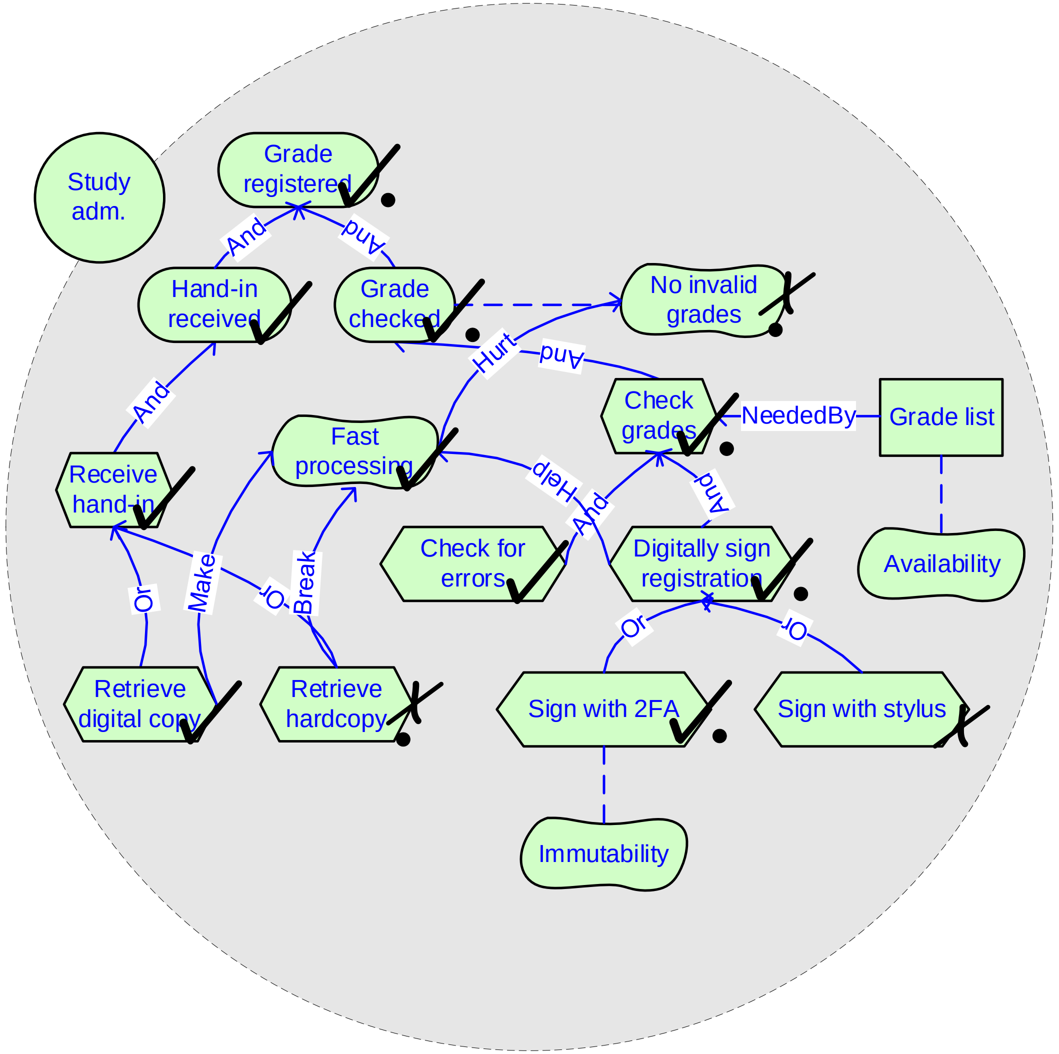


- When different contribution links affect the same quality, we may also get:
  - Unknown: it is unclear if the quality will be satisfied or not
  - Conflict: contributions contradict each other

Forward analysis – Example



Forward analysis – Results



Study material

- Books and articles:
  - Dalpiaz et al. - iStar 2.0 Language Guide
    - Available at: https://sites.google.com/site/istarlanguage/home
  - Lankhorst et al. - Enterprise Architectures at Work (4th Edition)
    - Available at: https://link.springer.com/book/10.1007/978-3-662-53933-0
    - Chapter 5.6
- Modeling tools:
  - piStar: http://www.cin.ufpe.br/~jhcp/pistar/
  - (alternatively) Leaf: https://www.cs.toronto.edu/~amgrubb/leaf-2.0/Tool.html
  - Archi: https://www.archimatetool.com/
  - (alternatively) SAP Signavio: https://academic.signavio.com/p/login

Exercises

Please answer all exercises to demonstrate your skills.

Solutions will be available at 11:45

Exercise 1 – Speedy

Speedy is an international delivery company that needs to refocus on which markets it should operate. Indeed, Speedy lacks a clear understanding on which markets are the most profitable. Thus, to address this issue, Speedy’s top management aims at improving their governance. To achieve this, the top management requires sales reports containing timely information. Thus, the top management decided to build a new reporting system to automatically produce such reports. After inspecting the sales reports, the top management may also need to query the existing ERP system to get detailed sales and HR information.

Prepare an ArchiMate and an I-star model that captures the formalizes Speedy’s new architecture.

- Can you model all the elements and relations described in this exercise with both languages? If not, which elements can be captured only in ArchiMate? Which ones in I-star?

Exercise 2 – IC

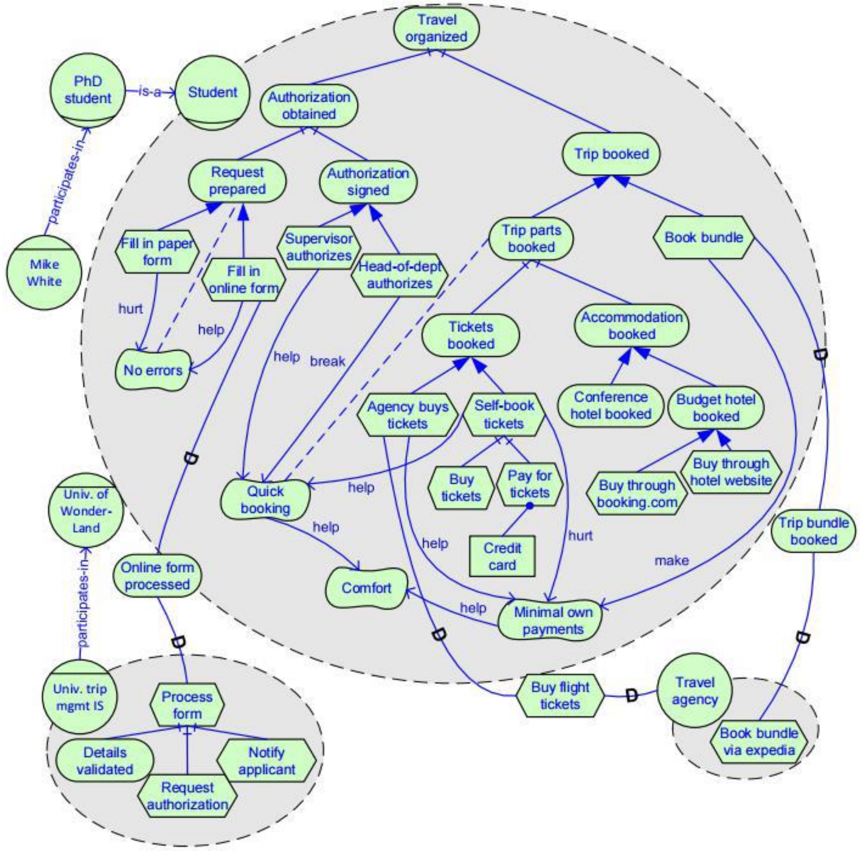
IC is an insurance company which wants to offer a new insurance service for small assets (<2000$) managed completely online for reliable customers.

To achieve this, a customer who wants his assets to be insured has to provide its credentials and photo of the asset and its details (serial number, purchase date) to IC. To ensure that the customer is reliable and the asset inexpensive, IC will then check the customers credentials and past history and estimate the asset’s price.

Prepare an ArchiMate and an I-star model that captures the formalizes IC’s requirements.

- Can you model all the elements and relations described in this exercise with both languages? If not, which elements can be captured only in ArchiMate? Which ones in I-star?

Exercise 3 – University travel reimbursement



Model from: F. Dalpiaz, X. Franch, and J. Horko – iStar 2.0 Language Guide
https://arxiv.org/abs/1605.07767

- This I-star model represents a university travel reimbursement system
- Prepare an ArchiMate model representing the same system
  - – Can you model all the elements and relations in this I-star model?
  - – If not, which elements and relations cannot be modeled?