Specification and Analysis of Information Systems

Lecture 4

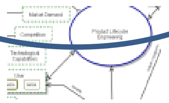
OPM Advanced

Dr. Niva Wengrowicz

Summary of Previous Lectures

System Specification & Analysis

- Specifying the system **function**, **structure** and **behavior**



This is prepared after detailed **communications** with the **project team** and **customer**

Modeling Behavior

System Behavior

- Behavior defines the **dynamic relations** between components (**calls** and **information exchanged**)
- System dynamics deals with system **changes over time**.
- The **dynamic aspects** of a system is the **complement to** the **static aspects**.

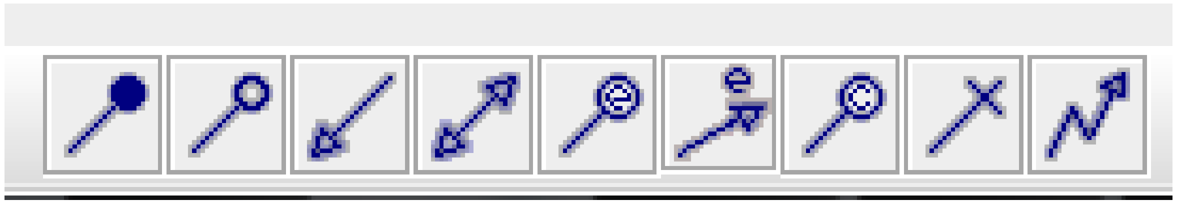
What Do We Want to Capture?

- **What** happens between the components of the system?
  - In standard environments
  - In exceptional situations
- **When** it happens?
  - What are the cause/effect relations
- **How** it happens?
  - How events are ordered
  - How events are related

Modeling Behavior in OPM

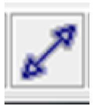
- There is **one diagram type** in OPM which is designed to incorporate structure and behavior in **one coherent frame of reference**.
- OPM **balances** between the **structural** and **procedural** aspects of the system.
- OPM address the **basic principles of the dynamics** aspect of a system - Its behavior and the changes it undergoes over time.
- How OPM can be used to model this aspect?
  - ✓ **procedural links** (Transforming links & Enabling links)
  - **Events** (object related, time related-process invocation links)
  - **Condition**
  - ✓ **Time line**: flow of control (implicit invocation links)

Procedural Links

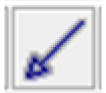


- **Transforming** – A transformee of a process is an object that undergoes a transformation as a result of the occurrence of the process. The transformation can be **construction**, **effect** (change of state) or **consumption**.

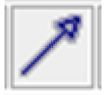
effect



consumption

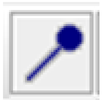


construction

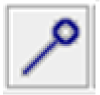


- **Enabling** - An enabler of a process is an object that **must be present** in order for that process to occur, but is **not transformed** as a result of the occurrence of the process.

agent



instrument

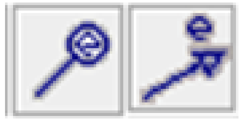


Event link

An Event Is What Triggers a Process

- A **process** **cannot start spontaneously**.
- For a process to start executing, it needs to be **triggered**.
- In other words, it must get a **signal** telling it "start executing!“
- An **event** is a **significant point in time** from the system's perspective.

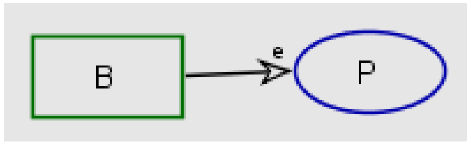
An Event Is What Triggers a Process



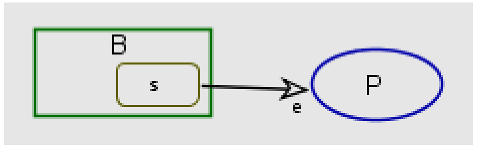
- By definition, an event is a **time-related concept**.
- Hence, **like** processes, **events happen** rather than exist.
- However, **unlike** a process, which takes a non-zero interval of time, an event is a **point in time**; it does not span across a time interval.

The Event Link

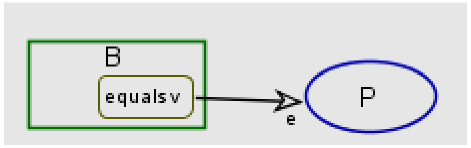
- An **event link** is an abstract **procedural link**
  - from an **object B** to a process **P**
  - from **state s** of **B** to a process **P**
  - from **value v** of **B** to a process **P**



As soon as the object becomes existent, it triggers the process



once an object enters the state, it triggers the process

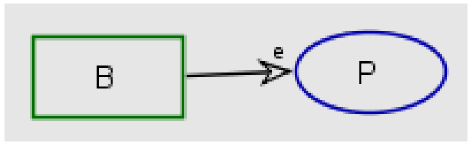


once an object equals the value, it triggers the process

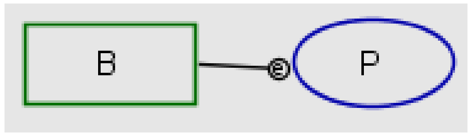
- The event link is denoted by small letter **e**, standing for event.

The Event Link

- There are two types of event links:
  - consumption event link
    Object is consumed
  - enabling event link
    Object remains unchanged



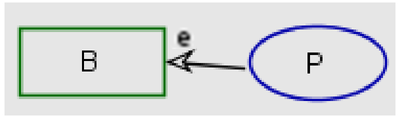
B triggers P.
P consumes B.

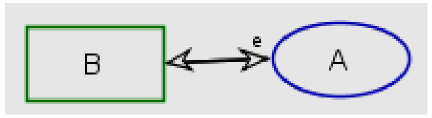


B triggers P.
P requires B.

- They are combinations of event link with a consumption and an enabling links, respectively.

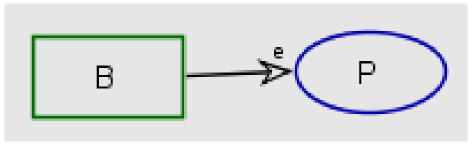
Can Construction or effect link serve as event links too? explain



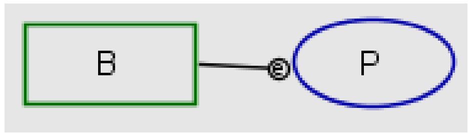


The Event Link

- There are two types of event links:
  - consumption event link
    Object is consumed
  - enabling event link
    Object remains unchanged



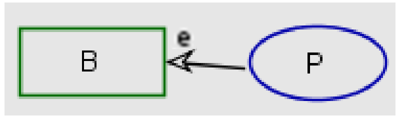
B triggers P.
P consumes B.

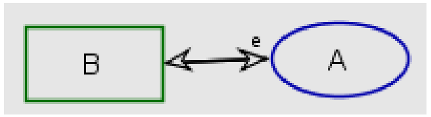


B triggers P.
P requires B.

- They are combinations of event link with a consumption and an enabling links, respectively.

Can Construction or effect link serve as event links too? explain

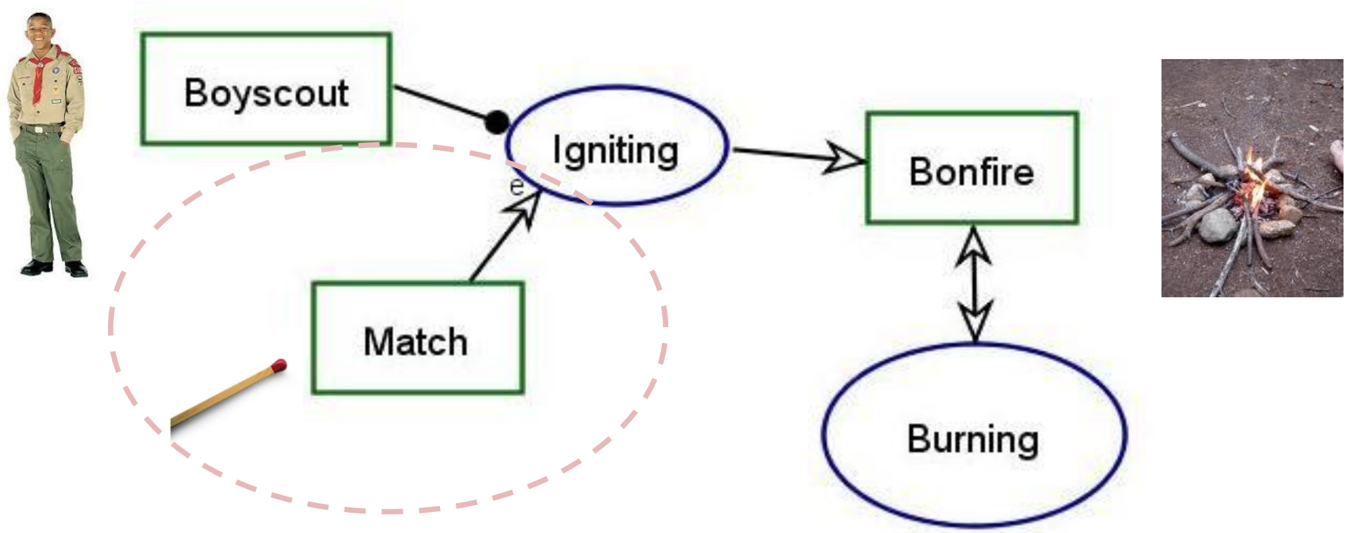




B triggers A.
A consumes B.
A yields B.

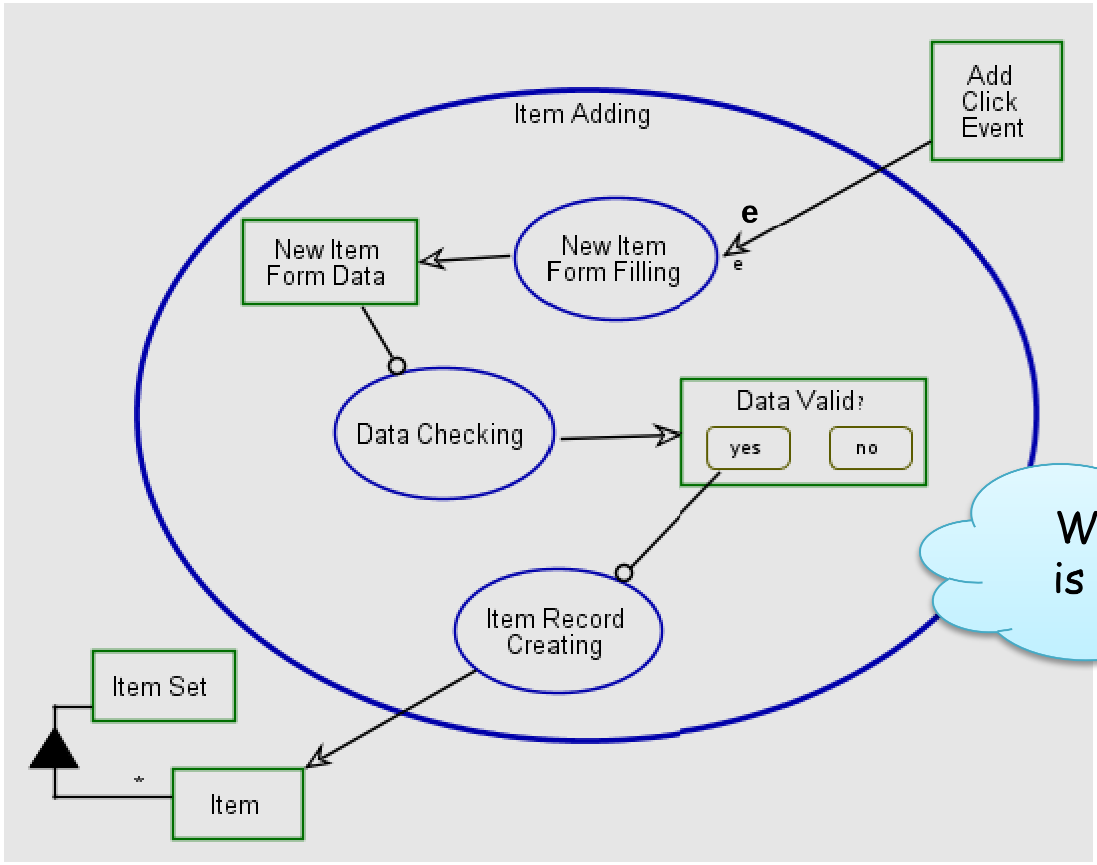
Consumption (Transforming) Event Link Example

- The first type of event link is the **consumption event link**.

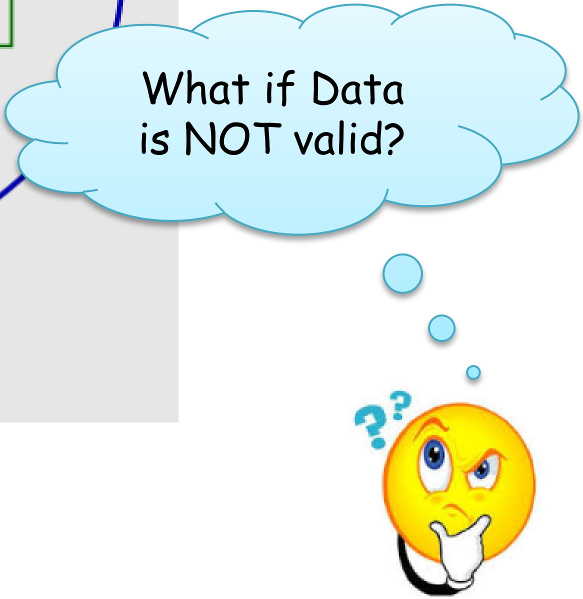


- The presence of **Match** is the event that triggers **Igniting**, which yields **Bonfire**.
- **Match is consumed**, as indicated by the consumption event link from Match to Igniting.

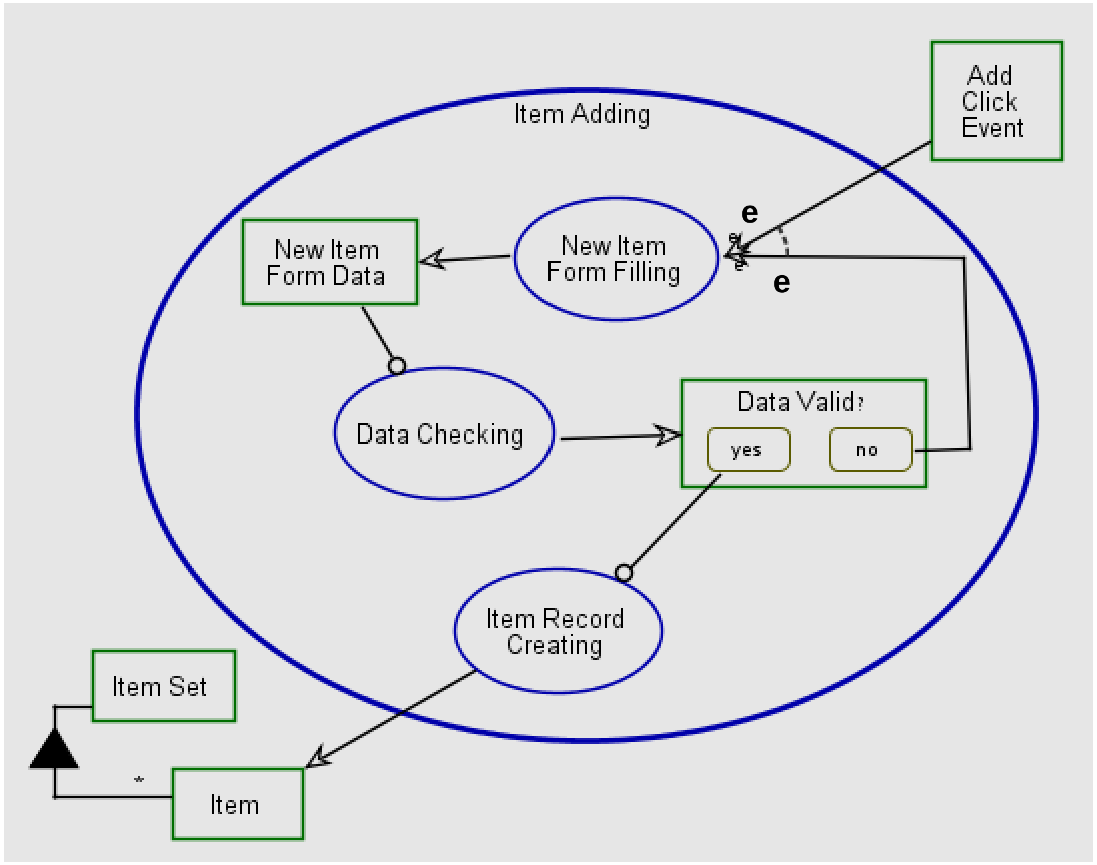
Consumption Event Link Example



Add Click Event triggers New Item Form Filling.



Consumption Event Link Example

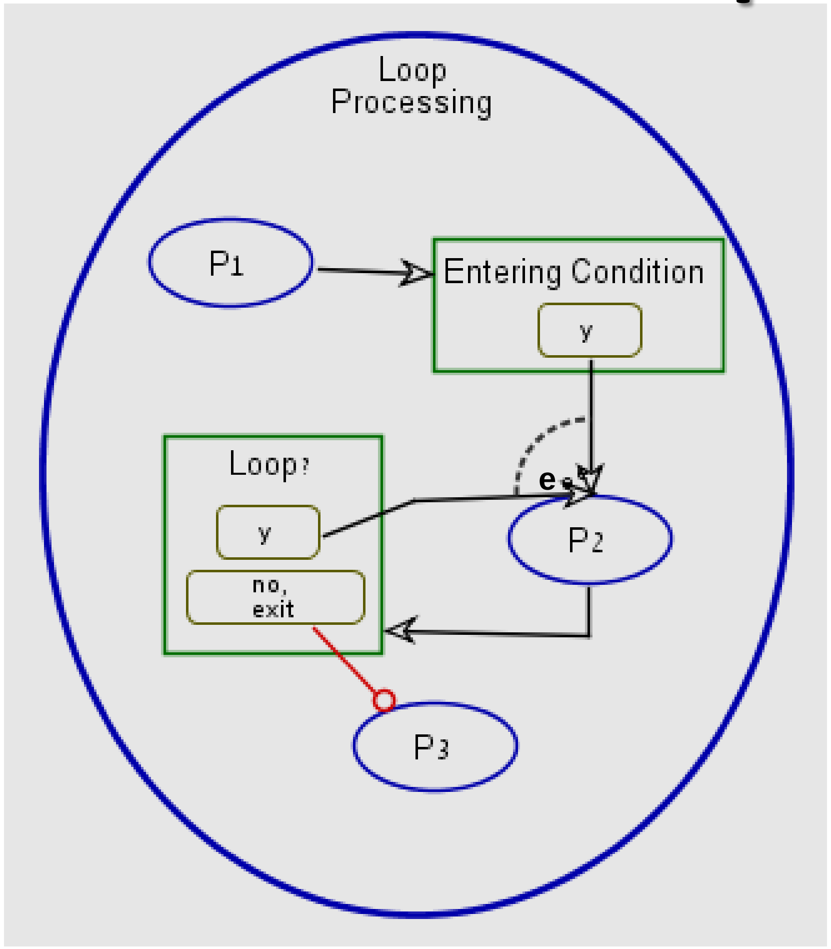


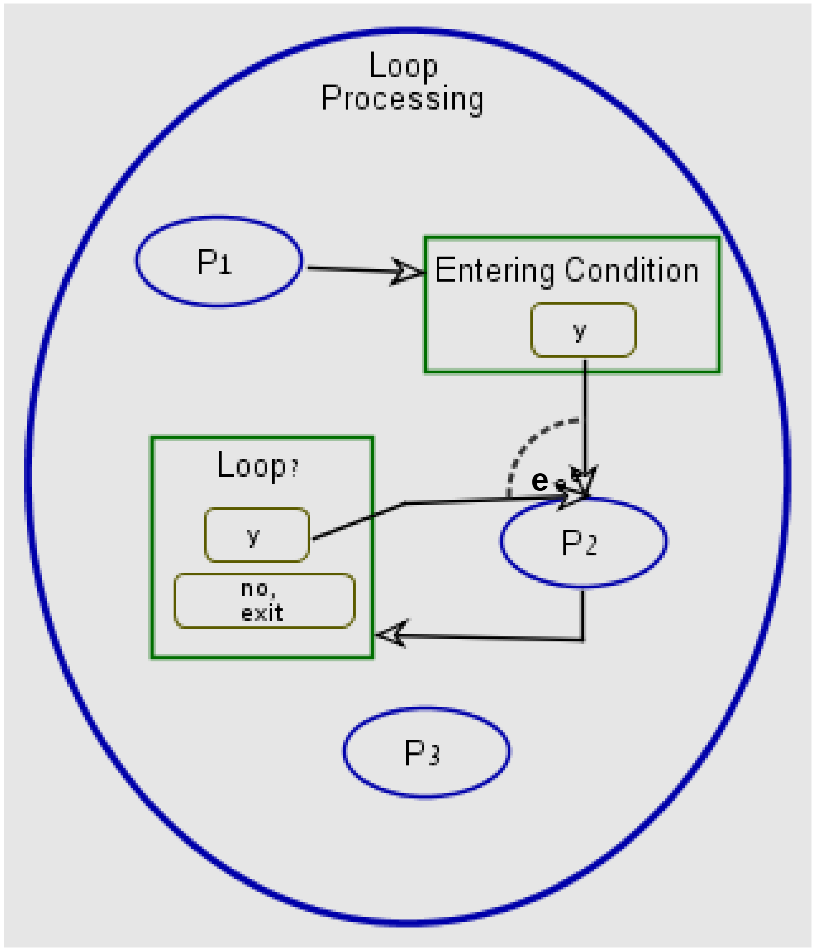
Add Click Event triggers New Item Form Filling.

Data Valid? triggers New Item Form Filling when it enters no.

New Item Form Filling consumes either no Data Valid? or Add Click Event.

Loop flow with consumption event links

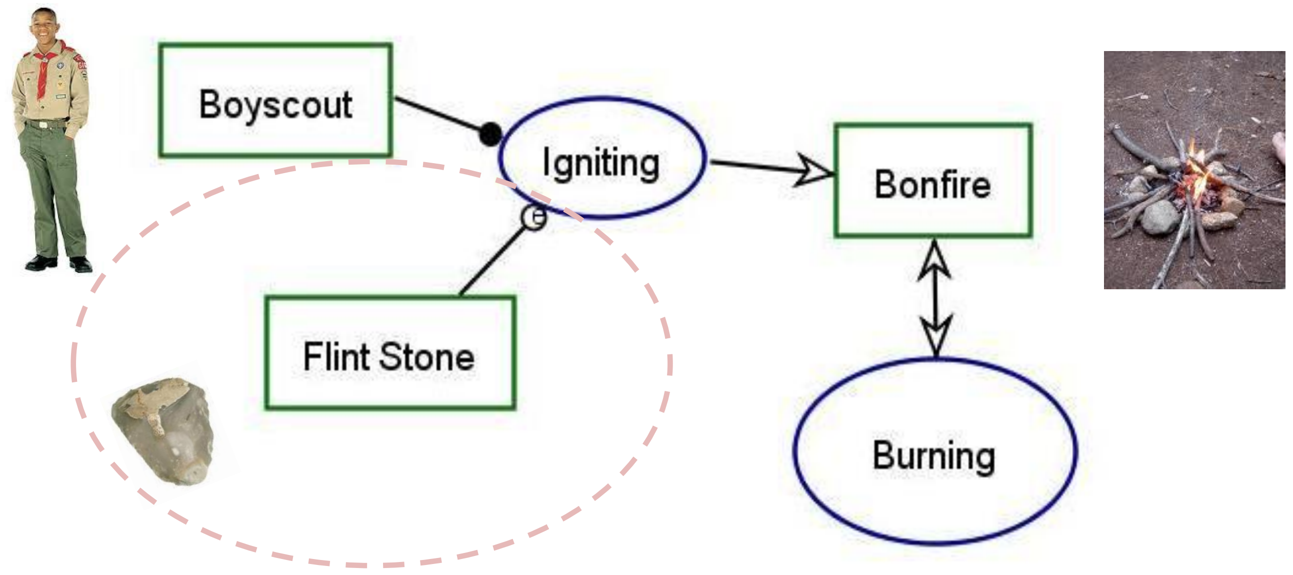




The same effect (Why?)

Instrument (Enabling) Event Link Example

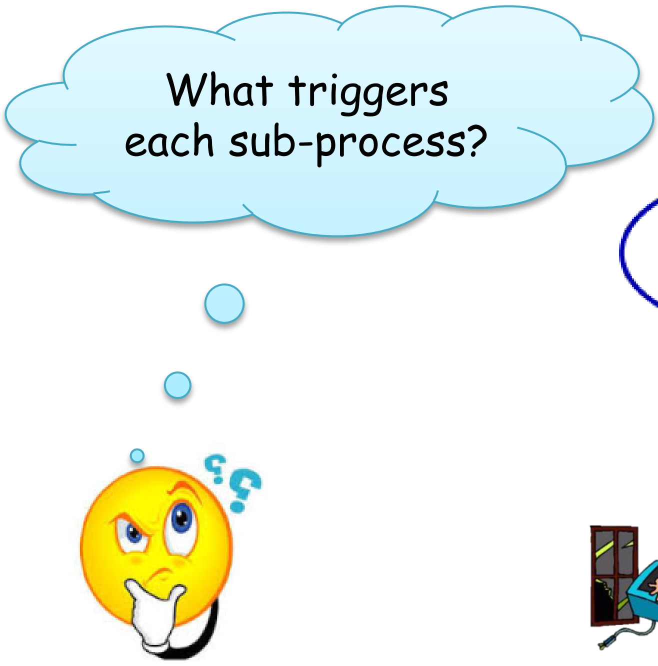
- The second type of event link is the **instrument event link**.
- The presence of **Flint Stone** is the event that triggers **Igniting**, handled by the agent **Boyscout**, which yields **Bonfire**.

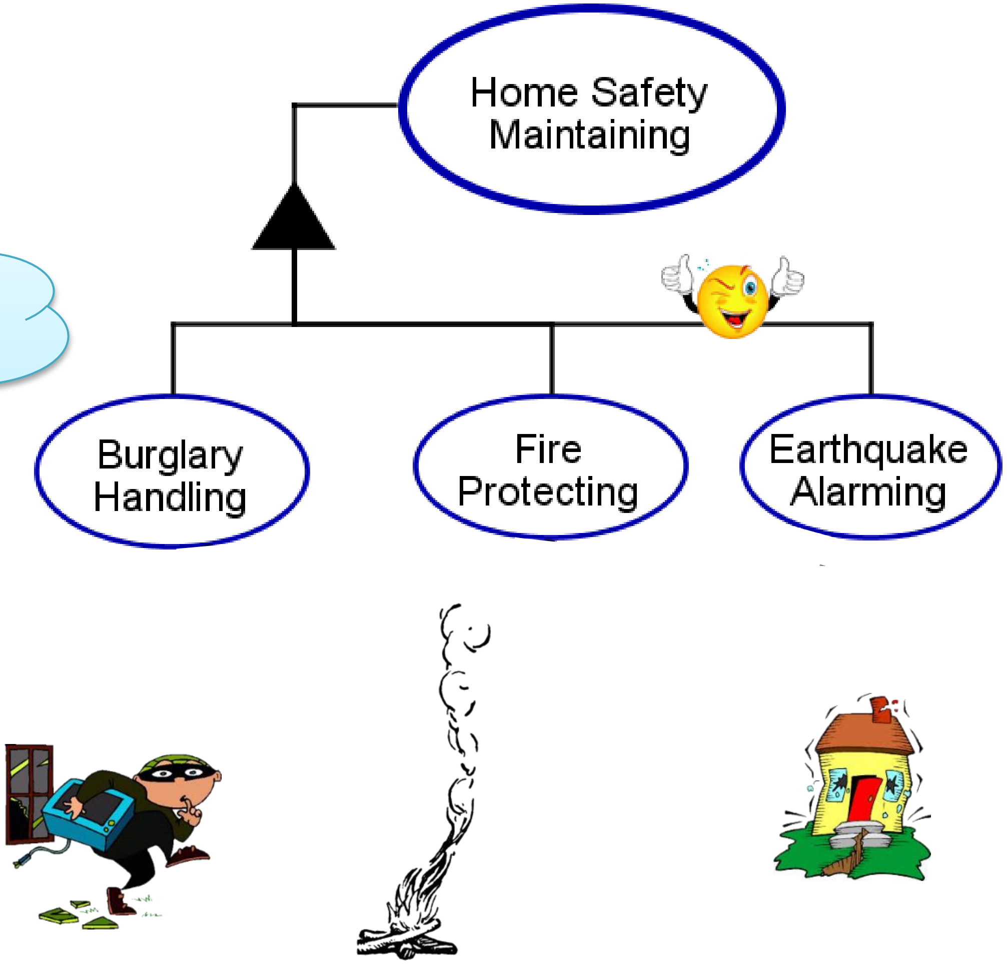


- Being an instrument, **Flint Stone** is not consumed, as indicated by the instrument event link.

Instrument Event Link Example

Asynchronous system
What is the execution order of these system?

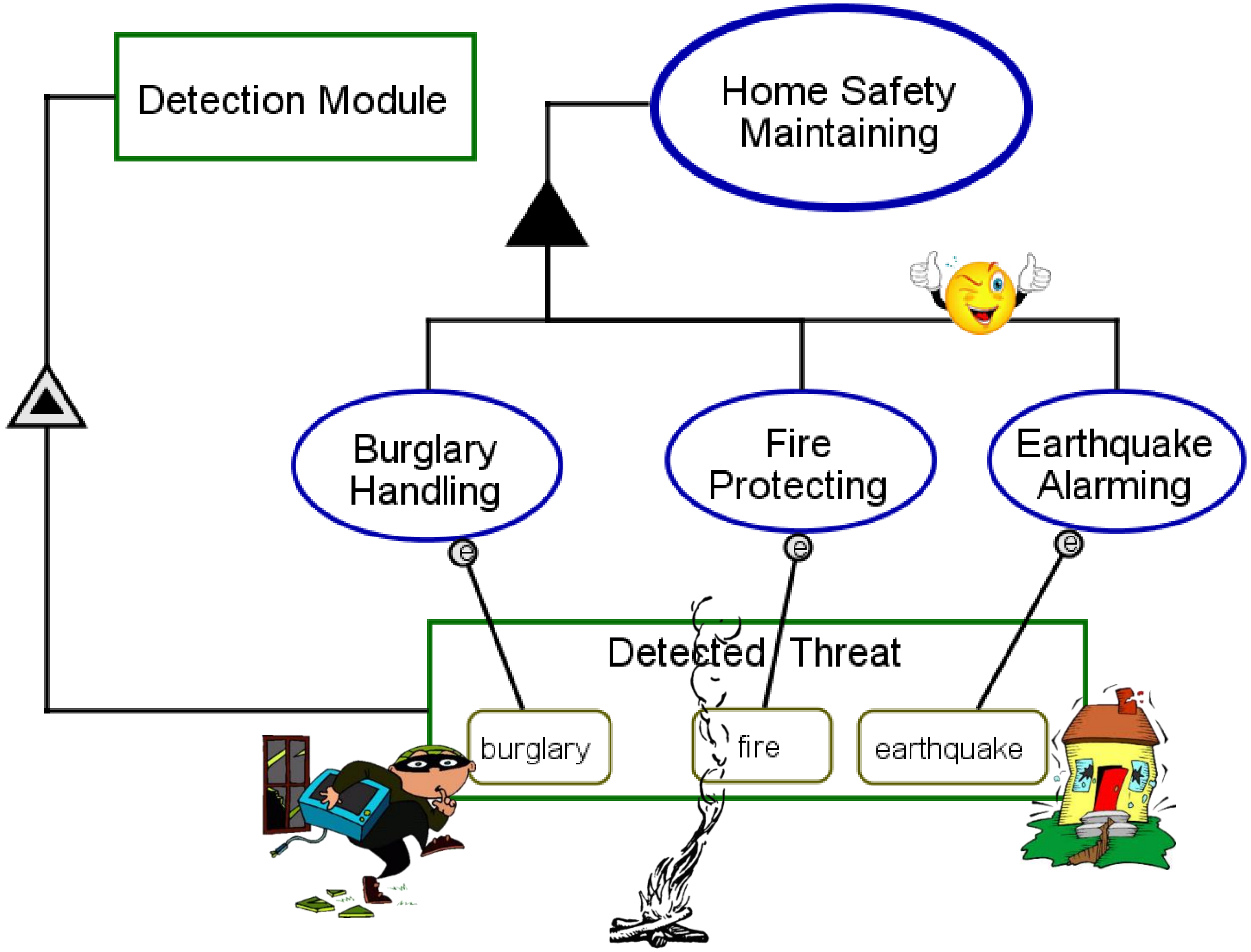




Instrument Event Link Example

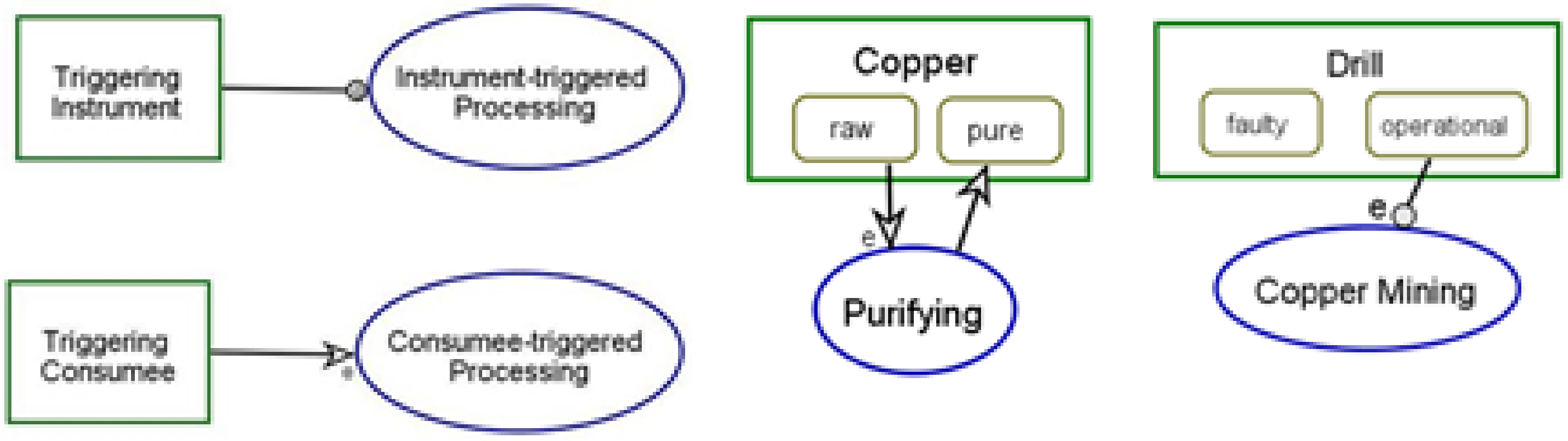
Asynchronous system

What is the execution order of these system?



Object-related Events

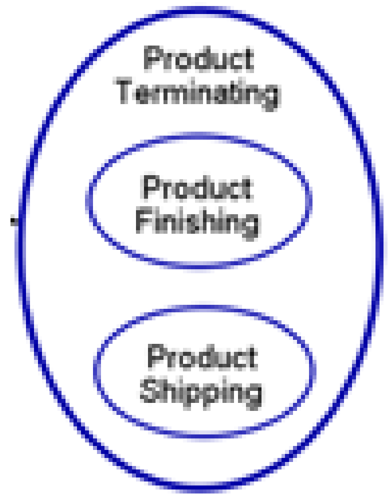
- The events we have seen in examples so far were **object-related events**.
- They happened when:
  - a specific object class became existent or available
  - a certain object entered a specific state or assumed a specific value.



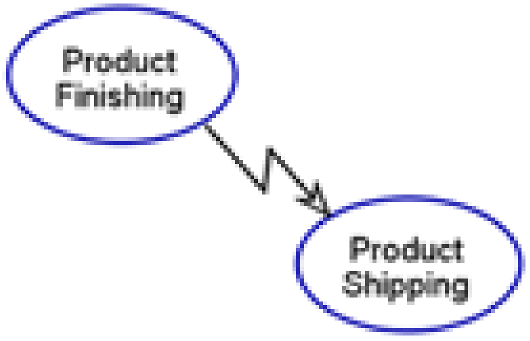
Time-related Events

- Events can also depend on a specific time in the system. These are **time-related events**.

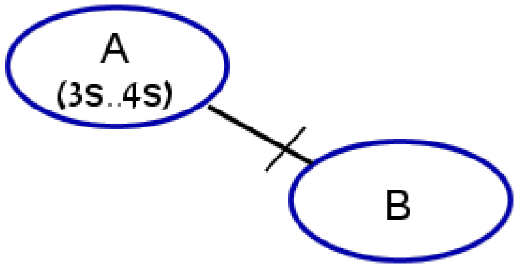
Implicit invocation (time-line)



Process invocation



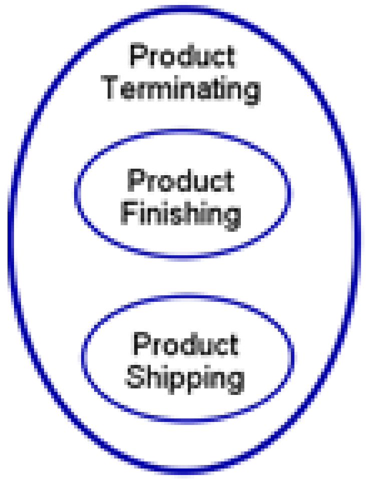
Exception invocation

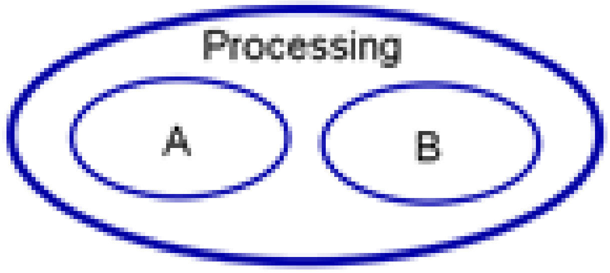


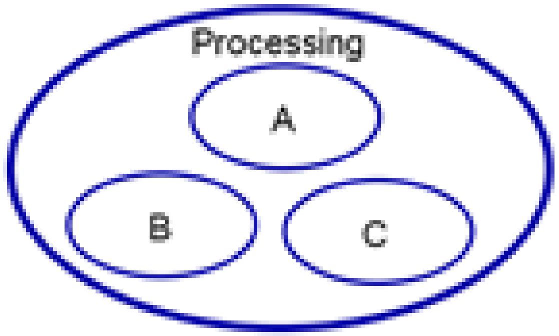
Time-related Events

Implicit Invocation links

- Implicit invocation is invocation of process upon process termination within the context of an in-zoomed process.
- Process **invokes the process**(es) immediately **below it**.







Product Terminating zooms into **Product Finishing** and **Product Shipping,** in that sequence**.**

Processing zooms into **parallel A** and **B.**

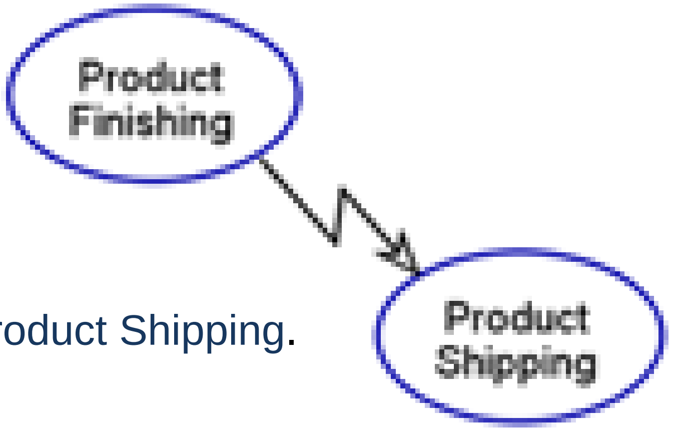
Processing zooms into **A** and parallel **B** and **C,** in that sequence**.**

Time-related Events

Process Invocation links

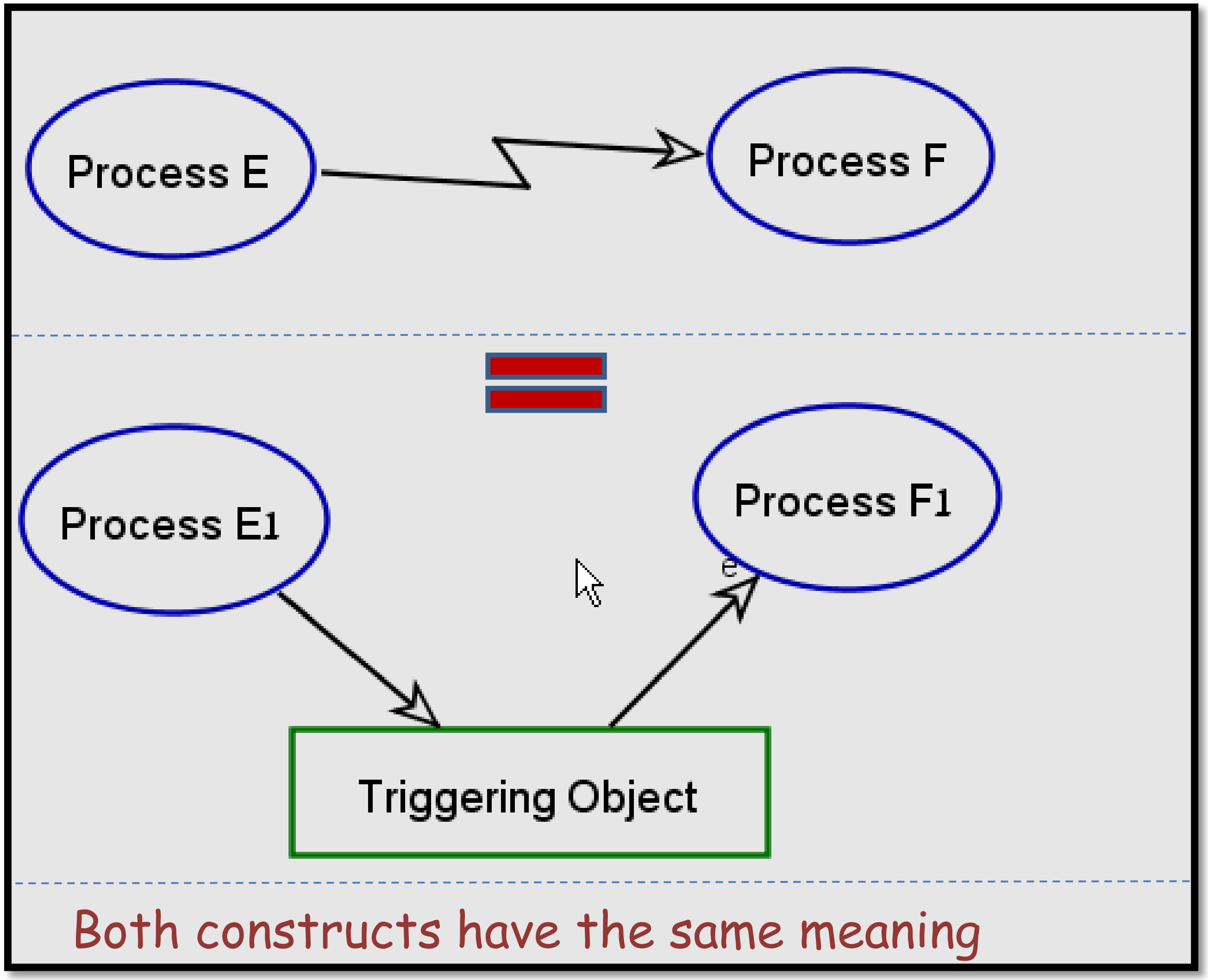
- **Process invocation** is an **event** that triggers a **process** by a **process**.

- An **invocation link** is a link from a **process to the process** that it invokes (triggers), meaning that when the **invoking process terminates**, it **immediately triggers** the process at the other end of the invocation link.



Product Finishing **invokes** Product Shipping.

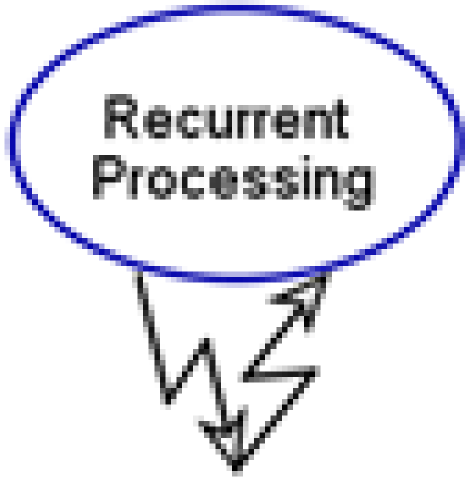
Time-related Events
Process Invocation links



Both constructs have the same meaning

Time-related Events
Self Invocation links

- **Self-invocation** is invocation of a **process by itself**
- When process terminate, the process immediately invokes itself.

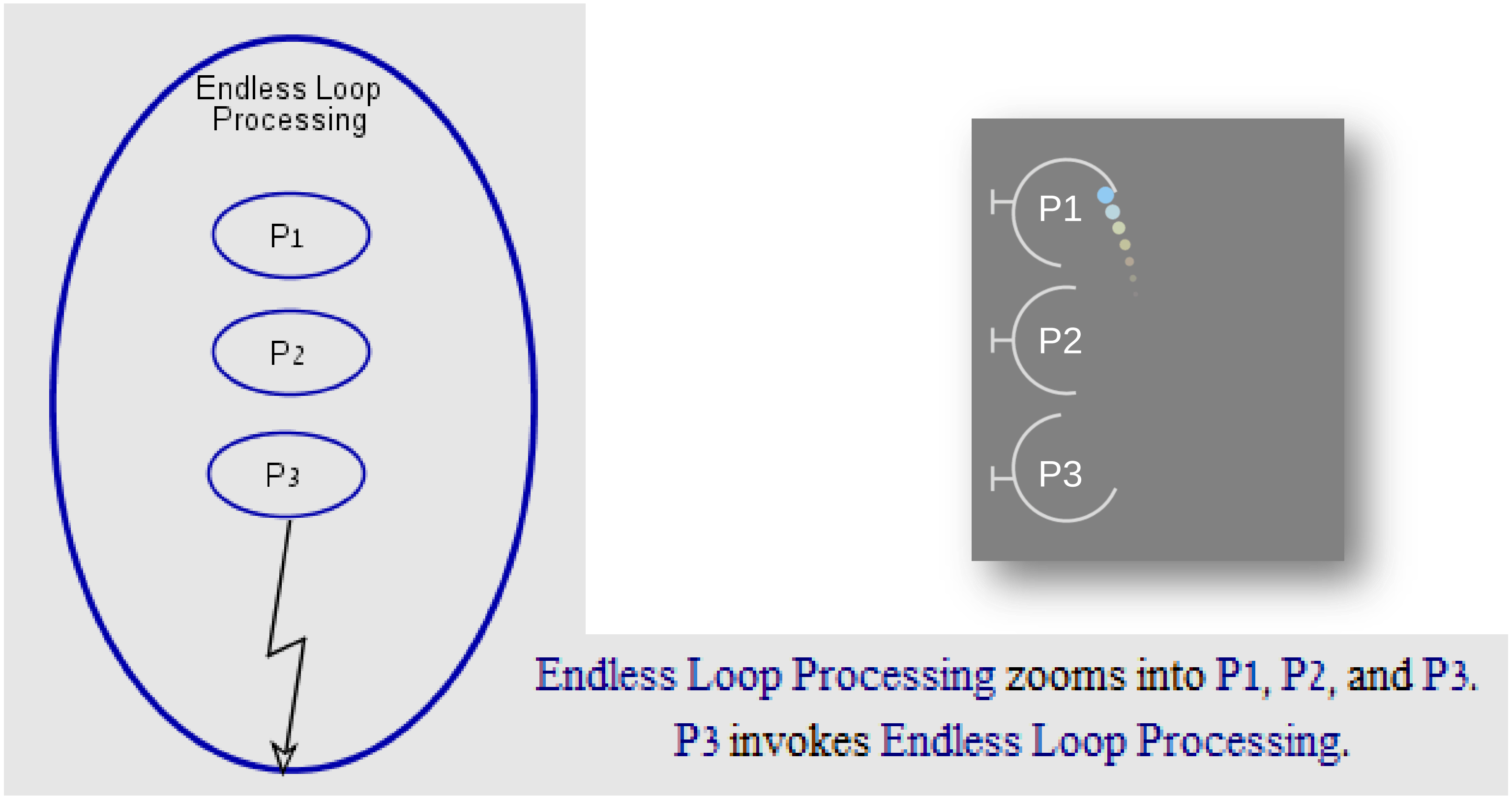


Recurrent Processing **invokes** itself.

Time-related Events

Endless Loop example

- Implicit Invocation links & Process Invocation links
  - Process invokes the process(es) immediately below it
  - when p3 **terminates**, it **immediately triggers** Endless Loop Processing which invokes P1



Endless Loop Processing zooms into P1, P2, and P3.
P3 invokes Endless Loop Processing.

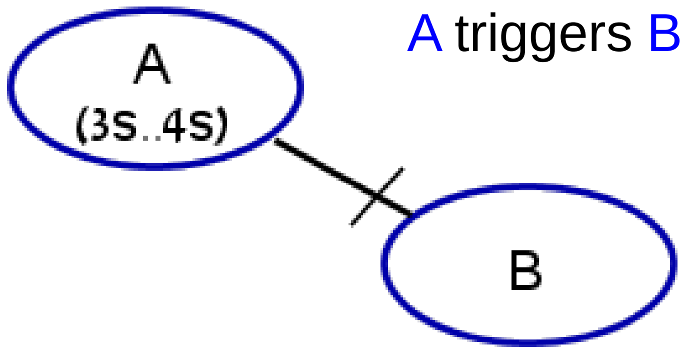
Time-related Events

Exception invocation Link

Process duration constraint

- **Exception invocation** is an **event** that triggers a **process** by timer.

  - The invoker can be **a process** that **has to be assigned with maximal** acceptable **time duration**, which, if exceeded, triggers the invoked process.



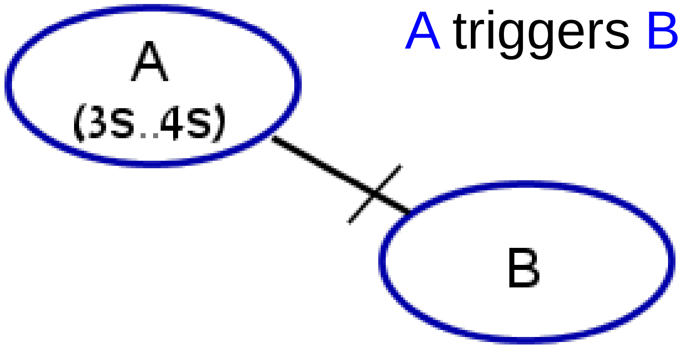
A triggers B when it lasts more than 4 seconds.

Time-related Events

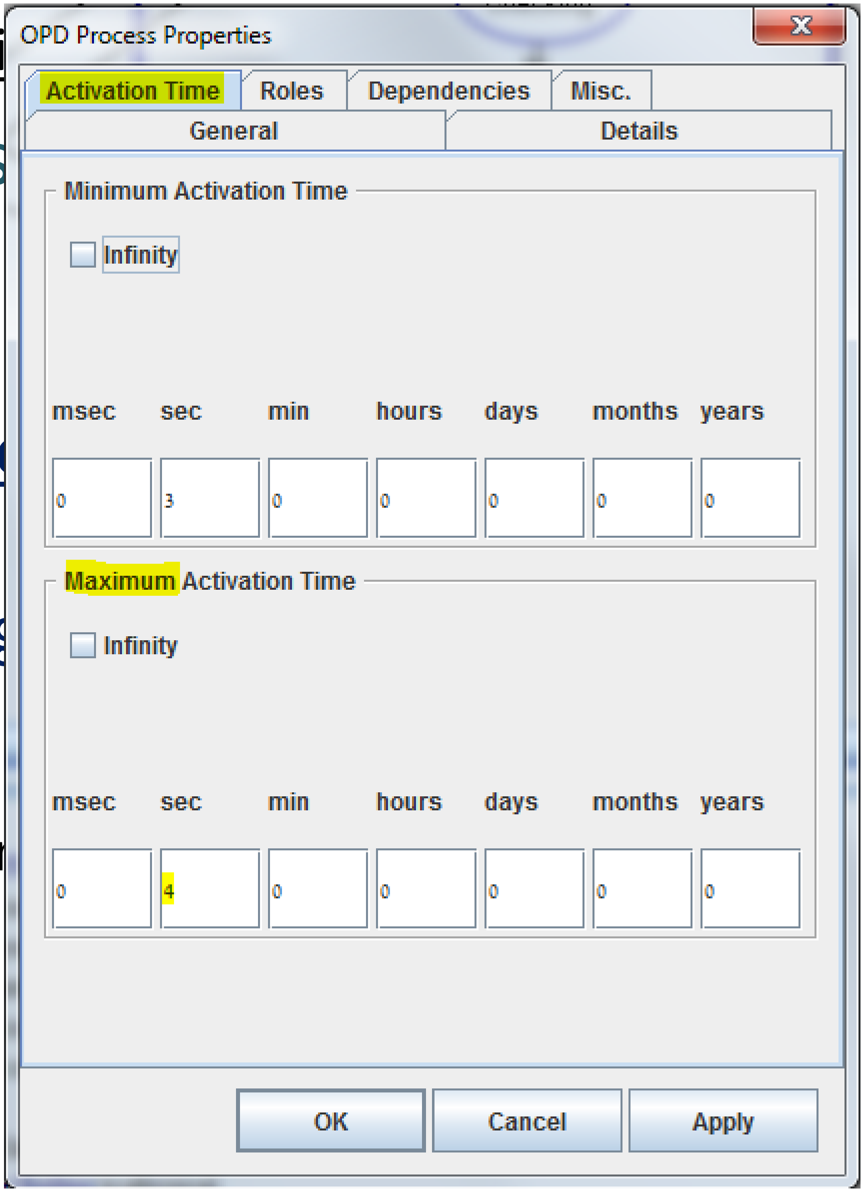
Exception invocation Link

Process durati

- **Exception invocation** is a **process** by timer.
  - The invoker can be **a pr assigned with maximal** which, if exceeded, trigg



A triggers B wher



Minimum Activation Time

☐ Infinity

msec	sec	min	hours	days	months	years
0	3	0	0	0	0	0

Maximum Activation Time

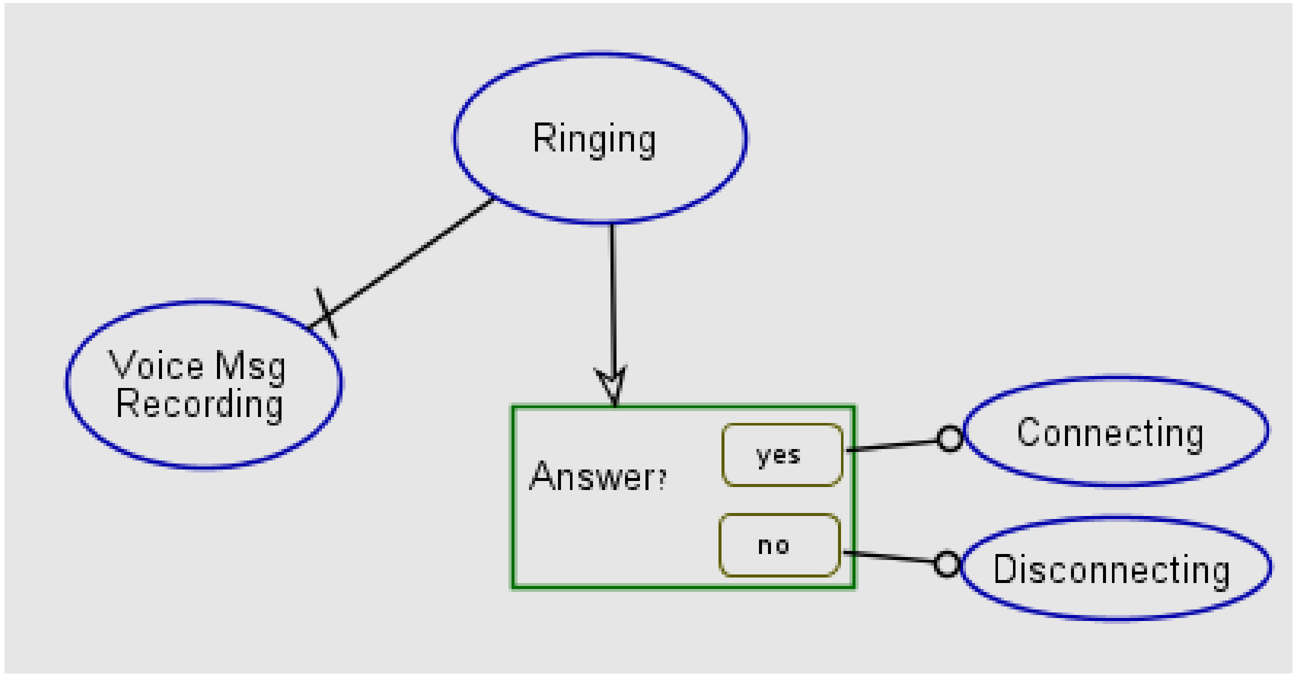
☐ Infinity

msec	sec	min	hours	days	months	years
0	4	0	0	0	0	0

Time-related Events

Exception invocation Link example

Process duration constraint

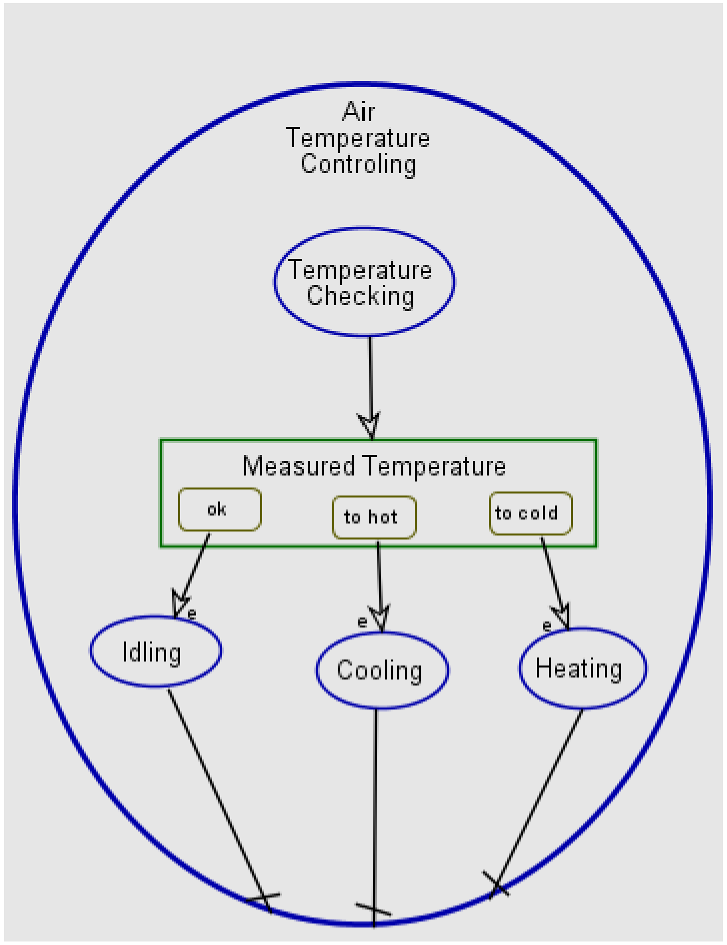


Ringing triggers Voice Msg Recording when it lasts more than 10 seconds.

Time-related Events

Exception invocation Link example

Process duration constraint



Measured Temperature can be to cold, to hot, or ok.

Measured Temperature triggers Heating when it enters to cold.

Measured Temperature triggers Cooling when it enters to hot.

Measured Temperature triggers Idling when it enters ok.

Heating triggers Air Temperature Controling when it lasts more than 1 minute.

Cooling triggers Air Temperature Controling when it lasts more than 1 minute.

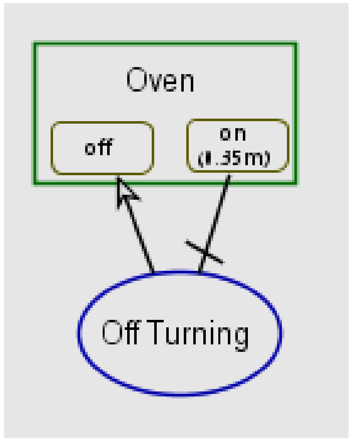
Idling triggers Air Temperature Controling when it lasts more than 10 seconds.

Time-related Events

Exception invocation Link

State duration constraint

- **Exception invocation** is an **event** that triggers a **process** by timer.
  - The invoker can be **an object state** that **last more than maximal** acceptable **time duration**, which, if exceeded, triggers the invoked process.



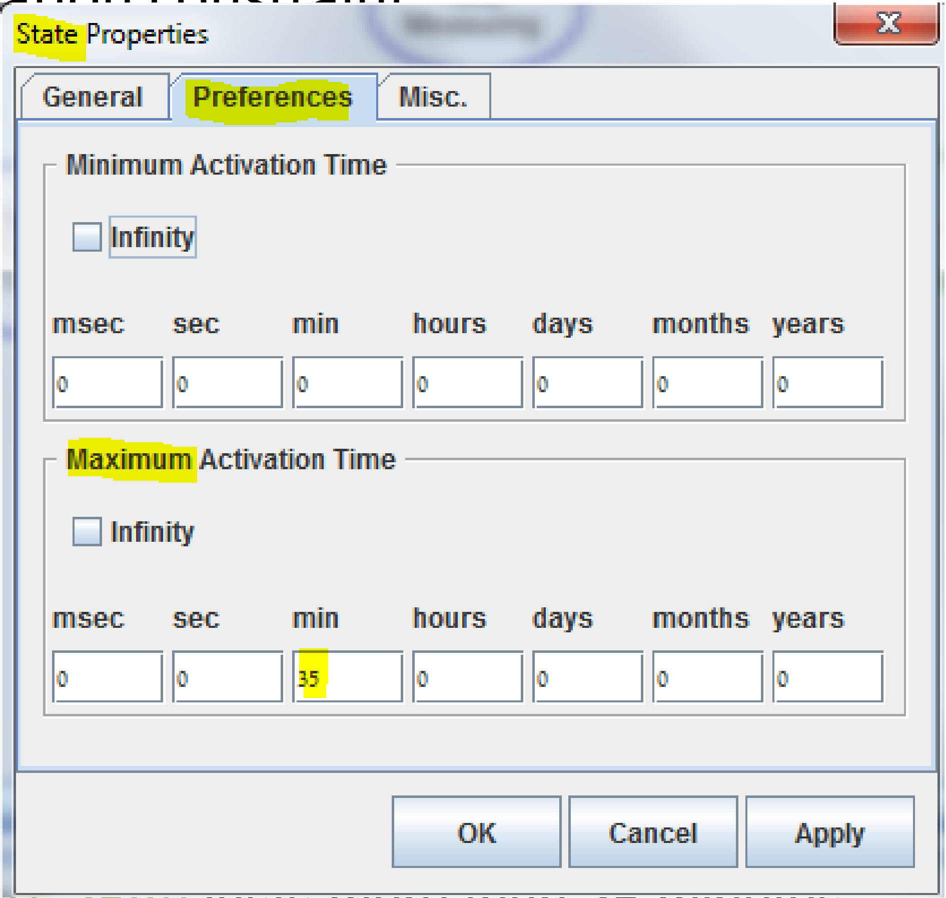
Oven triggers Off Turning
when on (0..35m) lasts more than 35 minutes.

Time-related Events

Exception invocation Link

State duration constraint

- **Exception invocatio**
  a **process** by timer.
  - The invoker can be **a**
    than maximal accep
    exceeded, triggers th



State Properties

General | Preferences | Misc.

Minimum Activation Time

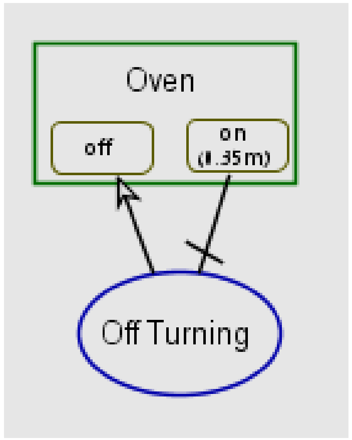
☐ Infinity

msec	sec	min	hours	days	months	years
0	0	0	0	0	0	0

Maximum Activation Time

☐ Infinity

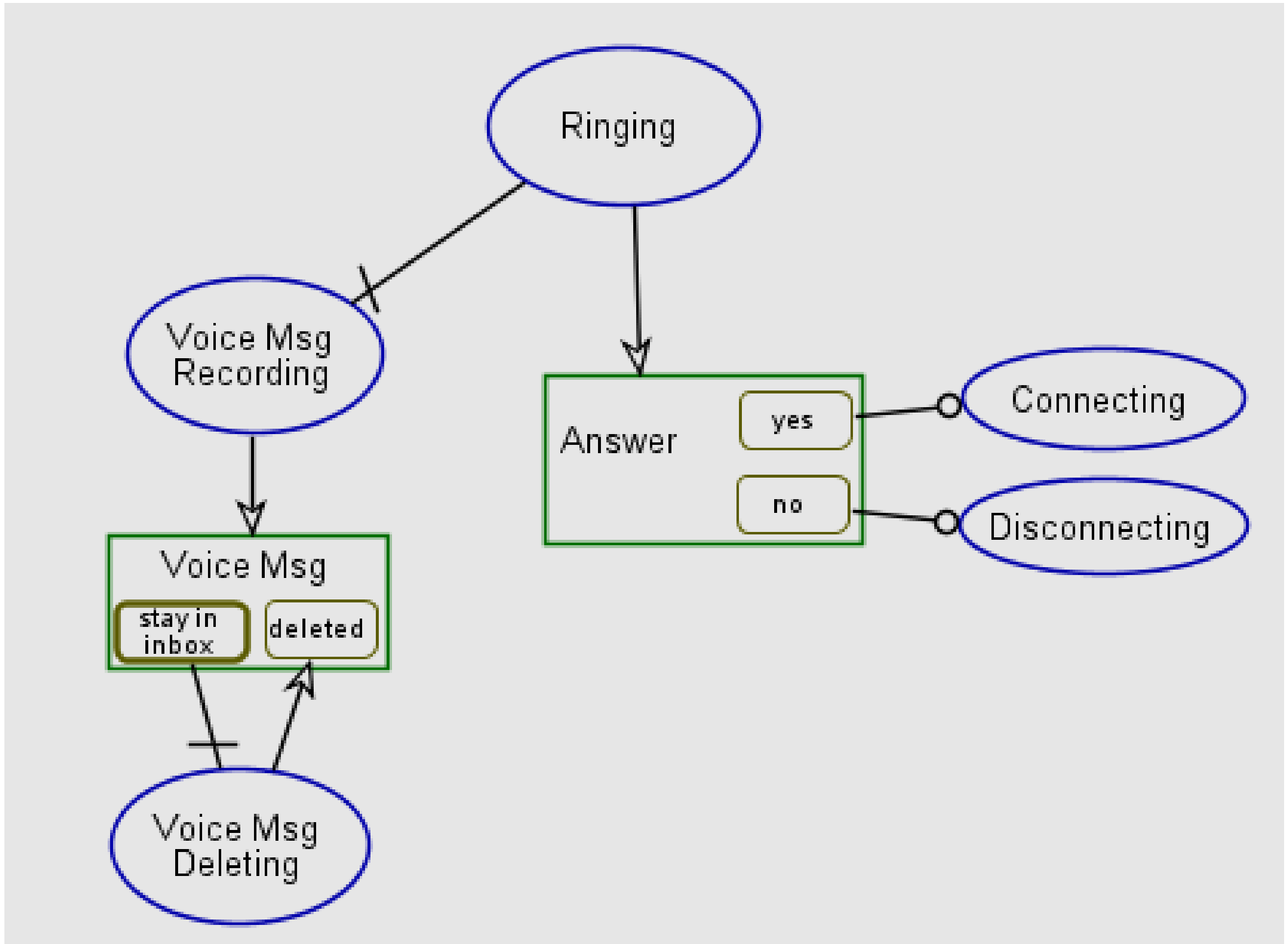
msec	sec	min	hours	days	months	years
0	0	35	0	0	0	0



Oven trig
when on (0..35m) lasts more than 35 minutes.

Time-related Events
Exception invocation Link example

State duration constraint

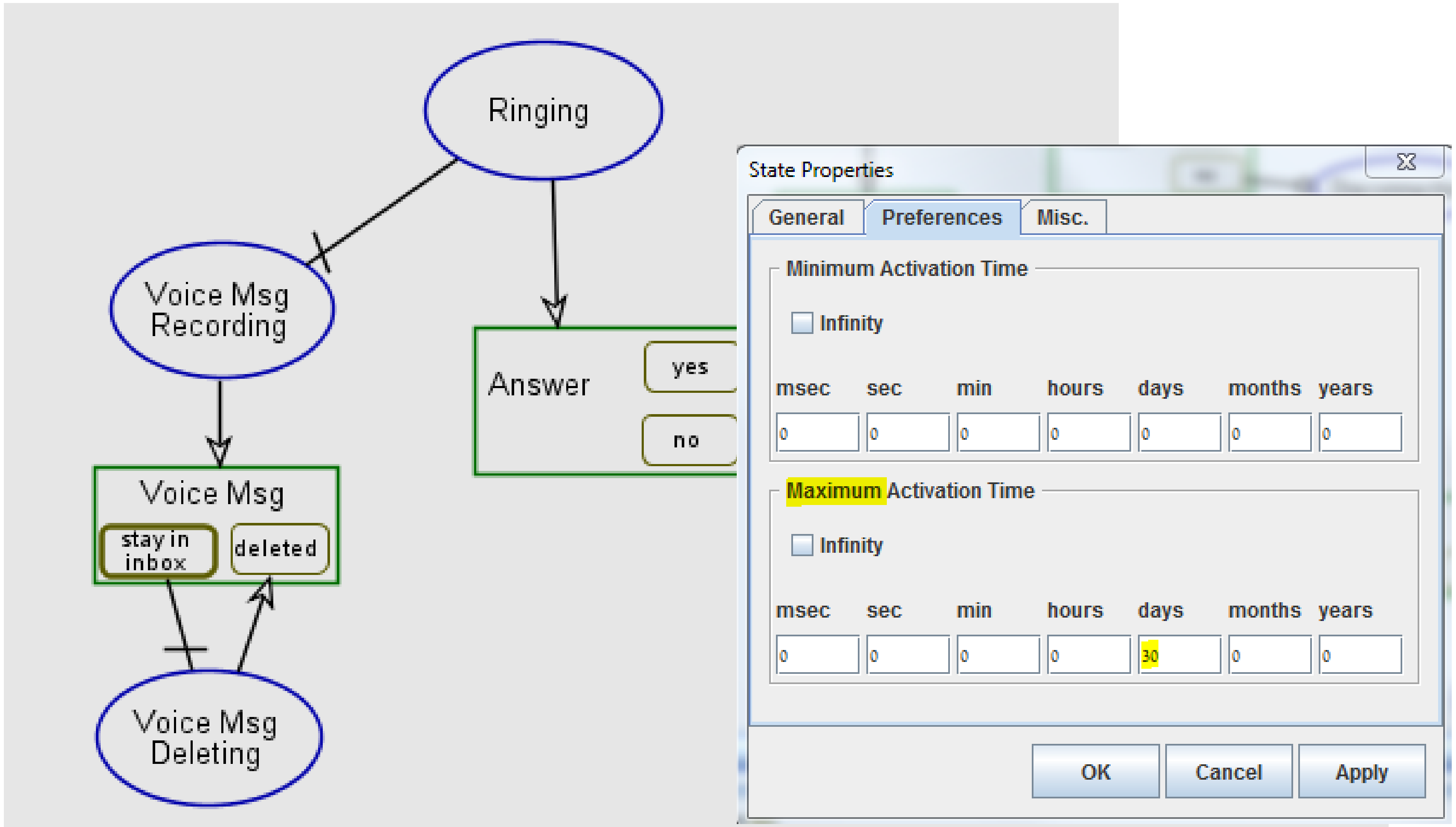


Voice Msg triggers Voice Msg Deleting when stay in inbox lasts more than 30 days.

Time-related Events

Exception invocation Link example

State duration constraint



Minimum Activation Time

- [ ] Infinity

msec	sec	min	hours	days	months	years
0	0	0	0	0	0	0

Maximum Activation Time

- [ ] Infinity

msec	sec	min	hours	days	months	years
0	0	0	0	30	0	0

Voice Msg triggers Voice Msg Deleting when stay in inbox lasts more than 30 days.

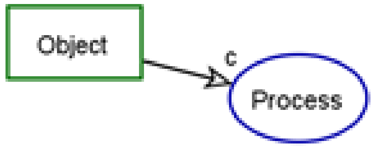
Condition link

Condition Link

- Enables conditional execution of a **process**:
  - **If** the object **exist** or **is at the state** from which the condition link originates, **then** **execution** of the target process is **attempted**.
  - **If** the object does **not exist** or is **not in the state** linked to the condition link, **then** the **process is simply skipped**.
- Semantically, the condition link is similar to an **"if…then"** command.

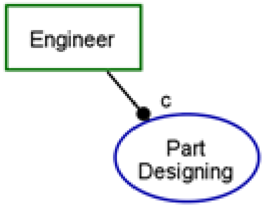
Condition Link

- There are two types of condition links:
  - Condition **transforming** links
    Object is changed



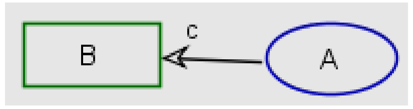
Process occurs if **Object** exists**,** in which case **Process** consumes **Object,** otherwise **Process** is skipped**.**

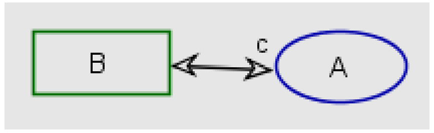
  - Condition **enabling** links
    Object remains unchanged



Engineer handles **Part Designing** if **Engineer** is present**,** otherwise **Part Designing** is skipped**.**

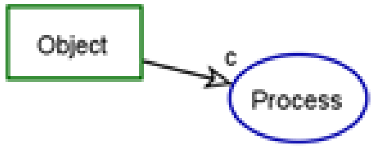
Can Construction or effect link serve as event condition links too? explain



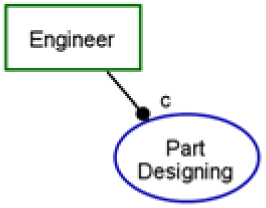


Condition Link

- There are two types of condition links:
  - Condition **transforming** links
    Object is changed
  - Condition **enabling** links
    Object remains unchanged

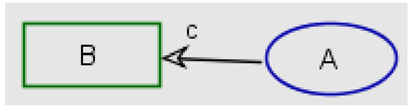


Process occurs if **Object** exists**,** in which case **Process** consumes **Object,** otherwise **Process** is skipped**.**

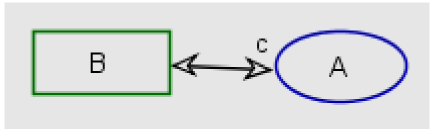


Engineer handles **Part Designing** if **Engineer** is present**,** otherwise **Part Designing** is skipped**.**

Can Construction or effect link serve as event condition links too? explain



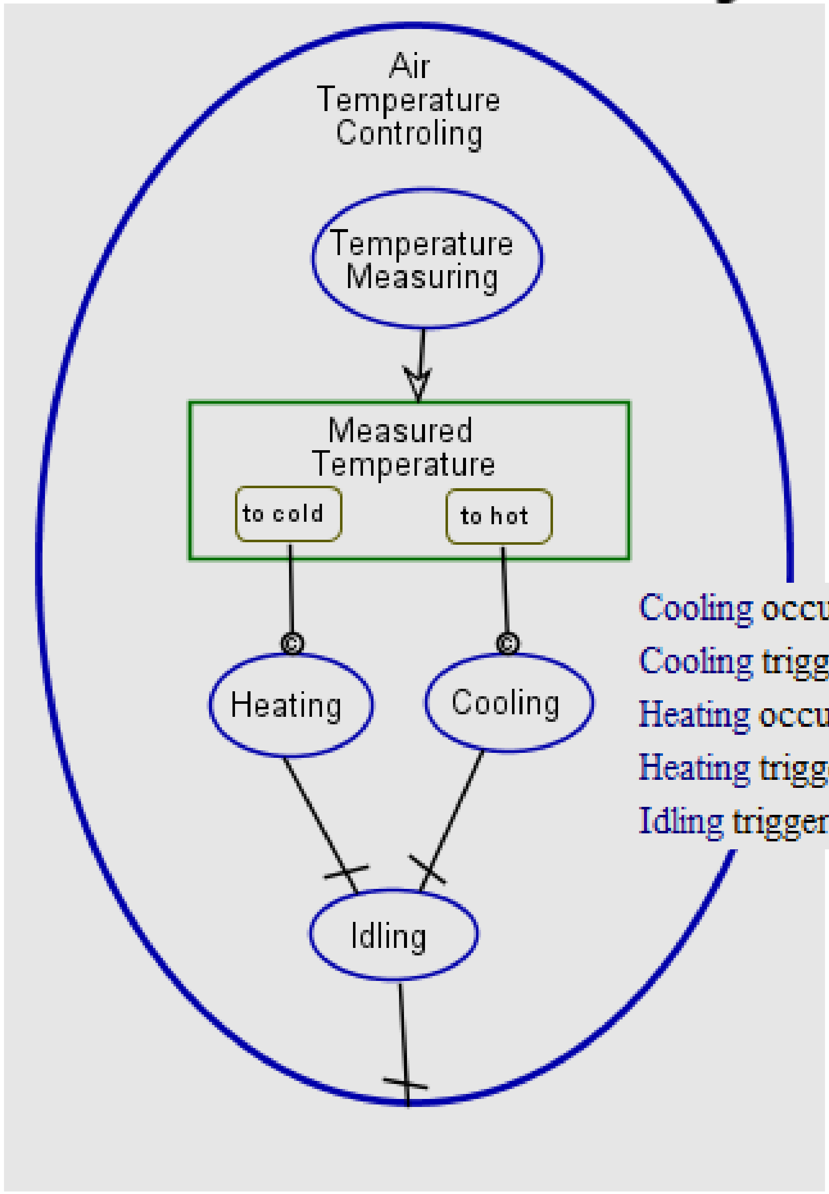
✗



A occurs if **B** exists**,** in which case **A** affects **B,** otherwise **A** is skipped**.**

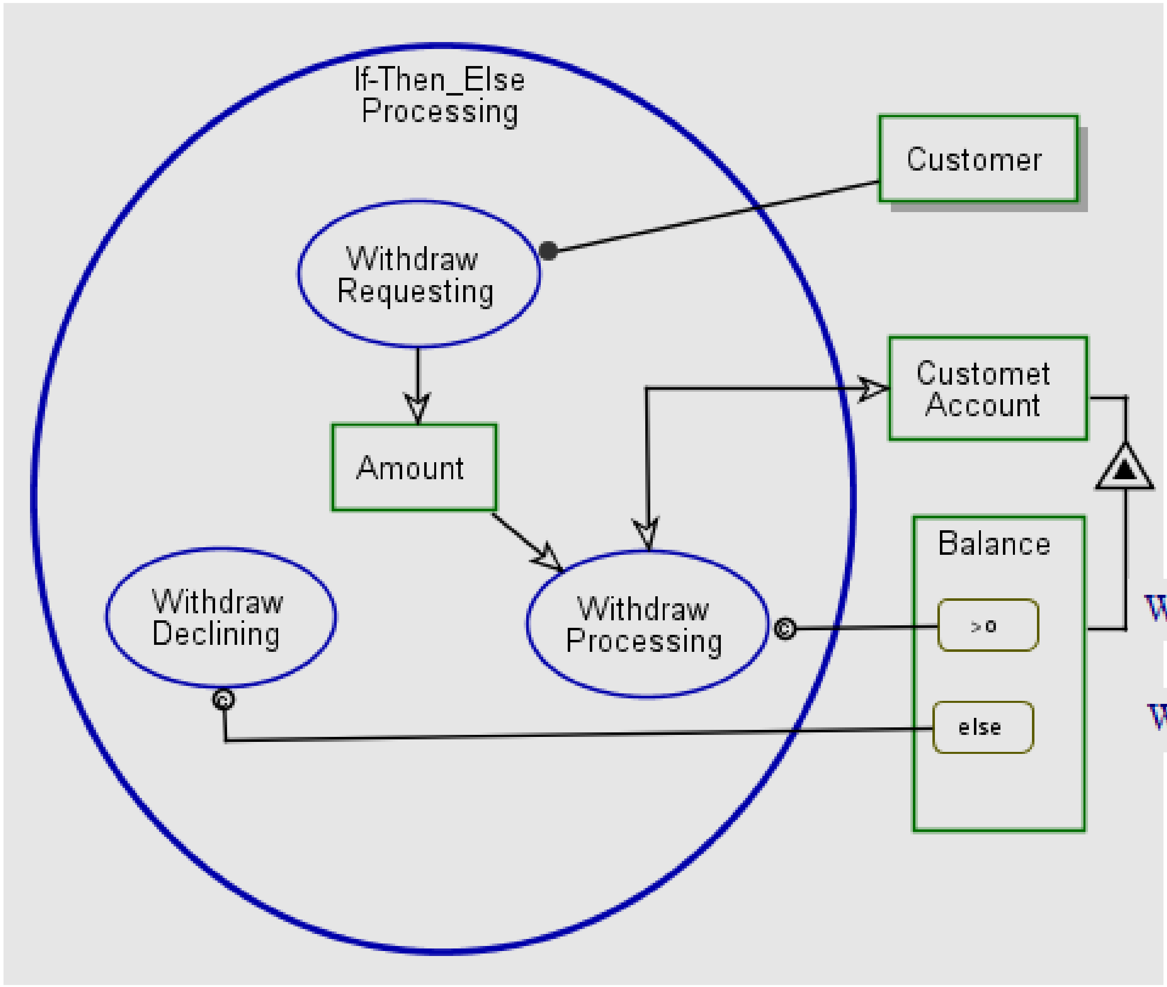
✓

Condition Link example



Cooling occurs if Measured Temperature is to hot.
Cooling triggers Idling when it lasts more than 1 minute.
Heating occurs if Measured Temperature is to cold.
Heating triggers Idling when it lasts more than 1 minute.
Idling triggers Air Temperature Controling when it lasts more than 10 seconds.

“If then Else” Flow Example



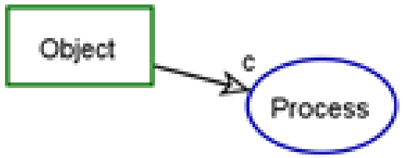
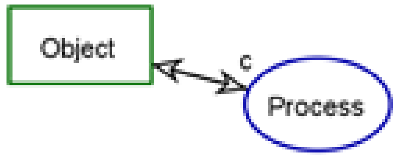
Withdraw Processing occurs if Balance is >0.

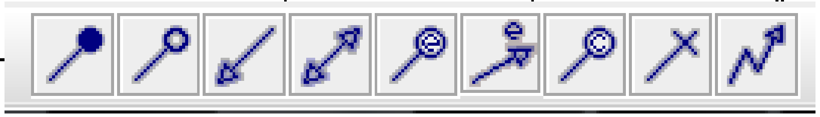
Withdraw Declining occurs if Balance is else.

Condition $\overset{c}{\nearrow}$ Link

Permitted but not available in OPCAT

Table 10 —Condition transforming link summary

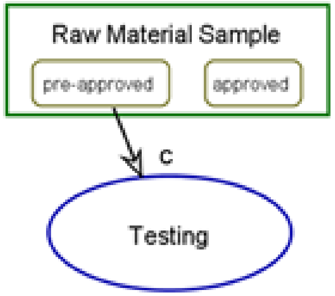
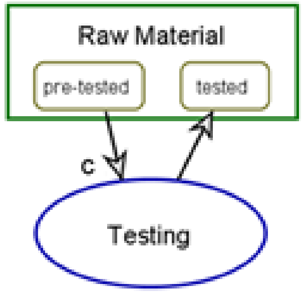
Name	Semantics	Sample OPD & OPL	Source	Destination
Condition consumption link	If an object instance exists and the rest of the process precondition is satisfied, then the process executes and consumes the object instance, otherwise the control advances to trigger the next sequential process or returns one level up.	 **Process** occurs if **Object** exists, in which case **Process** consumes **Object**, otherwise **Process** is skipped.	Conditioning object	Conditioned process
Condition effect link	If an object instance exists and the rest of the process precondition is satisfied, then the process executes and affects the object instance, otherwise the control advances to trigger the next sequential process or returns one level up.	 **Process** occurs if **Object** exists, in which case **Process** affects **Object**, otherwise **Process** is skipped.	Conditioning object	Conditioned process



Condition $\overset{c}{\nearrow}$ Link

Permitted but not available in OPCAT

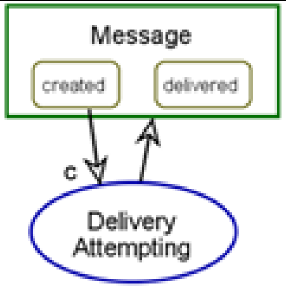
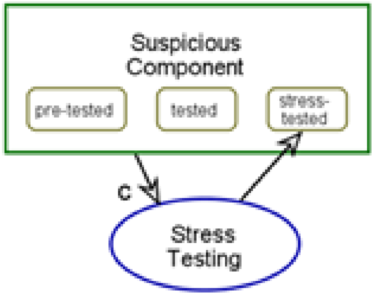
Table 12 — Condition state-specified transforming link summary

Name	Semantics	Sample OPD & OPL	Source	Destination
Condition state-specified consumption link	The process executes if the object is in the state from which the link originates, otherwise the process is skipped.	 Raw Material Sample pre-approved approved c Testing **Testing** occurs if **Raw Material Sample** is **pre-approved,** in which case **Raw Material Sample** is consumed, otherwise **Testing** is skipped.	Conditioning specified state of the object	Conditioned process
Condition input-output-specified effect link	The process executes if the object is in the input state (from which the link originates) and changes the object from its input state to its output state, otherwise the process is skipped.	 **Testing** occurs if **Raw Material** is **pre-tested,** in which case **Testing** changes **Raw Material** from **pre-tested** to **tested,** otherwise **Testing** is skipped.	Conditioning specified input state of the object	Triggered process

Condition $\overset{c}{\nearrow}$ Link

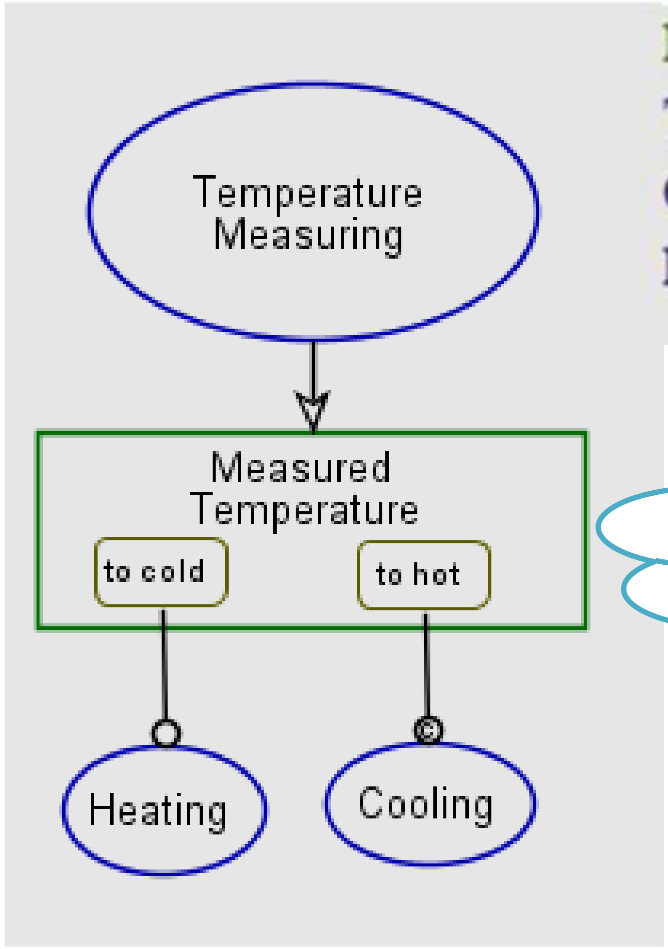
Permitted but not available in OPCAT

Table 12 — Condition state-specified transforming link summary

Name	Semantics	Sample OPD & OPL	Source	Destination
Condition input-specified effect link	The process executes if the object is in the input state (from which the link originates) and changes the object from its input state to any one of its states, otherwise the process is skipped.	 **Delivery Attempting** occurs if **Message** is **created**, in which case **Delivery Attempting** changes **Message** from **created,** otherwise **Delivery Attempting** is **skipped.**	Conditioning specified input state of the object	Triggered process
Condition output-specified effect link	The process executes if the object is in the input state (from which the link originates) and changes the object from its input state to any one of its states, otherwise the process is skipped.	 Suspicious Component pre-tested tested stress-tested c Stress Testing **Stress Testing** occurs if **Suspicious Component** exists, in which case **Stress Testing** changes **Suspicious Component** to **stress-tested,** otherwise **Stress Testing** is skipped.	Conditioning object	Triggered process

Skip vs Wait semantics

Condition Links vs. Non-condition Links



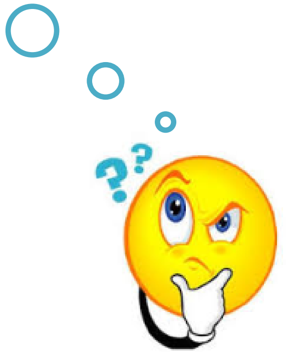
Measured Temperature can be to cold or to hot.

Temperature Measuring yields Measured Temperature.

Cooling occurs if Measured Temperature is to hot.

Heating requires to cold Measured Temperature.

What is the difference between Heating and Cooling In the diagram?



Skip vs Wait semantics
Condition Links vs. Non-condition Links

- **Condition** and **non-condition** links enables process execution only if the **object** instance from which the link originates **exists**.

  Similar If the precondition is true- execute the process

- The **difference** between the two link kinds is with respect to process execution in case of a **negative precondition evaluation**:

  - **condition** link - If a precondition evaluation fails, the process is **skipped**

    If the precondition is true, execute the process
    else skip

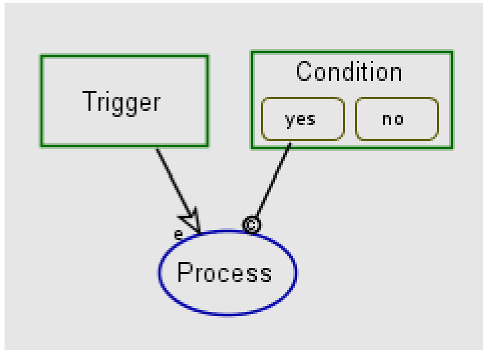
  - **Non-condition** - if a precondition evaluation fails, the process **wait**.

    If the precondition is true, execute the process
    else wait until the precondition becomes true*

* For this to happen, a new event must trigger the process again, causing the precondition evaluation to repeat.

Advanced event-condition Issues

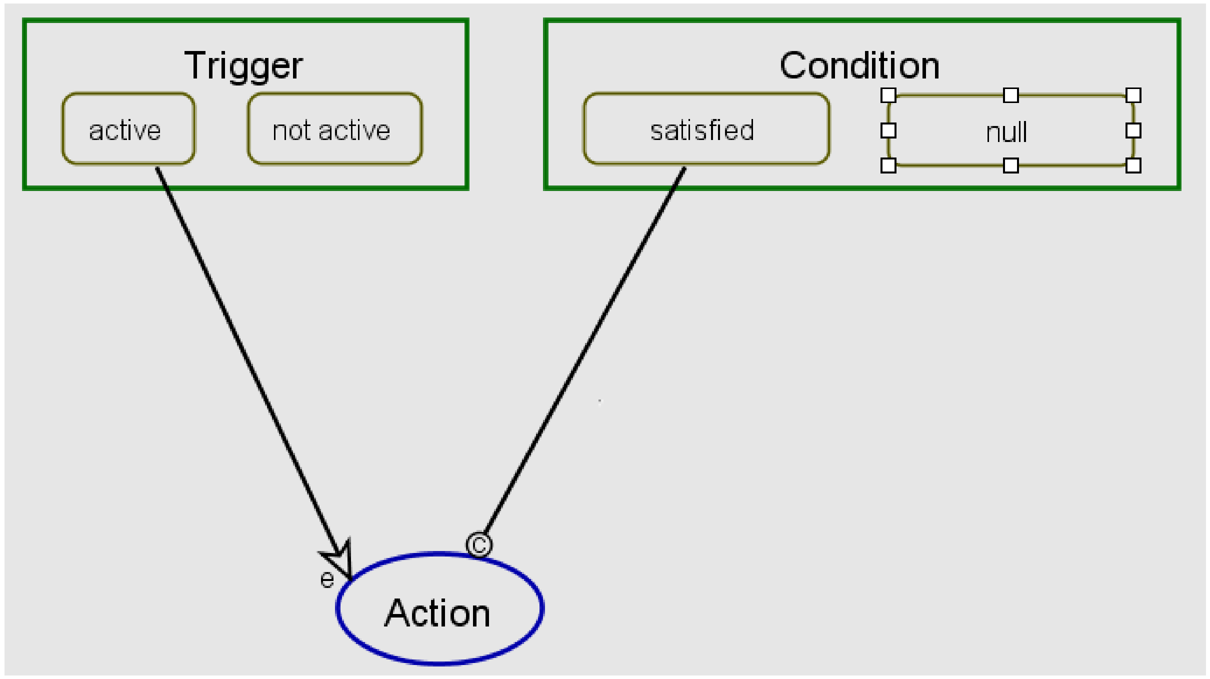
Event →Condition → Action paradigm



- An event link specify a **source event** and a **destination process** to activate on event occurrence.
- Triggering a process initiate an **attempt to execute** the process, but it **does NOT guarantee success** of this attempt.
- The **triggering event** forces an **evaluation of the process' precondition**, which, if and **only if** satisfied, allows process becomes active.
- **The event shall be lost** - If the precondition is not satisfied, process execution not occur **until another event** activates the process.

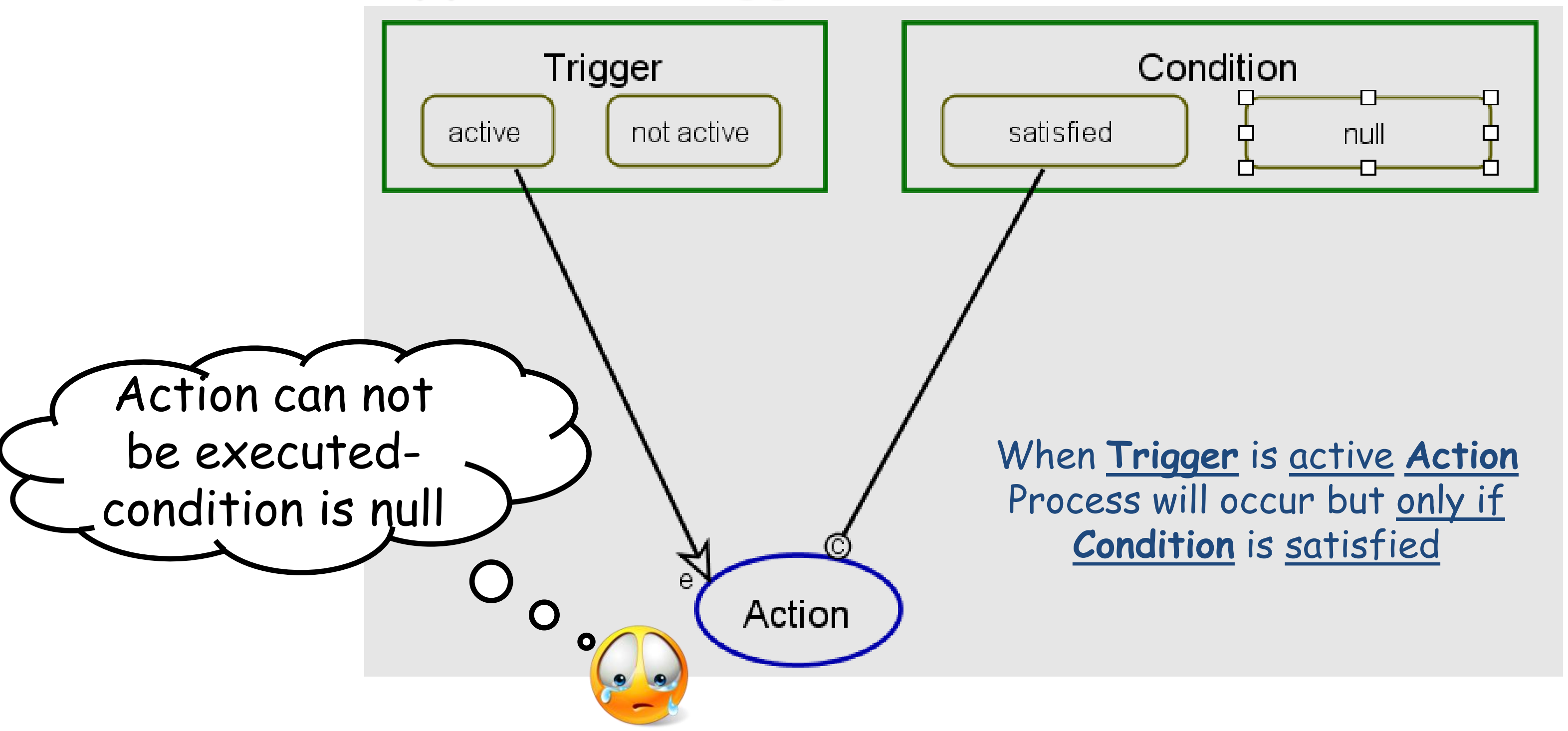
Event-Condition-Action - Example

What happens if trigger is active and condition is null?



Event-Condition-Action - Example

What happens if trigger is active and condition is null?

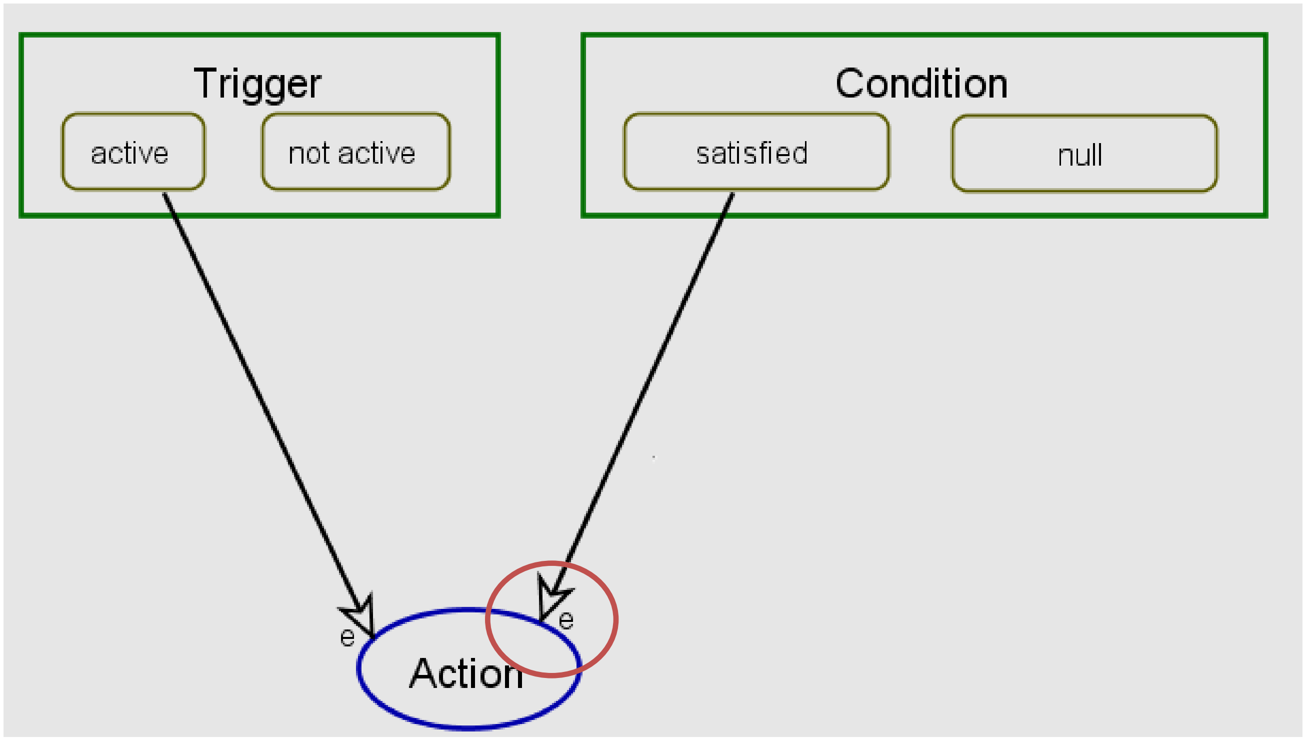

Trigger
active
not active
Condition
satisfied
null
e
©
Action
Action can not be executed- condition is null
When **Trigger** is active **Action** Process will occur but only if **Condition** is satisfied

Event is trying to trigger a process once. If the condition is not satisfied when the event happens, it is lost!

What is the right sentence here?

We change the **Condition Link** (from "satisfied") to **Event Link**

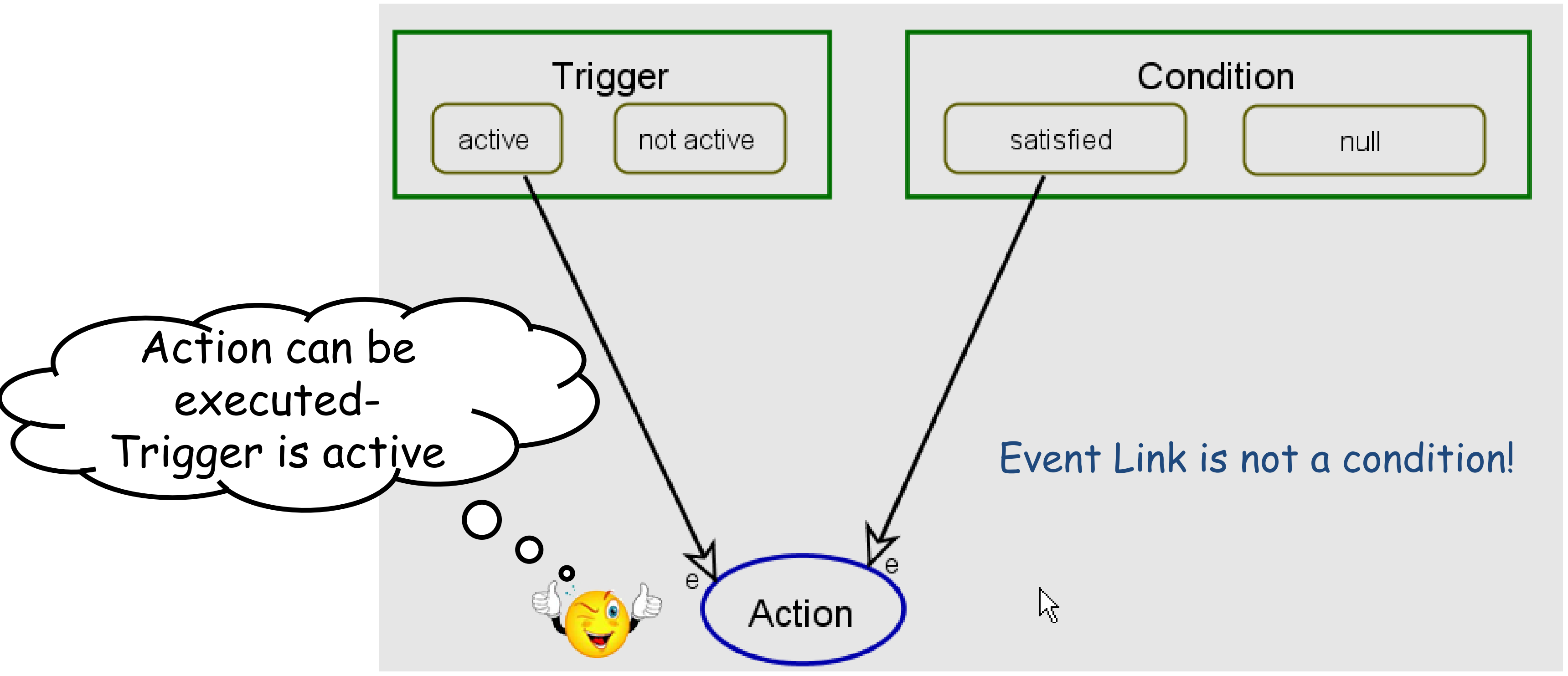
What happens if trigger is active and condition is null?



What is the right sentence here?

We change the **Condition Link** (from "satisfied") to **Event Link**

What happens if trigger is active and condition is null?

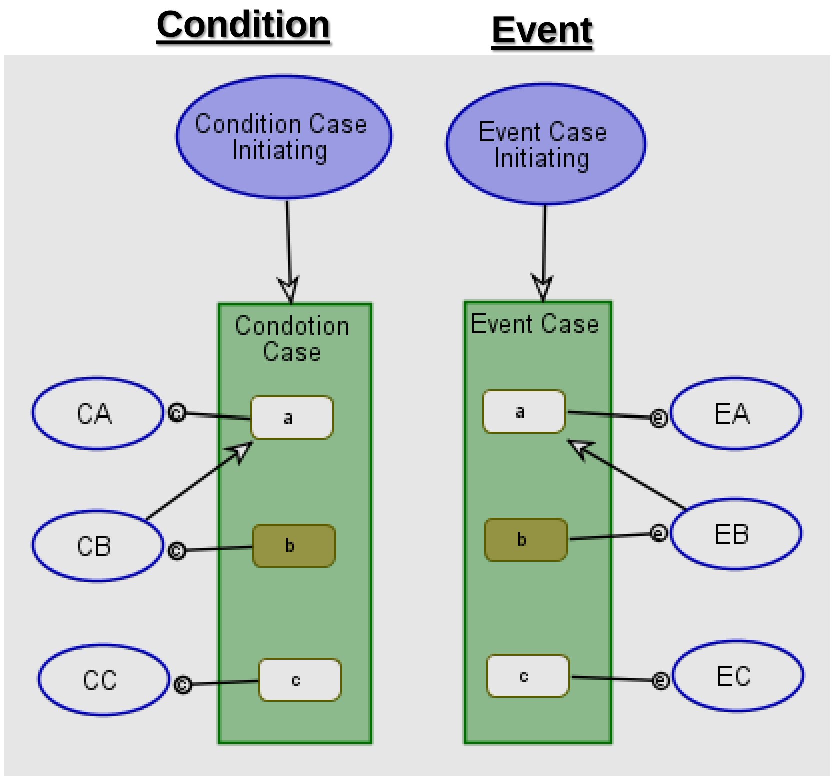

Trigger
active
not active
Condition
satisfied
null
e
e
Action
Event Link is not a condition!
Action can be executed- Trigger is active

Action is triggered if **Trigger** is active **or** **Condition** is satisfied

The relation among event links is always XOR!!!

Event vs. Condition Enabling Links

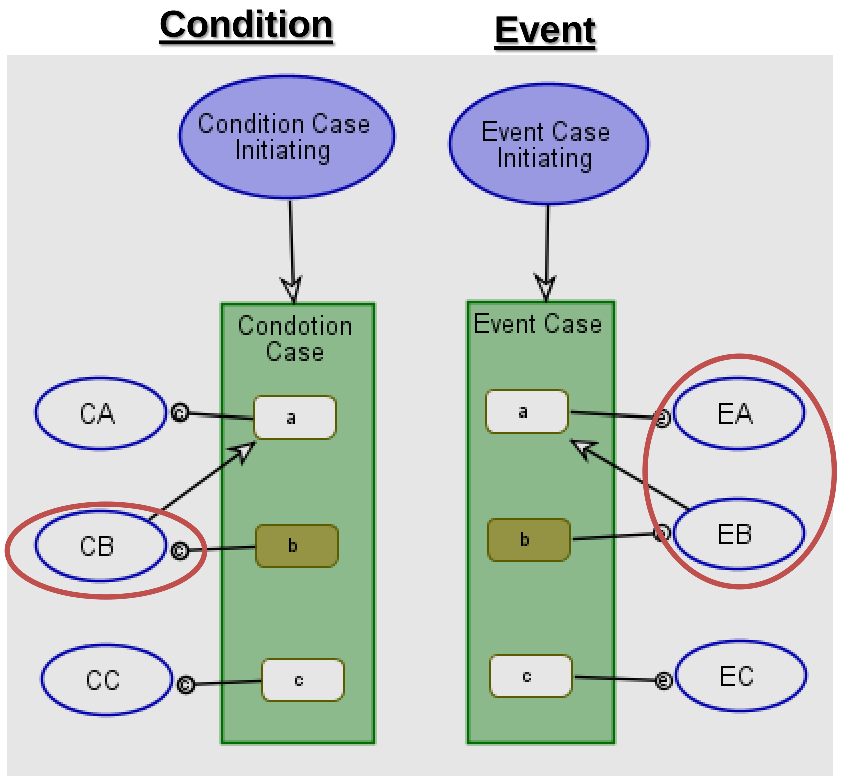
What are the differences between the models?



Event vs. Condition Enabling Links

What are the differences between the models?

- Starting process pre-condition evaluation depends on time line
- Case state **can** change any time **before** pre-condition process evaluation



- Starting process pre-condition evaluation **do not** depends on time line but on state change
- Case state can **not** be changed **any time before** pre-condition process evaluation.

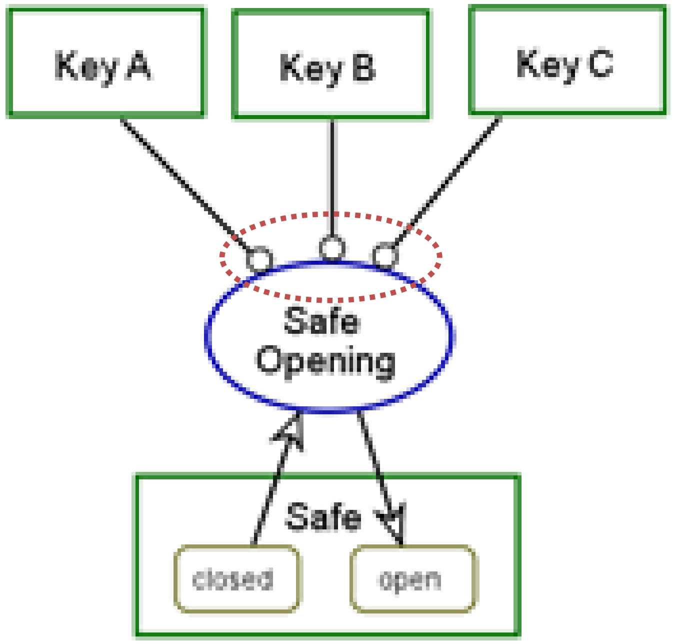
Only CB will be executed.
If CB would yields the c Condotion Case CC process would be executed too

EB and EA will be executed
In that order

Logical Operators

Boolean condition with Procedural Links

- Separate, non-touching links shall have the semantics of logical **AND**

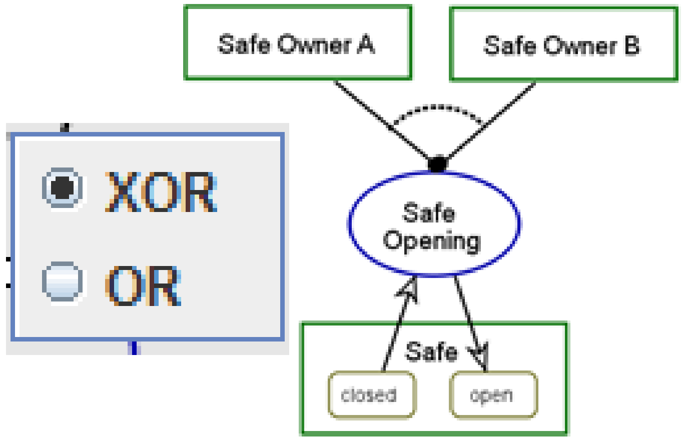


Pre condition

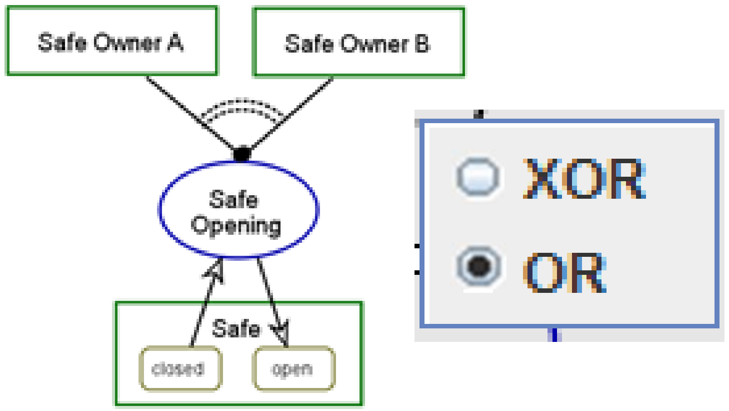
Key A **&** Key B **&** Key C **&** closed safe

Boolean condition with Procedural Links

- A group of two or more procedural links of the same kind that originate from, or arrive at, the same object represent **OR** / **XOR**



XOR operator mean
exactly one of the things



OR operator mean
at least one of the two or more

Table 17 — Summary of XOR and OR converging consumption and result links

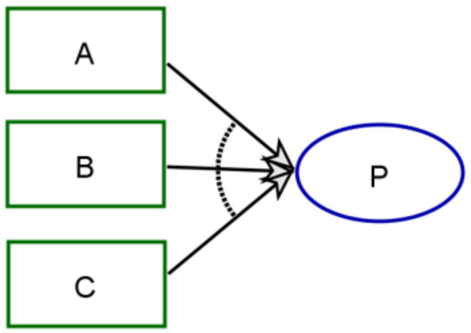
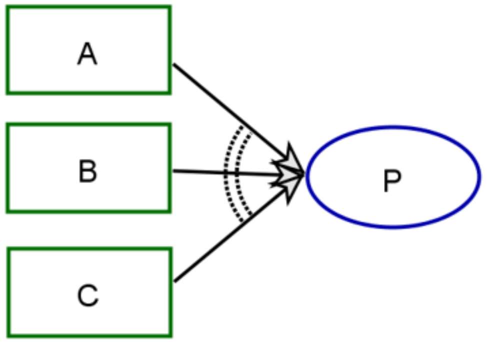
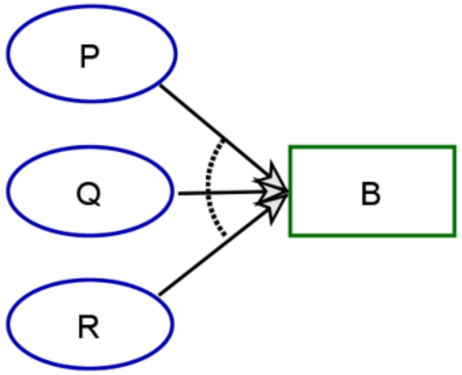
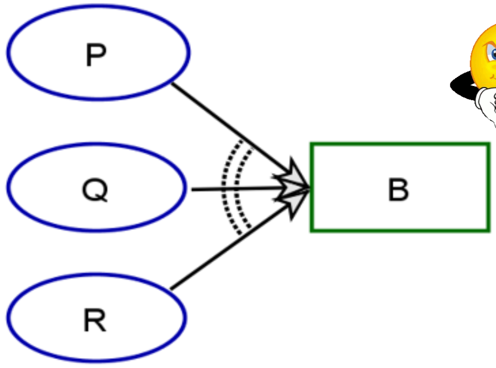
	XOR	OR
Converging consumption link fan	 **P** consumes exactly one of **A, B** , or **C.**	 **P** consumes at least one of **A, B** , or **C.**
Converging result link fan	 Exactly one of **P, Q,** or **R** yields **B.**	 At least one of **P, Q,** or **R** yields **B.**

Table 18 — Summary of XOR and OR diverging consumption and result link fans

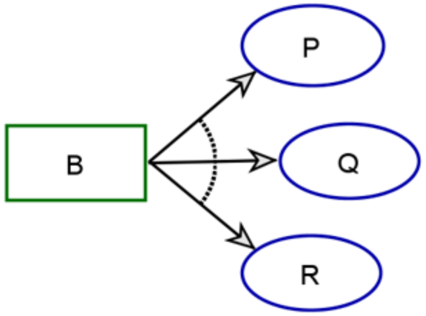
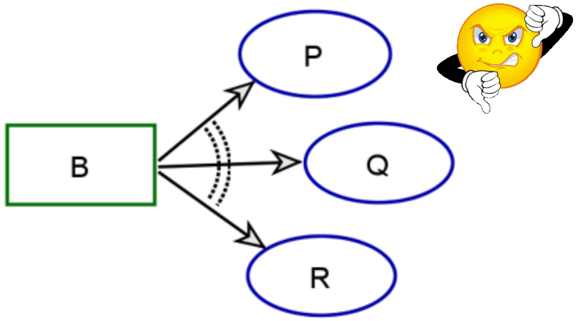
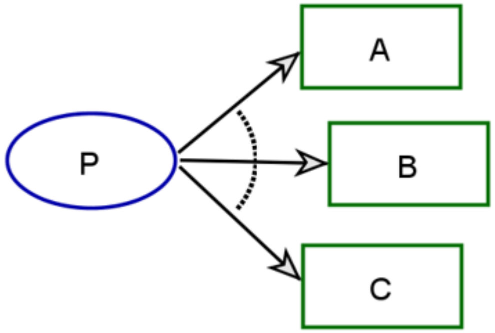
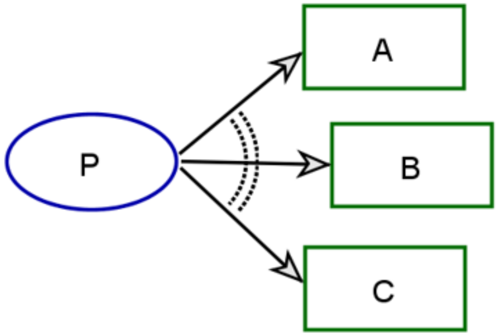
	XOR	OR
Diverging consumption link fan	 Exactly one of **P, Q** , or **R** consumes **B.**	 At least one of **P, Q** , or **R** consumes **B.**
Diverging result link fan	 **P** yields exactly one of **A, B,** or **C.**	 **P** yields at least one of **A, B,** or **C.**

Table 19 — Summary of XOR and OR joint effect link fans

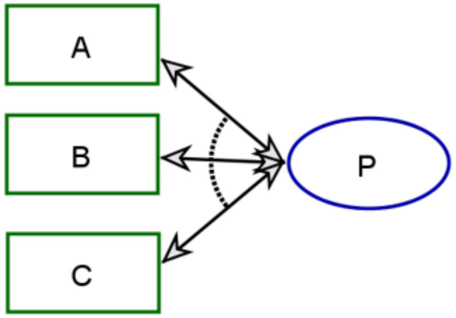
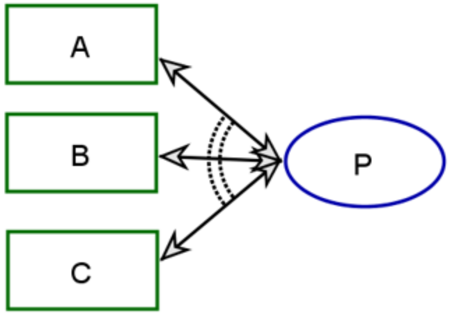
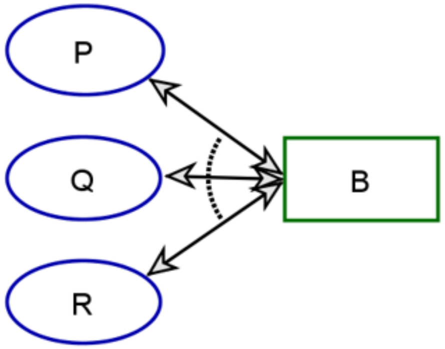
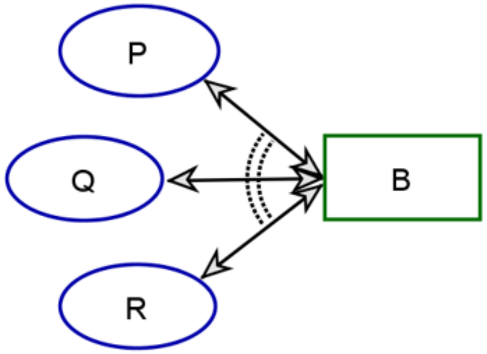
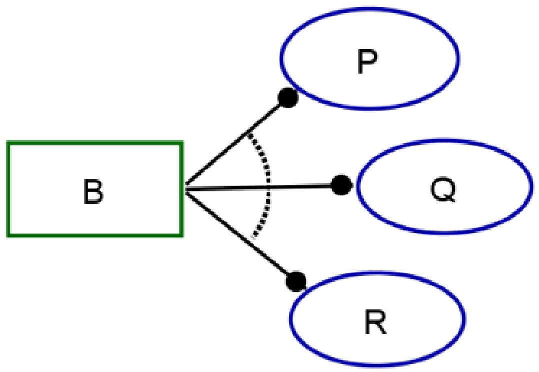
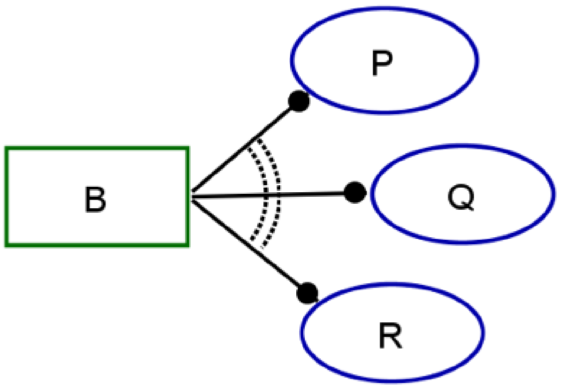
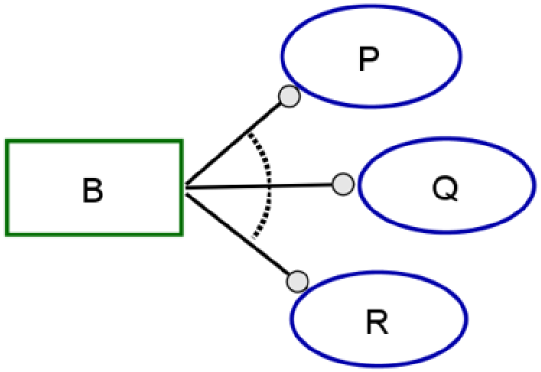
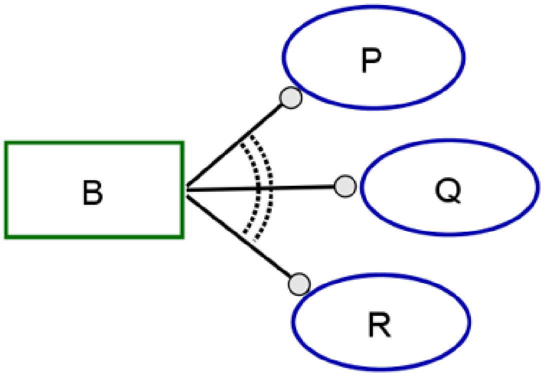
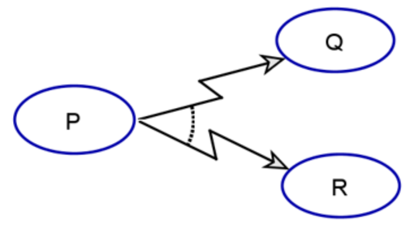
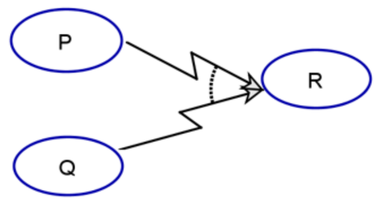
	XOR	OR
Multiple objects effect link fan	 **P** affects exactly one of **A**, **B**, or **C.**	 **P** affects at least one of **A**, **B**, or **C.**
Multiple processes effect link fan	 Exactly one of **P**, **Q**, or **R** affects **P.**	 At least one of **P**, **Q**, **R** affects **P.**

Table 20 — Agent and instrument link fans

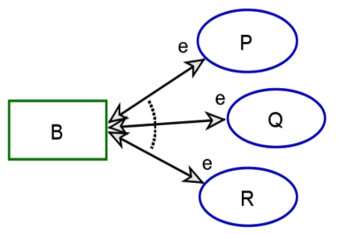
	XOR	OR
Agent link fan	 **B** handles exactly one of **P**, **Q**, or **R.**	 **B** handles at least one of **P**, **Q, R.**
Instrument link fan	 Exactly one of **P, Q**, or **R** requires **B.**	 At least one of **P, Q, R** requires **B.**



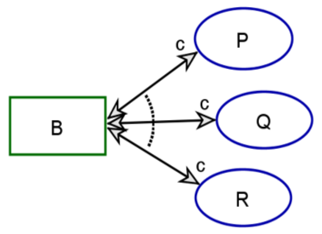
P invokes exactly one **Q** or **R.**



Exactly one of **P** or **Q** invokes **R.**

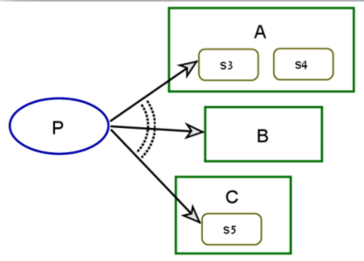


B triggers exactly one of **P**, **Q**, or **R**, which affects the occurring process.



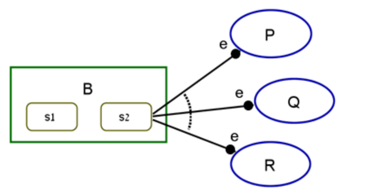
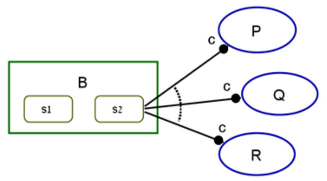
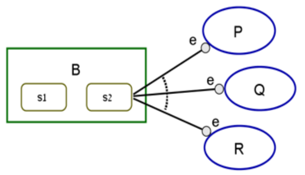
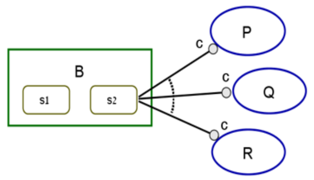
Exactly one of **P**, **Q**, or **R** occurs if **B** exists, in which case the occurring process affects **B**, otherwise these processes are skipped.

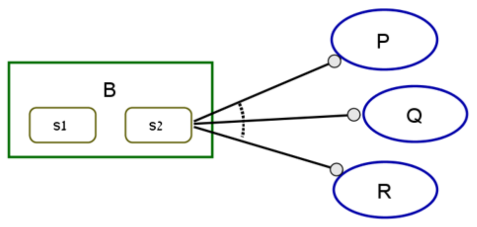
Same logic behavior for other types of procedural links such as invocation, event, condition



P yields at least one of **s3 A**, **B**, or **s5 C**.

Same logic behavior to/from object's state

Agent link fan	 **S2 B** triggers and handles exactly one of **P**, **Q**, or **R**. *The stateless case:* **B** triggers and handles exactly one of **P**, **Q**, or **R**.	 **B** handles exactly one of **P**, **Q**, or **R** if **B** is **s2**, otherwise these processes are skipped. *The stateless case:* **B** handles exactly one of **P**, **Q**, or **R** if **B** exists, otherwise these processes are skipped.
Instrument link fan	 **S2 B** triggers exactly one of **P**, **Q**, or **R**, which requires **s2 B.** *The stateless case:* **B** triggers exactly one of **P**, **Q**, or **R**, which requires **s2 B.**	 Exactly one of **P**, **Q**, or **R** requires that **B** is **s2**, otherwise these processes are skipped. *The stateless case:* Exactly one of **P**, **Q**, or **R** requires that **B** exists, otherwise these processes are skipped.

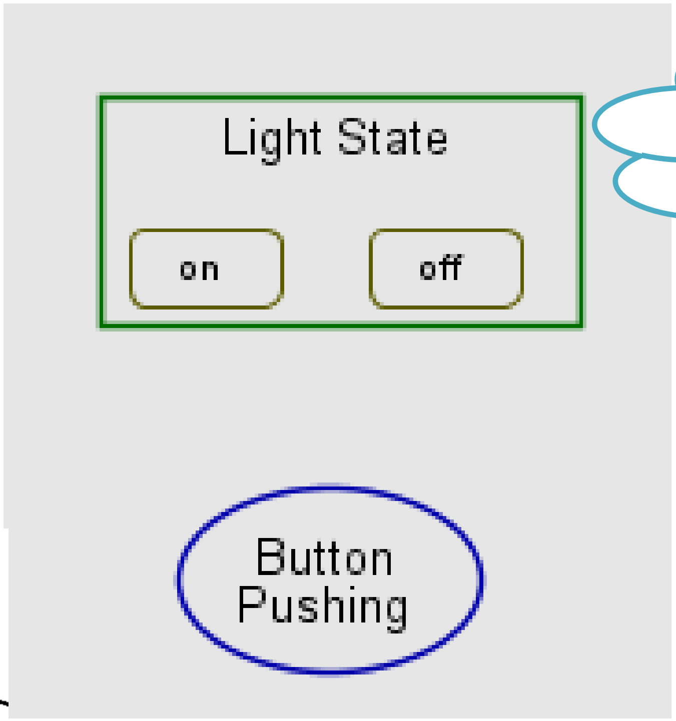


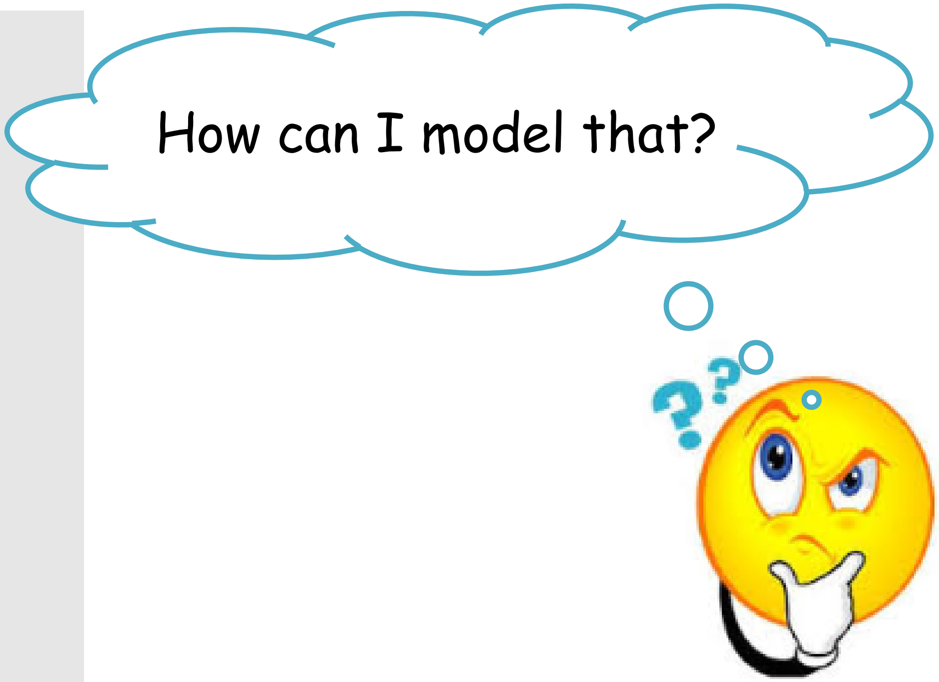
Exactly one of **P**, **Q**, or **R** requires **s2 B**.

Path

State Changes with Paths

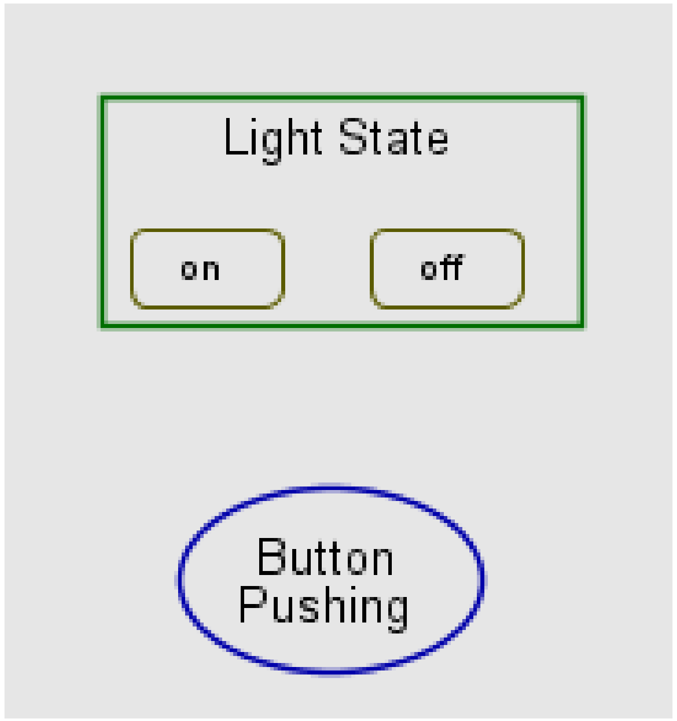
- Button Pushing changes Light state from on to off and vice versa

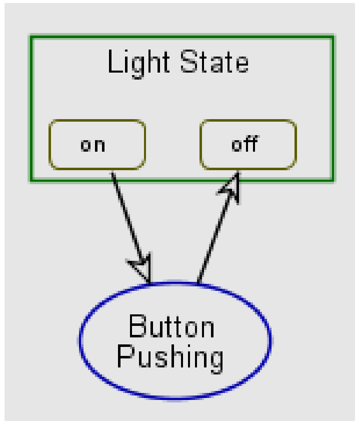


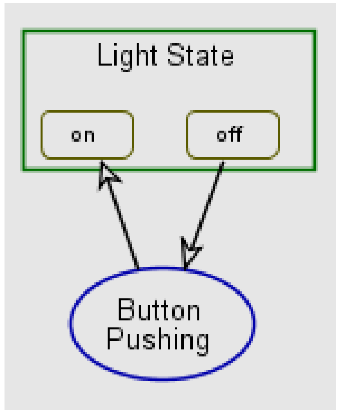


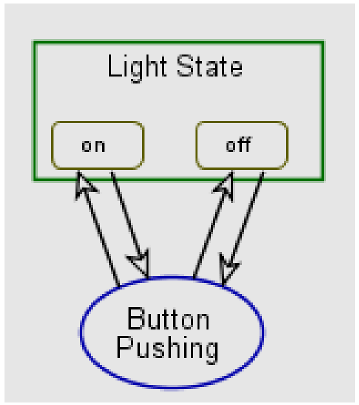
State Changes with Paths

- Button Pushing changes Light state from on to off and vice versa



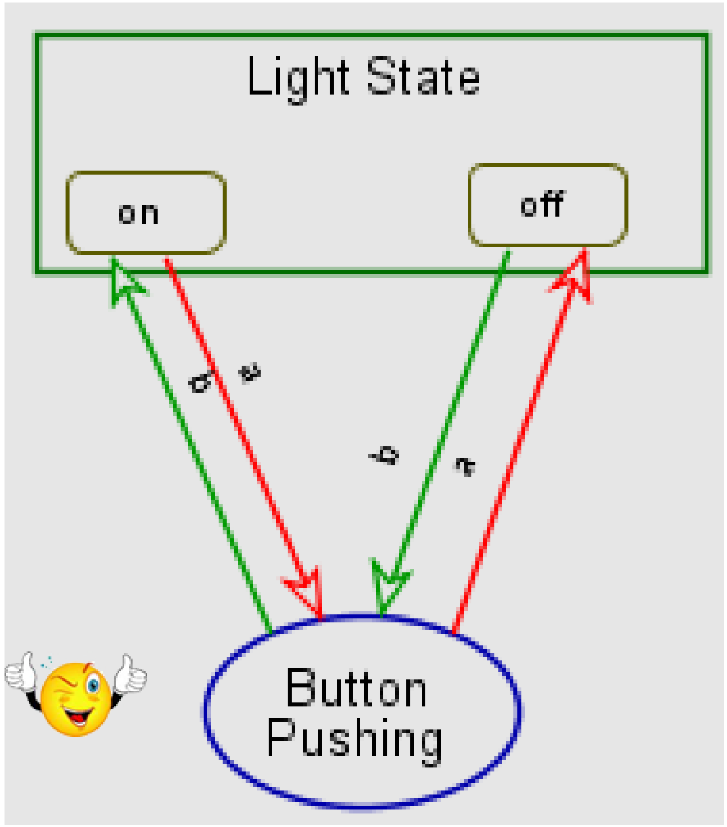




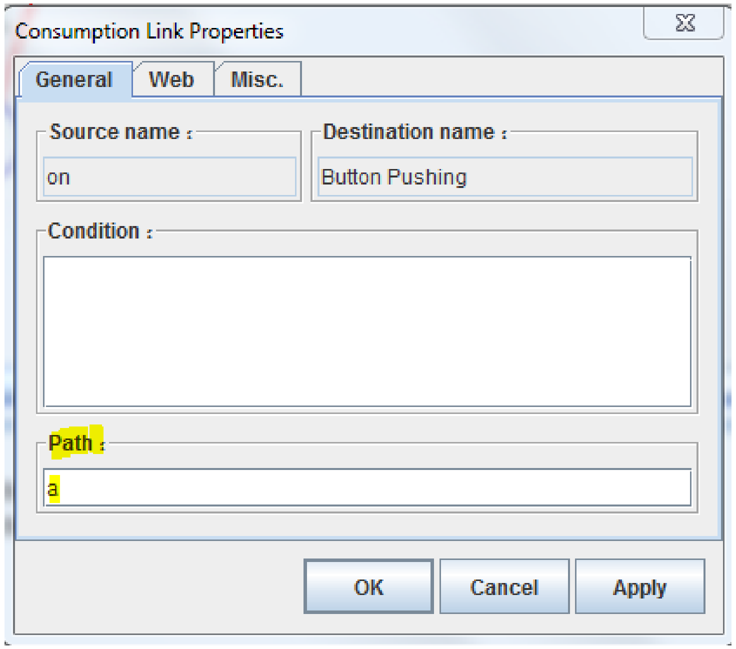


State Changes with Paths

- Button Pushing changes Light state from on to off and vice versa



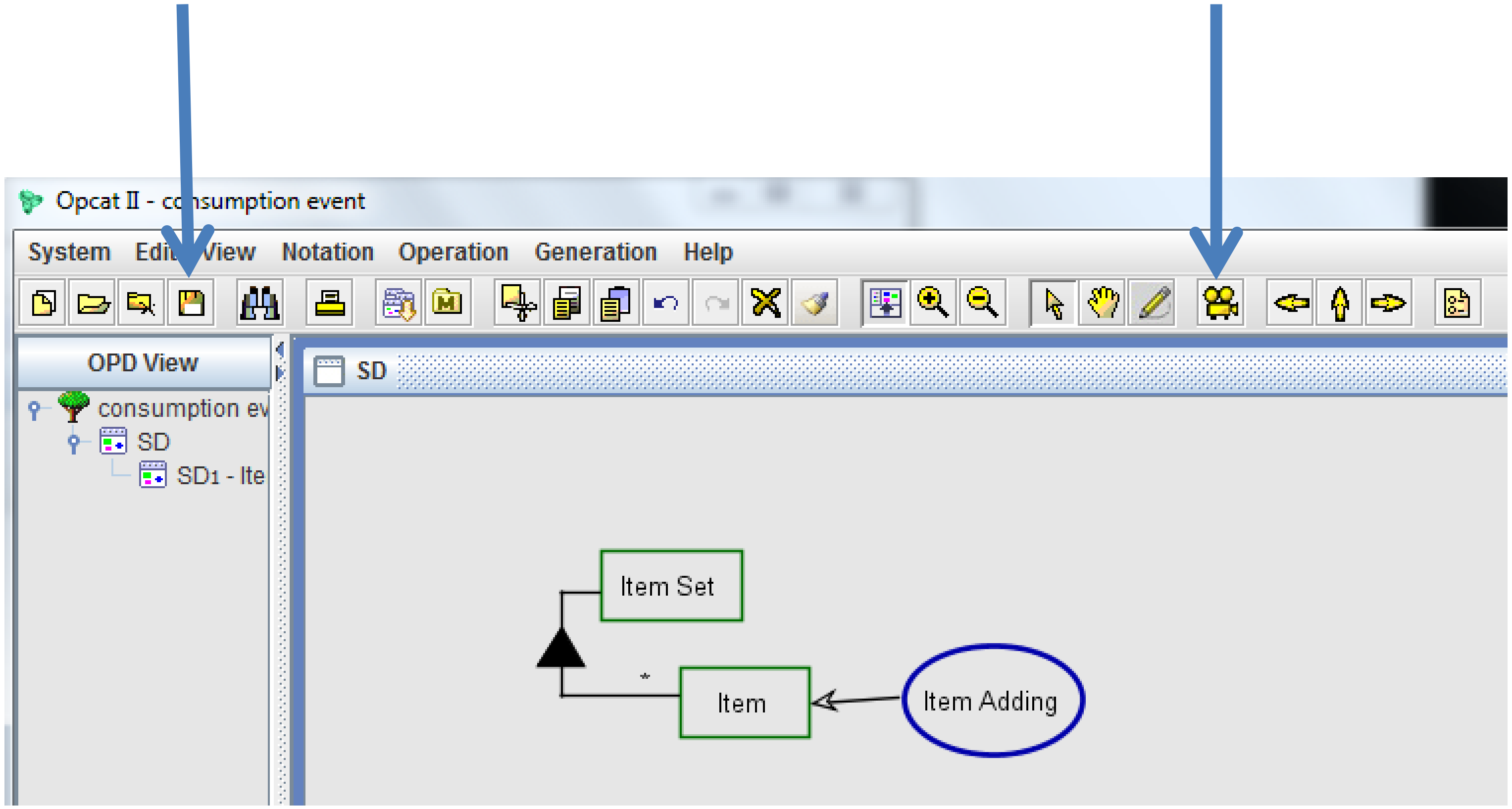
Light State can be on or off.
Following path a, Button Pushing changes Light State from on to off.
Following path b, Button Pushing changes Light State from off to on.



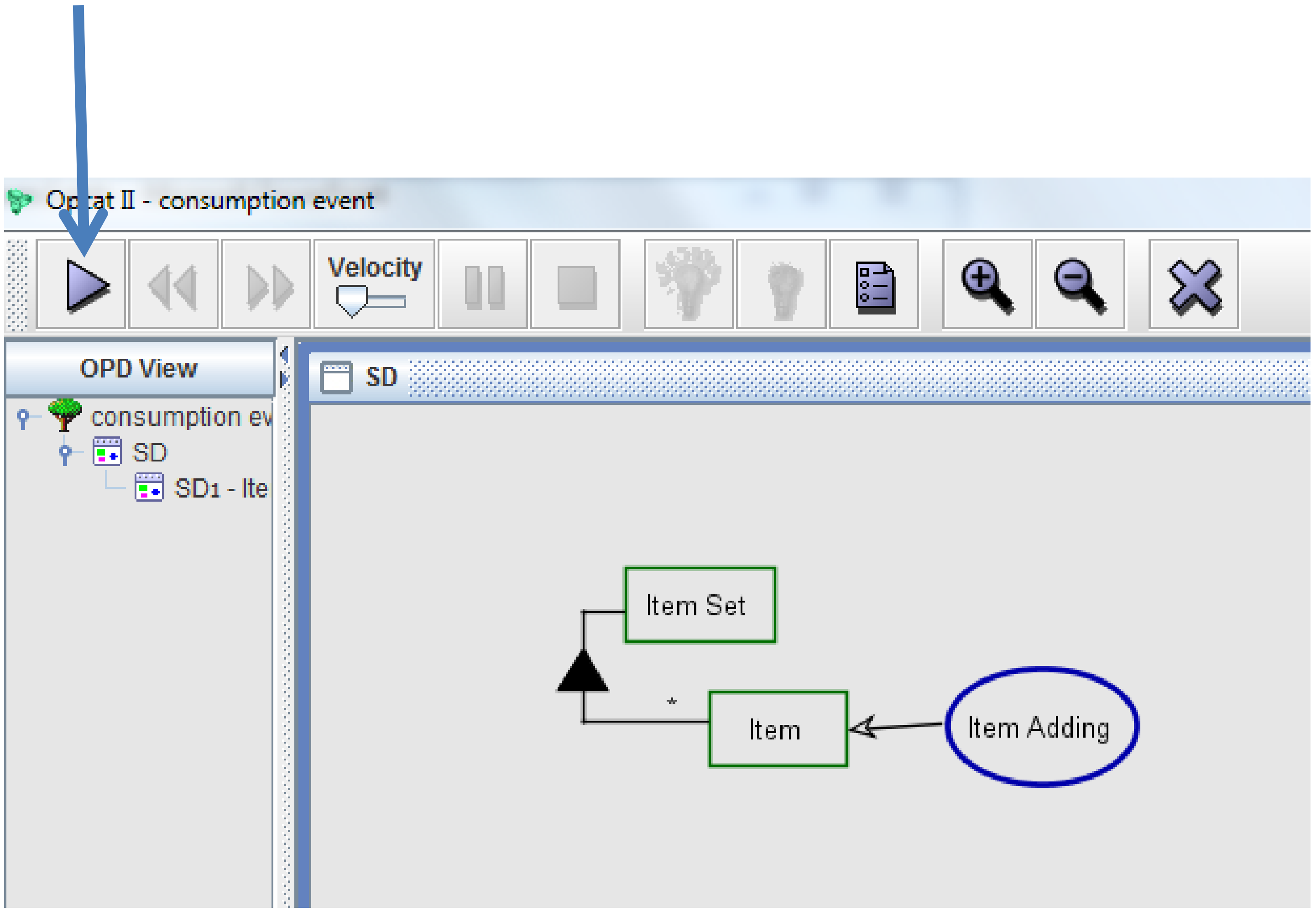
Source name : on
Destination name : Button Pushing
Condition :
Path : a

Simulation

Simulation

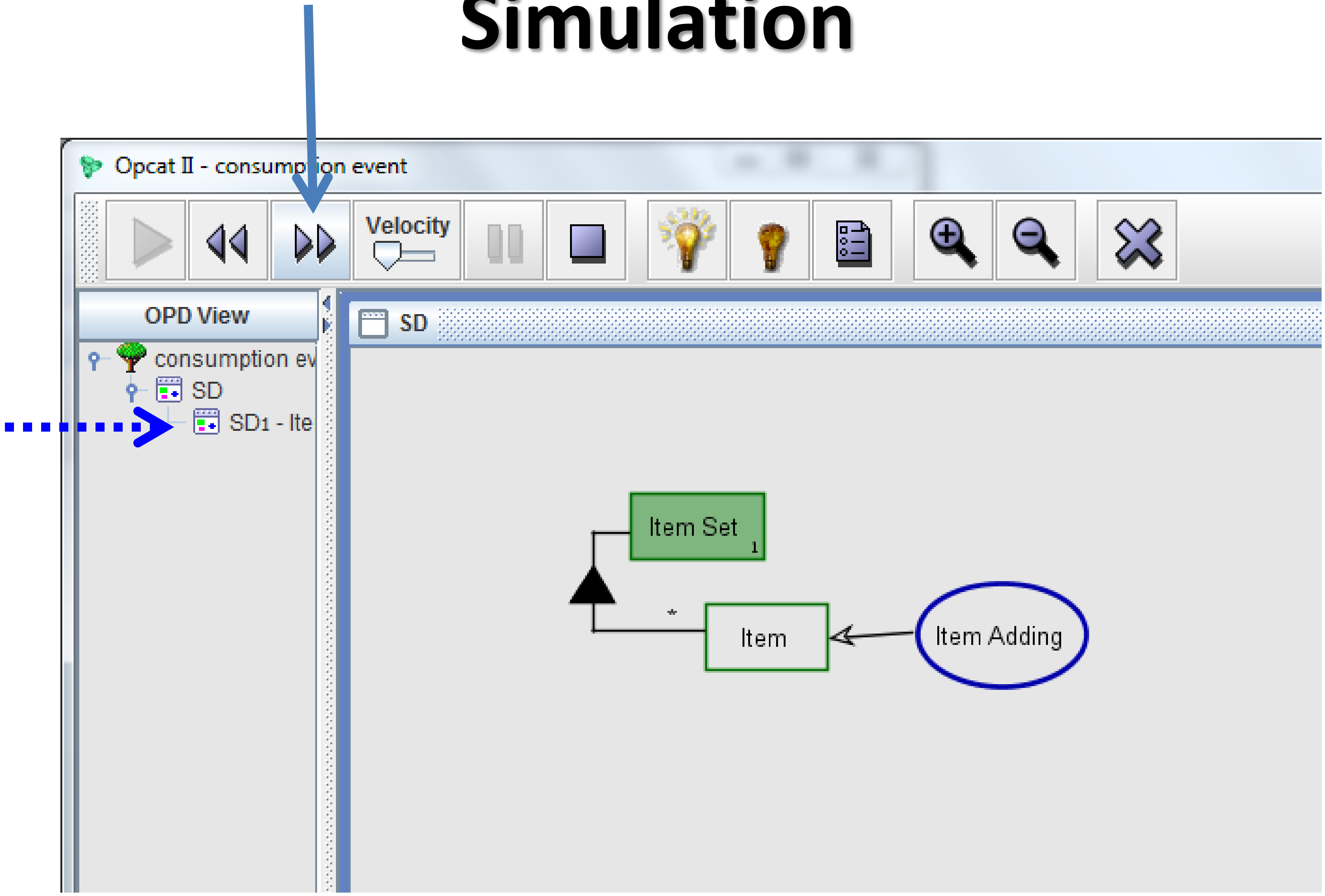

Opcat II - consumption event
System
Edit
View
Notation
Operation
Generation
Help
OPD View
consumption ev
SD
SD1 - Ite
SD
Item Set
Item
*
Item Adding

Simulation

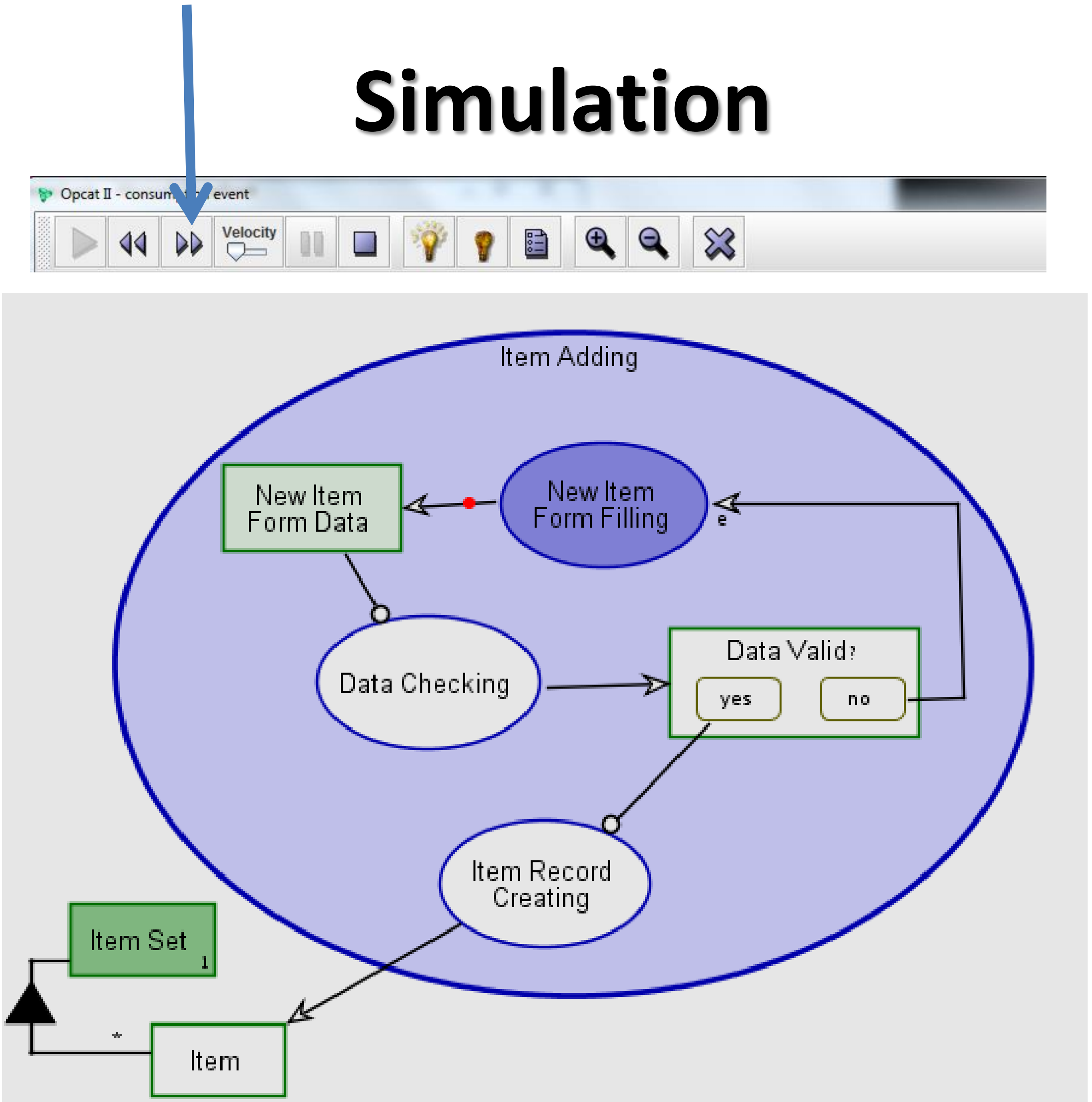


Opcat II - consumption event
Velocity
OPD View
consumption ev
SD
SD1 - Ite
SD
Item Set
Item
*
Item Adding

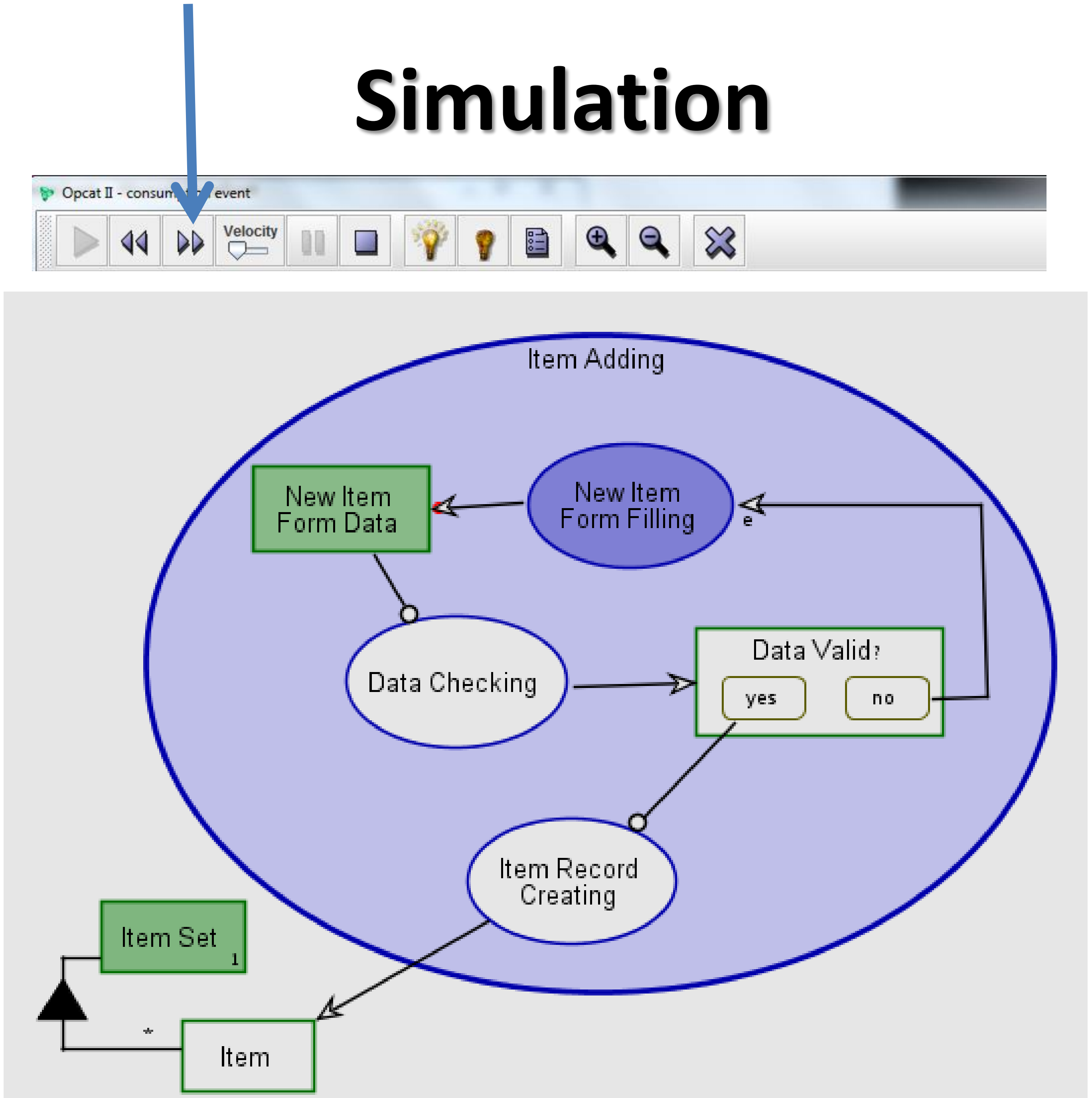
Simulation


Opcat II - consumption event
Velocity
OPD View
consumption ev
SD
SD1 - Ite
SD
Item Set
1
*
Item
Item Adding

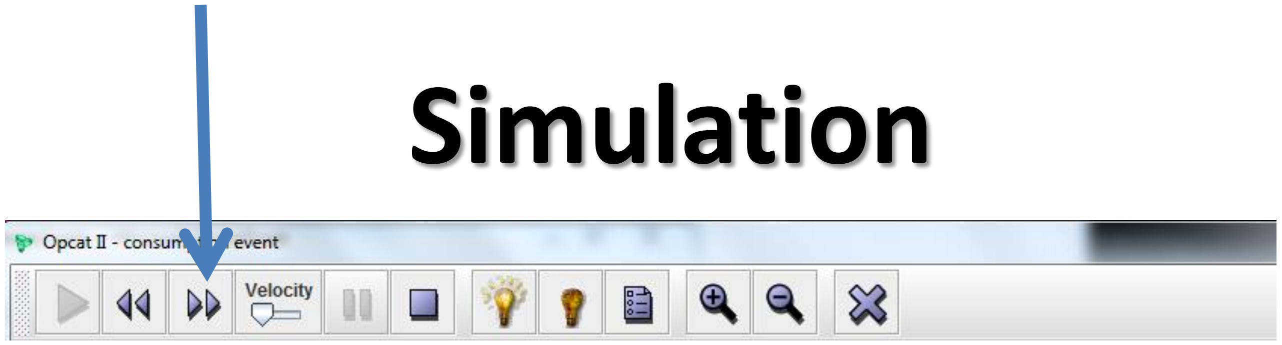
Simulation

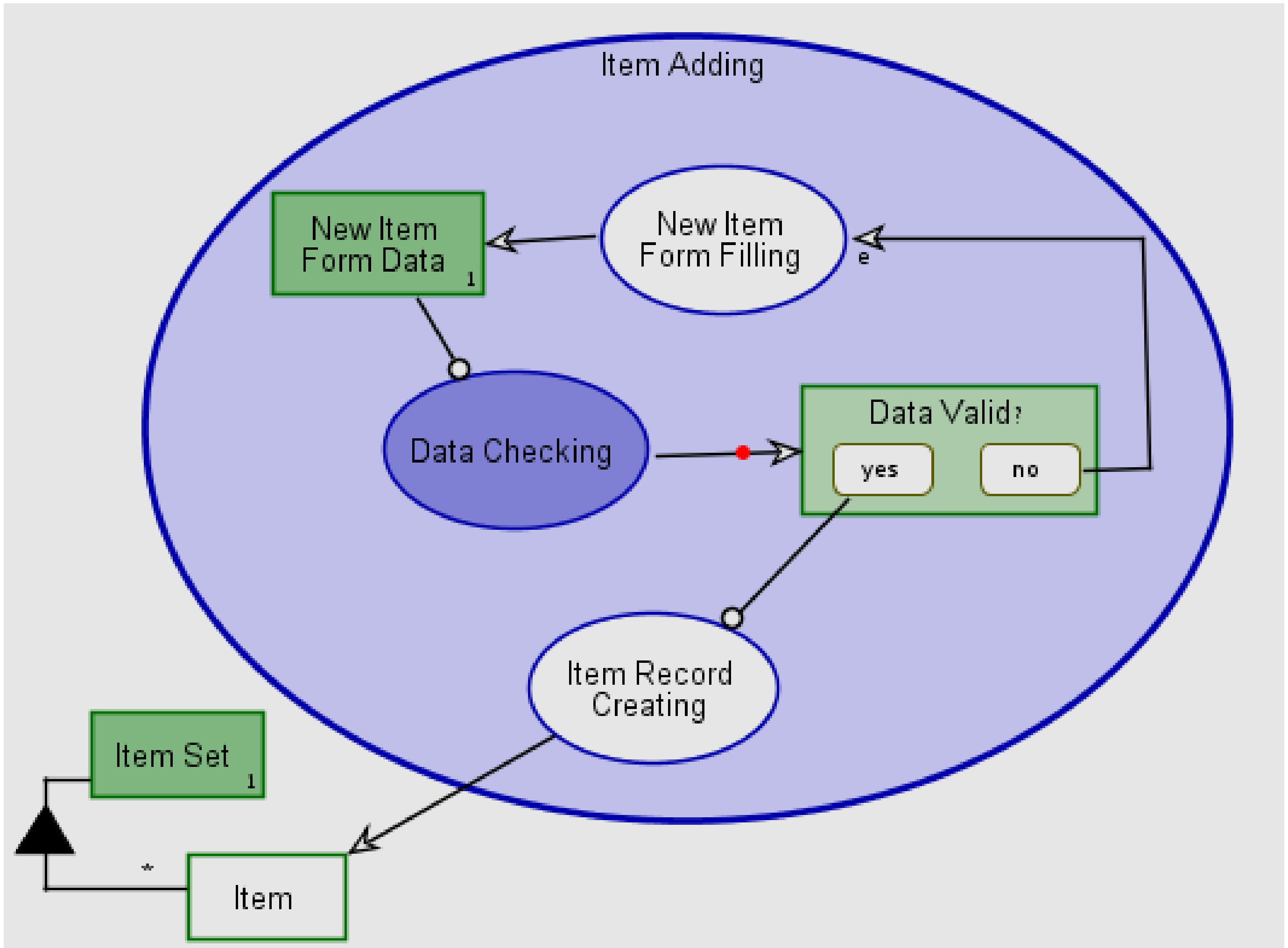

Opcat II - consum event
Velocity
Item Adding
New Item Form Data
New Item Form Filling
e
Data Checking
Data Valid?
yes
no
Item Record Creating
Item Set
1
*
Item

Simulation

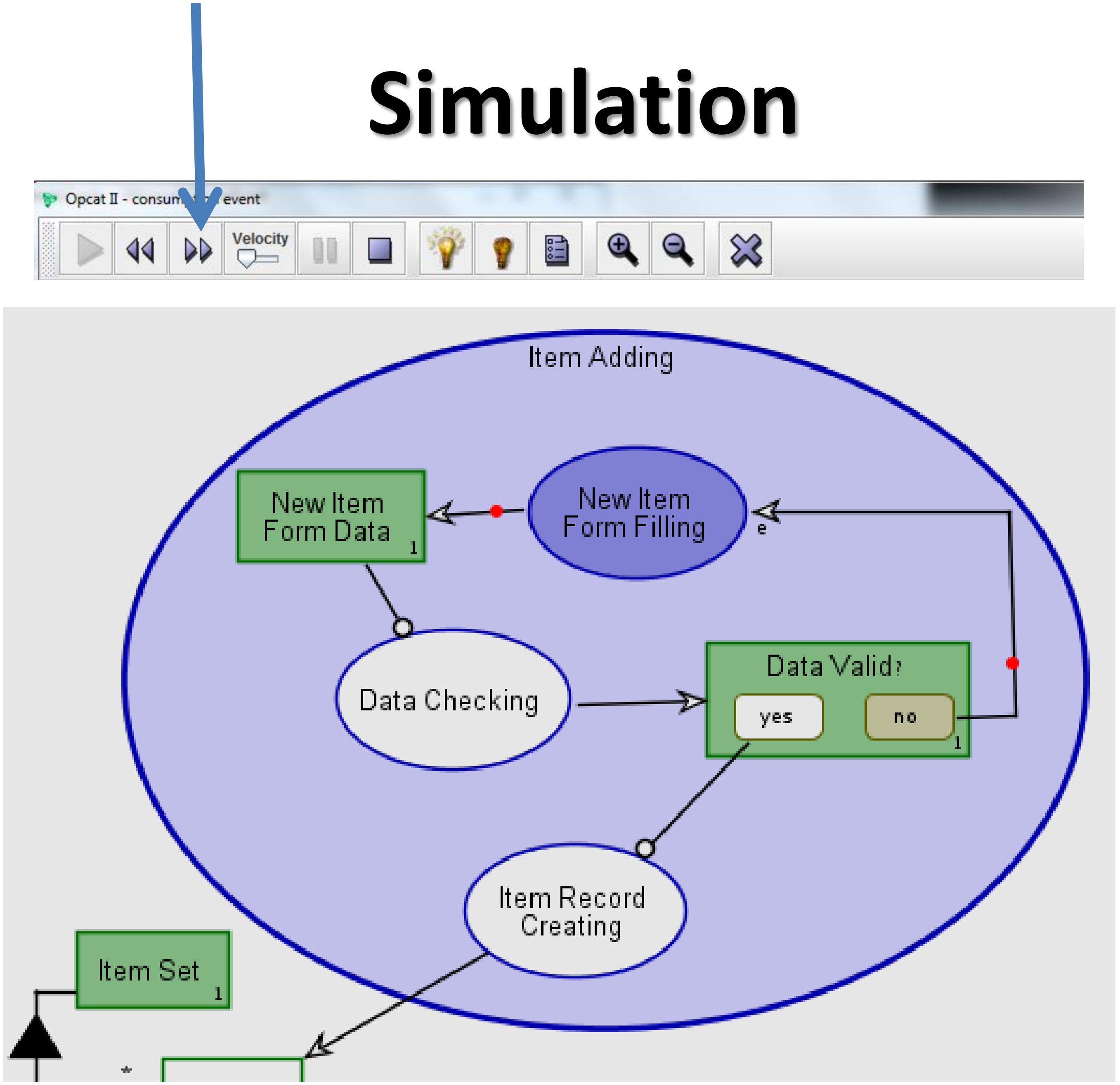

Opcat II - consum event
Velocity
Item Adding
New Item Form Data
New Item Form Filling
e
Data Checking
Data Valid?
yes
no
Item Record Creating
Item Set
1
*
Item

Simulation


Opcat II - consum event
Velocity

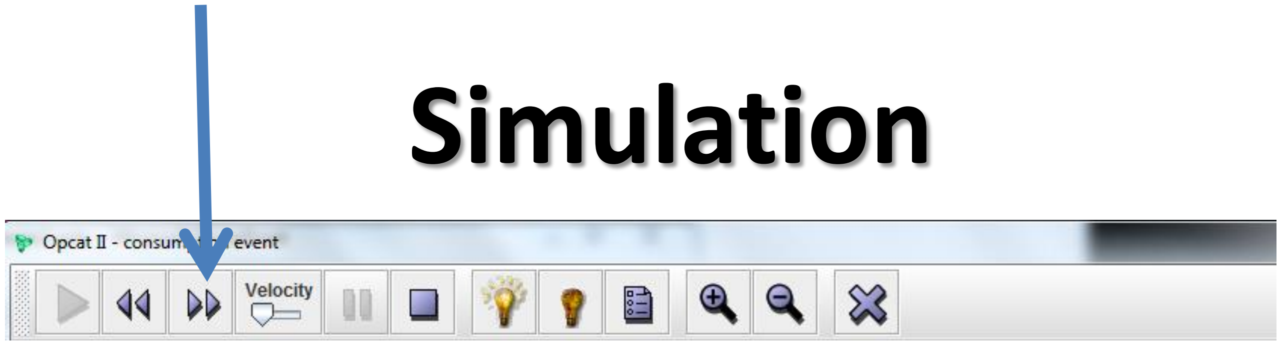

Item Adding
New Item Form Data
1
New Item Form Filling
e
Data Checking
Data Valid?
yes
no
Item Record Creating
Item Set
1
*
Item

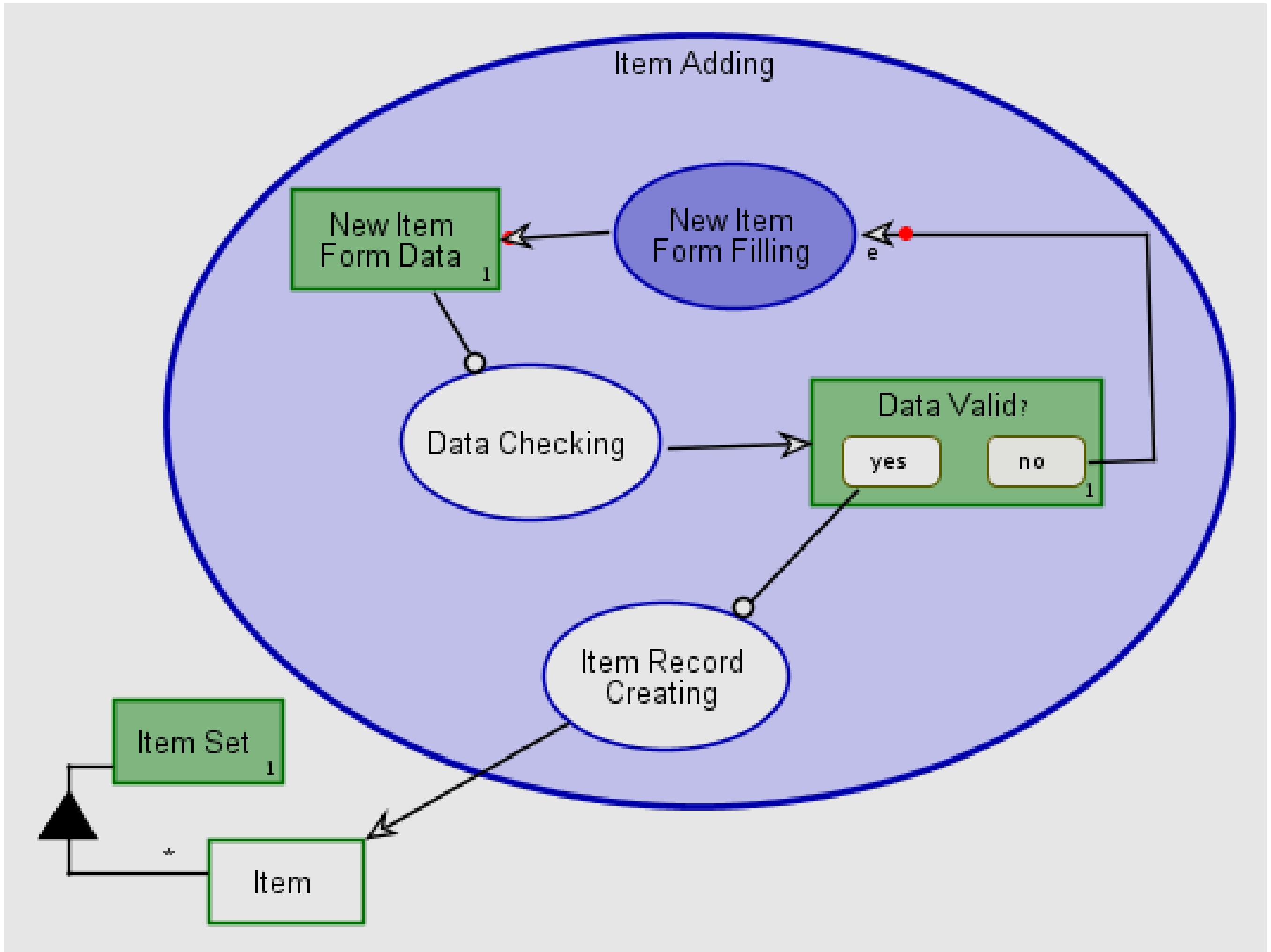
Simulation



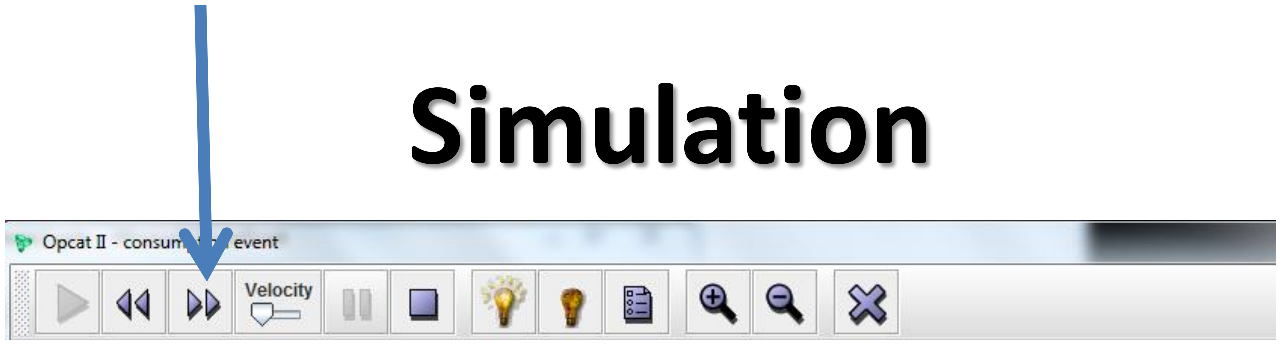
Instrument link is not satisfied because Object Data Valid? has no instances at State yes.

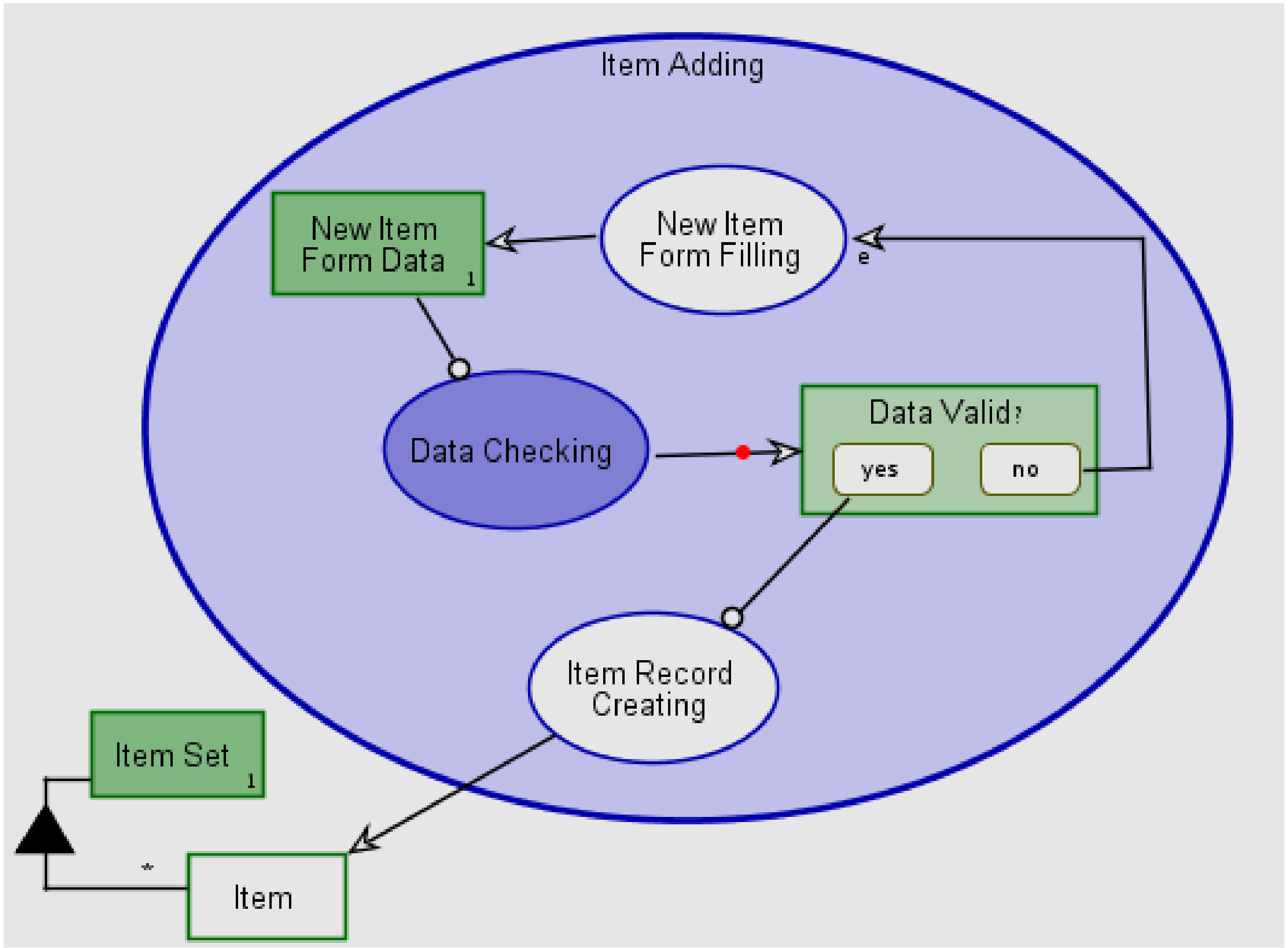
Simulation


Opcat II - consum... event
Velocity

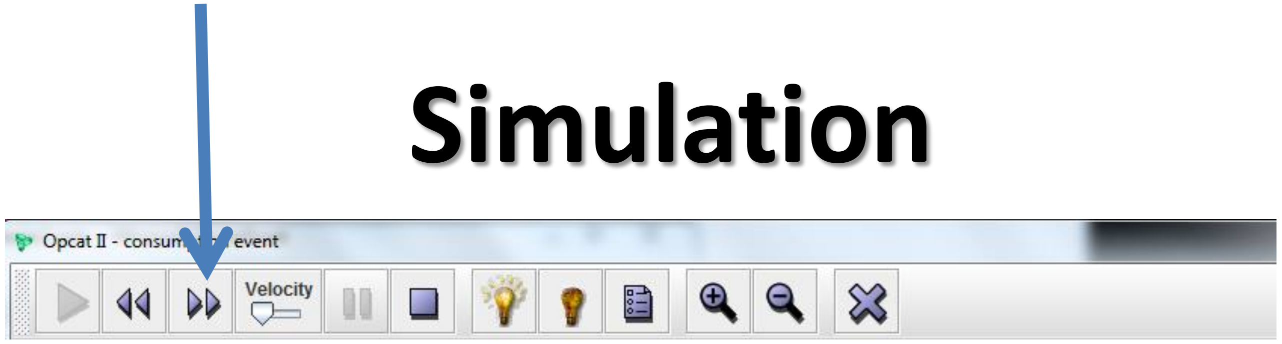

Item Adding
New Item Form Data
1
New Item Form Filling
e
Data Checking
Data Valid?
yes
no
1
Item Record Creating
Item Set
1
*
Item

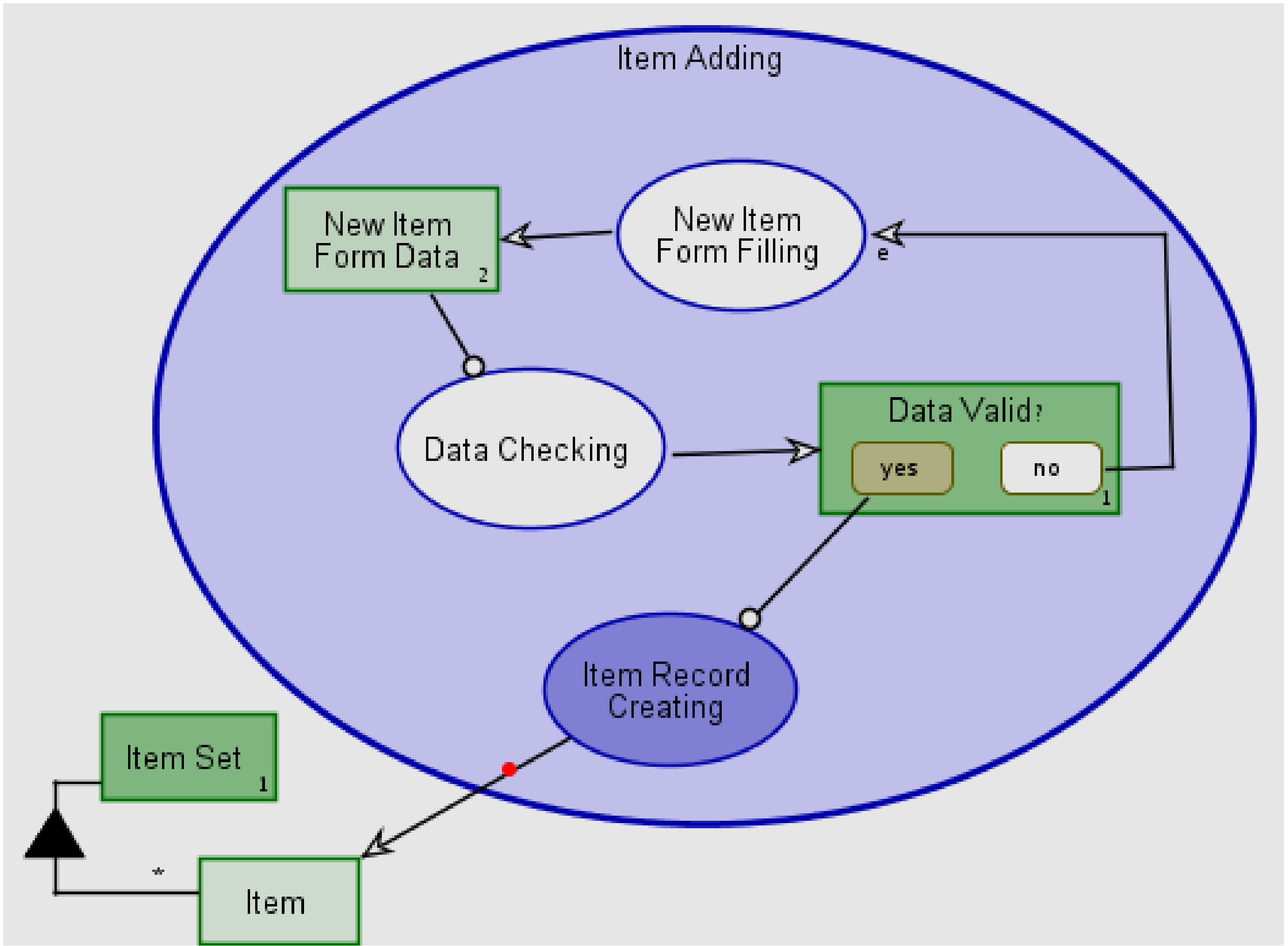
Simulation


Opcat II - consum... event
Velocity

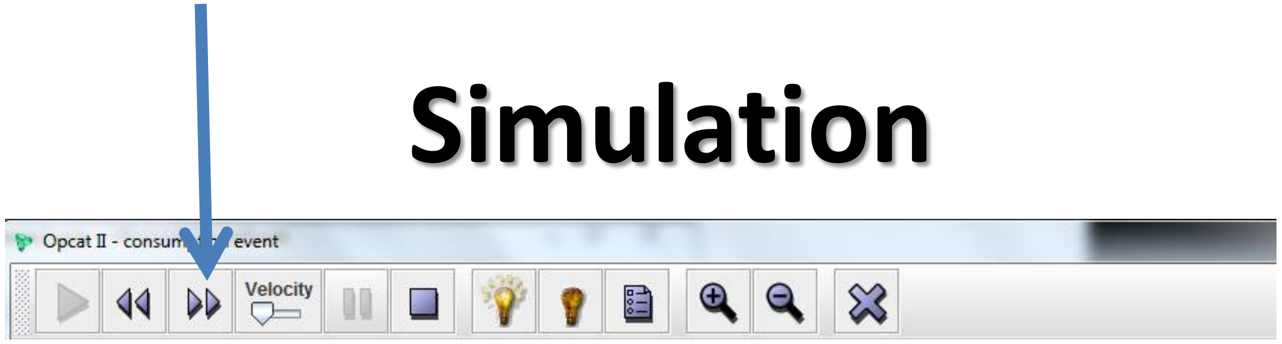

Item Adding
New Item Form Data
1
New Item Form Filling
e
Data Checking
Data Valid?
yes
no
Item Record Creating
Item Set
1
*
Item

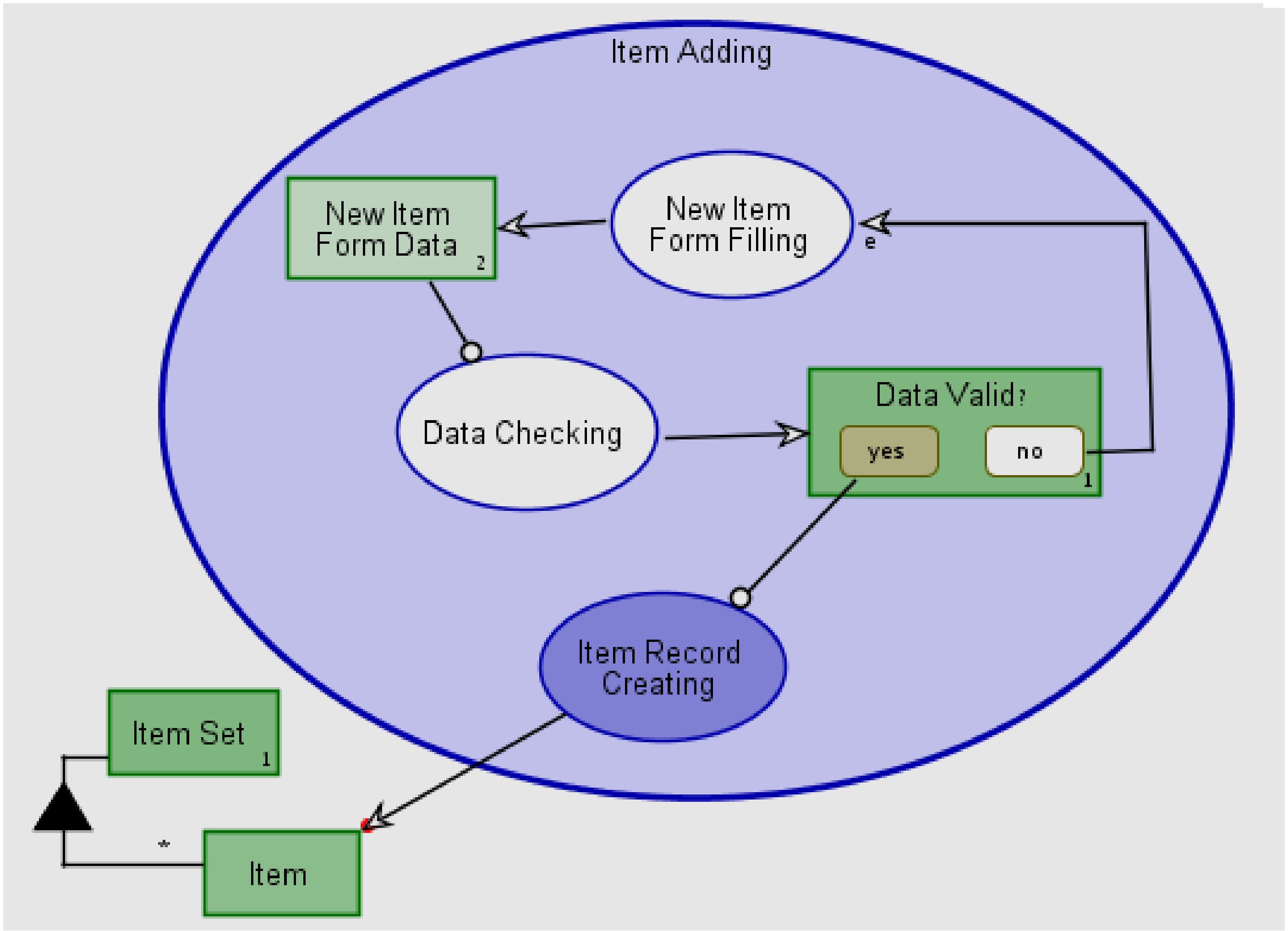
Simulation


Opcat II - consum event
Velocity

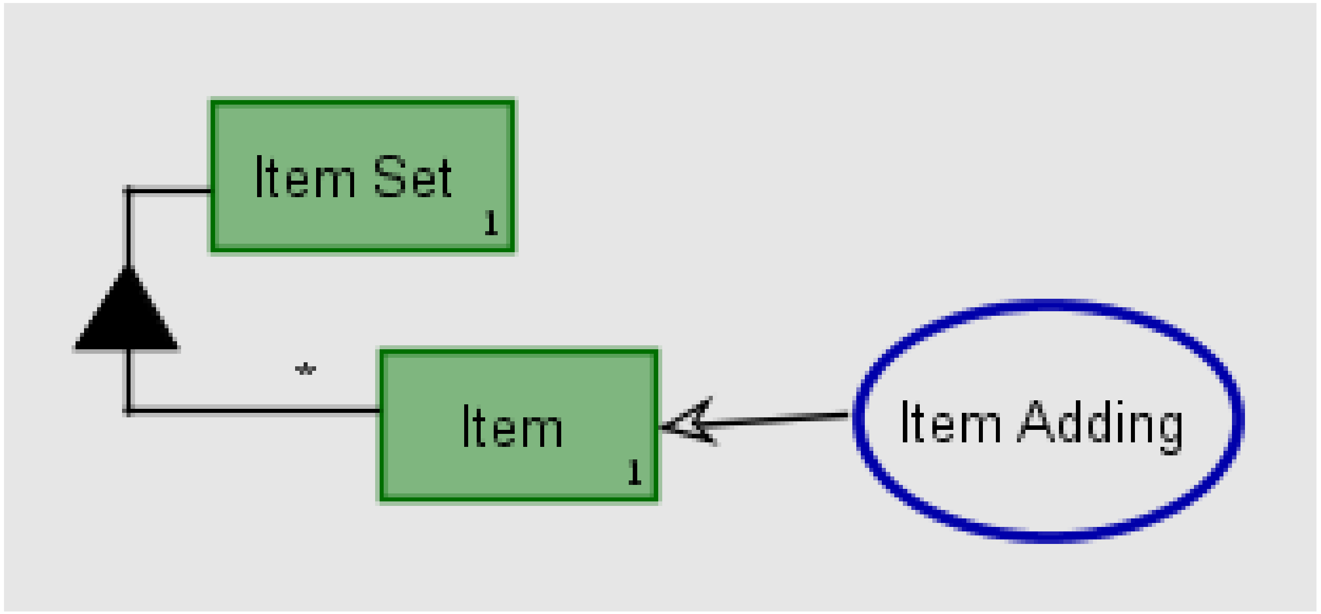

Item Adding
New Item Form Data
2
New Item Form Filling
e
Data Checking
Data Valid?
yes
no
1
Item Record Creating
Item Set
1
*
Item

Simulation


Opcat II - consum... event
Velocity


Item Adding
New Item Form Data
2
New Item Form Filling
e
Data Checking
Data Valid?
yes
no
1
Item Record Creating
Item Set
1
*
Item

Simulation



הנחיות להגשה ראשונה - מבנה העבודה

- **פרק מקדים של תיחום (scope) ודרישות (requirements), הכולל:**
  - **הקדמה** המתארת את מטרת המערכת והתיחום שלה (פסקה קצרה)
  - **תיאור הפונקציונליות והפונקציות המרכזיות** של המערכת שיאפשרו להשיג פונקציונליות זו (כחמש פונקציות מרכזיות).
  - הגדרת **מושגים מרכזיים** הקשורים לאונטולוגיה הרלוונטית למערכת (עבור מי שלא בקיא בעולם התוכן).
  - **טבלת הדרישות** מנותחות (בסביבות 10 דרישות מרכזיות בהתאם לתיחום שבחרתם). לכל דרישה יש להגדיר מהו התהליך שבו תמומש דרישה זו (בהתאם לתיחום הקיים בהגשה זו) וכן מהן וישויות המידע המרכזיות אשר נדרשות למימוש דרישה זו.

- **פרק ובו המודל של OPM אשר יכיל:**
  - דיאגרמת **SD** - אשר תגדיר את הפונקציונליות המרכזית של המערכת, בעלי העניין המרכזיים והערך שהמערכת מספקת למשתמשים.
  - דיאגרמת **SD1**
  - שלוש דיאגרמת נוספות אשר **יפרטו 3 תהליכים המופיעים ב SD1** (SD1.1; SD1.2; SD1.3) – הפירוט יכול להיות לתהליכים סינכרוניים (in-zooming) ו/או לתהליכים א-סינכרוניים (unfolding).
  - תיאור **המבנה של שתי ישויות מידע** מרכזיות (unfolding).

Model Evaluation

- Model clarity & understandability
- Model completeness
- Model correctness
- Documentation

Model clarity & understandability

- מודל שמבהיר מה עושה המערכת וכל תהליך בה
  - דיאגרמות לא עמוסות בתהליכים ואובייקטים – רק מה שנדרש בכל דיאגרמה
    - שימוש נבון ב scaling
      - state suppressing
      - in-zooming
      - unfolding
  - שמות משמעותיים שיוצרים משפטי OPL מובנים
  - הימנעות מספגטי של קשרים
  - פונטים קריאים
  - שימוש נבון בצבעים, גדלים, הערות וכד'
  - מרחקים נוחים בין "הדברים" (לא צפוף ולא מפוזר מידי)

Model completeness

- תאימות בין הדרישות למודל (אין דרישות שלא ברור היכן הם מתממשות, אין מימוש של דרישה שאיננה)
- תאימות בין רמות שונות של המודל (שימוש נכון בקדימויות/עדיפויות של קשרים)
- כל מה שהתבקשתם להגיש מופיע
  - 3 רמות של דיאגרמות (SD; SD1; 3*SD1.X)
  - הצגת דיאגרמה ייעודית עם המבנה של אובייקטים מרכזיים
  - כל מרכיבי האובייקט הידועים עד כה מופיעים בדיאגרמה ז

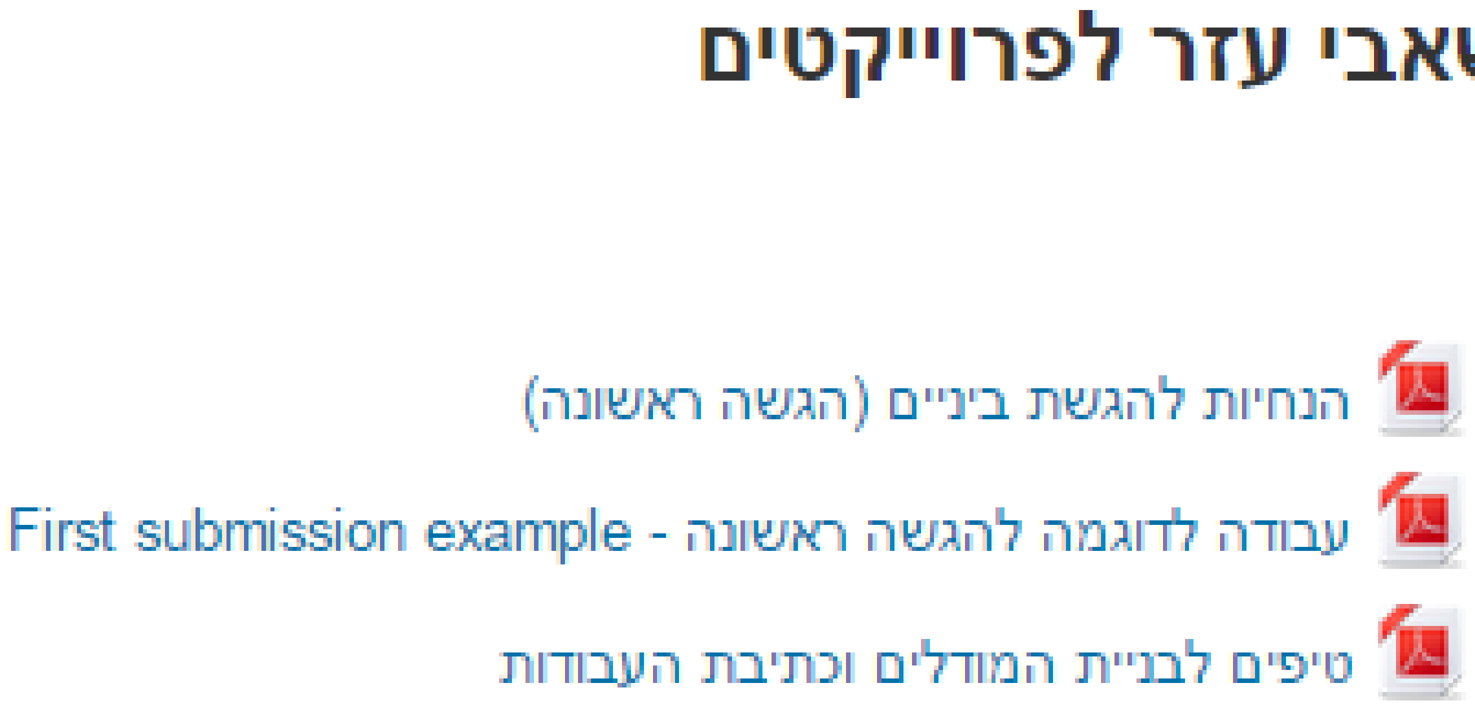
Model correctness

- שימוש נכון במבנה סינכרוני ואסנכרוני של תהליכים
- שמות תואמים לקונבנצית השמות
- כל התהליכים מחוברים לאובייקטים (ברור מה כל תהליך עושה)
- שימוש נכון בדברים פיזיים ומידעיים, מערכתיים וסביבתיים
- שימוש נכון באובייקטים בתוך תהליך ומחוצה לו.
- שימוש נכון בקשרים תהליכיים
- שימוש נכון בקשרים מבניים
- מימוש נכון של הדרישות – המערכת מתנהגת על פי מה שמבוקש

Documentation

- תיאור מילולי קצר של המערכת
- הצגת התיחום של המערכת עם הסבר לגבי המיקוד
- תיאור מילולי קצר של כל דיאגרמה – מה היא מכילה
- הסבר של שיקולי דעת במקומות רלוונטיים
- הוספת הערות מקומיות היכן שנדרש
- ניסוחים ברורים וקולחים – נוח לקריאה
- לא להכביר במילים – לכתוב רק מה שיש לו ערך מוסף לקריאה.

הנחיות ועזרה נוספת – באתר הקורס



משאבי עזר לפרוייקטים

- הנחיות להגשת ביניים (הגשה ראשונה)
- עבודה לדוגמה להגשה ראשונה - First submission example
- טיפים לבניית המודלים וכתיבת העבודות

יעוץ ושאלות

- פורום השאלות באתר הקורס עומד לרשותכם

- בשיעור הבא – מפגשי יעוץ קבוצתיים
  - יש להירשם לפגישת הייעוץ – עקבו אחר ההנחיות שיופצו במהלך השבוע.
  - יש להגיע מוכנים ולהכין שאלות מראש