Metamodellek a szoftverfejlesztésben

Action Semantics (AS)

Akció szemantika

- Action
  - a bemeneteket kimenetekké alakítja
- Akció szemantika
  - a számítástechnikai környezetet kivonjuk
    - végrehajtó gép, PL, implementáció
    - task struktúra, komponensek
    - végrehajtható specifikációt készíthetünk

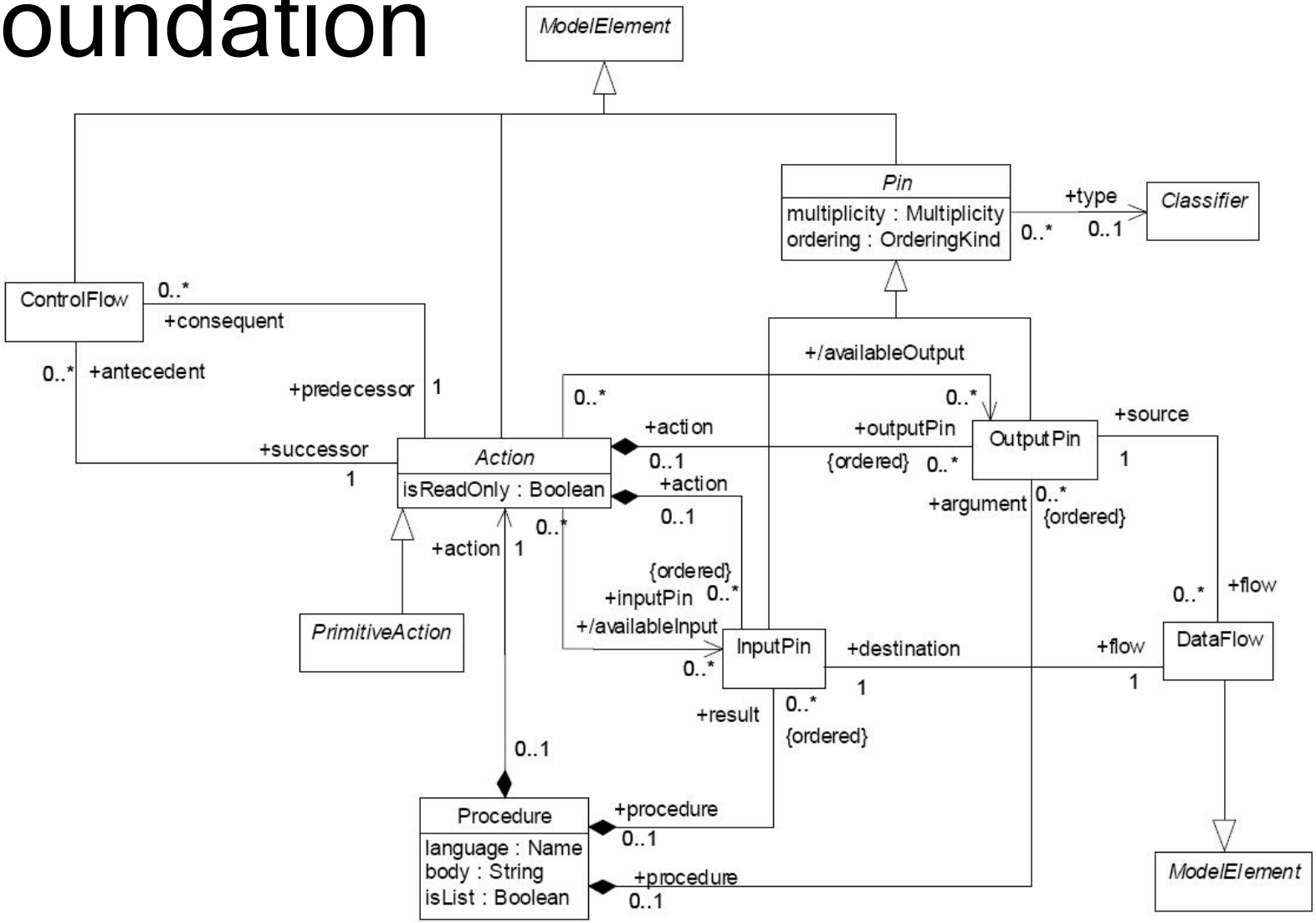
Akció szemantika

- Nincs normatív jelölésrendszer
- Vannak konkrét termekék
  - Action Specification Language (ASL)
  - BridgePoint Action Language(AL)
  - Kabira Action Semantics (Kabira AS)
  - action language subset of the Specification and Description Language (SDL)
  - nem koherensek az OCL-lel

Akció szemantika

- AS nyelv OCL-en (próbálkozás)
  - pl. Action Semantics Surface Language based on OCL Queries (ASOQ)
- Action Semantics
  - UML 1.5 (2002 Sept)
- Kód helyett model – model compiler
- Végrehajtható analízis model - xUML

Foundation


ModelElement
Pin
multiplicity : Multiplicity
ordering : OrderingKind
+type
0..1
0..*
Classifier
ControlFlow
0..*
+consequent
0..*
+antecedent
+predecessor
1
+successor
1
+/availableOutput
0..*
0..*
+action
0..1
Action
isReadOnly : Boolean
+action
0..1
0..*
+action
1
PrimitiveAction
{ordered}
+inputPin
0..*
+/availableInput
0..*
InputPin
+destination
1
+result
0..*
{ordered}
0..1
Procedure
language : Name
body : String
isList : Boolean
+procedure
0..1
+procedure
0..1
+outputPin
{ordered}
0..*
OutputPin
+source
1
+argument
0..*
{ordered}
0..*
+flow
+flow
1
DataFlow
ModelElement

Foundation

- Pin
  - kapcsolat a külvilággal – nincs hátsó ajtó !
  - típust és multiplicitást definiál – érték
  - kompatibilitás:
    - primitív és enumeráció – szigorú
    - egyéb – oo kompatibilitás

Foundation

- Data flow
  - kimenet más akció(k) bementére kerül, tárolás nélkül (src = out, dest = in)
  - fan out >1 mehet, de fan in>1 tilos
  - A cél pin kompatibilis a forrás pin-nel

Foundation

- Control flow
  - akciók közötti sorrend definiálása
  - a megelőző akciónak be kell fejeződnie a következő megkezdése előtt
  - a data flow által implikált sorrendet nem kell control flow-ra alakítani
  - data és control flow az akciók aciklikus irányított gráfját adját

Foundation

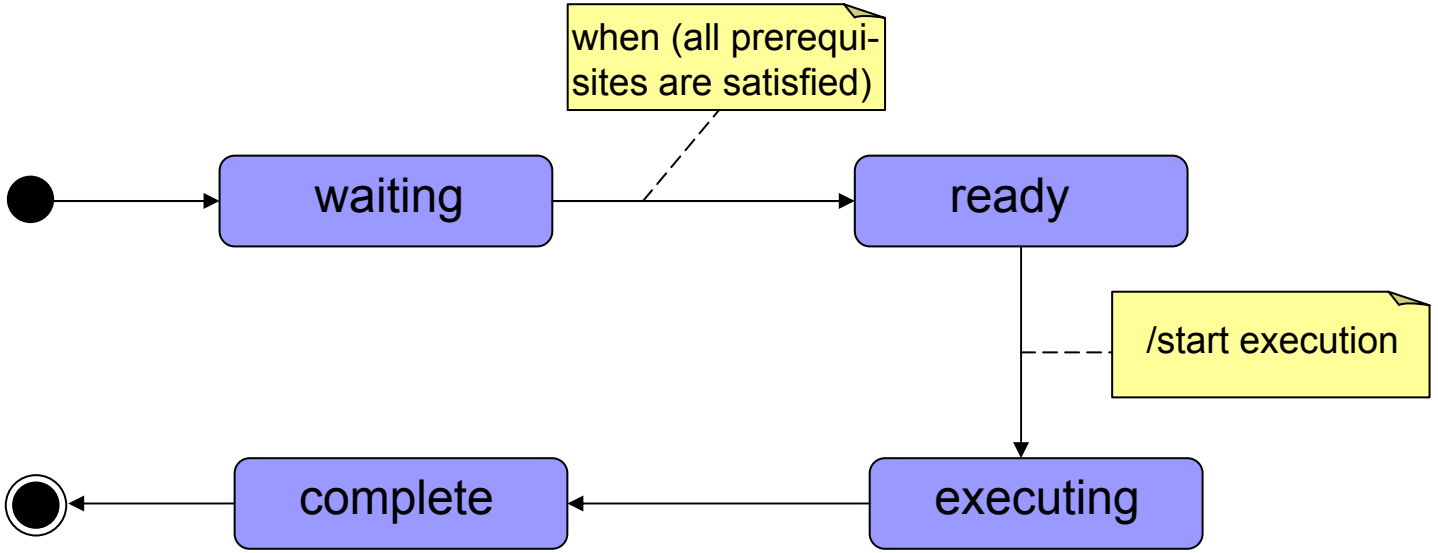
- Procedure
  - akciók halmaza, amely egységként kezelt
    - pl: metódustörzs
  - egy üzenet érkezésekor megkapja a request objectet
  - request object – pin vagy pin lista
  - result object

Foundation

- Host
  - a végrehajtás környezete
  - az éppen végrehajtandó metódust birtokló objektum példánya
  - a végrehajtás közben változatlan
  - static metódusnak nincs hostja

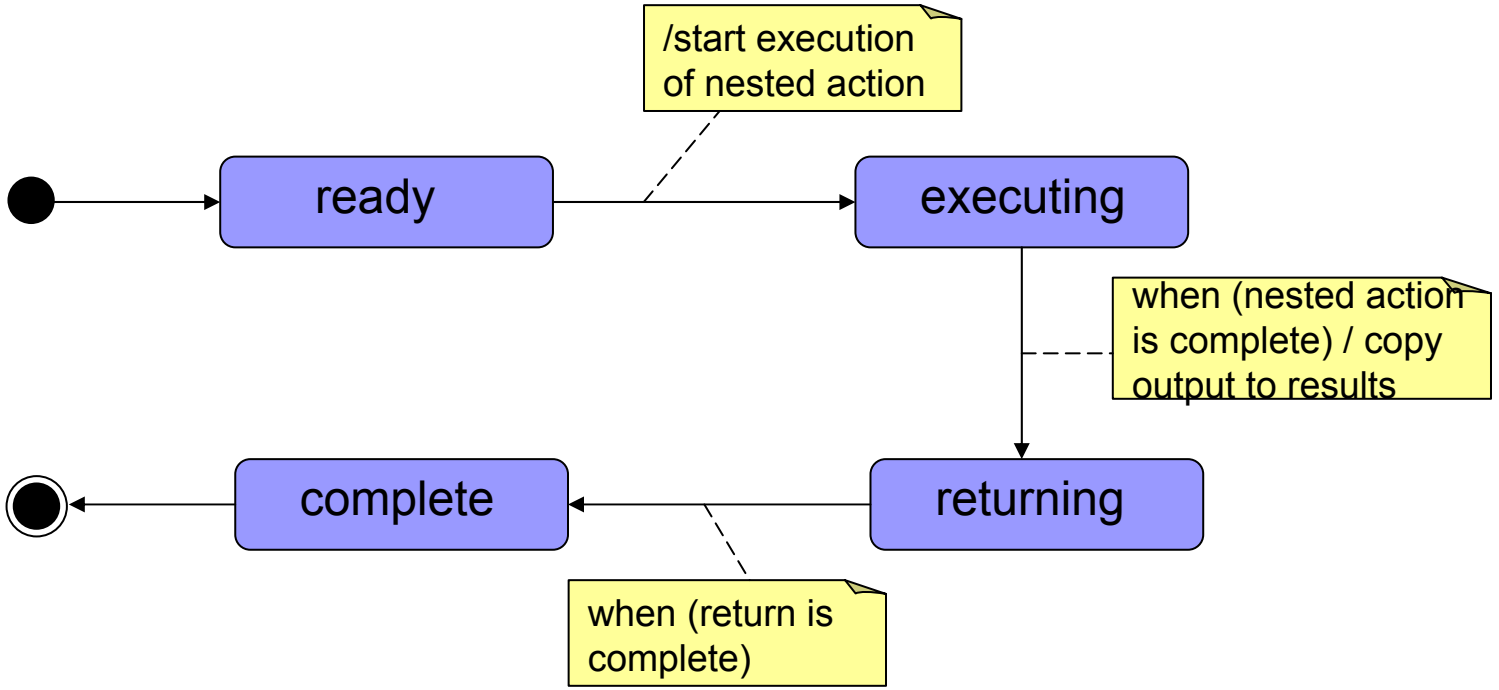
Foundation

- action execution

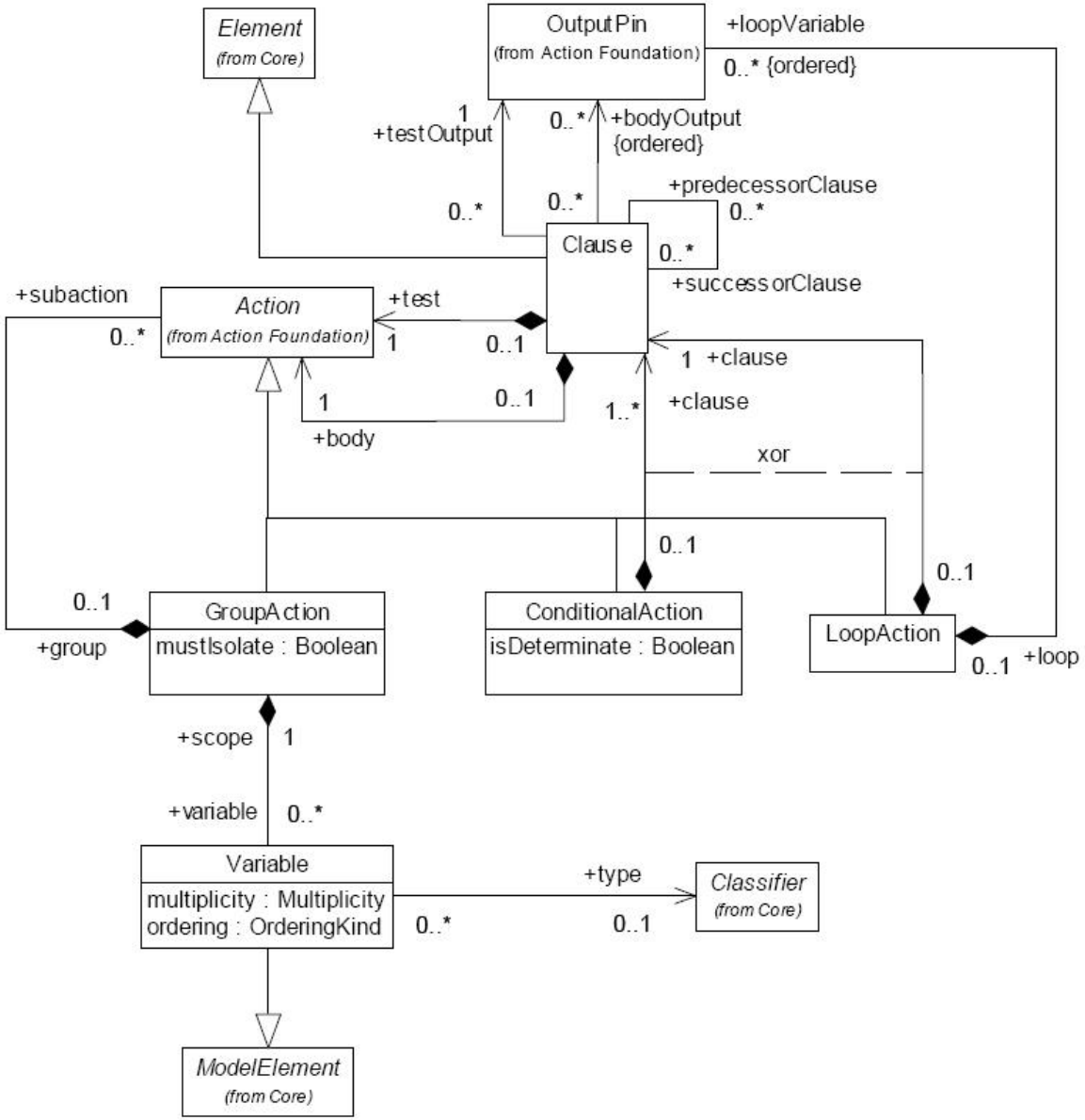


Foundation

- procedure execution



Composite


Element
(from Core)
OutputPin
(from Action Foundation)
+loopVariable
0..* {ordered}
1
+testOutput
0..*
+bodyOutput
{ordered}
+predecessorClause
0..*
0..*
0..*
Clause
0..*
+successorClause
+subaction
Action
(from Action Foundation)
+test
0..*
1
0..1
1 +clause
1
+body
0..1
1..*
+clause
xor
0..1
0..1
0..1
GroupAction
mustIsolate : Boolean
ConditionalAction
isDeterminate : Boolean
LoopAction
+group
0..1
0..1
+loop
+scope
1
+variable
0..*
Variable
multiplicity : Multiplicity
ordering : OrderingKind
+type
0..*
0..1
Classifier
(from Core)
ModelElement
(from Core)

Composite

- összetett és primitív akciókból áll
- GroupAction – csoportosítás
- ConditionalAction – futás közbeni feltételtől függő akciók
- LoopAction – futás közbeni feltételtől függő ismételt végrehajtás

Composite

- Available Inputs and Outputs
  - az összetett akciónak lehetnek saját pin-jei
  - lehetnek az al-akcióknak a külvilággal kapcsolatos pinjei
  - így a szerkezet black-box lesz
- Group Action
  - a fenti elv a control flow-ra is érvényes

Composite

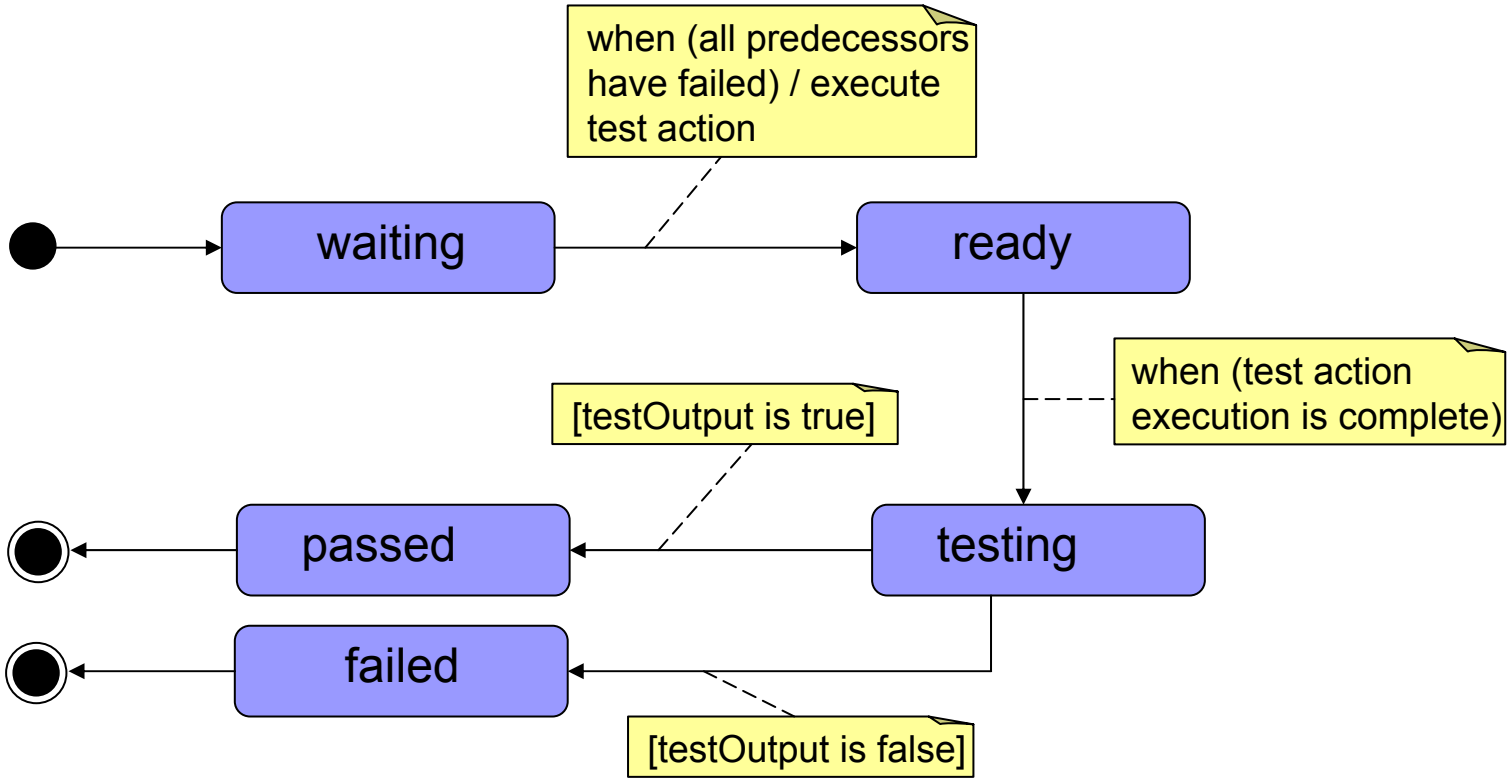
- Conditional Action
  - klóz-ok halmaza
  - klóz = test + body action pár
  - test-hez tartozik egy test output pin
    - ha true, akkor a body végrehajtható
  - valamennyi body ugyanazon output pin halmazt írja
  - kompatibilitás és látszólagos fan-in
  - csak egy body hajtódik végre

Composite

- Conditional Action
  - nincs saját bemente
  - kimenete a body kimenete
  - body bemenetei:
    - külső inputok és a saját test outputjai
    - különböző klóz-ok test és body-ja nem kapcsolódik
  - klóz-ok között sorrend állítható + véletlen

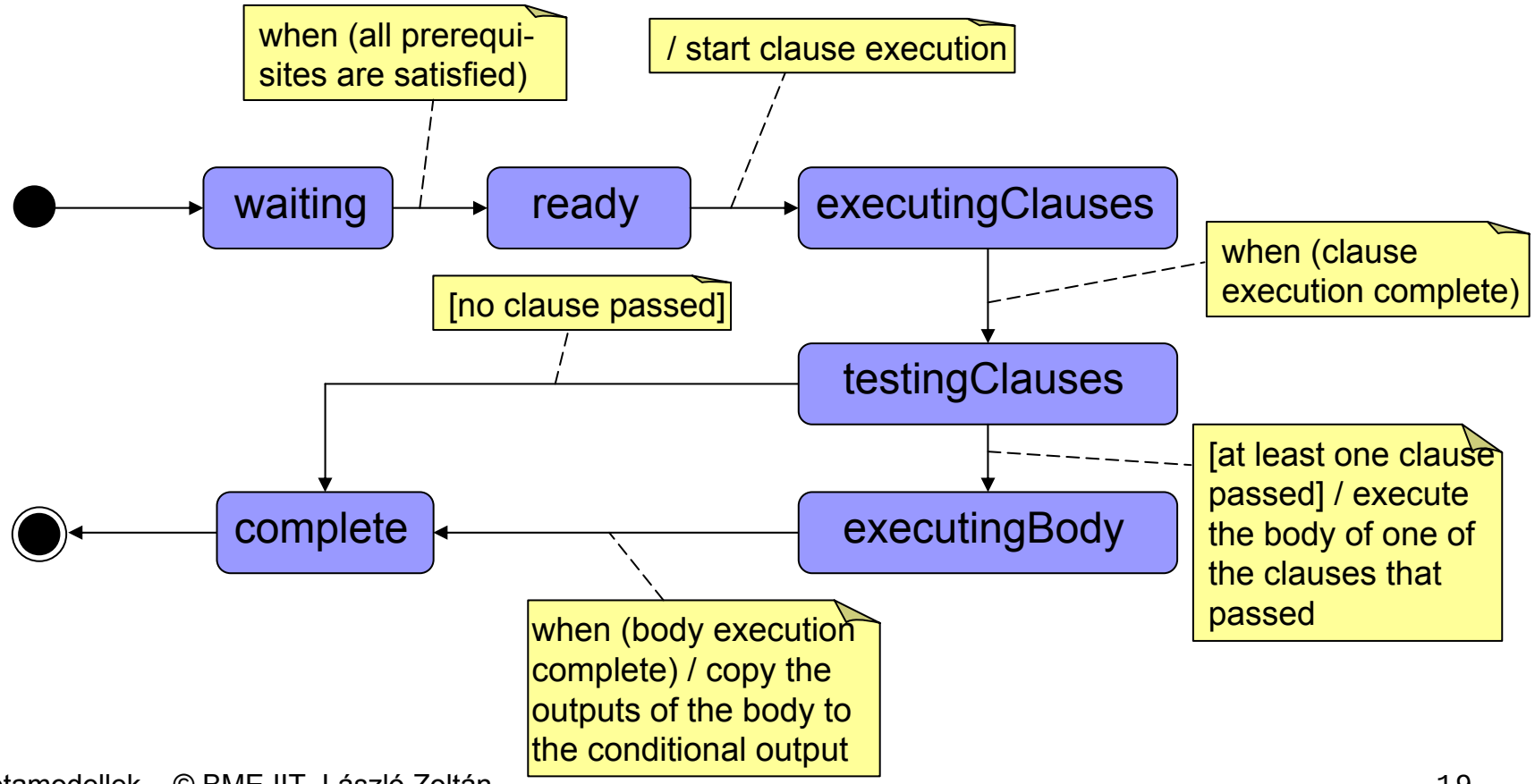
Composite

- klóz végrehajtás



Composite

- conditional végrehajtás

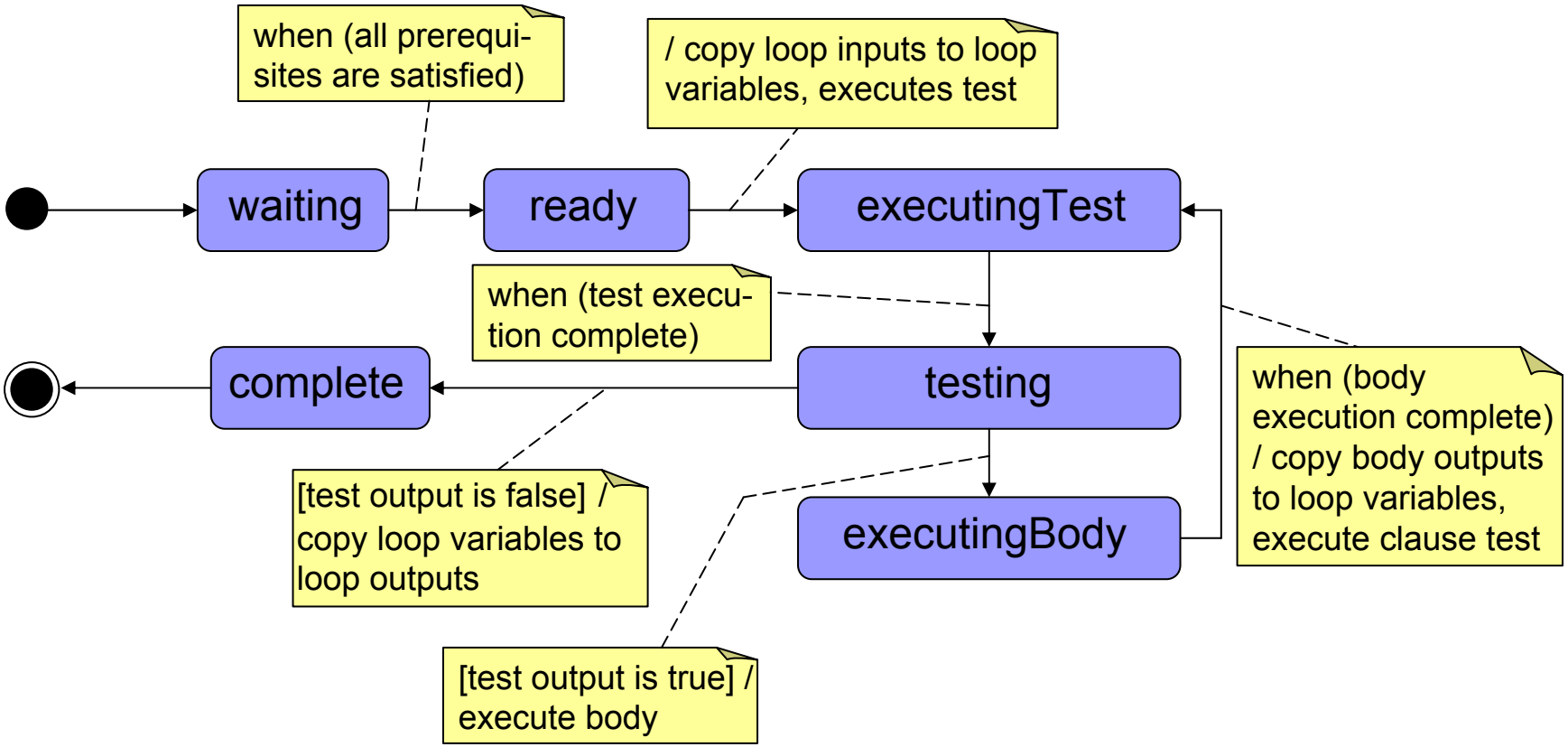


Composite

- Loop Action
  - egyetlen klóz-t tartalmaz
  - amíg a test igaz, addig a body-t végrehajtjuk
  - loop variables (output pins) rákötheők mind a test, mind a body input pin-jeire

Composite

- loop végrehajtás



Composite

- Local Variables
  - belső változó, kívülről nem látszik
  - az imperatív nyelvekben szokásosan
- Isolation
  - hagyományos "tranzakció"

Read & Write Actions

- Object Actions
  - CreateObjectAction
  - DestroyObjectAction
- Attribute Actions
  - Read AttributeAction
  - WriteAttributeAction

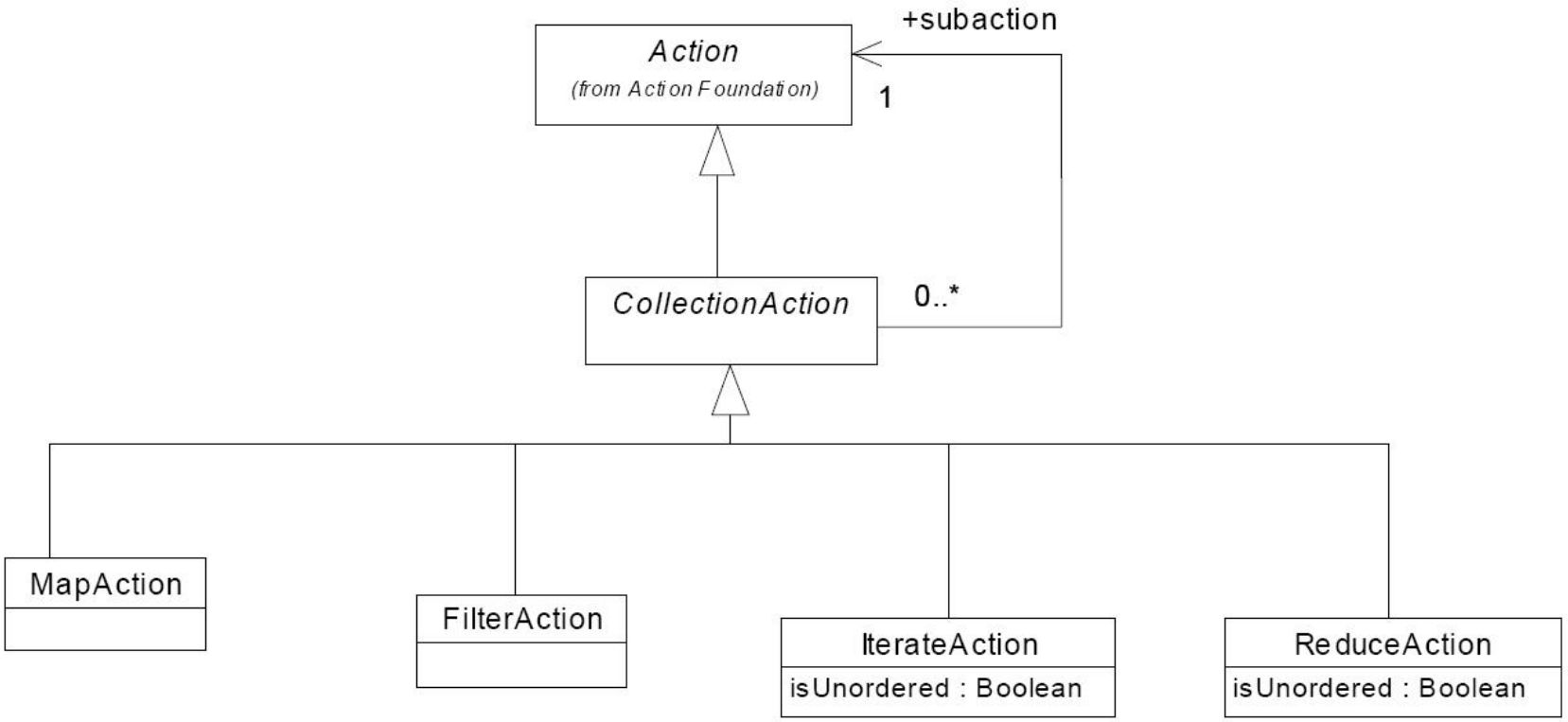
Read & Write Actions

- Association Actions
  - link önmagában nem kezelhető
  - a kapcsolódó végek érdekesek
  - navigálás, kapcsolat építés, törlés
- Variable Actions
  - olvasás, írás

Computation Actions

- Számítások
  - ApplyFunctionAction
  - ArgumentSpecification
  - CodeAction
    - kilépés az UML-ből
  - LiteralValueAction
  - MarshalAction - UnmarshalAction

Collection


Action
(from Action Foundation)
+subaction
1
CollectionAction
0..*
MapAction
FilterAction
IterateAction
isUnordered : Boolean
ReduceAction
isUnordered : Boolean

Collection Actions

- Filter
  - a kollekció minden elemét ugyanazon függvény alapján szűrjük - párhuzamos
- Iterate
  - minden elemen ugyanazt az akciót hajtjuk végre, loop variable-t használunk akkumulátorként - soros

Collection Actions

- Map
  - mint filter, de nem csak szűr az action - párhuzamos
- Reduce
  - a kollekció egymást követő elemeit "eredmény"-nyel helyettesítjük, amíg skalárt nem kapunk

Messaging Actions

- Egy objektum egy request-et (reply-t) küld másik (target) objektum(ok)nak
- A fogadó a requestnek megfelelő procedure-t hajtja végre (resolution)
- Procedure célja
  - operáció végrehajtása
  - állapotátmenet triggerelése

Messaging Actions

- Aszinkron hívás
  - kérő request-et készít és elküldi célnak
  - kérő folytatja működését
  - a request átmegy a közegen
  - request megérkezik célba és vár
  - a cél feldolgozza a requestet
  - a request típusa alapján procedure-t keres
  - procedure-t végrehajt

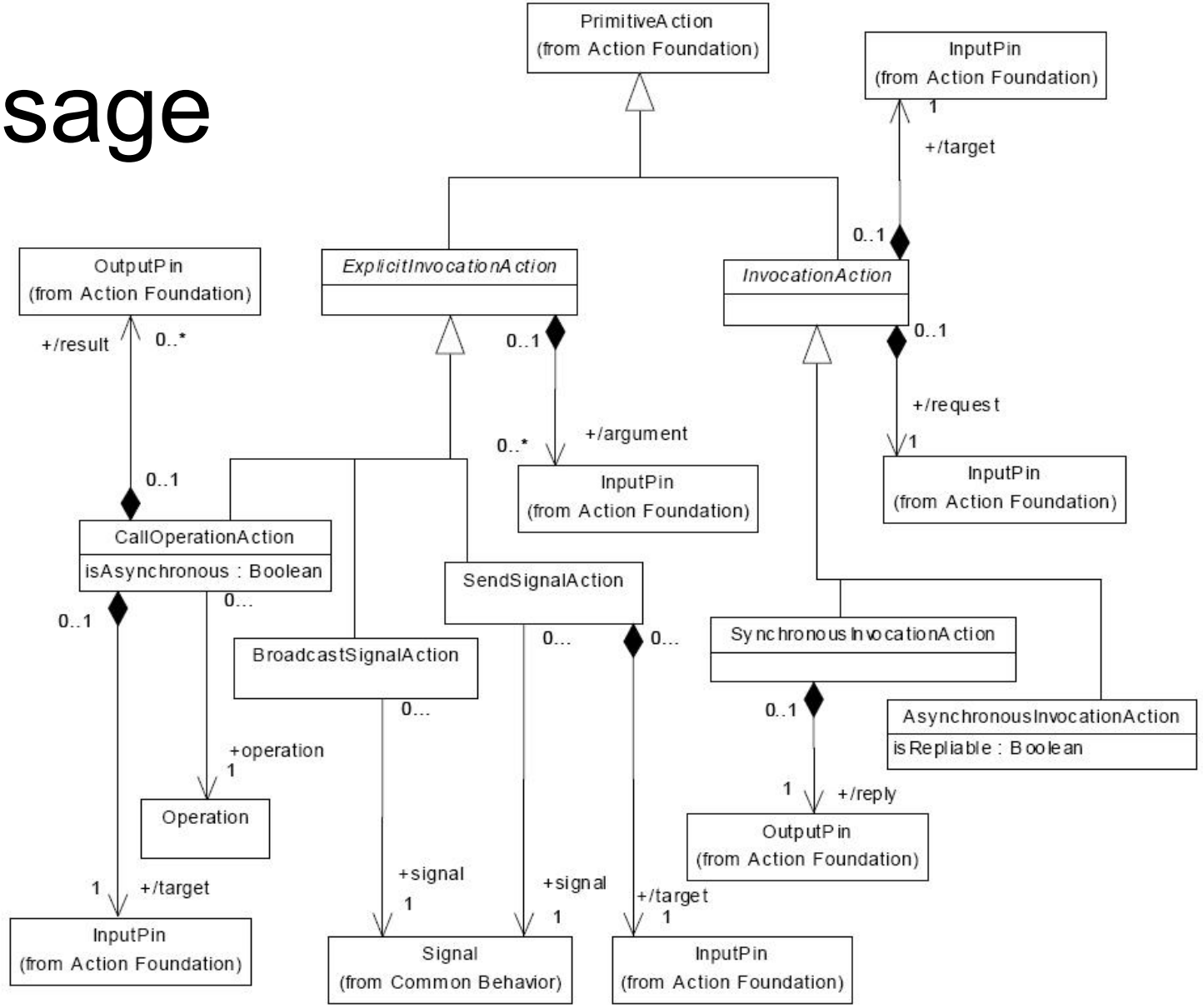
Messaging Actions

- Szinkron hívás
  - kérő request-et készít és elküldi célnak, majd blokkolódik
  - a request átmegy a közegen
  - request megérkezik célba és vár
  - a cél feldolgozza a requestet
  - a request típusa alapján procedure-t keres
  - procedure-t végrehajt, közben reply-t készít, elküld
  - a reply átmegy a közegen
  - megérkezik a blokkolt kérőhöz, aki folytatódik

Messaging Actions

- Request kezelése
  - a cél objektum határozza meg
  - szekvenciális (guarded) vagy konkurens
  - sorbanállás saját hatáskörben
  - reply küldése lehet külön akció vagy implicit
  - reply visszatalál
  - üzenet – tiszta érték (adat vagy referencia)

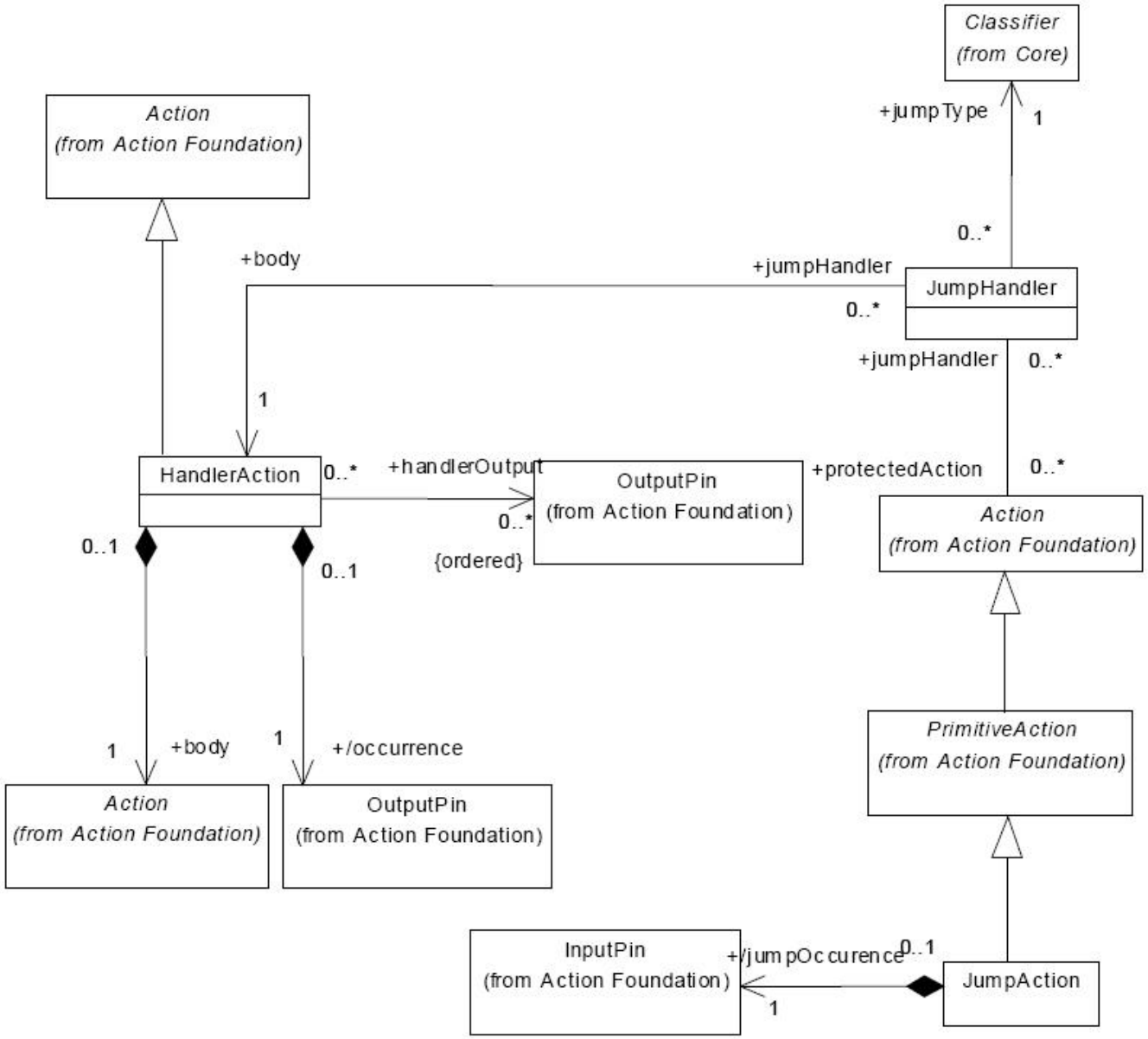
Message


PrimitiveAction
(from Action Foundation)
InputPin
(from Action Foundation)
1
+/target
0..1
OutputPin
(from Action Foundation)
ExplicitInvocationAction
InvocationAction
+/result
0..*
0..1
0..1
+/request
1
0..*
+/argument
InputPin
(from Action Foundation)
InputPin
(from Action Foundation)
0..1
CallOperationAction
isAsynchronous : Boolean
SendSignalAction
SynchronousInvocationAction
0..1
0...
0...
0...
BroadcastSignalAction
0...
0..1
AsynchronousInvocationAction
isRepliable : Boolean
+operation
1
1
+/reply
Operation
OutputPin
(from Action Foundation)
1
+/target
+signal
1
+signal
1
+/target
1
InputPin
(from Action Foundation)
Signal
(from Common Behavior)
InputPin
(from Action Foundation)

Jump Actions

- eltérés a normál végrehajtási sorrendtől
  - futó akció félbeszakad
  - alternatív akció(k) hajtódik végre (ha van)
  - folytatódik a félbeszakadt akciót követő akcióval (output pin !)
- Jump type – tetszőleges
  - példánya: mi a jump oka
  - átadjuk a Jump Handlernek

Jump


Classifier
(from Core)
+jumpType
1
0..*
Action
(from Action Foundation)
+body
+jumpHandler
JumpHandler
0..*
+jumpHandler
0..*
1
HandlerAction
0..*
+handlerOutput
OutputPin
(from Action Foundation)
+protectedAction
0..*
0..*
{ordered}
Action
(from Action Foundation)
0..1
0..1
PrimitiveAction
(from Action Foundation)
1
+body
1
+/occurrence
Action
(from Action Foundation)
OutputPin
(from Action Foundation)
InputPin
(from Action Foundation)
+/jumpOccurence
0..1
1
JumpAction

Jump Actions

- Jump Handler hozzáköt Handler akciókat
  - ha nincs akció, akkor feljebb propagál
  - legkülső szinten – reply
- handler output – normál eredmény
- occurance
  - a body számára elérhető jump példány