# *Formal Methods for Interactive Systems*

Part 6 — Cognitive Models

Antonio Cerone

*United Nations University*

*International Institute for Software Technology*

*Macau SAR China*

email: `antonio@iist.unu.edu`

web: `www.iist.unu.edu`

# *Analysis of Interactive Systems*

Purposes:

- Understand the internal cognitive processes of a person performing a task
  $\Longrightarrow$ analyse complexity, learnability, ...

# *Analysis of Interactive Systems*

Purposes:

- Understand the internal cognitive processes of a person performing a task
  $\Longrightarrow$ analyse complexity, learnability, ...
- Look at the observable behaviour of a person performing a task
  $\Longrightarrow$ analyse requirement capture, interface design, system design, documentation

# *Cognitive Models*

of the user

# *Cognitive Models*

of the user

- competence models represent expected behaviour

# *Cognitive Models*

of the user

- competence models represent expected behaviour
- performance models represent and analyse routine behaviour

# *Cognitive Models*

of the user

- competence models represent expected behaviour
- performance models represent and analyse routine behaviour

deal with three different levels:

- goal and task hierarchies: GOMS, Cognitive Complexity Theory (CCT)

# *Cognitive Models*

of the user

- competence models represent expected behaviour
- performance models represent and analyse routine behaviour

deal with three different levels:

- goal and task hierarchies: GOMS, Cognitive Complexity Theory (CCT)
- human understanding: BNF, Task-action Grammar (TAG)

# *Cognitive Models*

of the user

- competence models represent expected behaviour
- performance models represent and analyse routine behaviour

deal with three different levels:

- goal and task hierarchies: GOMS, Cognitive Complexity Theory (CCT)
- human understanding: BNF, Task-action Grammar (TAG)
- physical/device: Keystroke-level Model (KLM)

# *Model and Meta-models*

- developed when most interactive systems were command line or at most keybord and cursor based

# *Model and Meta-models*

- developed when most interactive systems were command line or at most keybord and cursor based
- how can they deal with windowed and mouse-driven interface

# *Model and Meta-models*

- developed when most interactive systems were command line or at most keybord and cursor based
- how can they deal with windowed and mouse-driven interface
- incorporate implicit and explicit models of cognitive processing

# *Model and Meta-models*

- developed when most interactive systems were command line or at most keybord and cursor based
- how can they deal with windowed and mouse-driven interface
- incorporate implicit and explicit models of cognitive processing
- tend to be competence models

# *Model and Meta-models*

- developed when most interactive systems were command line or at most keybord and cursor based
- how can they deal with windowed and mouse-driven interface
- incorporate implicit and explicit models of cognitive processing
- tend to be competence models
- seldom include the system

# *Model and Meta-models*

- developed when most interactive systems were command line or at most keybord and cursor based
- how can they deal with windowed and mouse-driven interface
- incorporate implicit and explicit models of cognitive processing
- tend to be competence models
- seldom include the system

$\Longrightarrow$ need for meta-model, expressive and flexible and able to address perfomance and be instantiated as overall system model

# *Formal Methods*

- Z and B
- Statecharts
- Petri nets
- Process Algebras
- Temporal Logic
- ...

# *FM Advantages*

- expressive and flexible

# *FM Advantages*

- expressive and flexible
- can describe both User and System

# *FM Advantages*

- expressive and flexible
- can describe both User and System
- address verification ($\Longrightarrow$ performance models)

# *FM Advantages*

- expressive and flexible
- can describe both User and System
- address verification ($\Longrightarrow$ performance models)
- address safety and security standards

# *Operator Choice Model*

Scanning: The operator searches the interface for a certain property.

# *Operator Choice Model*

**Scanning:** The operator searches the interface for a certain property.

**Identification:** The operator identifies part of the interface that may represent the property.

# *Operator Choice Model*

**Scanning:** The operator searches the interface for a certain property.

**Identification:** The operator identifies part of the interface that may represent the property.

**Classification:** The operator

- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

# *Operator Choice Model*

**Scanning:** The operator searches the interface for a certain property.

**Identification:** The operator identifies part of the interface that may represent the property.

**Classification:** The operator

- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

**Decision** on how to resolve the situation.

# *Operator Choice Model*

**Scanning:** The operator searches the interface for a certain property.

**Identification:** The operator identifies part of the interface that may represent the property.

**Classification:** The operator

- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

**Decision** on how to resolve the situation.

**Action** to be performed as a series of interaction with the interface.

# *OCM for Nuclear Plant*

**Scanning:** The operator scans among each of the individual reactor readouts on the interface searching for any anomalies.

**Identification:** The operator identifies a particular readout.

**Classification:** The operator

- assesses whether the identified readout describes a normal or abnormal operation of the plant;
- if abnormal, gives a priority to the operation according to its urgency to be resolved.

**Decision** on how to resolve the abnormal situation.

**Action** to be performed as a series of interaction with the interface and with internal and/or external authorities.

# *OCM for Air Traffic Control*

**Scanning:** The operator scans among each pair of aircraft searching for a pair that may violate separation.

**Identification:** The operator identifies a pair of aircraft.

**Classification:** The operator

- assesses whether the identified pair of aircraft will eventually violate separation (in conflict) or not (not in conflict);
- if so, gives a priority to the conflict according to its urgency to be resolved.

**Decision** on how to resolve the conflict.

**Action** to be performed as a series of interaction with the interface.

# *Operator Choice Model*

**Scanning:** The operator searches the interface for a certain property.

**Identification:** The operator identifies part of the interface that may represent the property.

**Classification:** The operator

- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

**Decision** on how to resolve the situation.

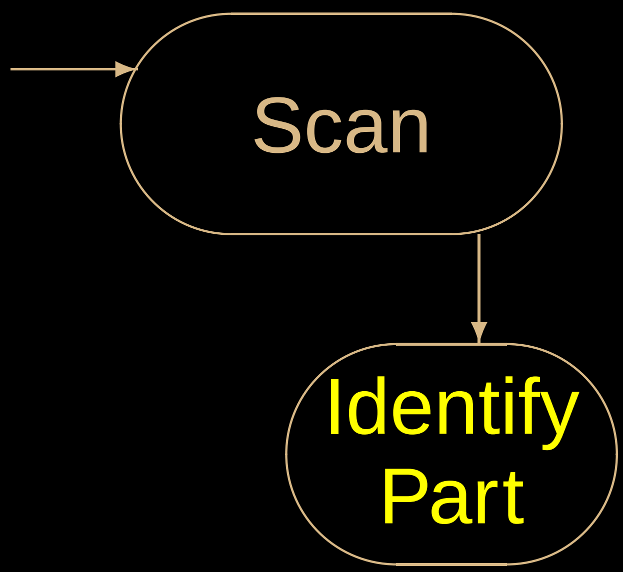
**Action** to be performed as a series of interaction with the interface.

# *Scanning*


Scan:
The operator searches the interface for a certain property.

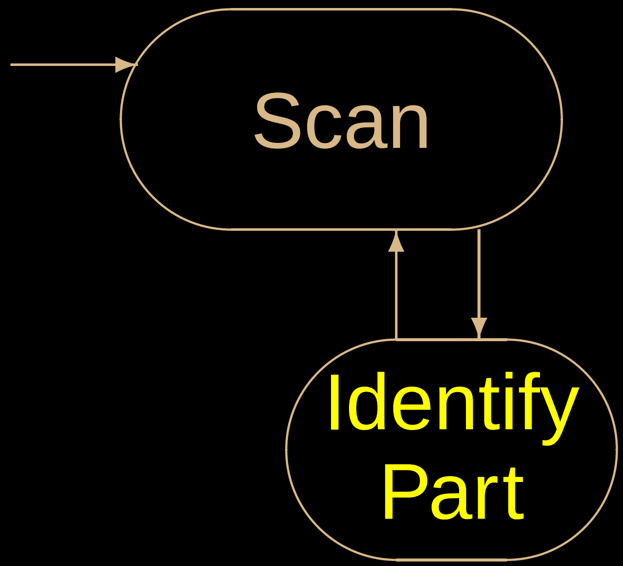
# *Identifying Part*

# *Identifying Part*



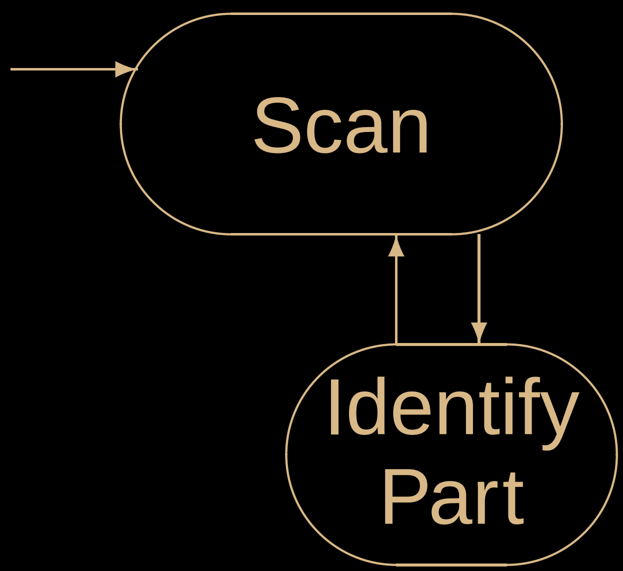

Identify Part:
The operator identifies a part $p$ of the interface that may represent the property of interest.

# *Identifying Part*

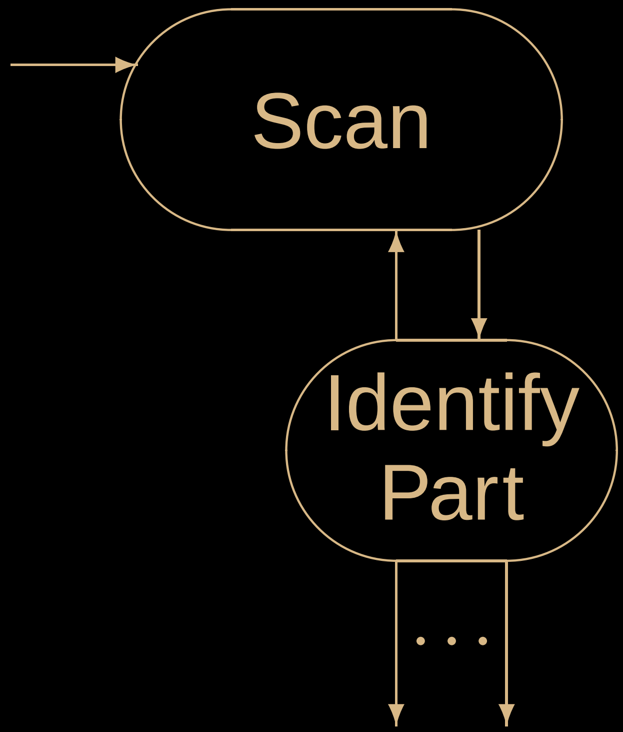



Identify Part:
The operator identifies a part $p$ of the interface that may represent the property of interest.

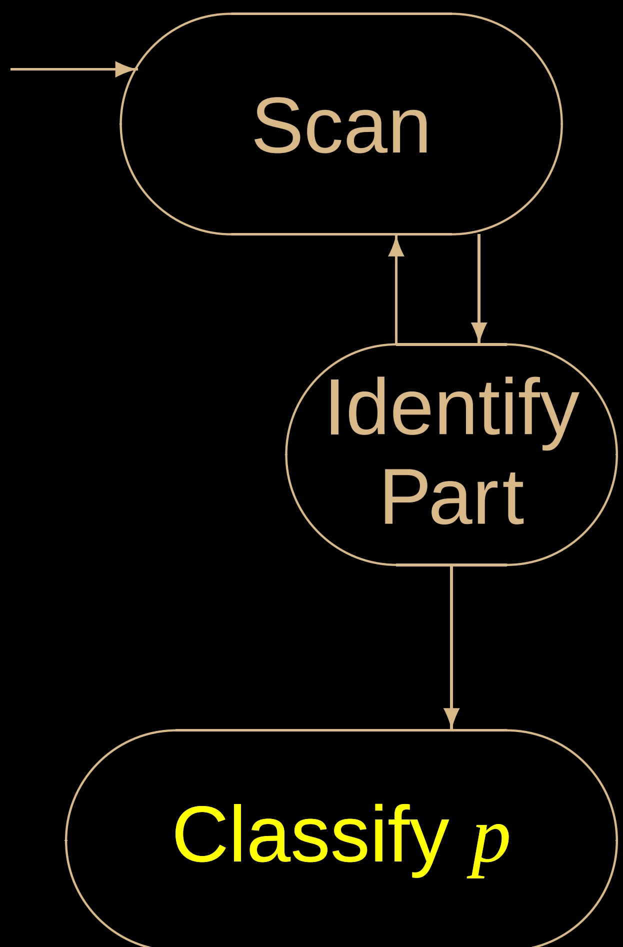
# *Classification*



# *Classification*

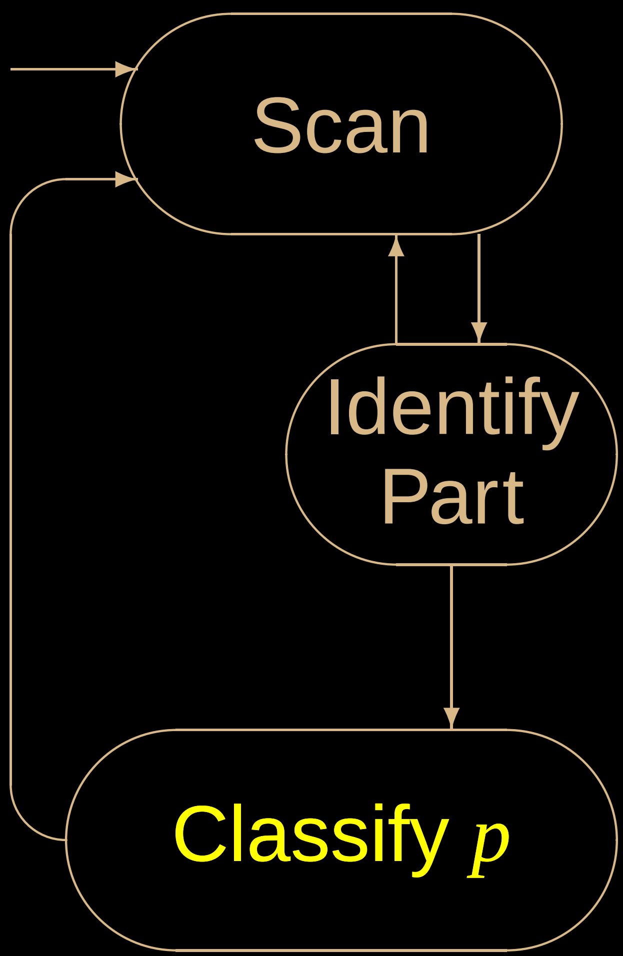


# *Classification*



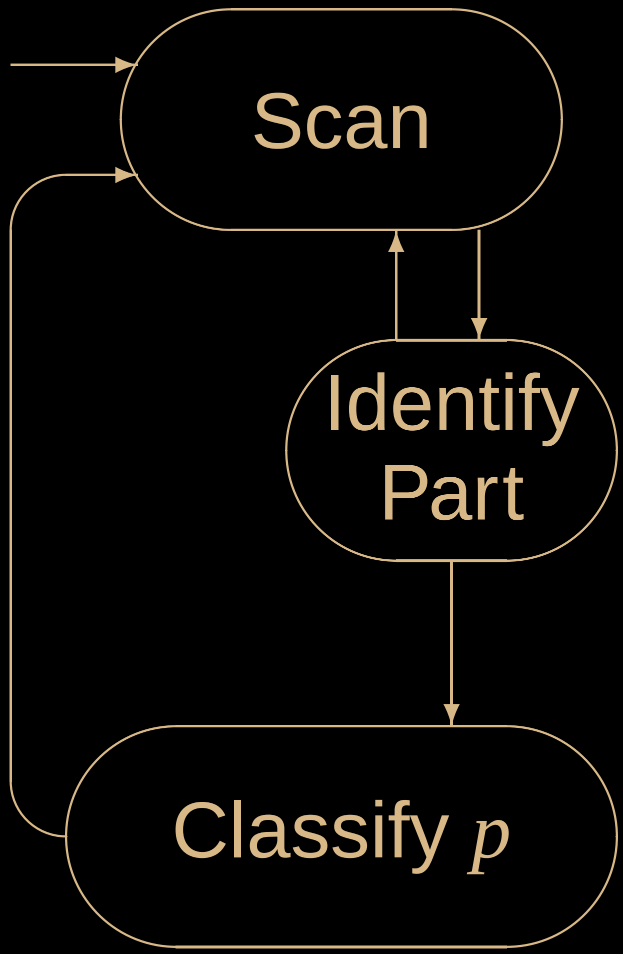

Classify $p$:
The operator
- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

# *Classification*

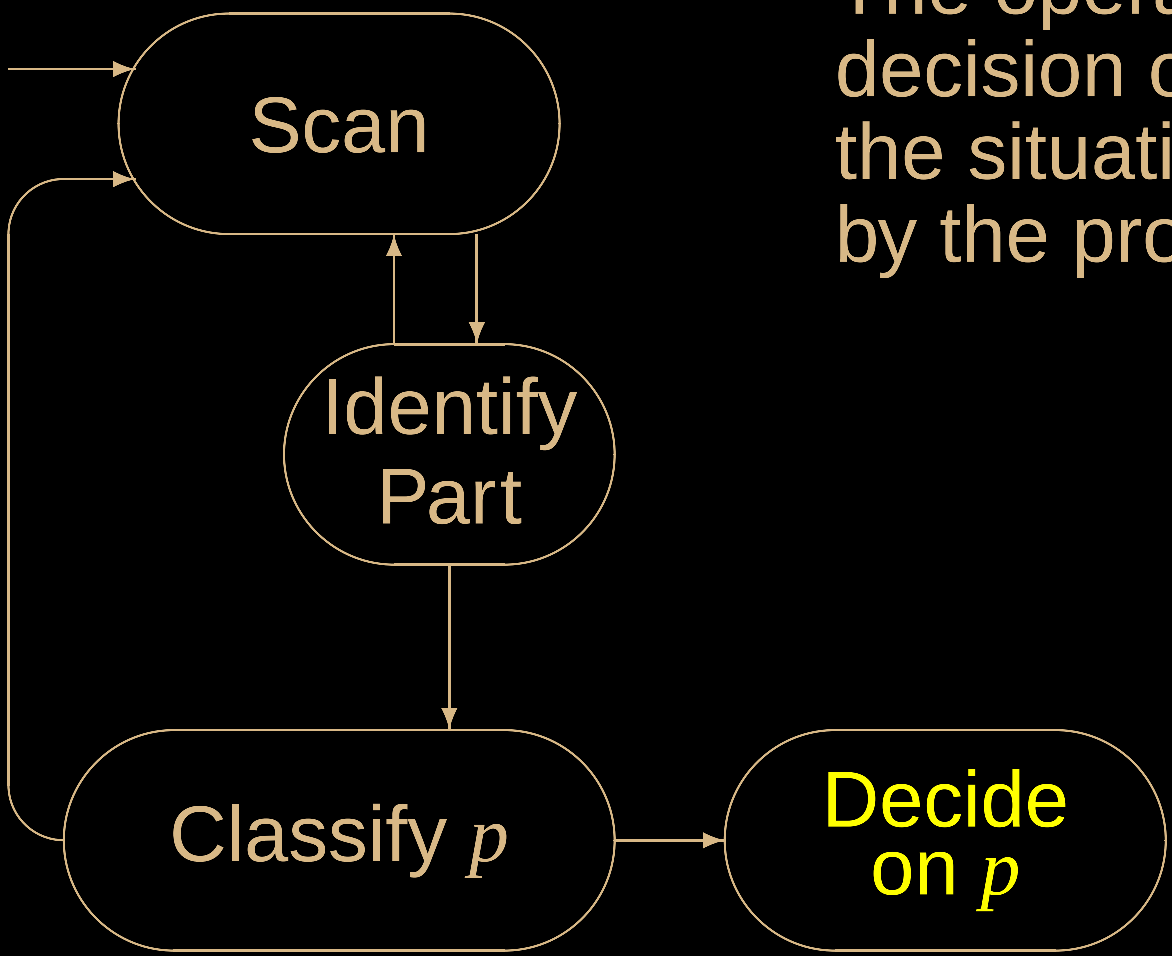



Classify $p$:
The operator
- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

# *Decision Making*

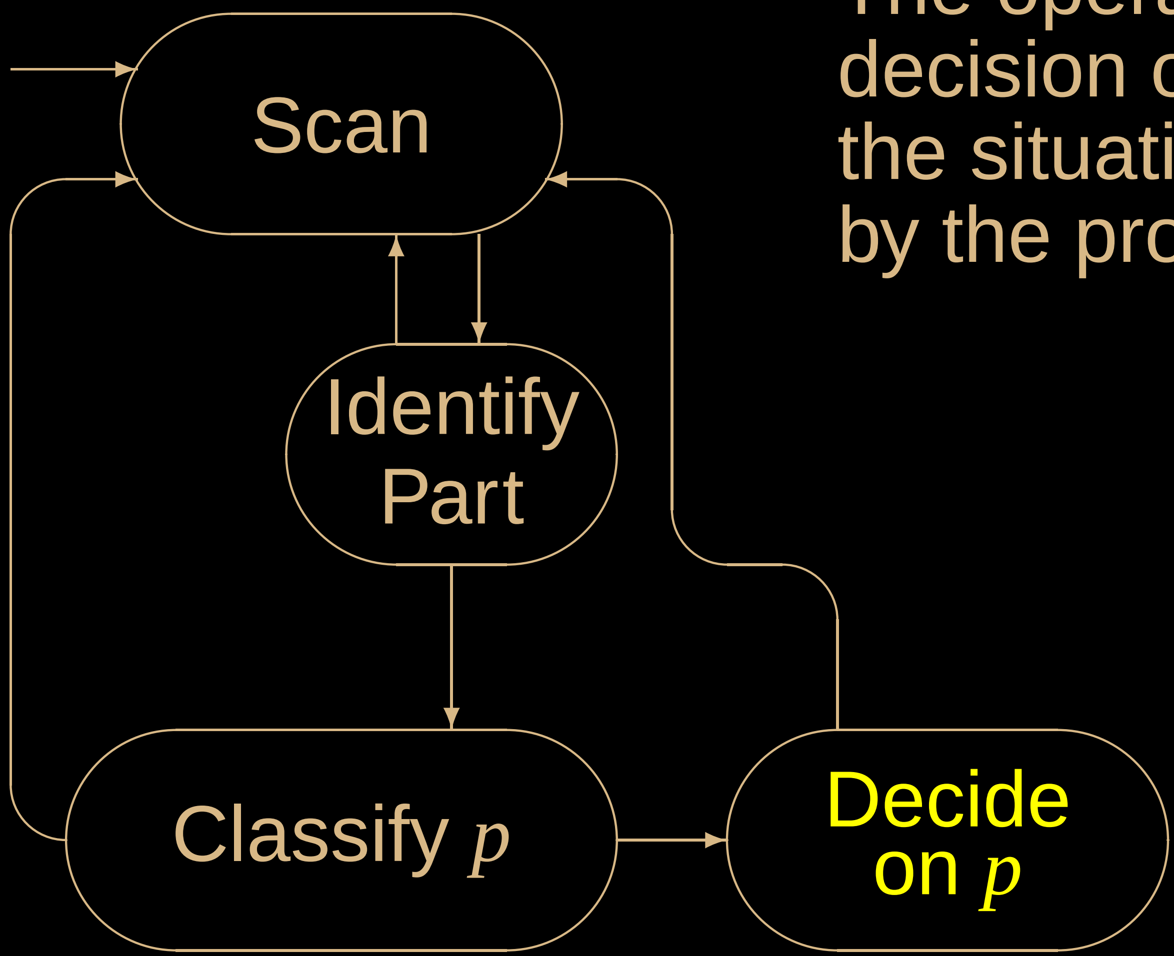


# *Decision Making*



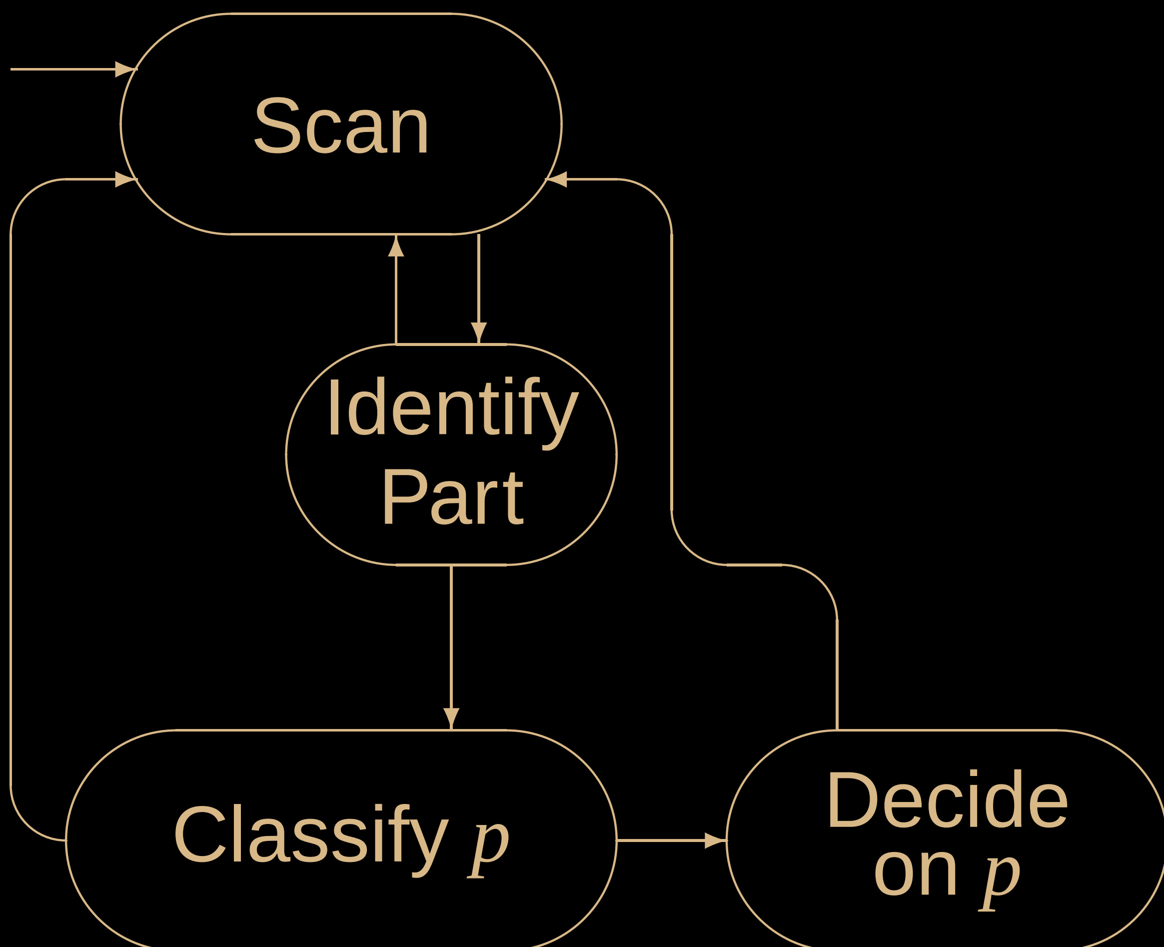

Make Decison on $p$:
The operator makes a decision on how to resolve the situation determined by the property of $p$.

# *Decision Making*



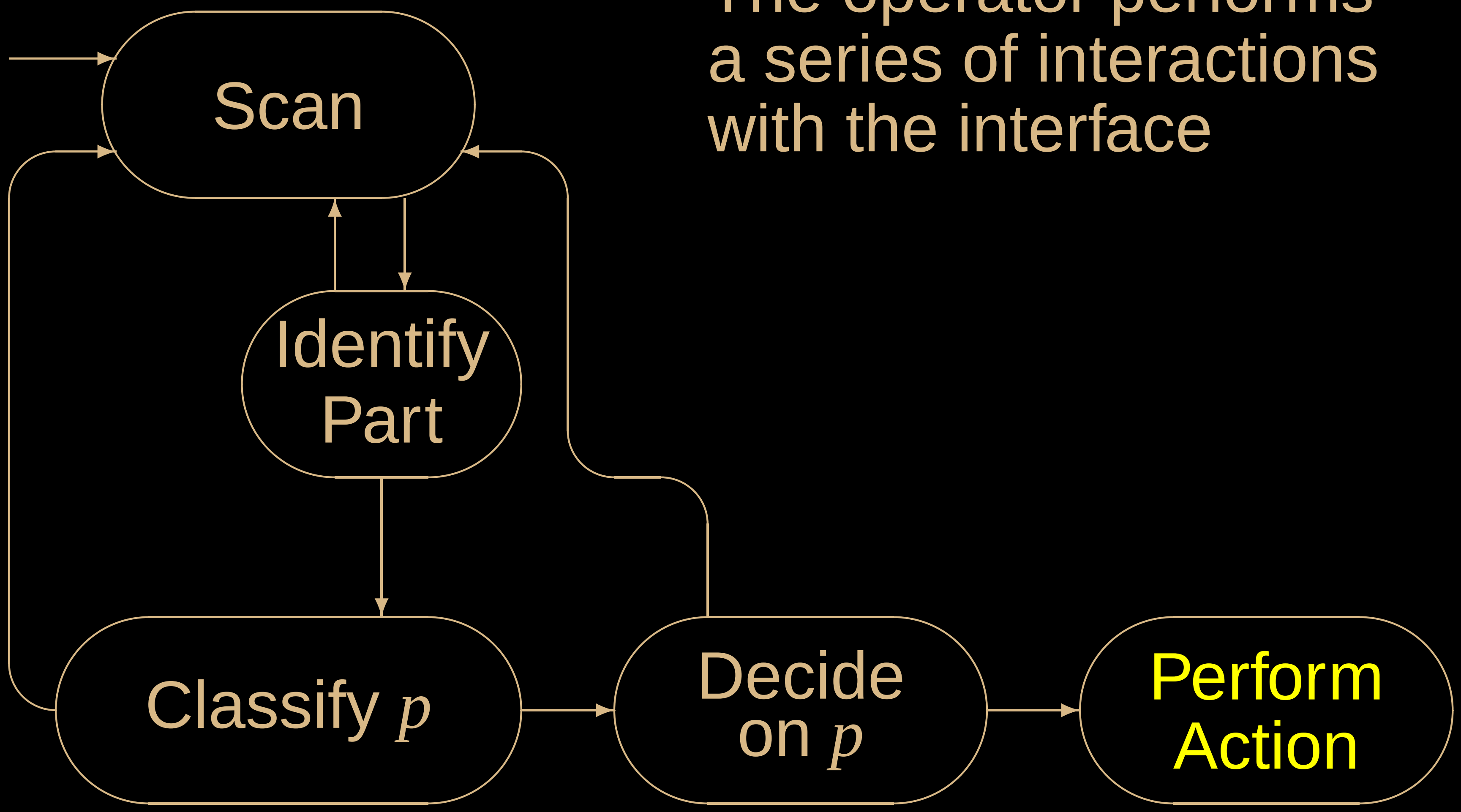

Make Decison on $p$:
The operator makes a decision on how to resolve the situation determined by the property of $p$.

# *Action*

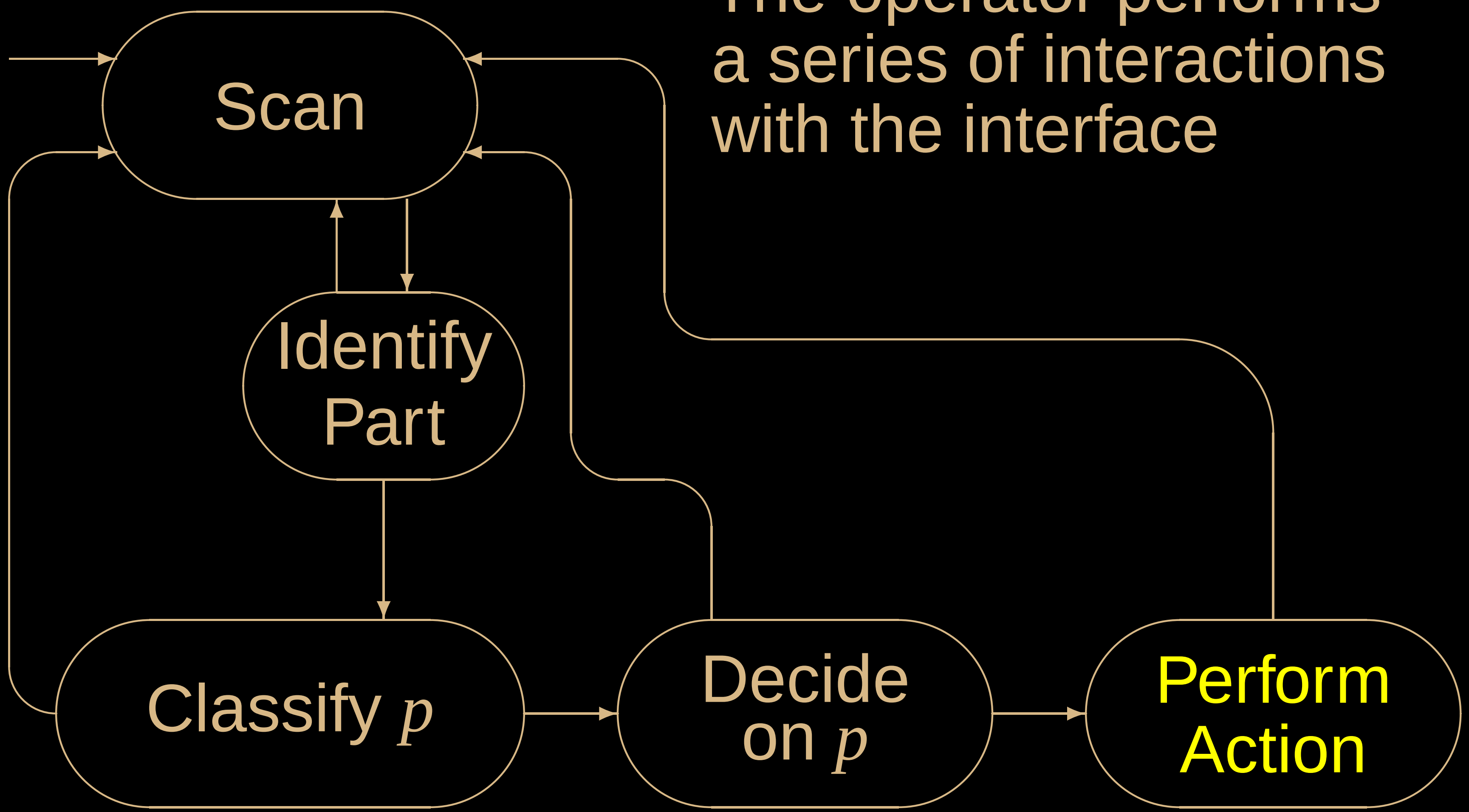


# *Action*

Perform Action:
The operator performs a series of interactions with the interface

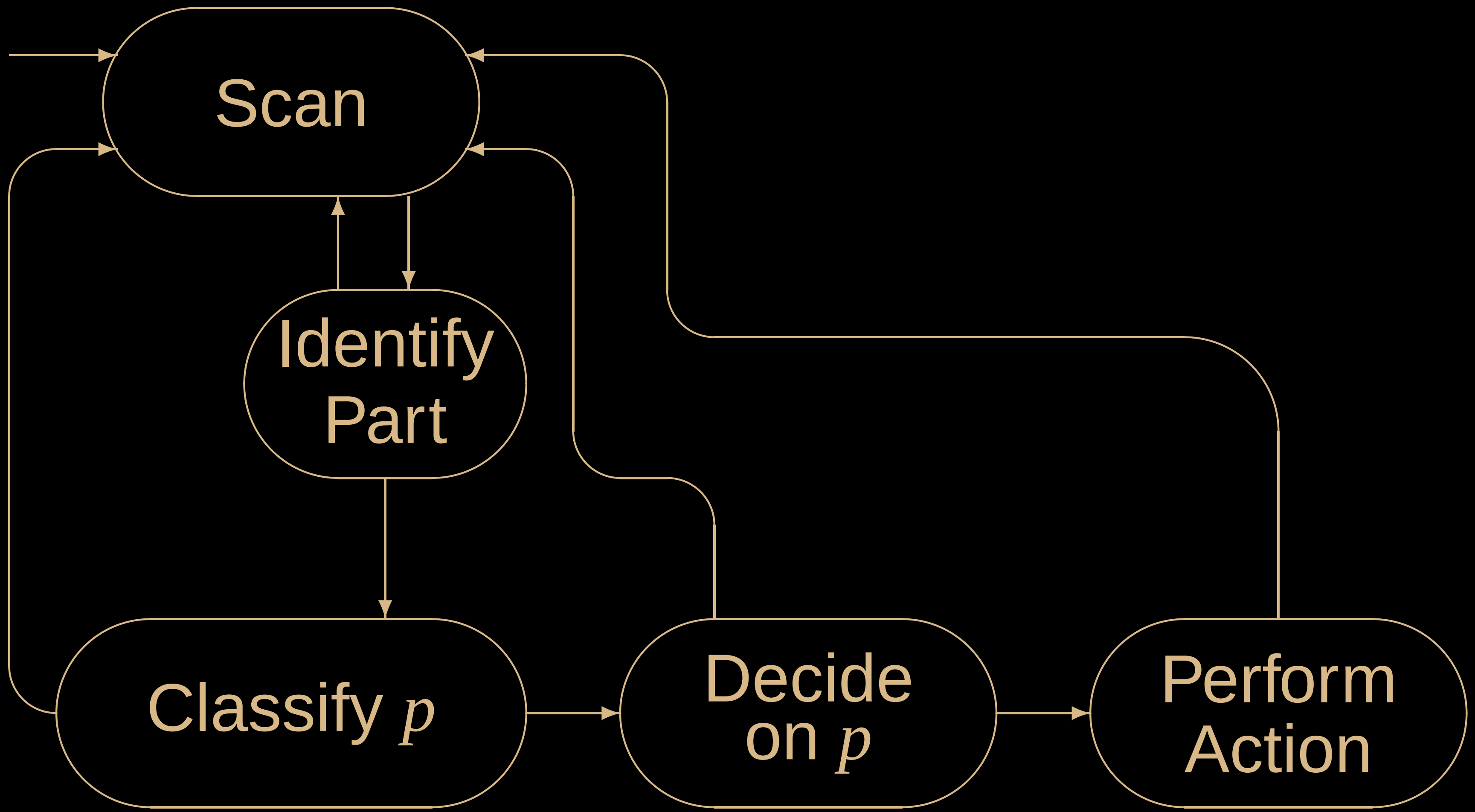


# *Action*



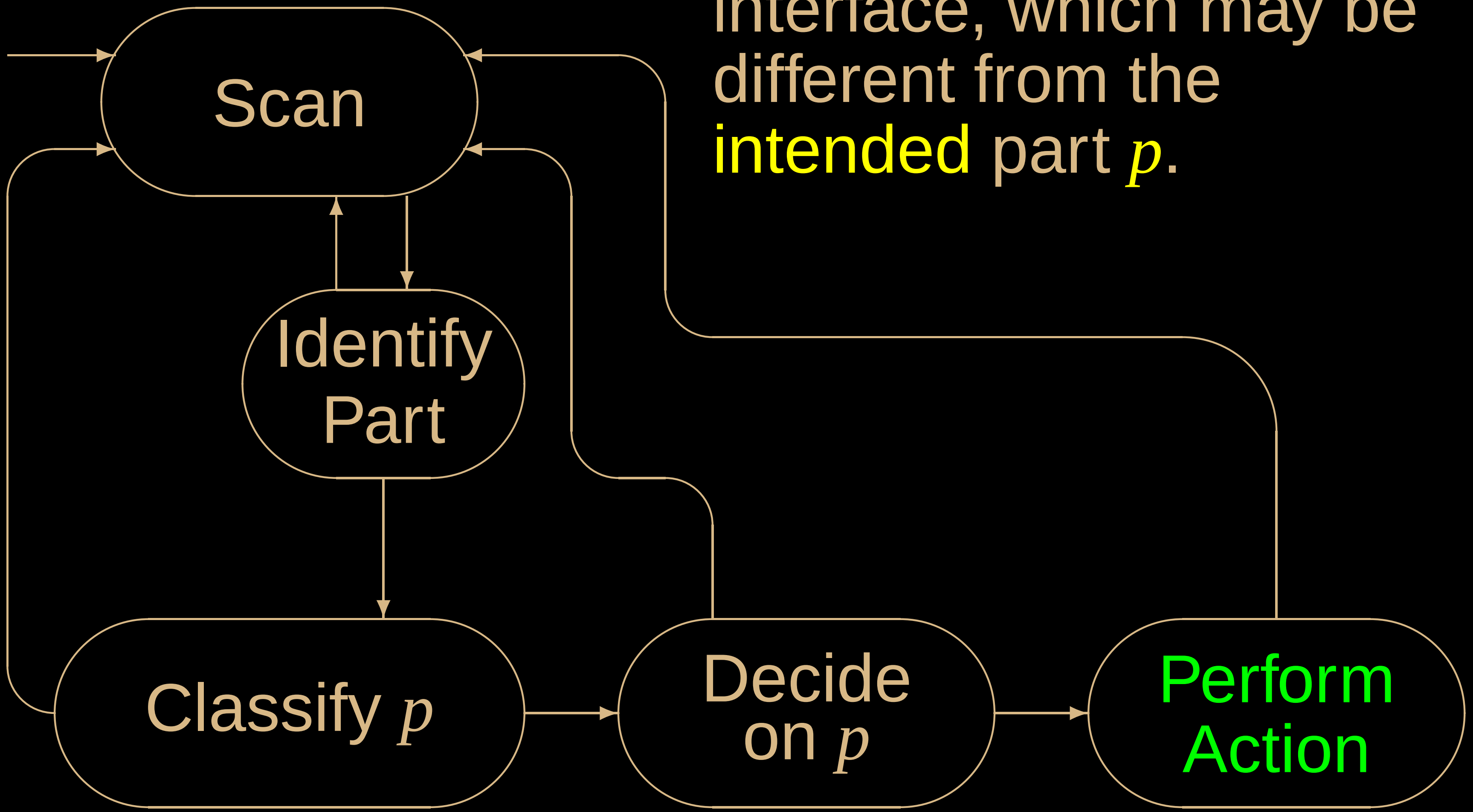

Perform Action:
The operator performs a series of interactions with the interface

# *Intention vs. Action*



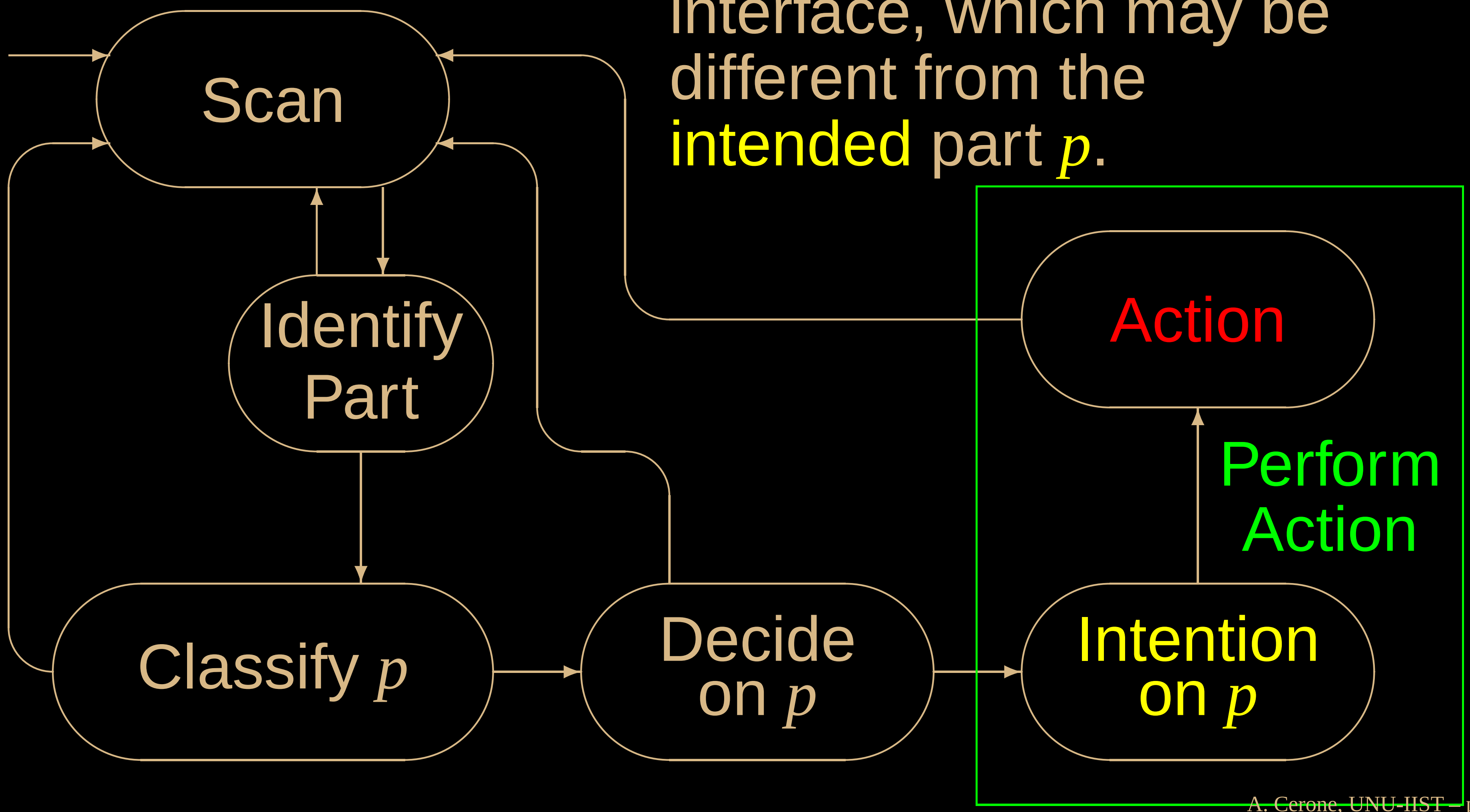
# *Intention vs. Action*

The Action is performed on the selected part of the interface, which may be different from the intended part $p$.



# *Intention vs. Action*

The Action is performed on the selected part of the interface, which may be different from the intended part $p$.



# *CSP Notation*

Communication Sequential Processes

Actions: $a$, $b$, $c$, ...

Processes: $P$, $Q$, $R$, ...

# *CSP Notation*

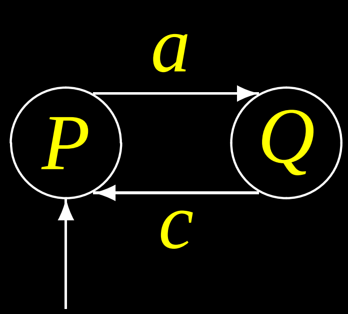
Communication Sequential Processes

Actions: $a$, $b$, $c$, ...

Processes: $P$, $Q$, $R$, ...

Basic Operators

Prefix: $\rightarrow$ $$P = a \rightarrow Q, \quad Q = c \rightarrow P$$



# *CSP Notation*

Communication Sequential Processes

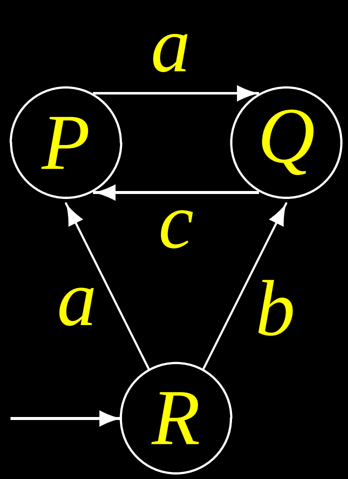
Actions: $a$, $b$, $c$, ...

Processes: $P$, $Q$, $R$, ...

Basic Operators

Prefix: $\rightarrow$ $\qquad$ $P = a \rightarrow Q, \quad Q = c \rightarrow P$

Choice: $\Box$ $\qquad$ $R = (a \rightarrow P) \Box (b \rightarrow Q)$



# *CSP Notation*

Communication Sequential Processes

Actions: $a$, $b$, $c$, ...

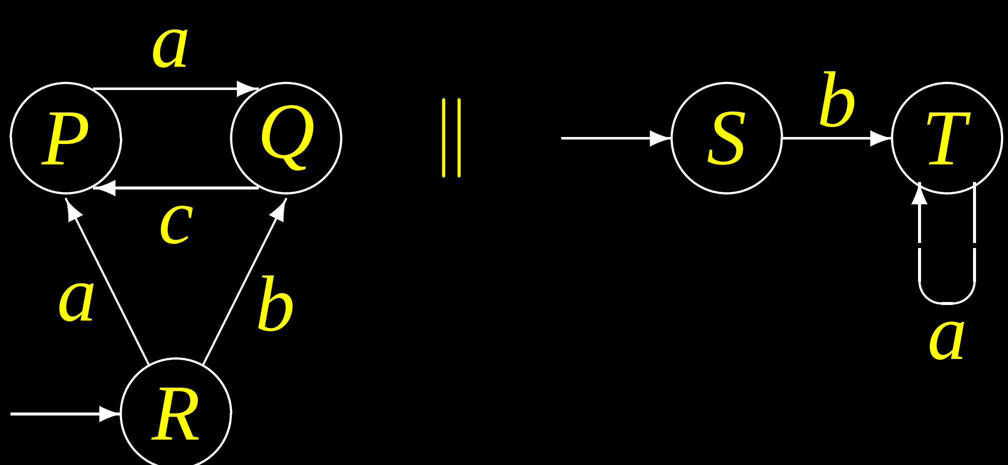
Processes: $P$, $Q$, $R$, ...

Basic Operators

Prefix: $\rightarrow$ $\quad P = a \rightarrow Q, \quad Q = c \rightarrow P$

Choice: $\Box$ $\quad R = (a \rightarrow P) \Box (b \rightarrow Q)$

Parallel: $\parallel$ $\quad R \parallel S, \quad S = b \rightarrow T, \quad T = a \rightarrow T$



# *CSP Notation*

Communication Sequential Processes

Actions: $a$, $b$, $c$, ...

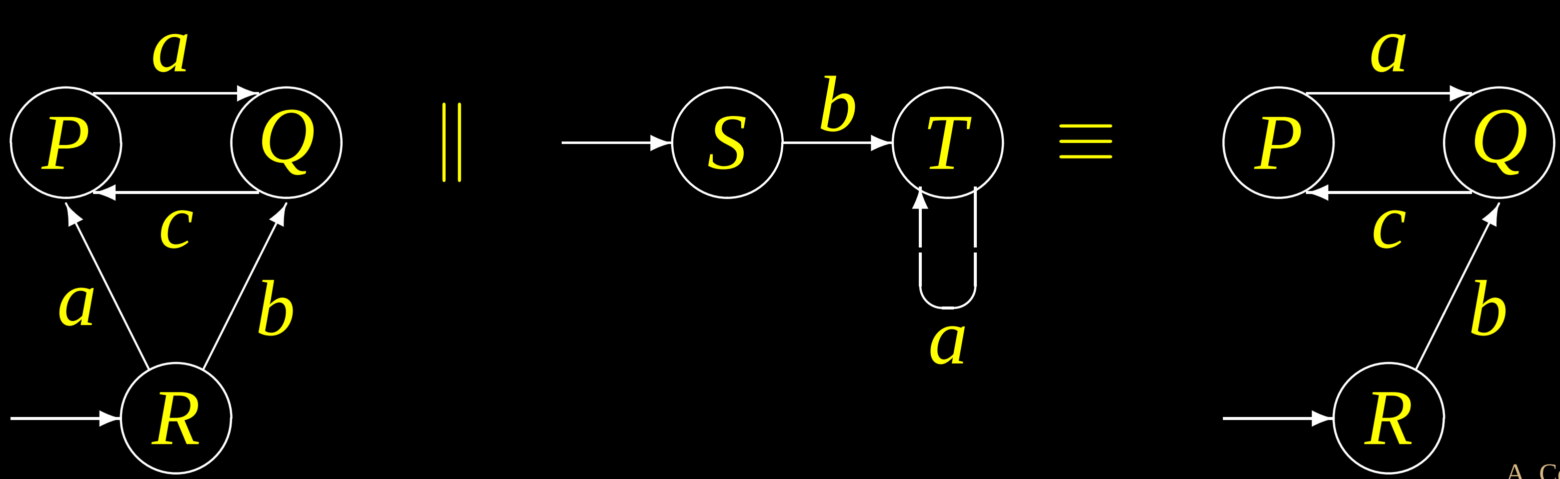
Processes: $P$, $Q$, $R$, ...

Basic Operators

Prefix: $\rightarrow$ $\quad P = a \rightarrow Q, \quad Q = c \rightarrow P$

Choice: $\square$ $\quad R = (a \rightarrow P) \square (b \rightarrow Q)$

Parallel: $\|$ $\quad R \;\|\; S, \quad S = b \rightarrow T, \quad T = a \rightarrow T$



# *CSP: Parallel Composition*

used to

- compose parts of the system

# *CSP: Parallel Composition*

used to

- compose parts of the system
- compose user and system/environment

# *CSP: Parallel Composition*

used to

- compose parts of the system
- compose user and system/environment
- define assumptions

# *CSP: Parallel Composition*

used to

- compose parts of the system
- compose user and system/environment
- define assumptions

$Assumptions \;\|\; User \;\|\; System$
where $System = System_1 \;\|\; \ldots \;\|\; System_n$

# CSP: Parallel Composition

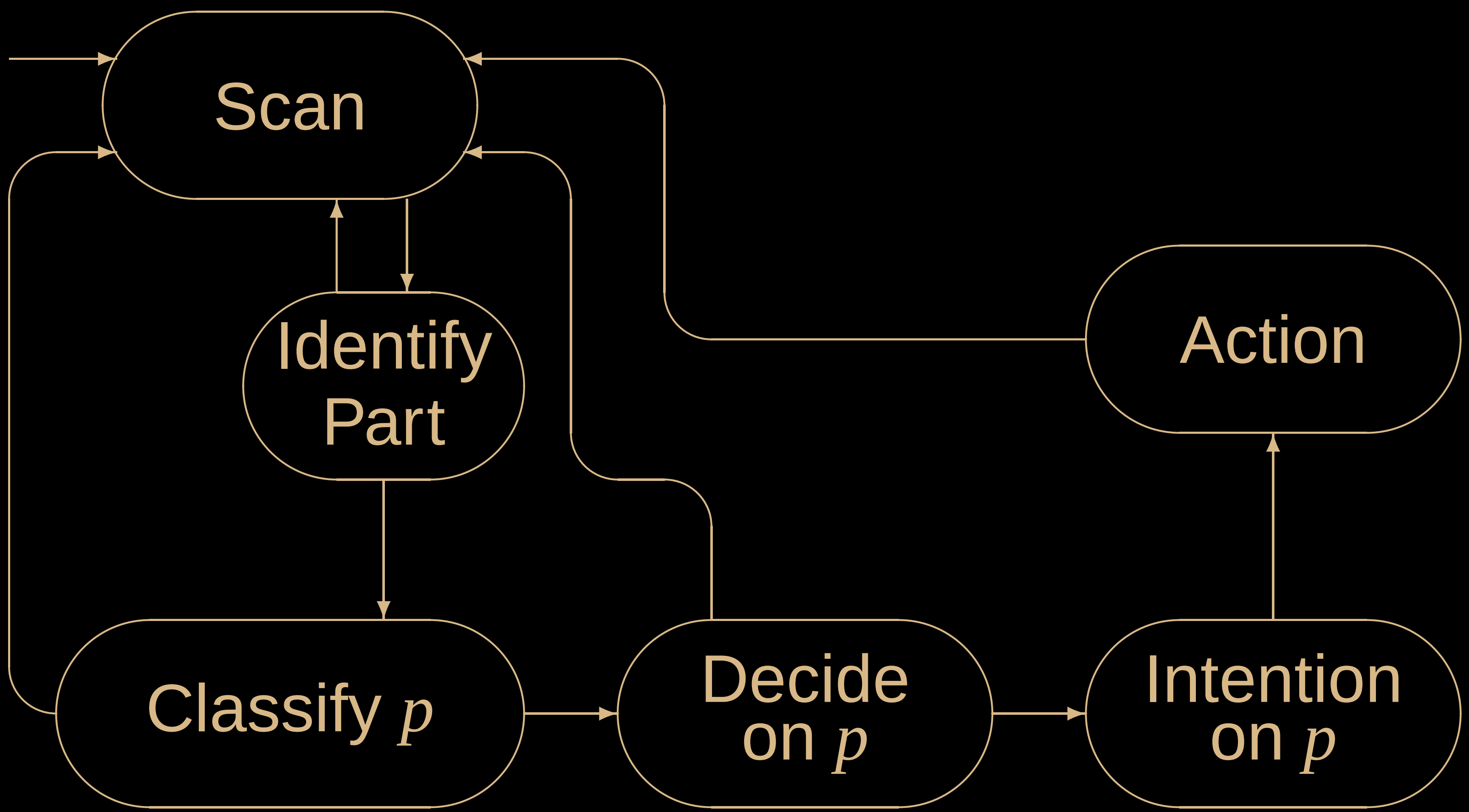
used to

- compose parts of the system
- compose user and system/environment
- define assumptions

$Assumptions \parallel User \parallel System$
where $System = System_1 \parallel \ldots \parallel System_n$

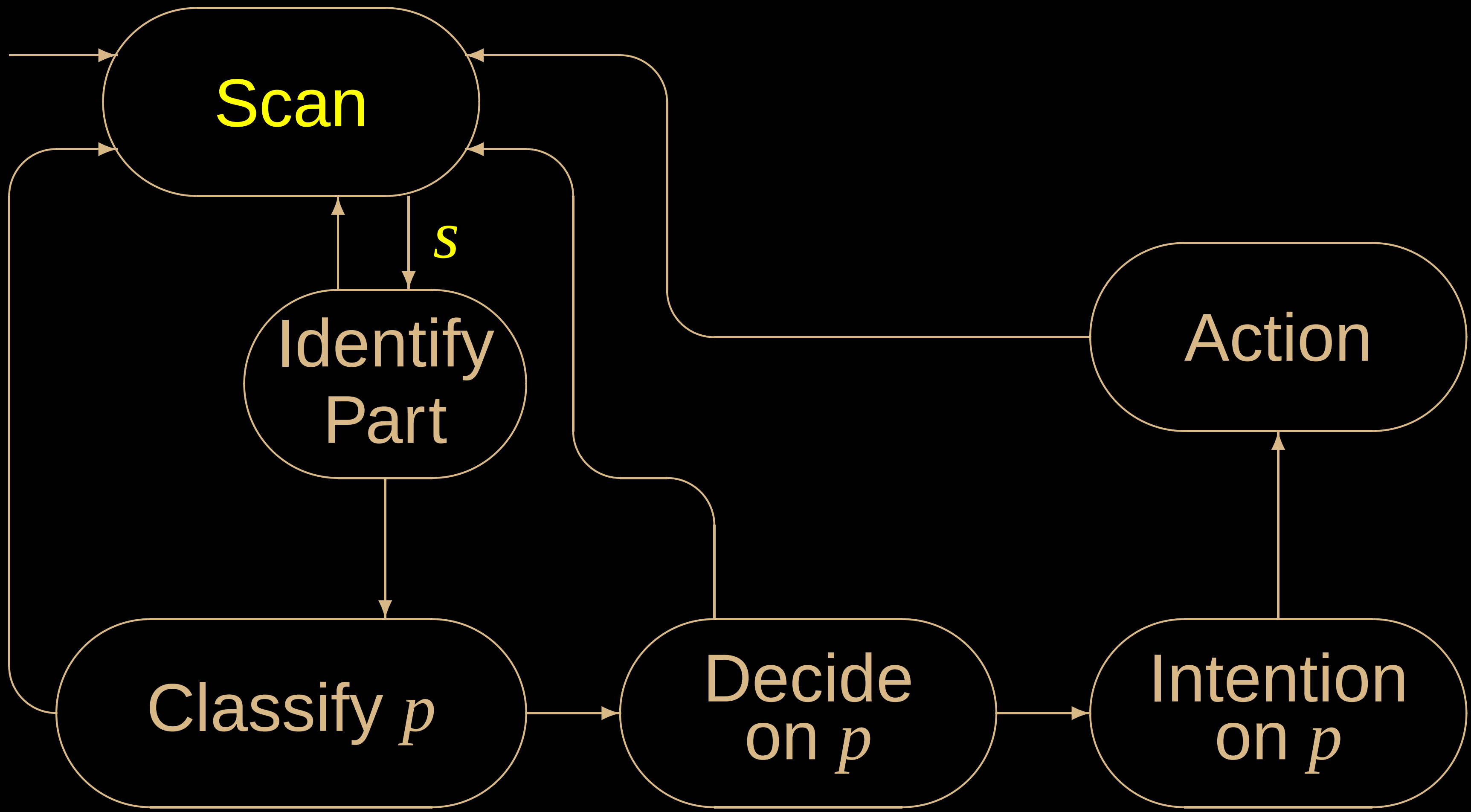
$User \parallel System$
$User \parallel Environment$
with possibly $User = User_1 \parallel \ldots \parallel User_m$

# *Formal Model using CSP*



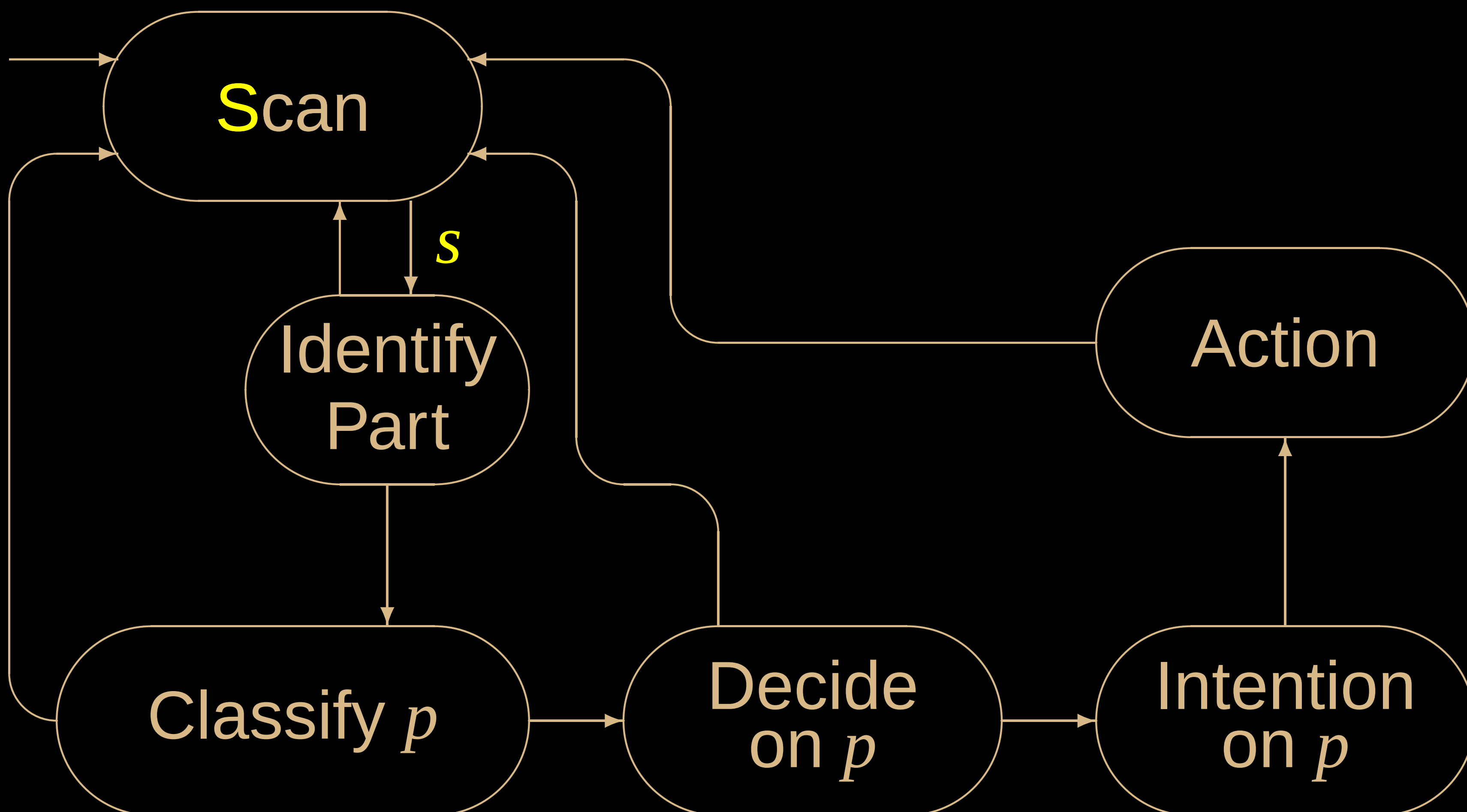
# *Formal Model using CSP*

Scanning: The operator searches the interface for a certain property



# *Formal Model using CSP*

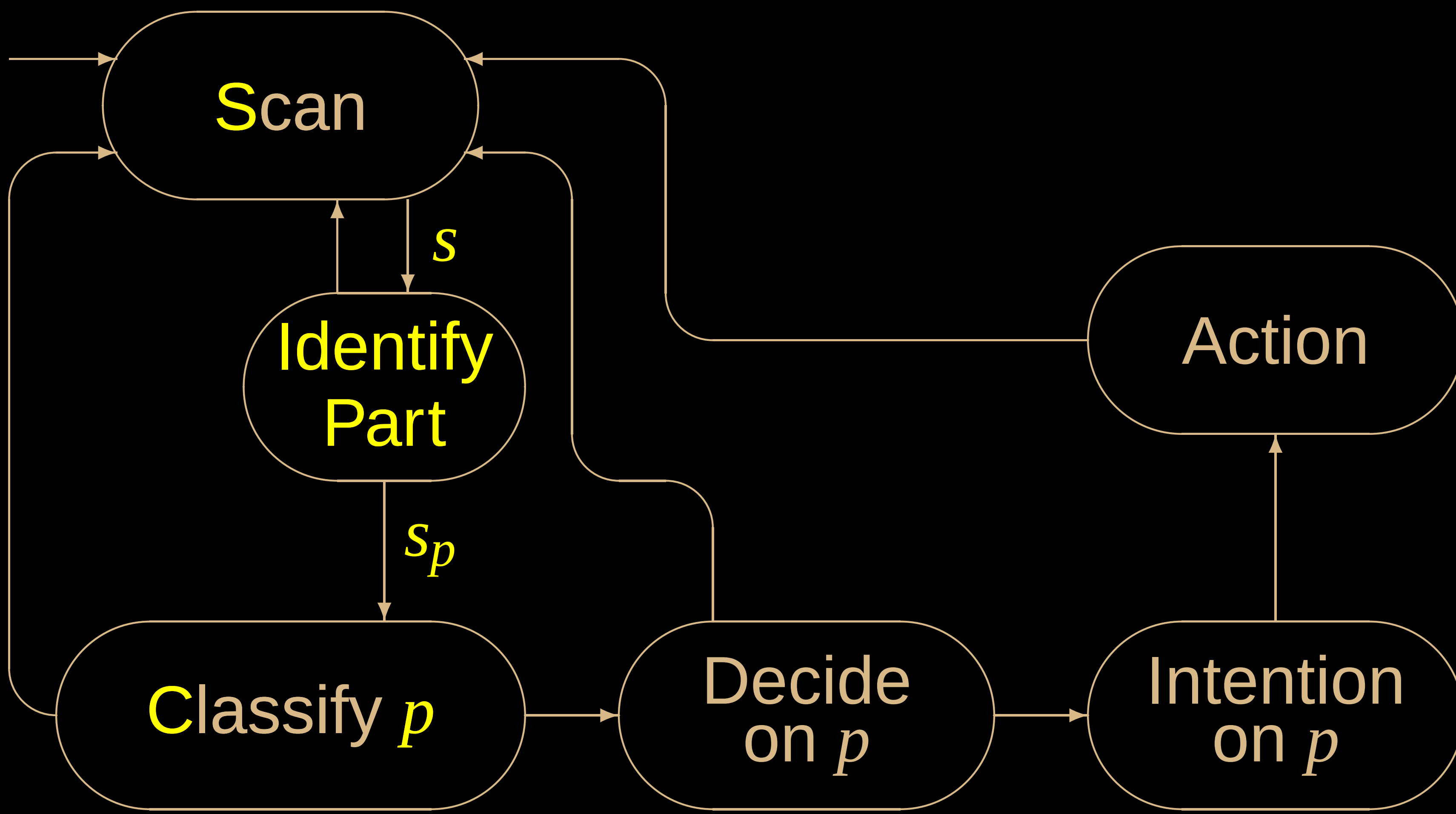
$$S = s \rightarrow ((\Box_{p:Part} \cdots) \Box S)$$



# *Formal Model using CSP*

Identification: The operator identifies part of the interface that may represent the property

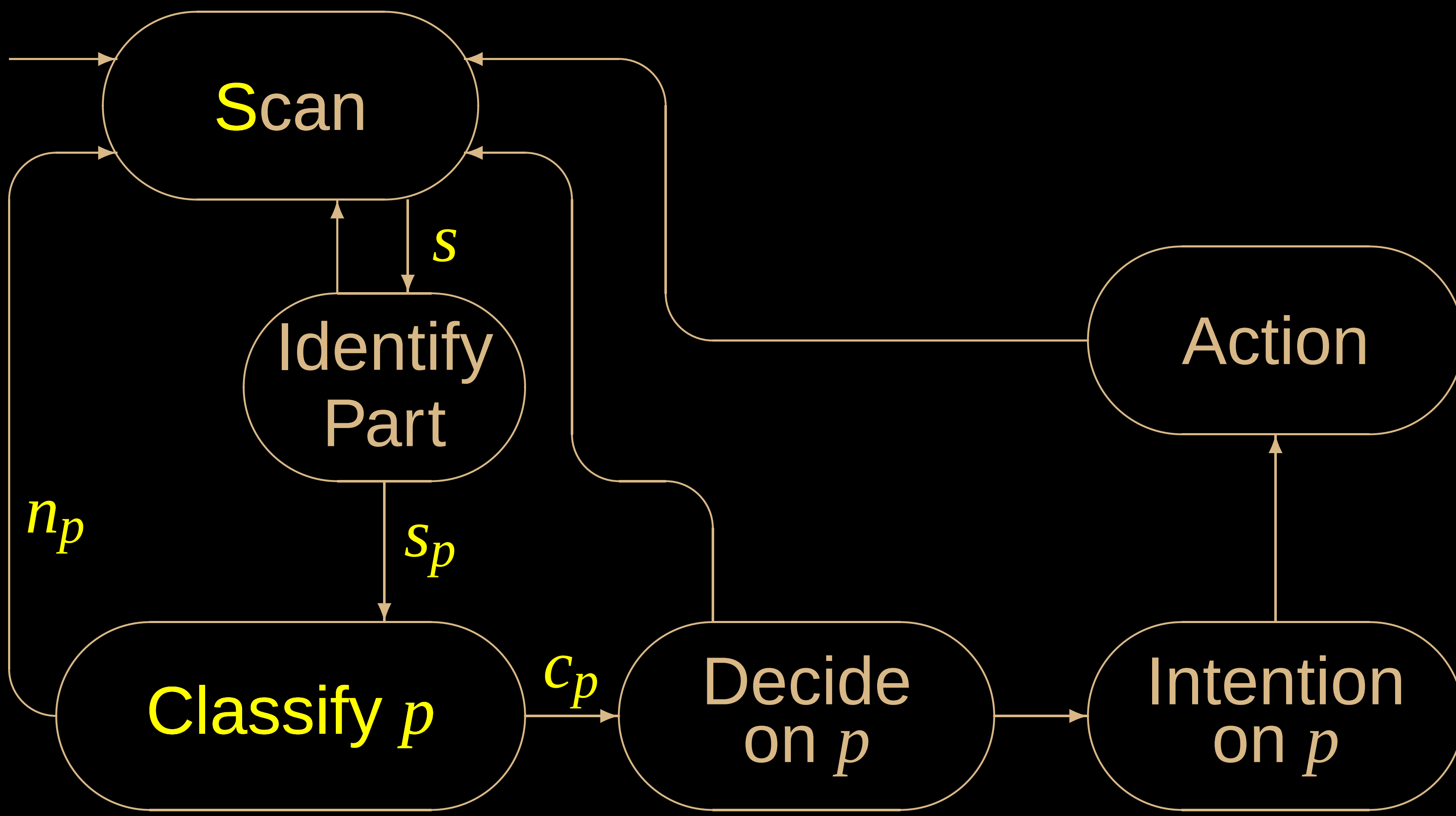
$$S = s \rightarrow ((\Box_{p:Part}(s_p \rightarrow C_p)) \Box S)$$



# *Formal Model using CSP*

Classification: The operator assesses whether the property is in need of further interest

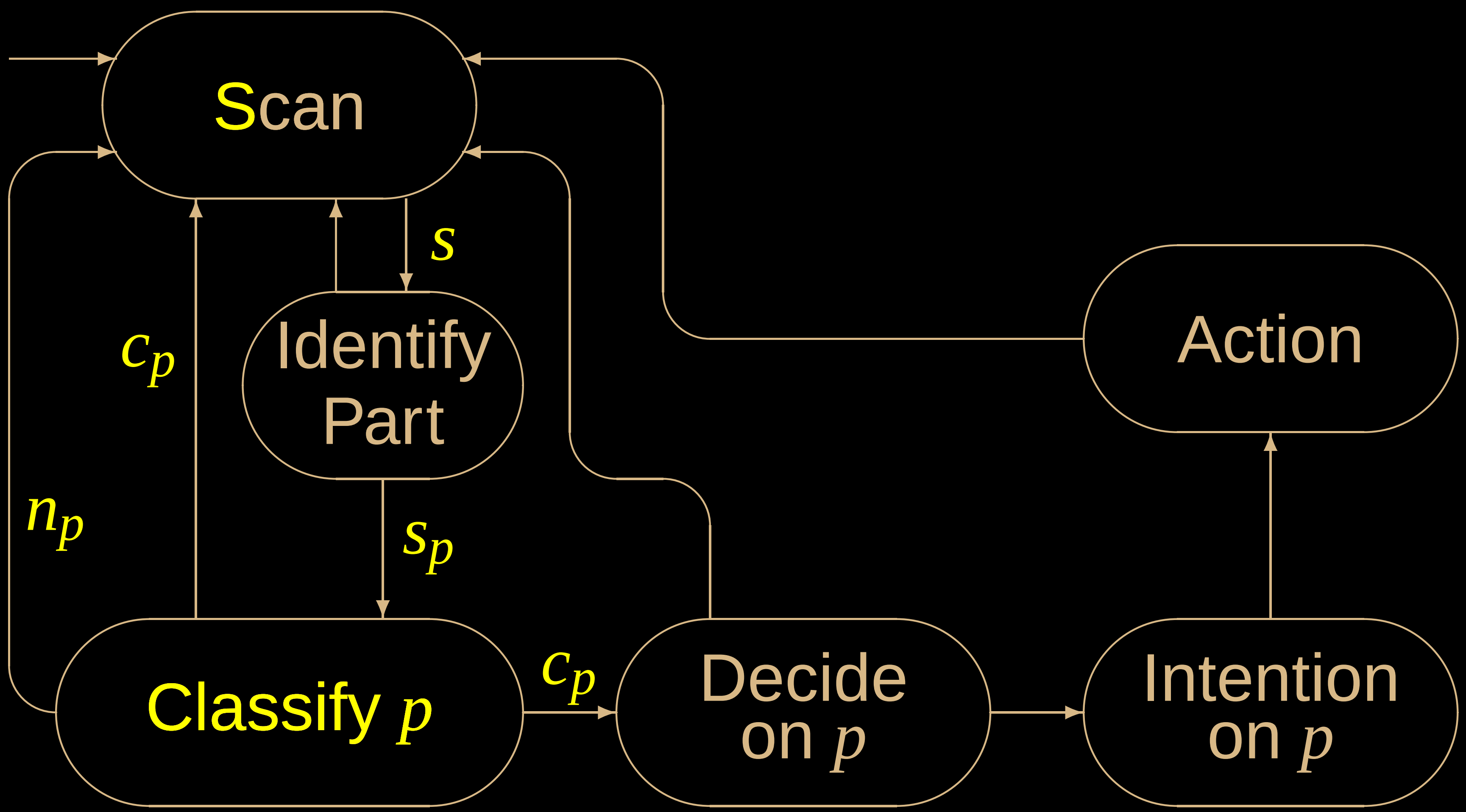
$$S = s \rightarrow ((\Box_{p:Part}(s_p \rightarrow C_p)) \Box S)$$



# *Formal Model using CSP*

Classification: If so, the operator gives some form of priority to the property

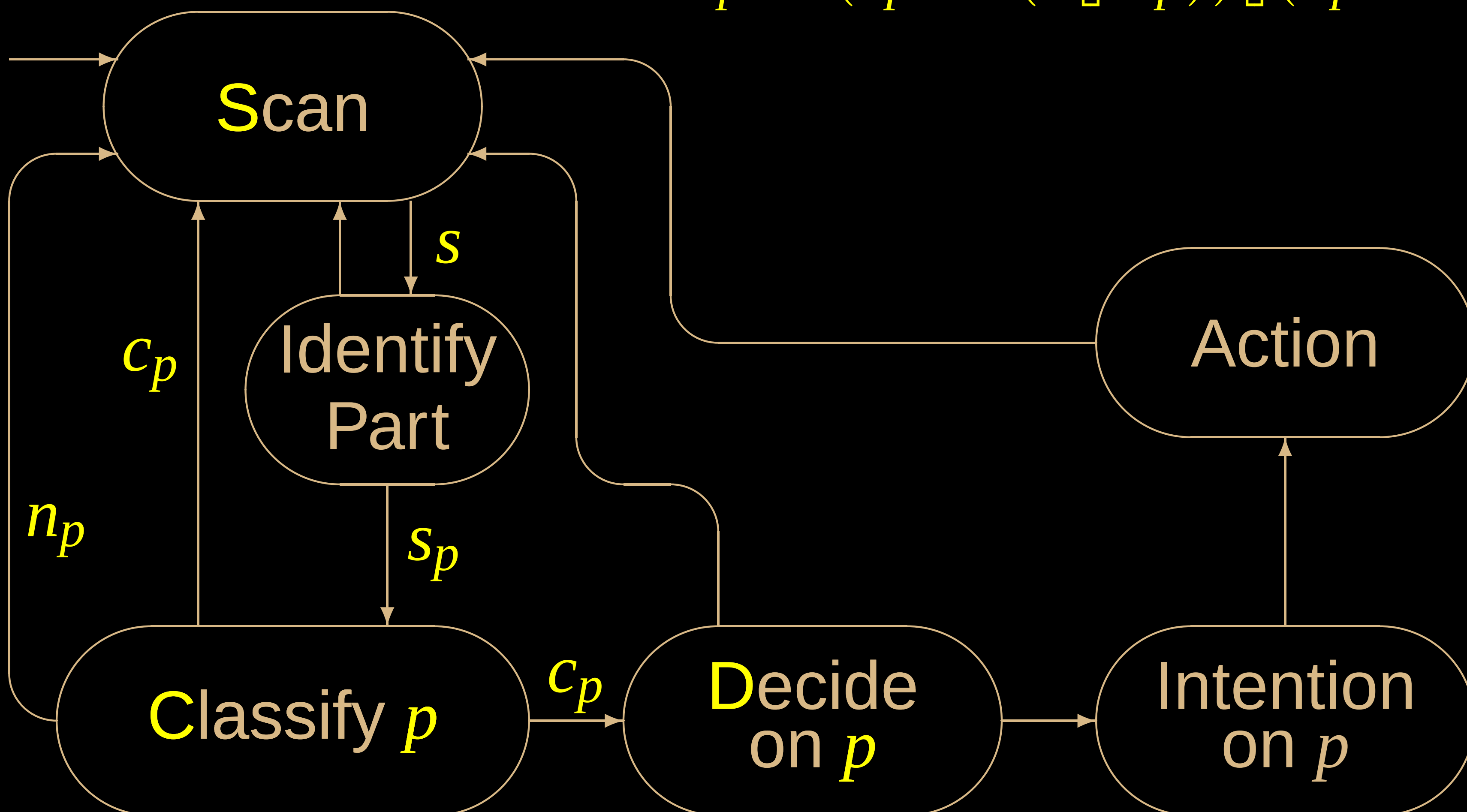
$$S = s \to ((\Box_{p:Part}(s_p \to C_p)) \Box S)$$



# *Formal Model using CSP*

$$S = s \rightarrow ((\Box_{p:Part}(s_p \rightarrow C_p)) \Box S)$$

$$C_p = (c_p \rightarrow (S \Box D_p)) \Box (n_p \rightarrow S)$$

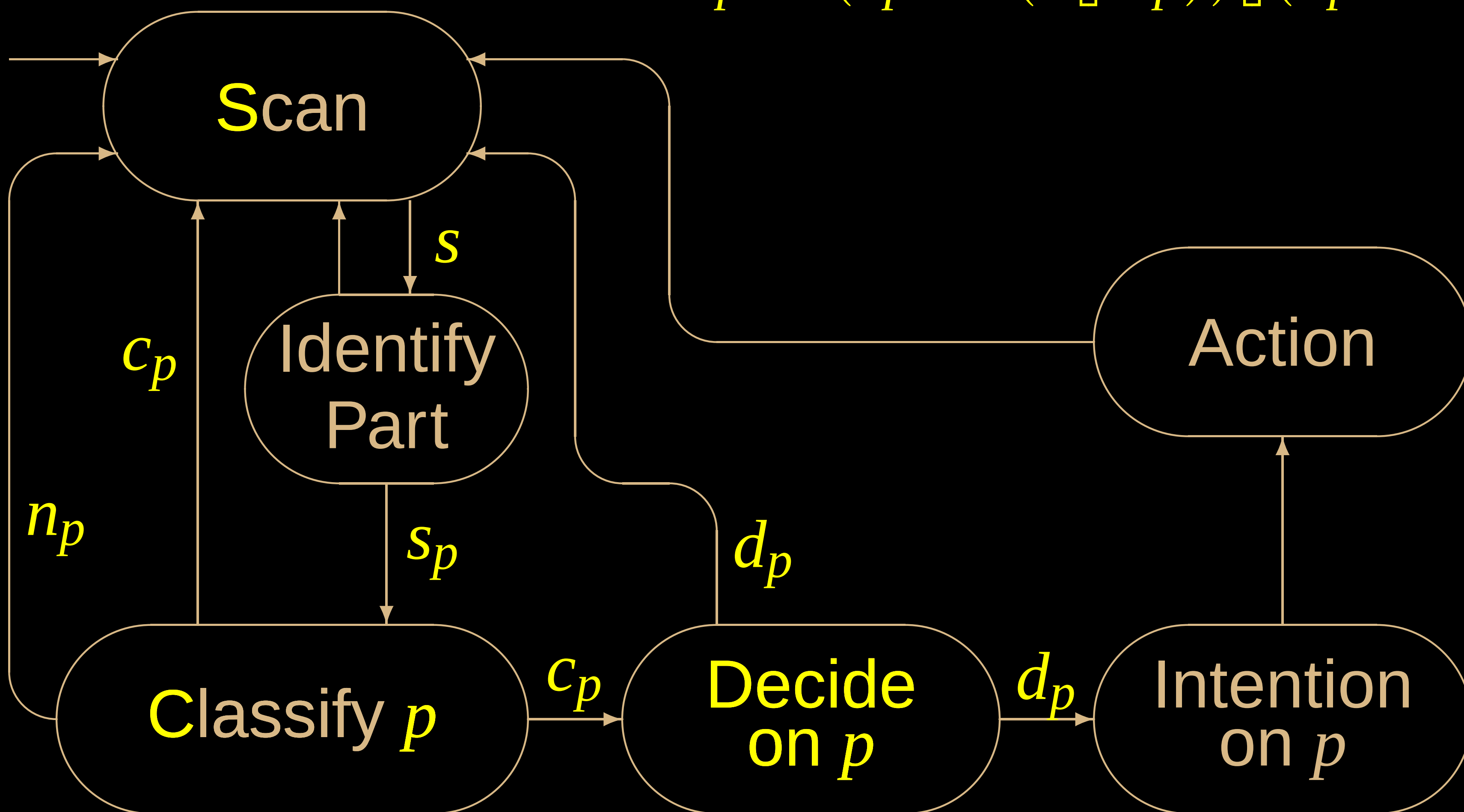


# *Formal Model using CSP*

Decision on how to resolve the situation

$$S = s \to ((\Box_{p:Part}(s_p \to C_p))\Box S)$$

$$C_p = (c_p \to (S \Box D_p)) \Box (n_p \to S)$$



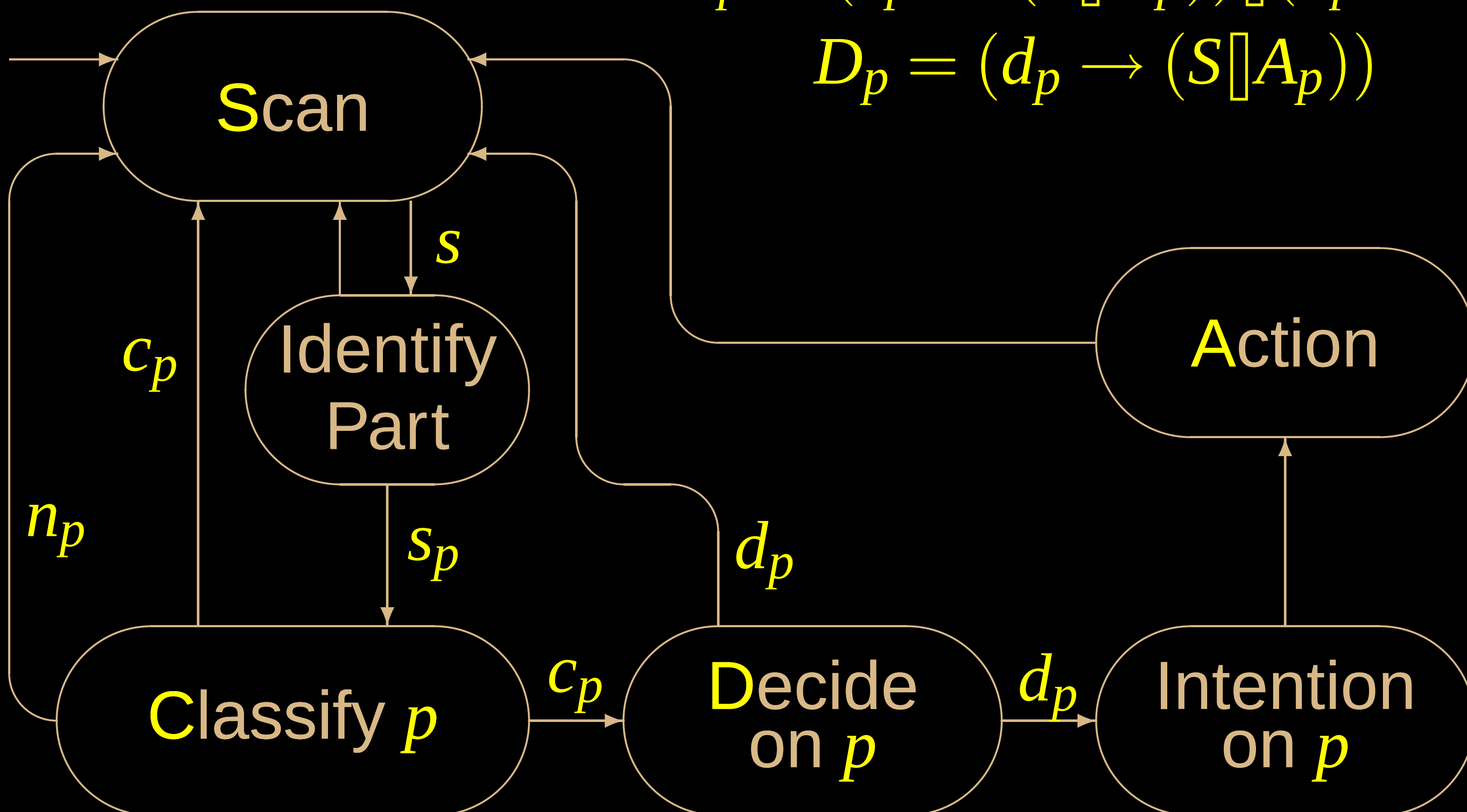
# *Formal Model using CSP*

Decision on how to resolve the situation

$$S = s \rightarrow ((\Box_{p:Part}(s_p \rightarrow C_p)) \Box S)$$

$$C_p = (c_p \rightarrow (S \Box D_p)) \Box (n_p \rightarrow S)$$

$$D_p = (d_p \rightarrow (S \Box A_p))$$

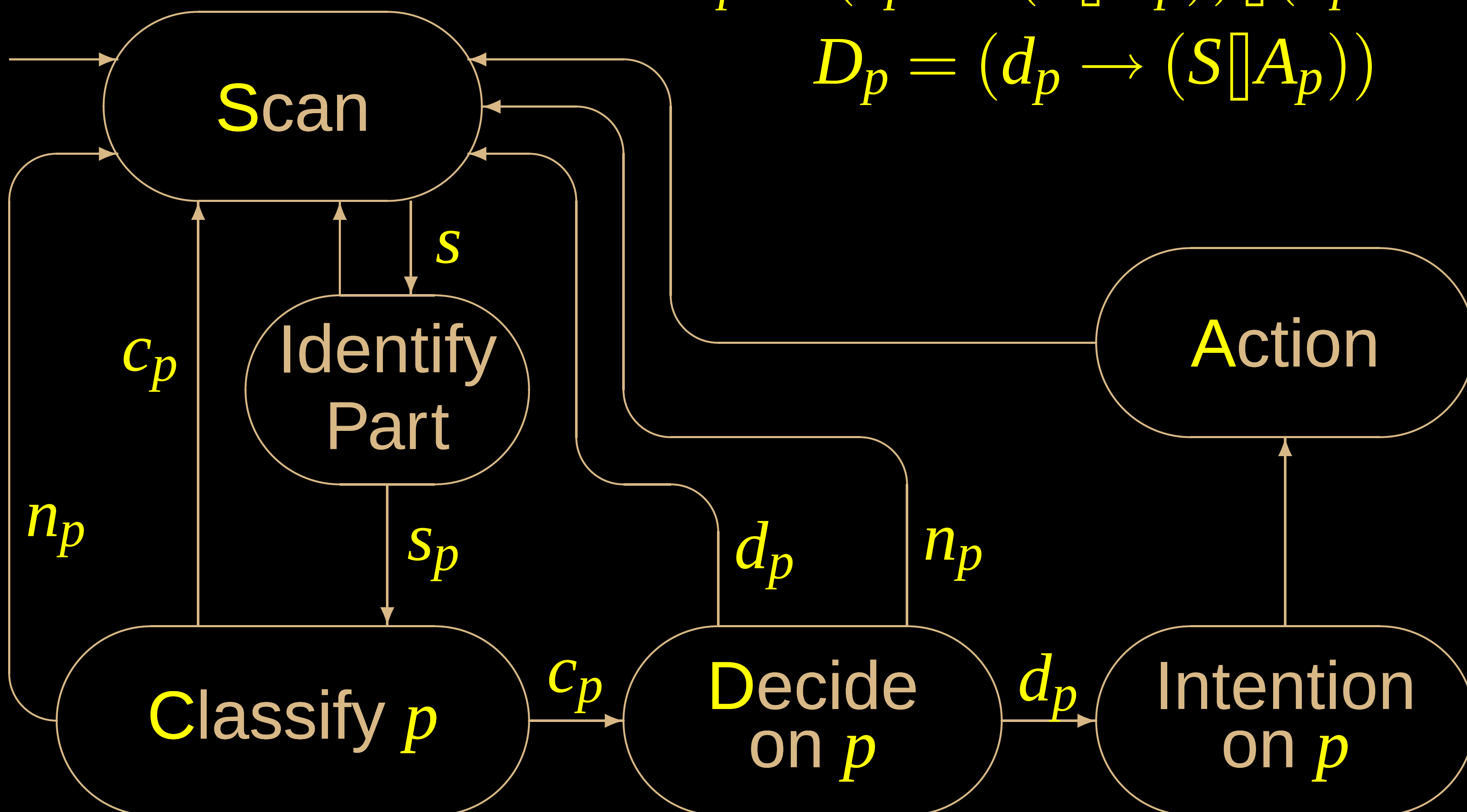


# *Formal Model using CSP*

$$S = s \rightarrow ((\Box_{p:Part}(s_p \rightarrow C_p)) \Box S)$$

$$C_p = (c_p \rightarrow (S \Box D_p)) \Box (n_p \rightarrow S)$$

$$D_p = (d_p \rightarrow (S \Box A_p))$$

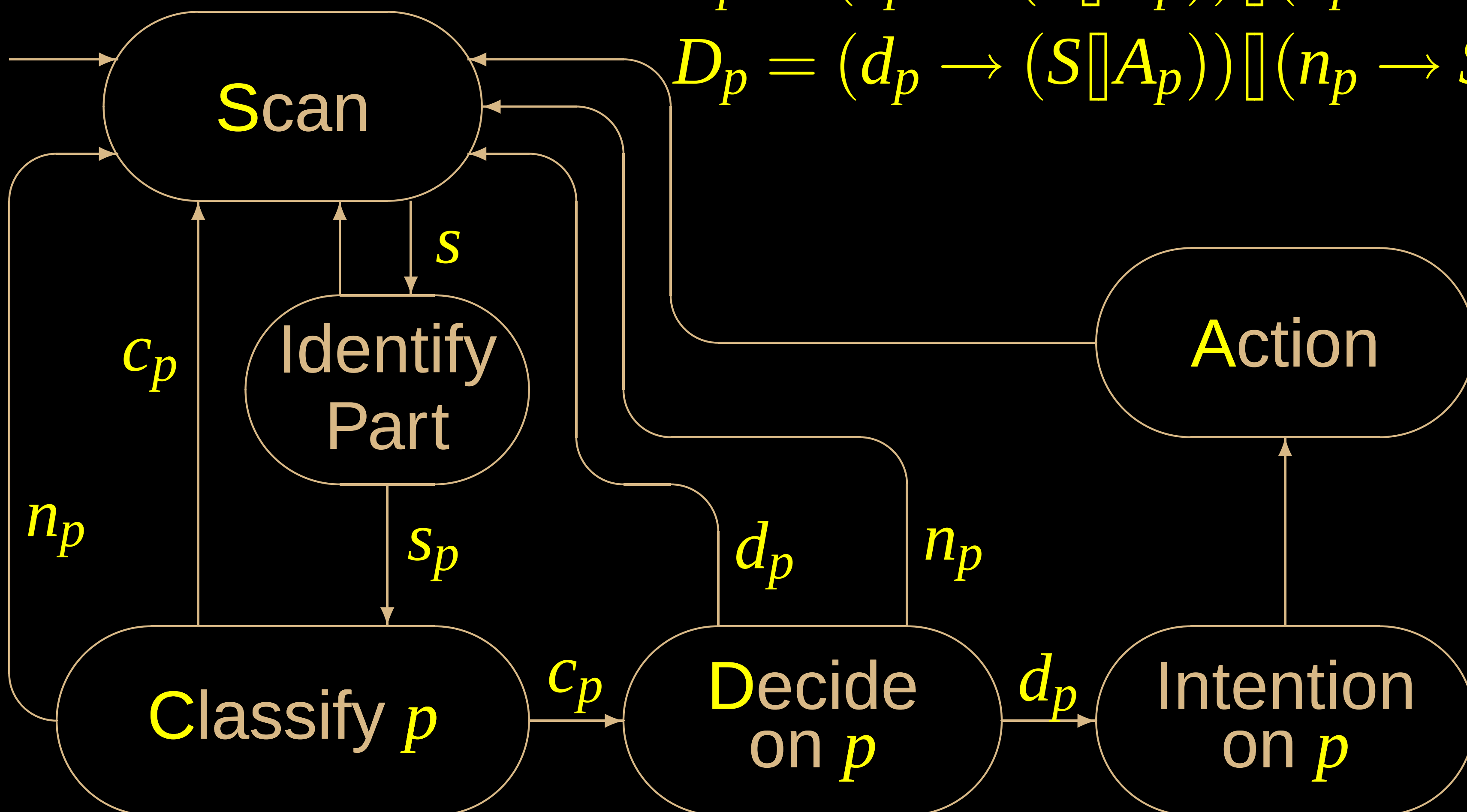


# *Formal Model using CSP*

$$S = s \rightarrow ((\Box_{p:Part}(s_p \rightarrow C_p)) \Box S)$$

$$C_p = (c_p \rightarrow (S \Box D_p)) \Box (n_p \rightarrow S)$$

$$D_p = (d_p \rightarrow (S \Box A_p)) \Box (n_p \rightarrow S)$$



# *Formal Model using CSP*

Action to be performed as a series of interactions with the interface

$$S = s \rightarrow ((\Box_{p:Part}(s_p \rightarrow C_p)) \Box S)$$

$$C_p = (c_p \rightarrow (S \Box D_p)) \Box (n_p \rightarrow S)$$

$$D_p = (d_p \rightarrow (S \Box A_p)) \Box (n_p \rightarrow S)$$

$$A_p = (i_p \rightarrow a \rightarrow S)$$