Practical Reasoning Agents

Dominic Hallau // Tim Preuß

Practical Reasoning

Intention

Means-Ends-Reasoning

Implementierung eines Practical Reasoning Agents

Procedural Reasoning System

Theoretical Reasoning

beliefs

Practical Reasoning

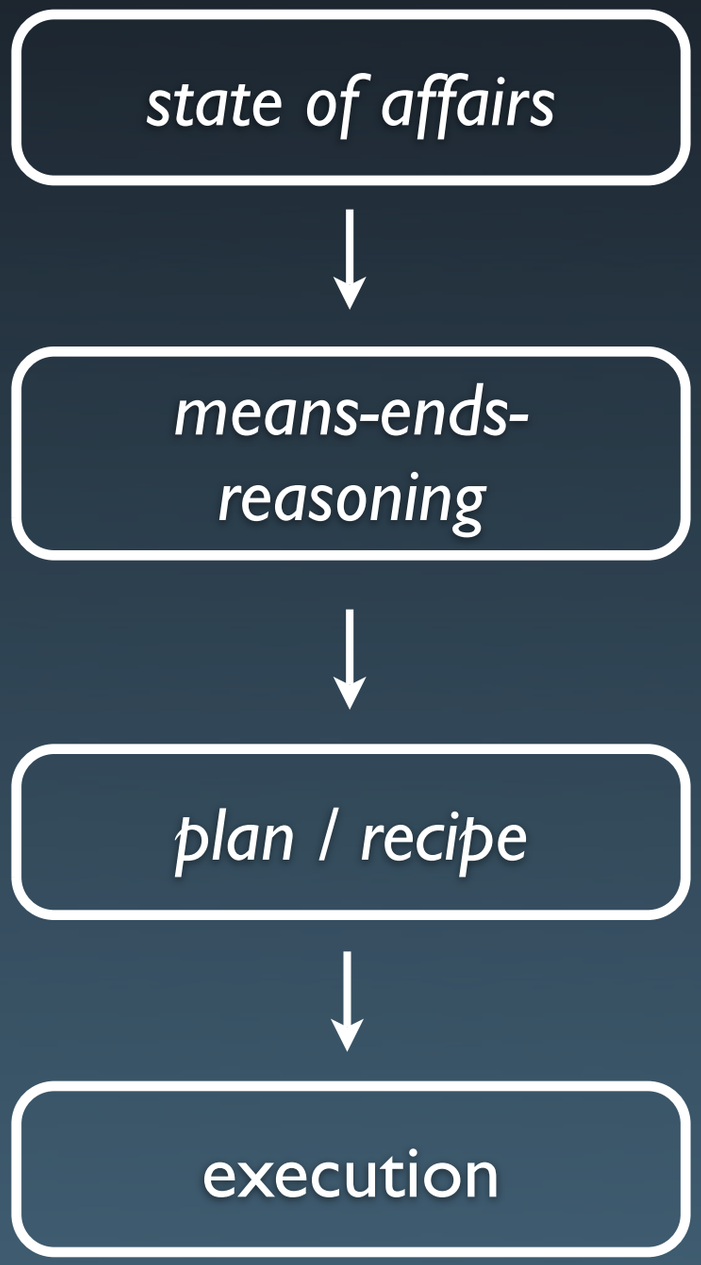
actions

practical reasoning

Human Practical Reasoning

Welchen Stand der Dinge will ich erreichen?
(deliberation)

Wie will ich diesen Stand der Dinge erreichen?
(means-ends-reasoning)



practical reasoning

Beschränkung

deliberation / means-ends-reasoning sind Rechenprozesse

time constraints

Agenten können nicht unendlich lange Entscheidungen abwägen

Intention

actions

states of mind
(pro attitudes)

intentions

Intentionen sind der Auslöser für practical reasoning

Intentionen sind beständig
(persistance)

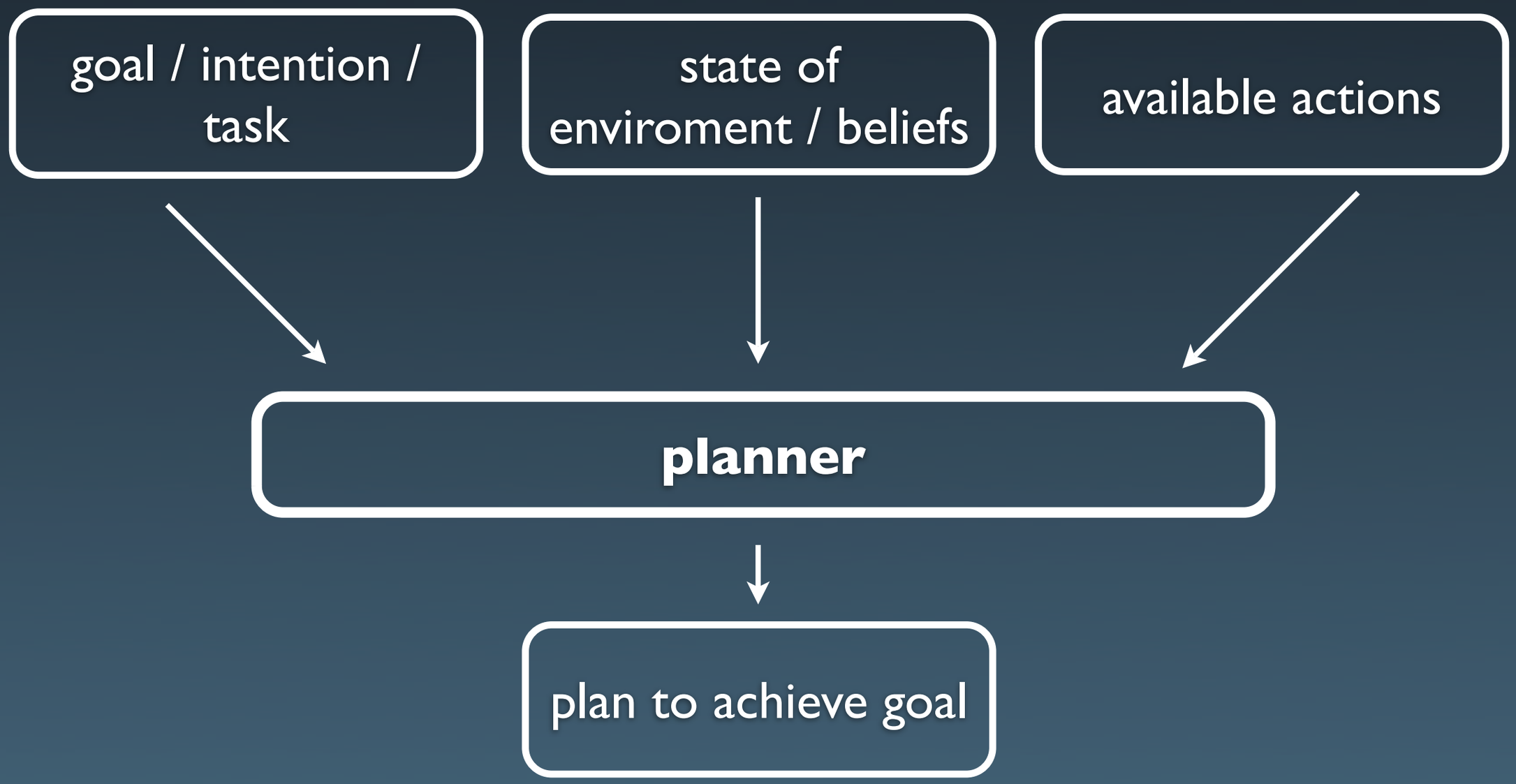
Intentionen beschränken
die Handlungsmöglichkeiten eines Agenten
(deliberation)

Intentionen beeinflussen die zukünftige Sicht der Dinge
(beliefs)

means-ends-reasoning

(planning)

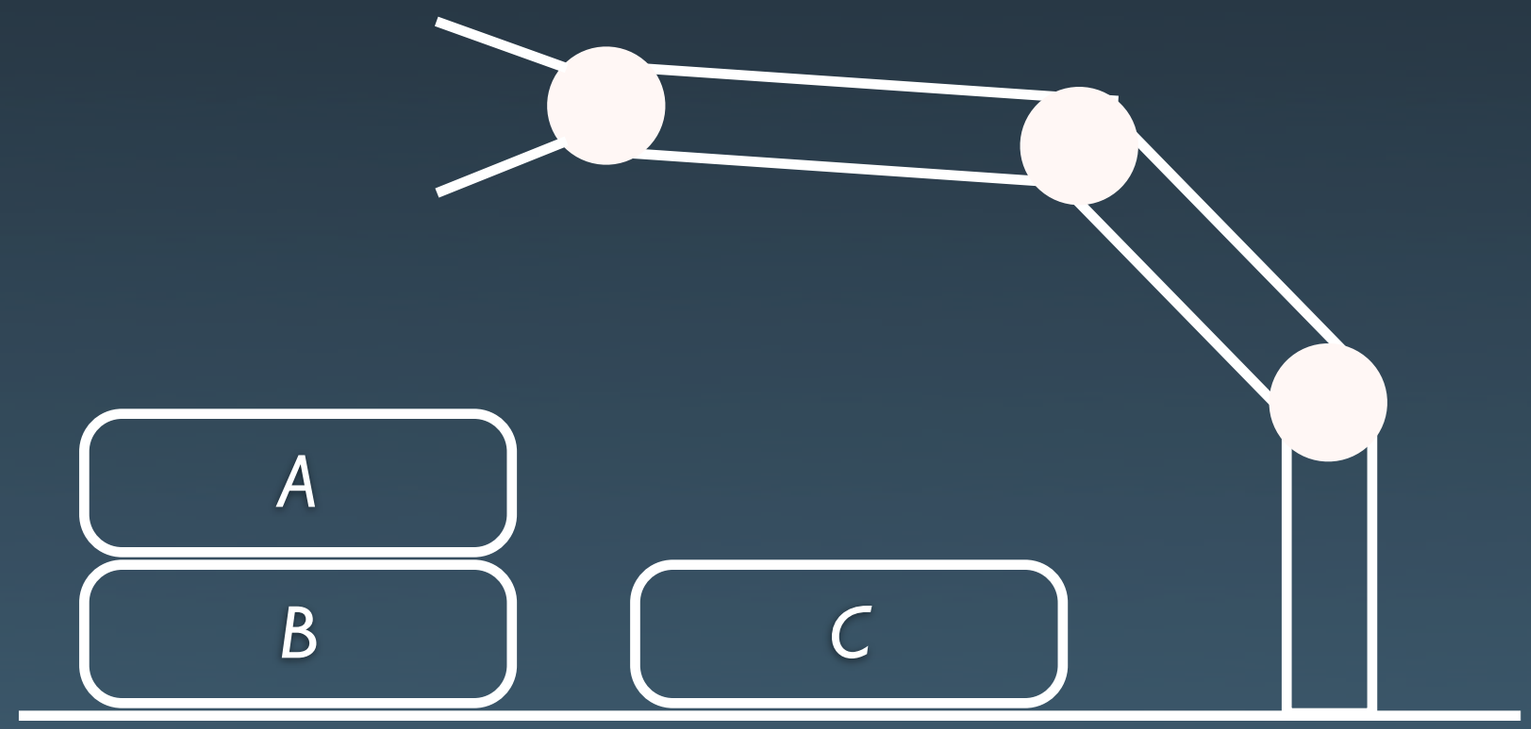
STRIPS



means-ends-reasoning

STRIPS: The Blocks World

{Clear(A), On(A,B), OnTable(B) , OnTable(C), Clear(C)}



goal: {OnTable(A), OnTable(B), OnTable(C)}

STRIPS-Notation

name	Name mit Argumenten
precondition list	Vorbedinungen
delete list	“falsch-werdende” Fakten
add list	“wahr-werdene” Fakten

Beispiele:

Stack(x,y)

pre	{Clear(y), Holding(x)}
del	{Clear(y), Holding(x)}
add	{ArmEmpty, On(x,y)}

Unstack(x,y)

pre	{On(x,y), Clear(x), ArmEmpty}
del	{On(x,y), ArmEmpty}
add	{Holding(x), Clear(y)}

PutDown(x)

pre	{Holding(x)}
del	{Holding(x)}
add	{ArmEmpty, OnTable(x)}

PickUp(x)

pre	{Clear(x), OnTable(x), ArmEmpty}
del	{OnTable(x), ArmEmpty}
add	{Holding(x)}

```
B <- B0;     // B0 sind Anfangs-Annahmen über die Umwelt
I <- I0;     // I0 sind Anfangs-Intentionen

while true do
    get next percept ρ through see(...) function;
    B <- brf(B,ρ);
    D <- options(B,I);
    I <- filter(B,D,I);
    π <- plan(B,I,Ac);

    while not (empty(π) or succeeded(I,B) or impossible(I,B)) do
        α <- head(π);
        execute(α);
        π <- tail(π);

        get next percept ρ through see(...) function;
        B <- brf(B,ρ);
        if reconsider(I,B) then
            D <- options(B,I);
            I <- filter(B,D,I);
        end if
        if not sound(π,I,B) then
            π <- plan(B,I,Ac);
        end if
    end while
end while
```

practical reasoning agent: implemention

Commitment
(führt zu *temporal persistence)*

Blind commitment

Single-minded commitment

Open-minded commitment

```
B <- B0;     // B0 sind Anfangs-Annahmen über die Umwelt
I <- I0;     // I0 sind Anfangs-Intentionen

while true do
    get next percept ρ through see(...) function;
    B <- brf(B,ρ);
    D <- options(B,I);
    I <- filter(B,D,I);
    π <- plan(B,I,Ac);

    while not (empty(π) or succeeded(I,B) or impossible(I,B)) do
        α <- head(π);
        execute(α);
        π <- tail(π);

        get next percept ρ through see(...) function;
        B <- brf(B,ρ);
        if reconsider(I,B) then
            D <- options(B,I);
            I <- filter(B,D,I);
        end if
        if not sound(π,I,B) then
            π <- plan(B,I,Ac);
        end if
    end while
end while
```

reconsider

```
B <- B0;     // B0 sind Anfangs-Annahmen über die Umwelt
I <- I0;     // I0 sind Anfangs-Intentionen

while true do
    get next percept ρ through see(...) function;
    B <- brf(B,ρ);
    D <- options(B,I);
    I <- filter(B,D,I);
    π <- plan(B,I,Ac);

    while not (empty(π) or succeeded(I,B) or impossible(I,B)) do
        α <- head(π);
        execute(α);
        π <- tail(π);
        get next percept ρ through see(...) function;
        B <- brf(B,ρ);
        if reconsider(I,B) then
            D <- options(B,I);
            I <- filter(B,D,I);
        end if
        if not sound(π,I,B) then
            π <- plan(B,I,Ac);
        end if
    end while
end while
```

reconsider

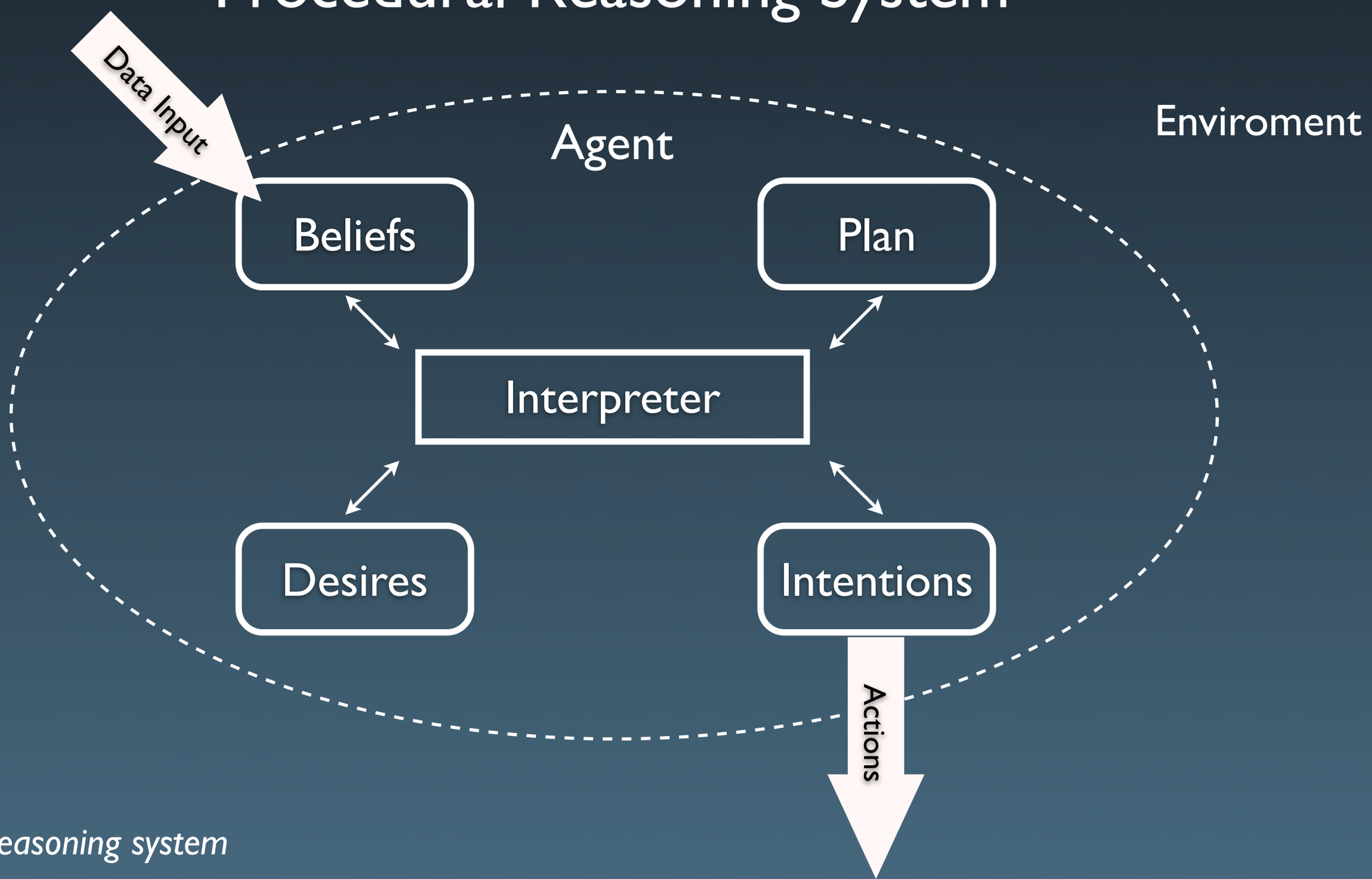
Chose to deliberate?	Changed intentions?	Would have changed intentions?	*reconsider* optimal?
No	-	No	Yes
No	-	Yes	No
Yes	No	-	No
Yes	Yes	-	Yes

optimal:

deliberate und Intentionen geändert

kein *deliberate* und hätte Intentionen nicht geändert

Procedural Reasoning System



procedural reasoning system

Die Blocks World in Jam!

```
GOALS:
    ACHIEVE blocks_stacked;

FACTS:
    FACT ON “Block5” “Block4”;  FACT ON “Block4” “Block3”;
    FACT ON “Block1” “Block1”;  FACT ON “Block2” “Table”;
    FACT ON “Block3” “Table”;   FACT CLEAR “Block1”;
    FACT CLEAR “Block5”         FACT CLEAR “Table”;
```

```
Plan: {
    NAME:    "Top-Level-Plan"
    GOAL:    ACHIEVE blocks_stacked;
    CONTEXT:
    BODY:    ACHIEVE ON "Block3" "Table";
             ACHIEVE ON "Block2" "Block3";
             ACHIEVE ON "Block1" "Block2";
}

Plan: {
    NAME:    "Stack blocks that are already clear"
    GOAL:    ACHIEVE ON $OBJ1 $OBJ2;
    CONTEXT:
    BODY:    ACHIEVE CLEAR $OBJ1;
             ACHIEVE CLEAR $OBJ2;
             ACHIEVE move $OBJ1 $OBJ2;
    UTILITY: 10;
    FAILURE: EXECUTE print "\n\nStack blocks failed!\n\n";
}

Plan: {
    NAME:    "Clear a block"
    GOAL:    ACHIEVE CLEAR $OBJ;
    CONTEXT: FACT ON $OBJ1 $OBJ2;
    BODY:    ACHIEVE ON $OBJ2 "Table";
    EFFECTS: RETRACT ON $OBJ2 $OBJ};
    FAILURE: EXECUTE print "\n\nClearing block failed!\n\n";
}
```

procedural reasoning system

Stack(x,y)
pre {Clear(y), Holding(x)}
del {Clear(y), Holding(x)}
add {ArmEmpty, On(x,y)}

```
Plan: {
    NAME:    "Stack blocks that are already clear"
    GOAL:    ACHIEVE ON $OBJ1 $OBJ2;
    CONTEXT:
    BODY:    ACHIEVE CLEAR $OBJ1;
             ACHIEVE CLEAR $OBJ2;
             ACHIEVE move $OBJ1 $OBJ2;
    UTILITY: 10;
    FAILURE: EXECUTE print "\n\nStack blocks failed!\n\n";
}
```

procedural reasoning system

Fazit
("für Mama")

Practical Reasoning Agents

haben Intentionen und planen diese zu verwirklichen
(means-ends-reasoning)

können Intentionen überdenken und Pläne abändern
(reconsider)

sind zum Agieren in der realen Welt ausgelegt

bewerten und filtern ihre Handlungsoptionen