Multiagentové systémy II

RNDr. Andrej Lúčny

MicroStep-MIS

andy@microstep-mis.com

http://www.microstep-mis.sk/~andy

Subsumpčná architektúra

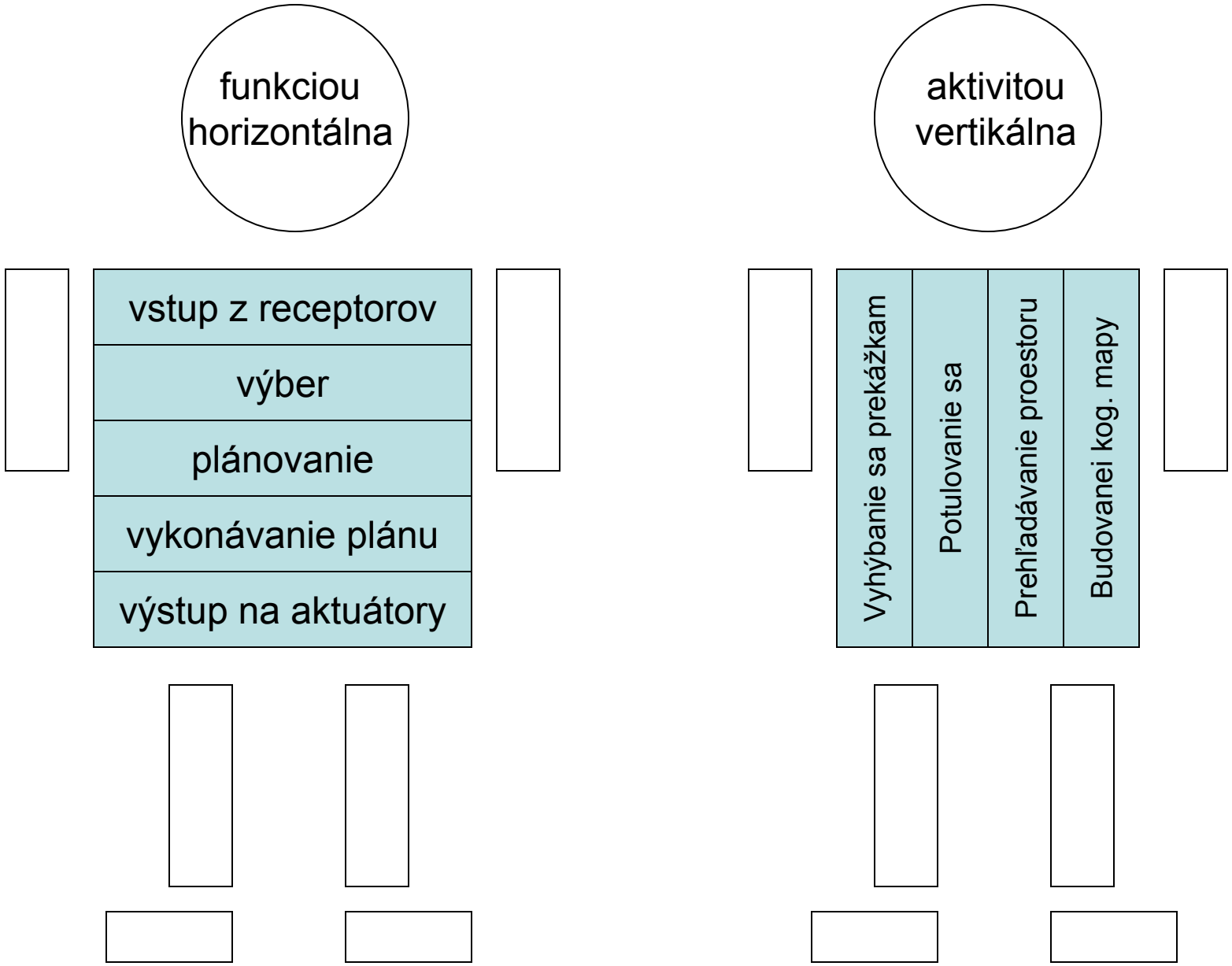
- **inkrementálny prístup**
- **vývoj zdola nahor, hierarchia**
- **stelesnenosť, ladenie**
- **vtieranie sa vyššej vrstvy do nižšej**

Dekompozícia

- Funkciou = spolu (z hľadiska hierarchických vrstiev) sú kódy realizujúce jednu funkciu, napr. spracovanie obrazu

- Aktivitou = spolu sú kódy realizujúce jednu aktivitu, napr. vyhýbanie sa prekážkam

Subsumčná architektúra je založená na dekompozícii aktivitou


funkciou
horizontálna
vstup z receptorov
výber
plánovanie
vykonávanie plánu
výstup na aktuátory
aktivitou
vertikálna
Vyhýbanie sa prekážkam
Potulovanie sa
Prehľadávanie proestoru
Budovanei kog. mapy

Subsumpcia

Ako môže vyššia vrstva pôsobiť na nižšiu ?

Ako sa môže vtierať do jej činnosťi ?

Zvážme, že vyššia vrstva sa navrhuje až keď je nižšia už hotová, takže tam nemôžeme mať nachystané interface-y umožňujúce toto vtieranie

Nižšia vrstva nie je otvorená

Subsumpcia

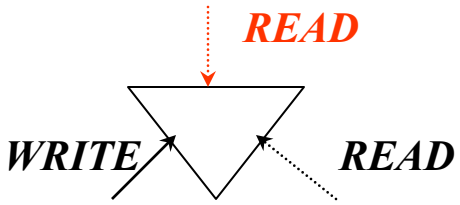
Ako sa vyššia môže vtierať do činnosti nižšej ?

Sú na tri mechanizmy:

- Odpočúvanie
- Supresia
- Inhibícia

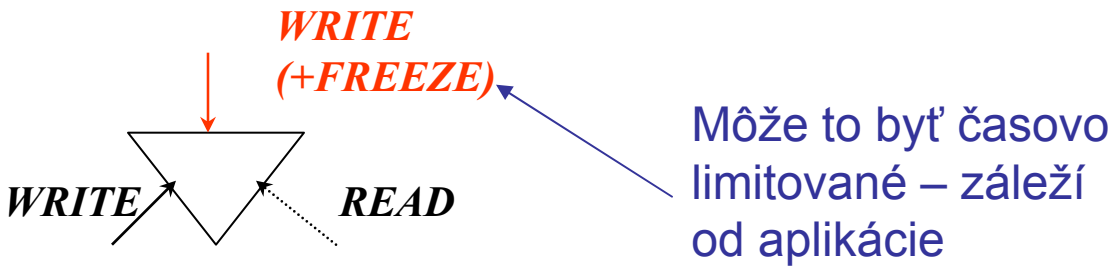
Odpočúvanie

- Vyššia vrstva nedeštruktívne odoberá údaje z nižšej, sleduje čo sa tam deje



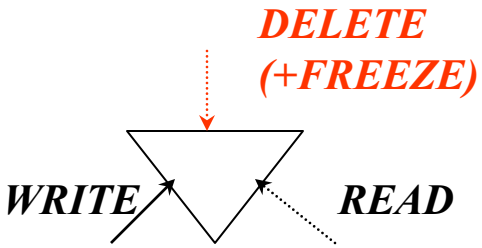
Supresia

- Vyššia vrstva nahradí údaj plynúci v nižšej vlastnou hodnotou



Inhibícia

- Vyššia vrstva zmaže údaj v nižšej vrstve . Zastaví tam dátový tok



Mobilné roboty skonštruované na základe subsumčnej architekúry

ALLEN – prechádzanie priestoru

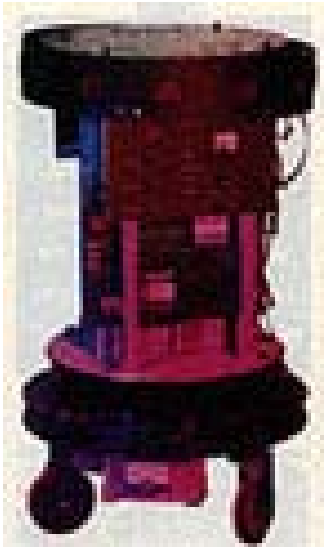
HERBERT – odnášanie prázdnych plechoviek do koša

TOTO – budovanie kognitívnej mapy

METATOTO – prichádzanie na určené miesto

COG – (humanoidný) učenie napodobňovaním (sledovanie pohľadu)

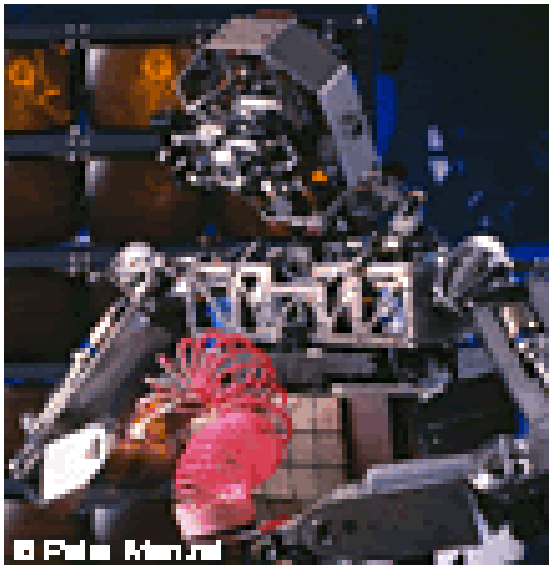
Mobilné roboty skonštruované na základe subsumčnej architekúry



ALLEN

HERBERT

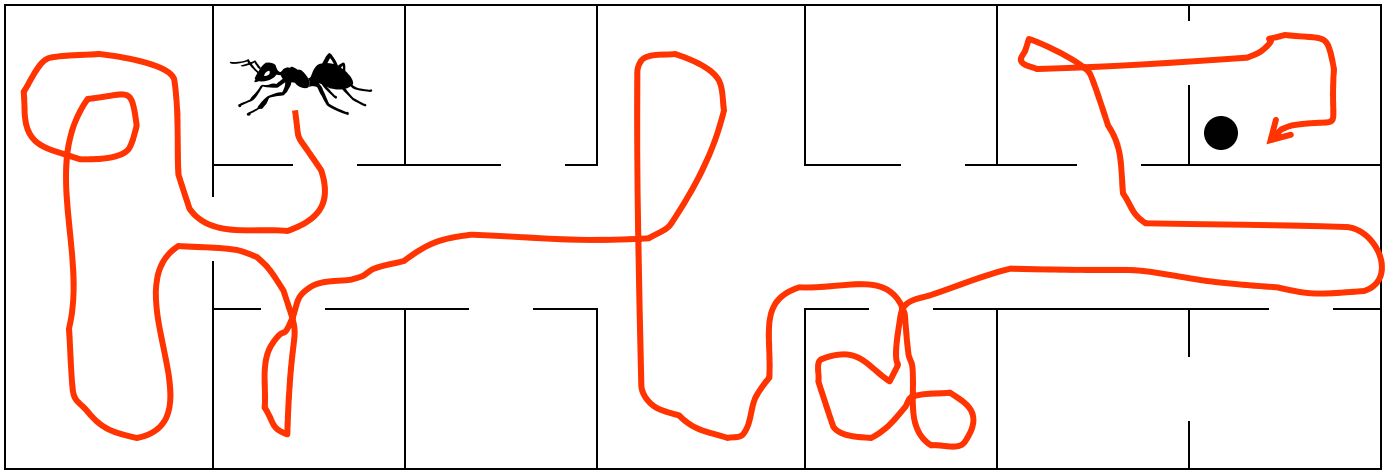
TOTO



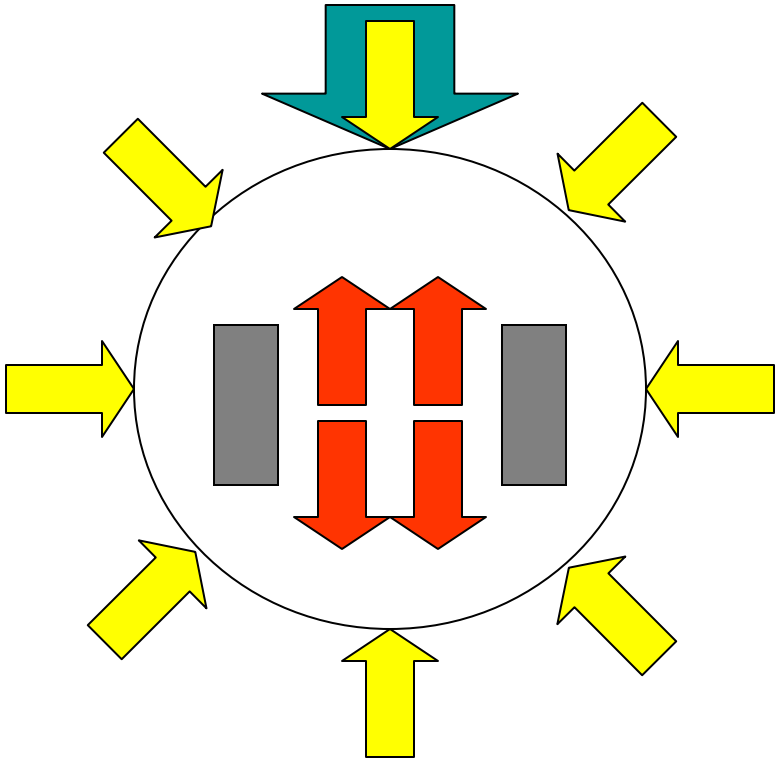
COG

Riadiaci systém robota “ALLEN”

Úloha: urobiť robota, ktorý dokáže nájsť na zem umiestnenú kovovú guličku niekde na 4 poschodí bloku D.



Senzory a aktuátory



Sonary na detekciu najbližej prekážky

Kolesá na pohyb vpred, vzad a otáčanie

Detektor guličky

Návrh systému

Podľa princípu subsumpcie

STOP – Zastavenie pri guličke

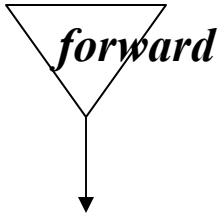
EXPLORE – Prehľadávnie priestoru

WANDER - Potulovanie sa

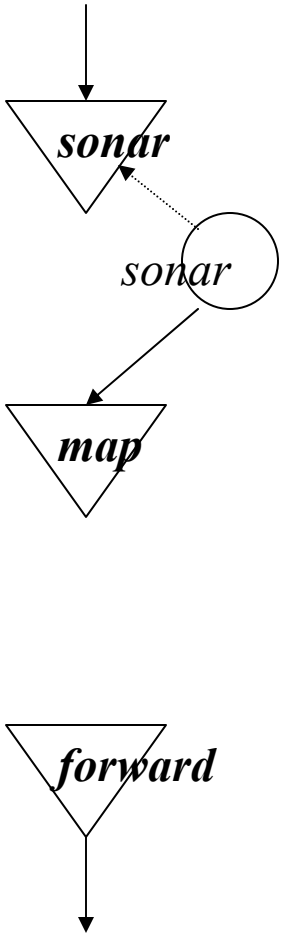
AVOID - Výhýbanie sa prekážkam

AVOID

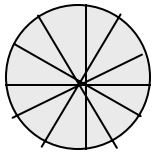
Za normálnych okolností, ideme stále vpred



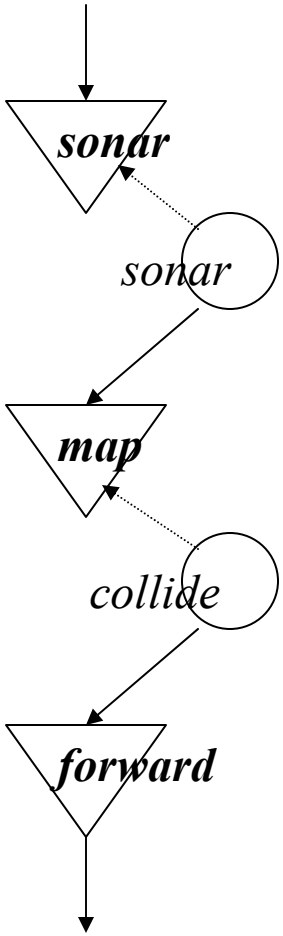
AVOID



Sonar zostaví pole vzdialeností najbližších prekážok v rôznych absolútnuch výsečiach



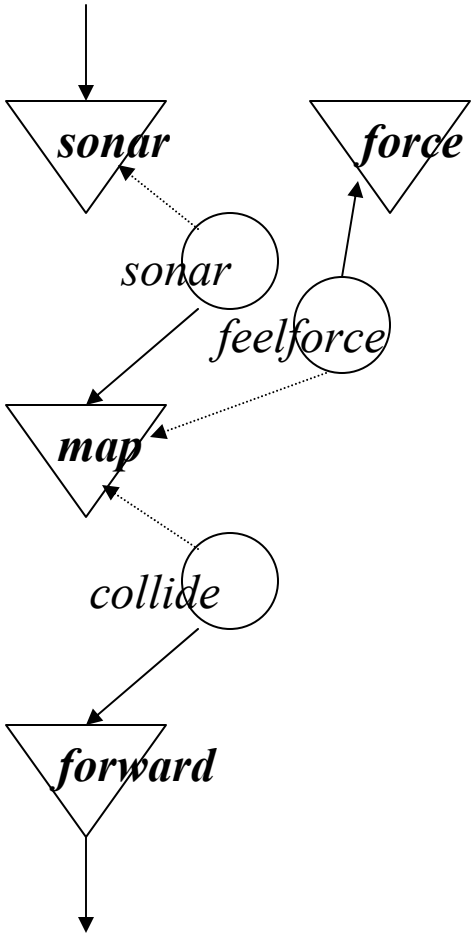
AVOID



Ked hrozí náraz, collide zastaví forward. Zastaví ho na základe údajov z dopredného sonaru

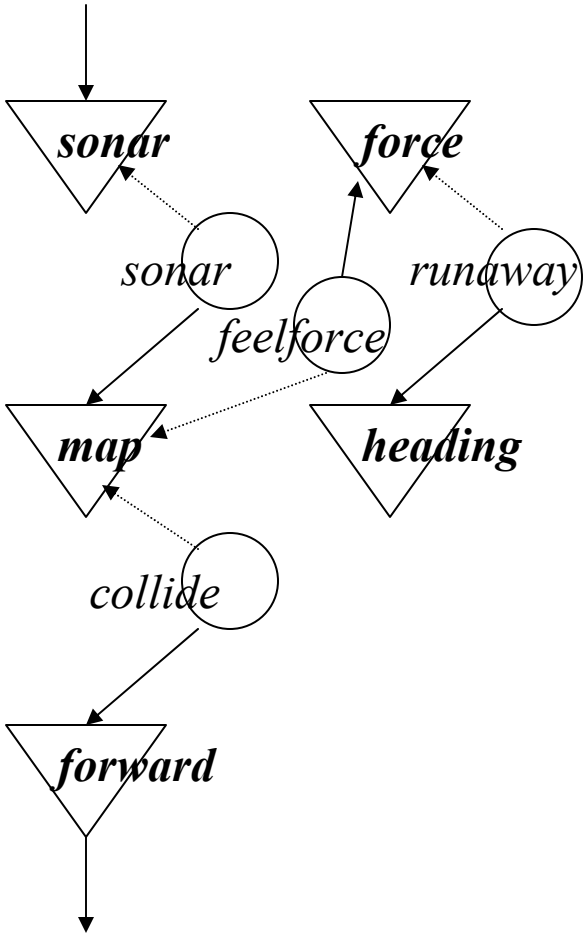
Tým pádom ideme rovno až kým nehrozí náraz, potom zastavíme

AVOID



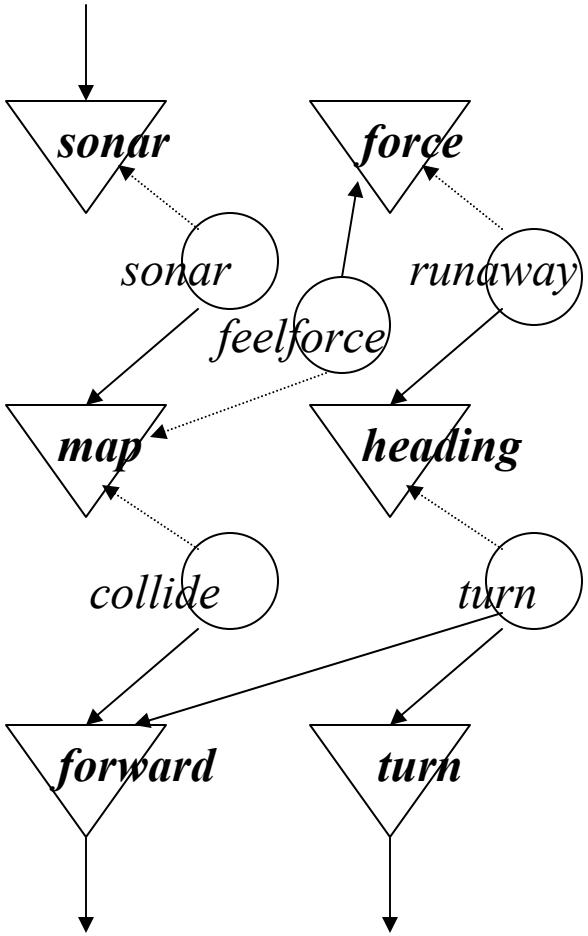
Feelforce vyhodnocuje smer v ktorom najviac hrozí zrážka

AVOID



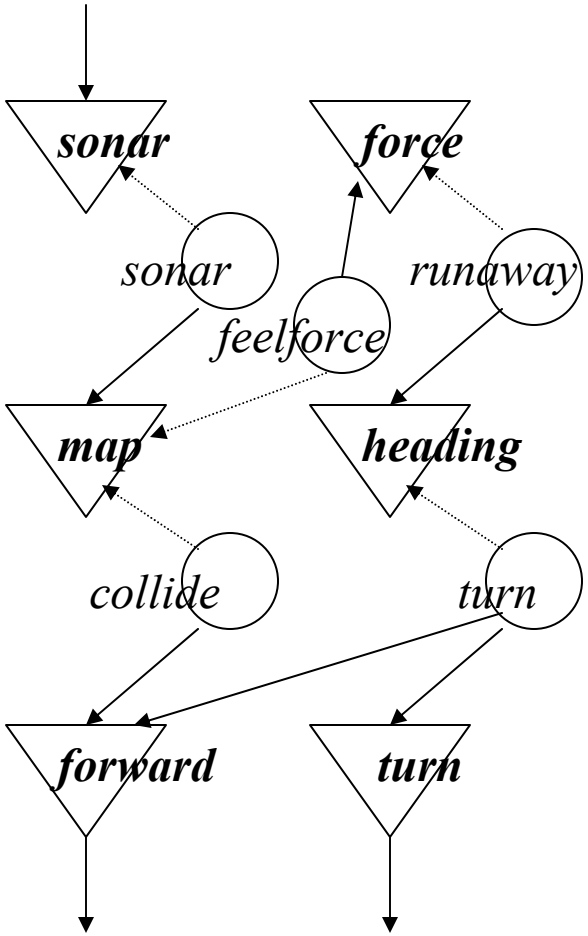
Runaway zavelí uberať sa opačným smerom

AVOID



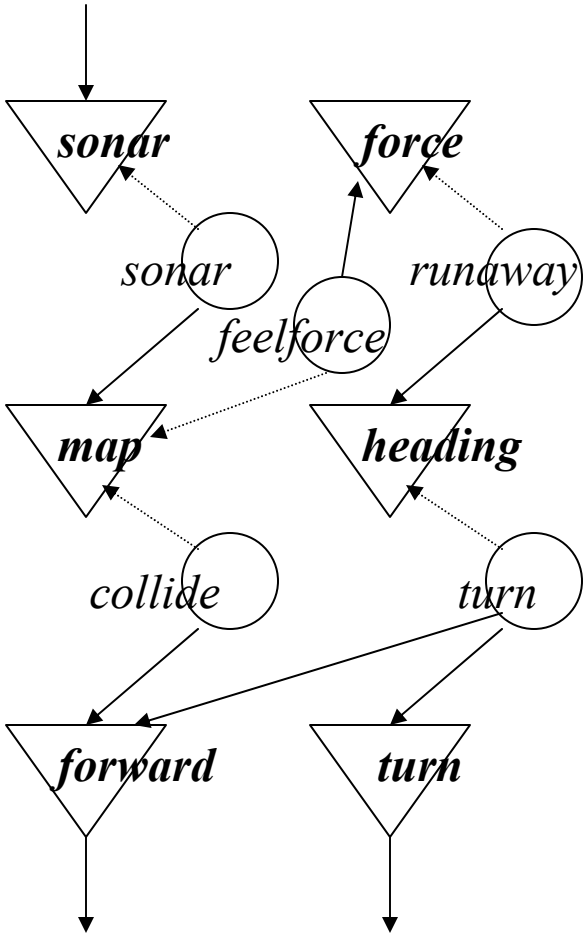
A turn podľa toho smeru riadi otáčanie kolies, prípadne aj cúvanie

AVOID



Uvedomme si, že toto otáčanie má spätne vplyv na obsah map a tým pádom na heading. Čím viac sa robot odkloní od smeru v ktorom hrozí zrážka, tým menší je odklon požadovaný v heading

AVOID

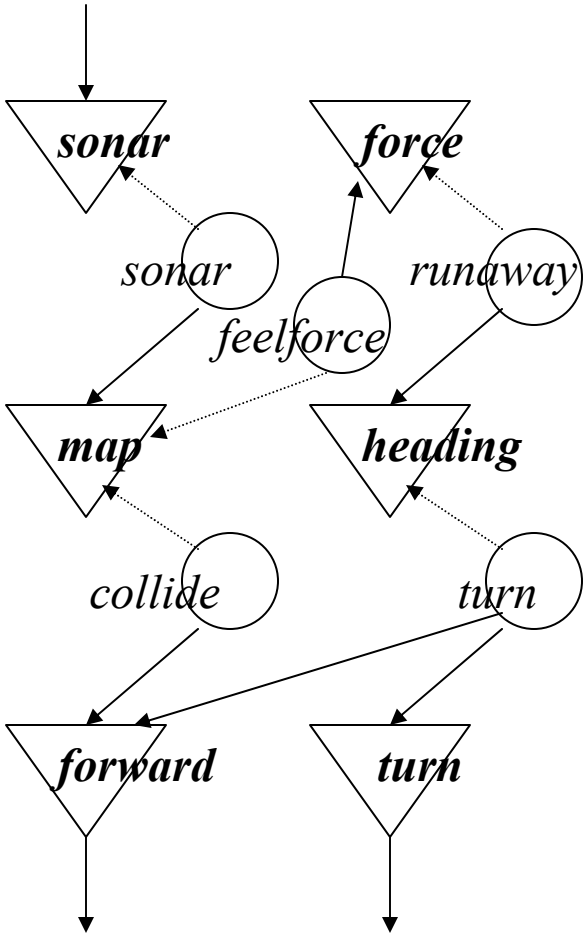


Tým pádom pojdeme rovno až kým nehrozí náraz. Potom sa žačneme vyhýbať, a potom sa opäť pohybujeme rovno.

Pri tomto pohybe sa pomerne ľahko dostaneme do nejakého cyklu, ale na vrstvu AVOID to stačí.

AVOID je hotová.

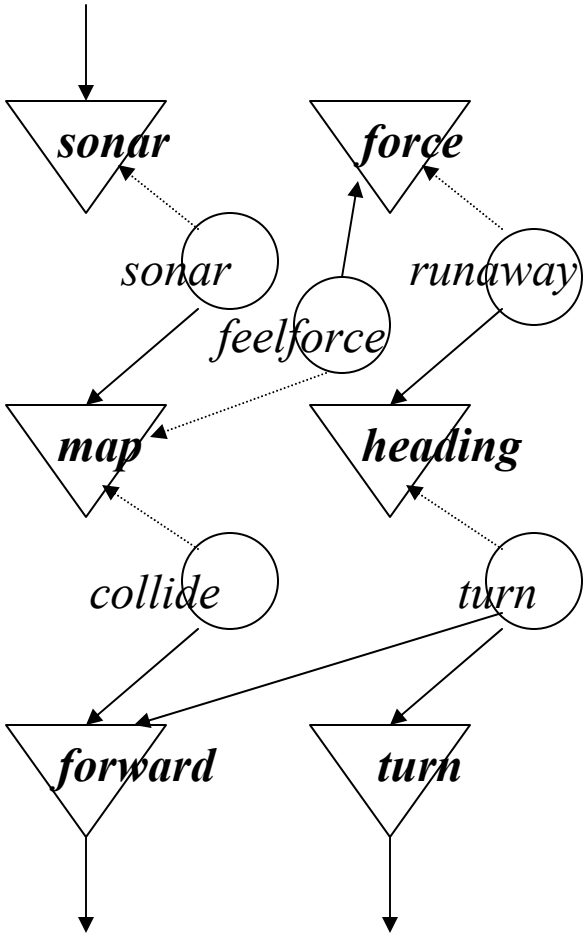
AVOID



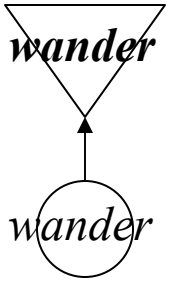
Všimnime si, že realizáciou vyhýbania sa statickým prekážkam, sme zároveň realizovali výbanie sa pohybujúcim objektom, ktoré nebezpečne križujú smer nášho pohybu.

Podobné prípady sa niekedy označujú ako tzv. emergencia.

AVOID



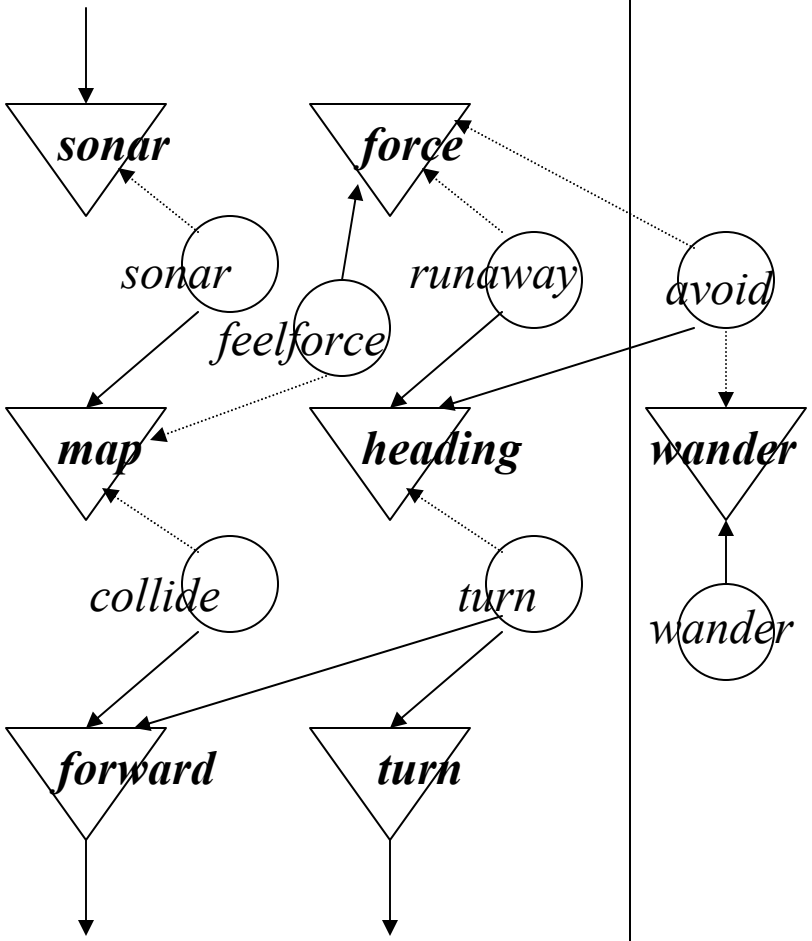
WAN DER



Teraz by sme chceli aby sem-tam porušil prípadné cyklenie náhodným pohybom

Preto agent wander s veľkým sleepom sem-navrhne určitý odklon smeru pohybu

AVOID



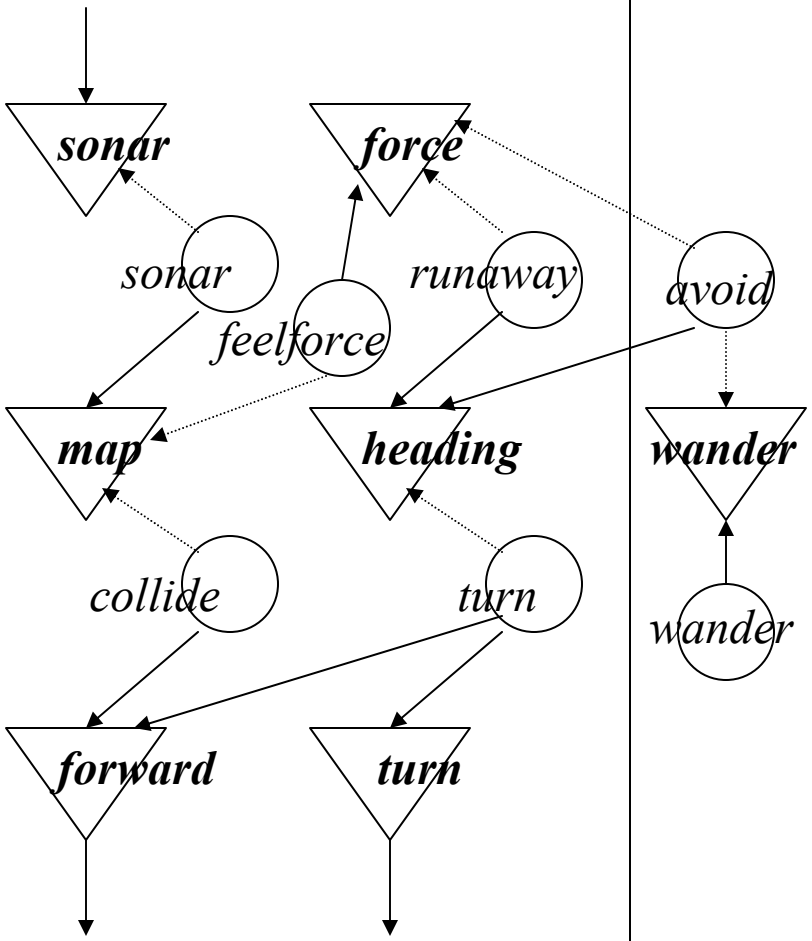
WAN DER

A agent avoid, ak sa cez force presvedčí, že momentálne nehrozí zrážka, prepíše navhrnutým smerom heading, tak ako keby nejaká zrážka hrozila.

Tým pádom využije vyhýbanie sa prekážkam na potulovanie sa

AVOID

WAN
DER



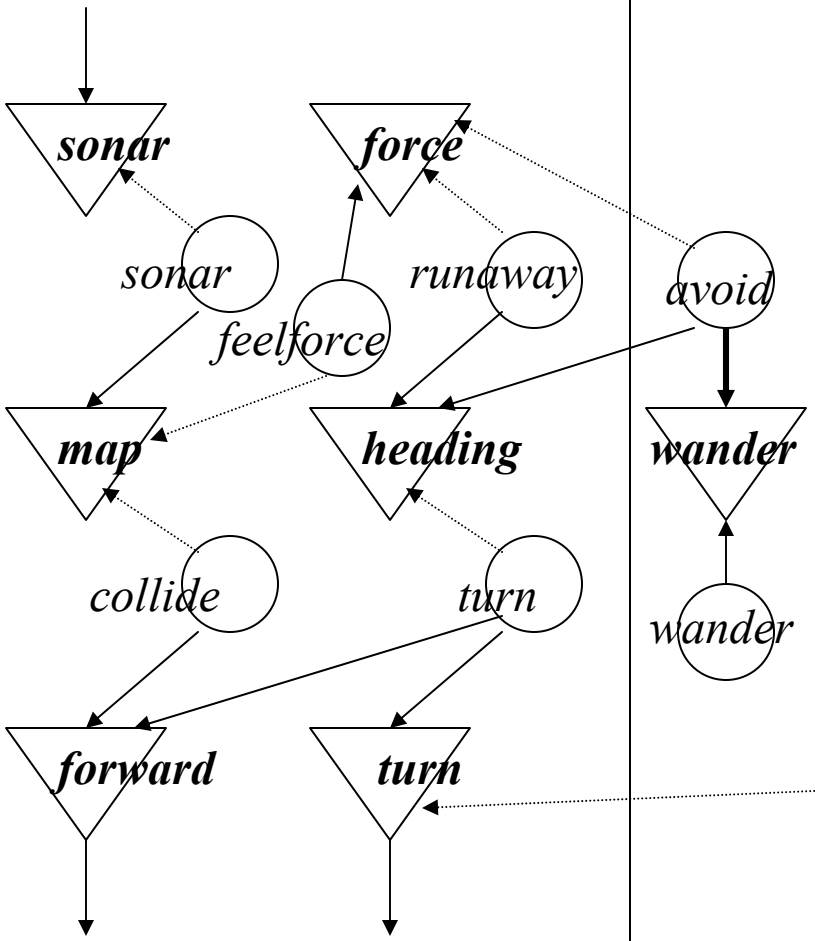
Tým pádom sa nielen vyhýbame prekážkam, ale sa aj potulujeme a pohyb nášho robota už nie je predvídateľný.

Ale zatiaľ sa dosť motá na mieste.

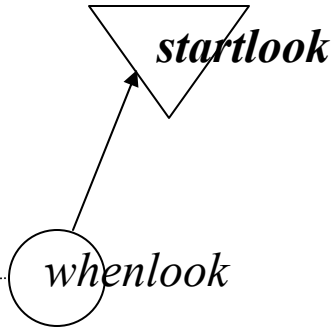
AVOID

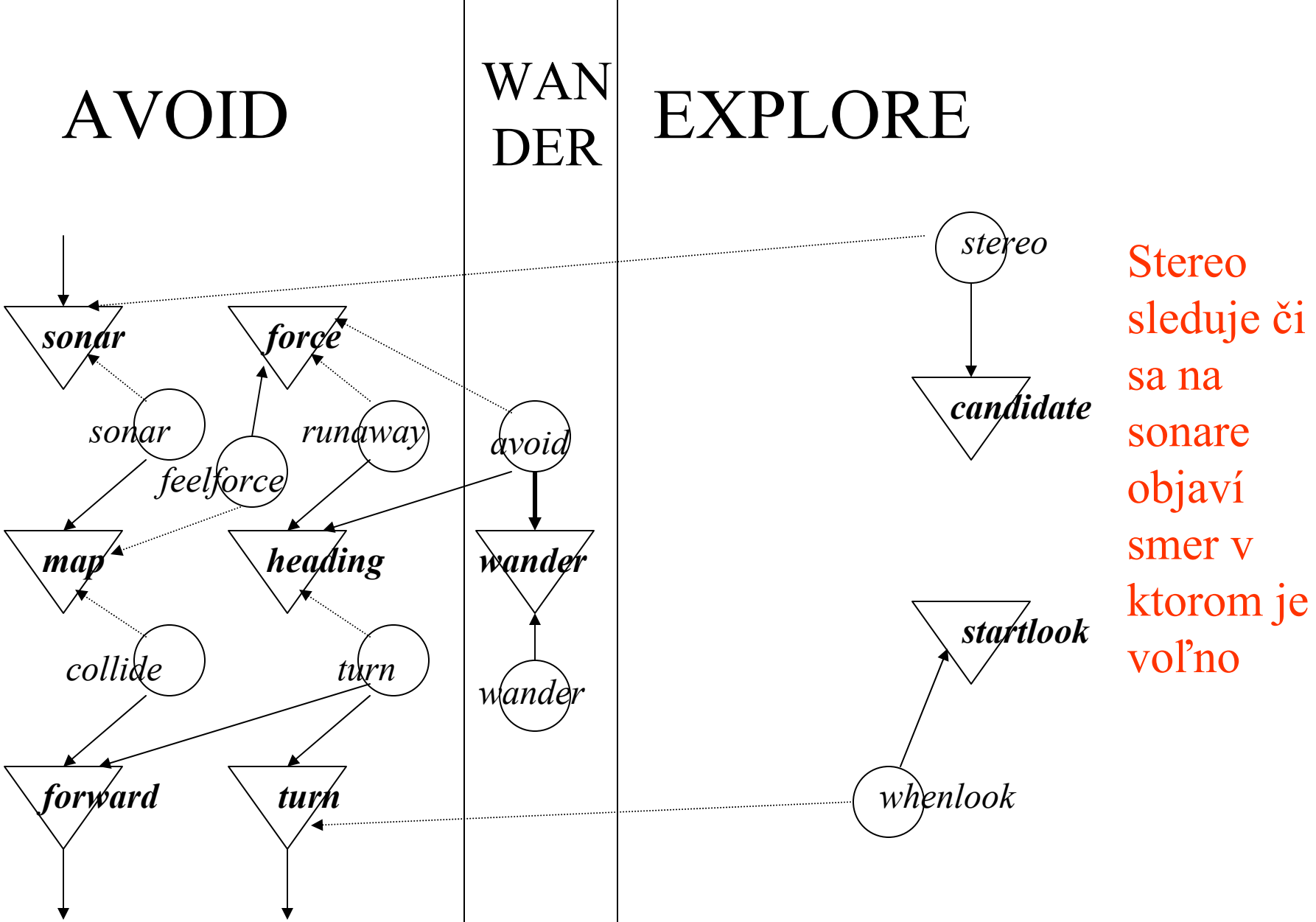
WAN DER

EXPLORE

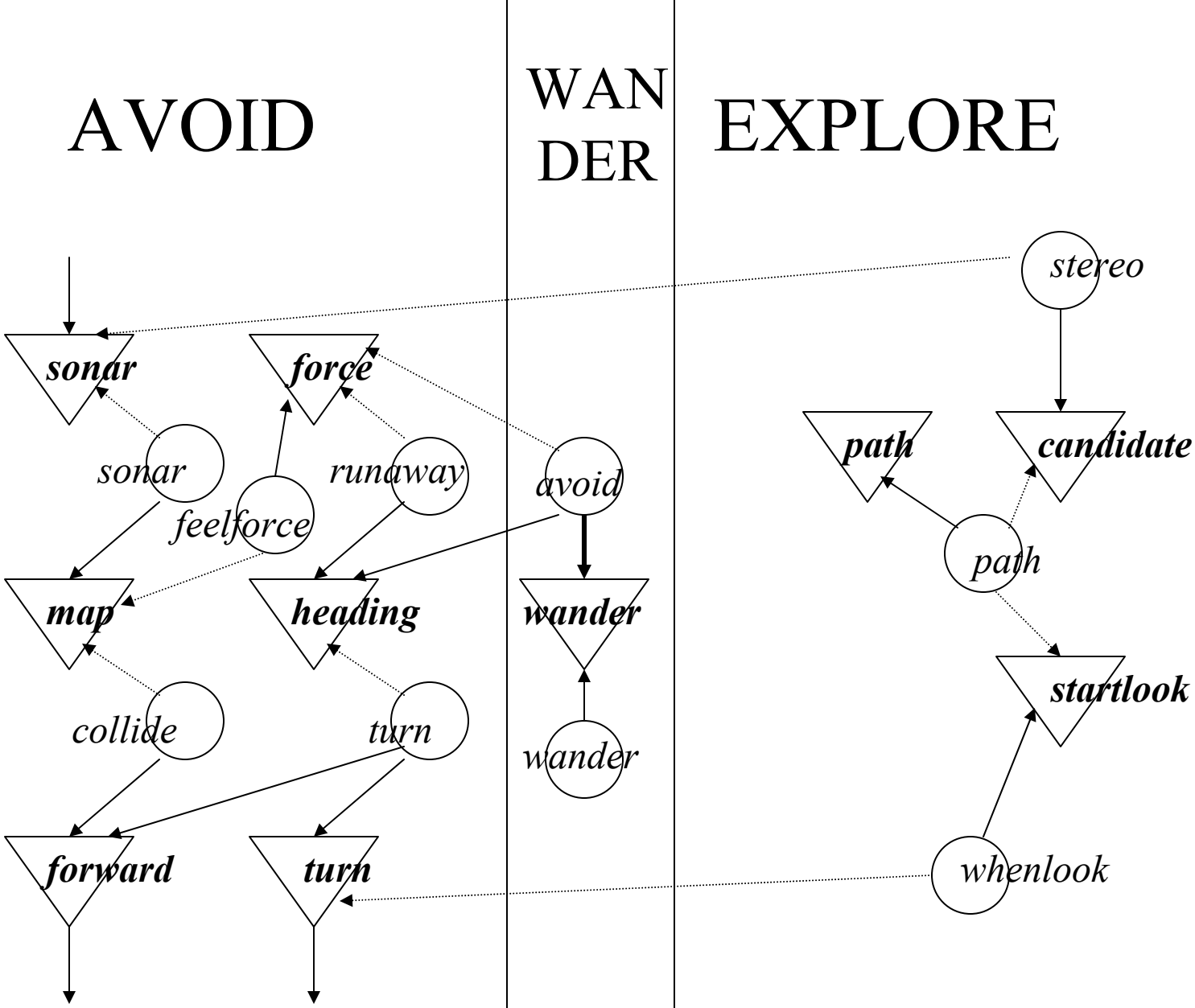


whenlook sleduje či sa nemotáme. Ak usúdi, že už je toho priveľa, dá povel na presun do inej oblasti

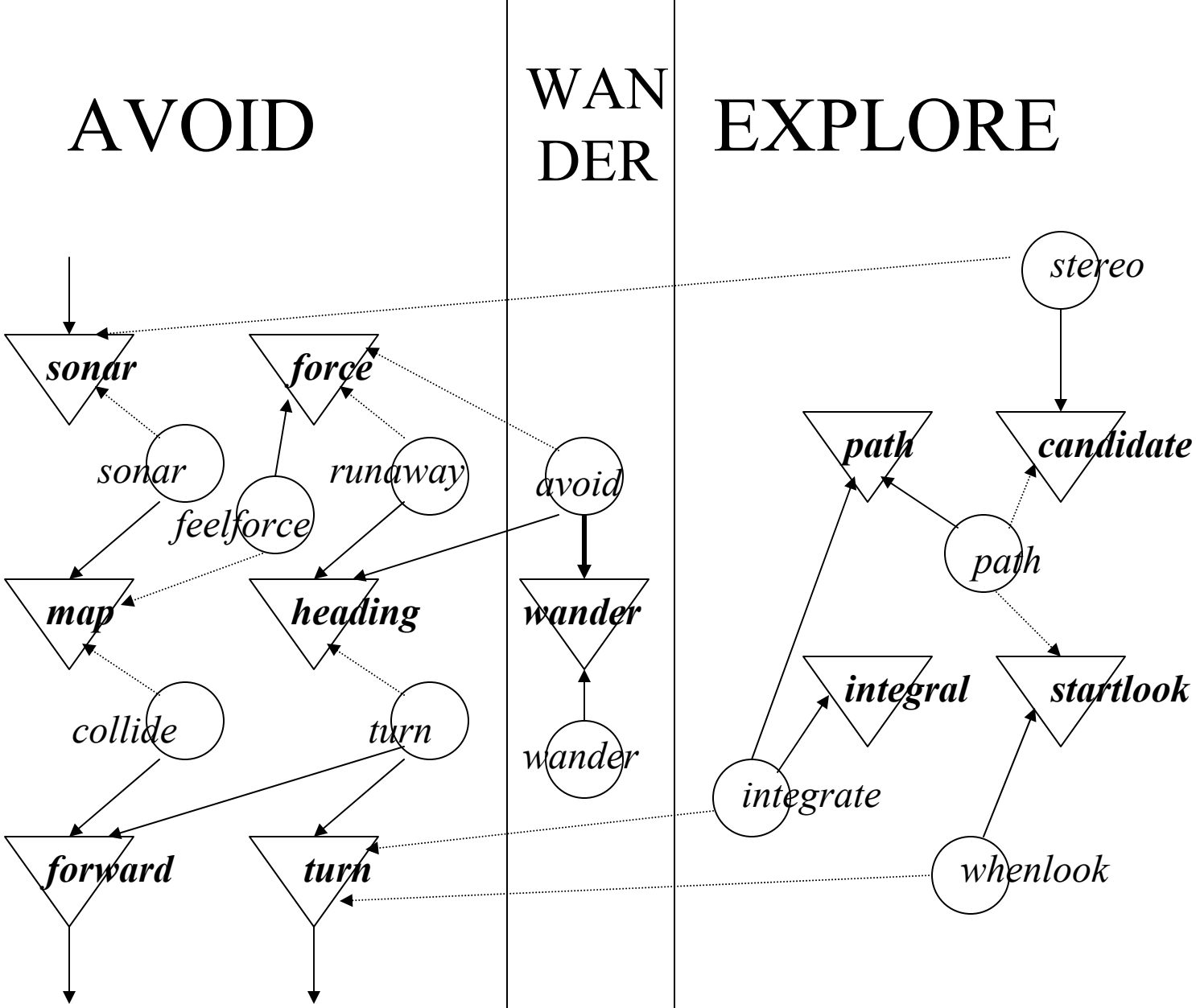




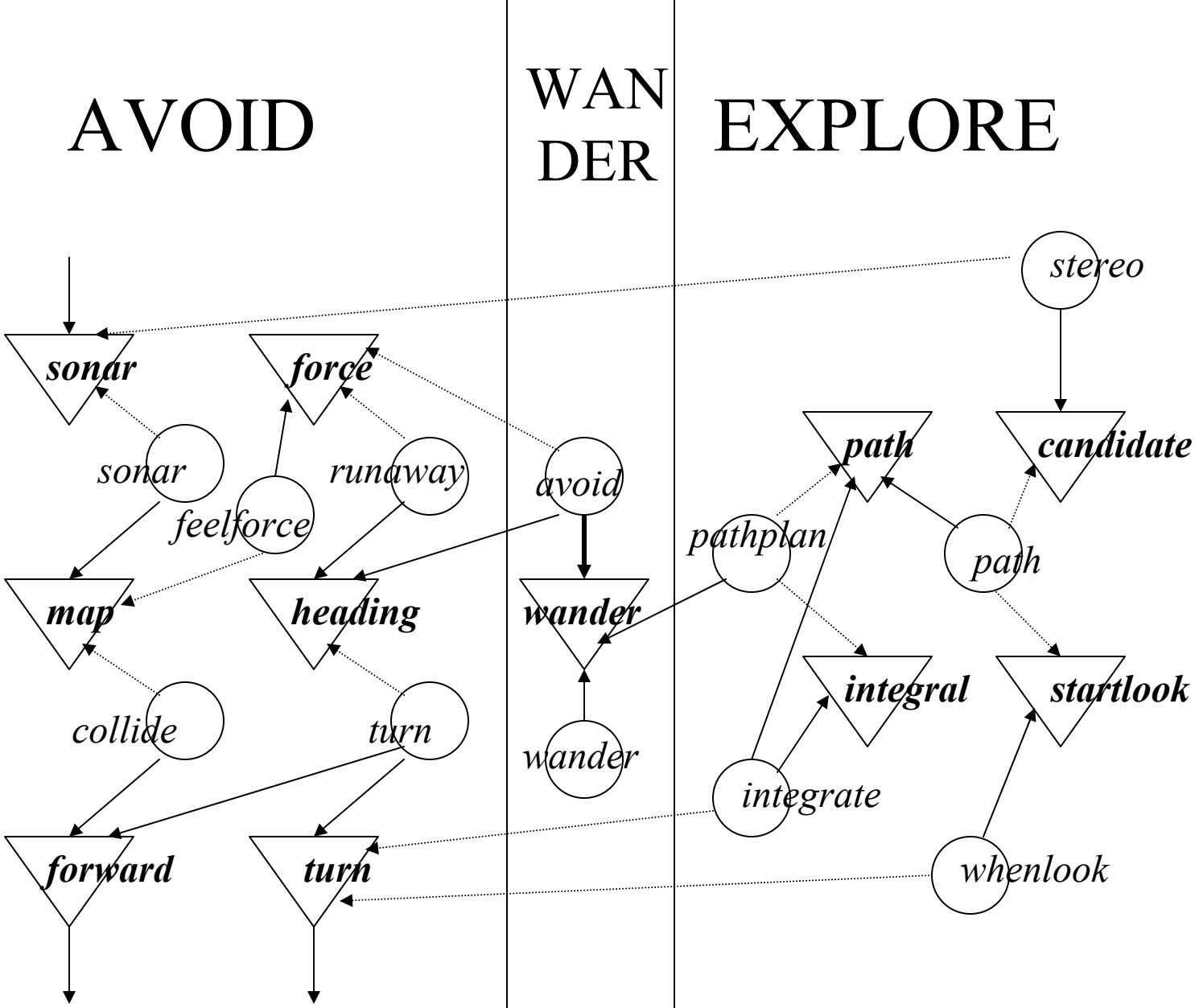
Stereo sleduje či sa na sonare objaví smer v ktorom je voľno



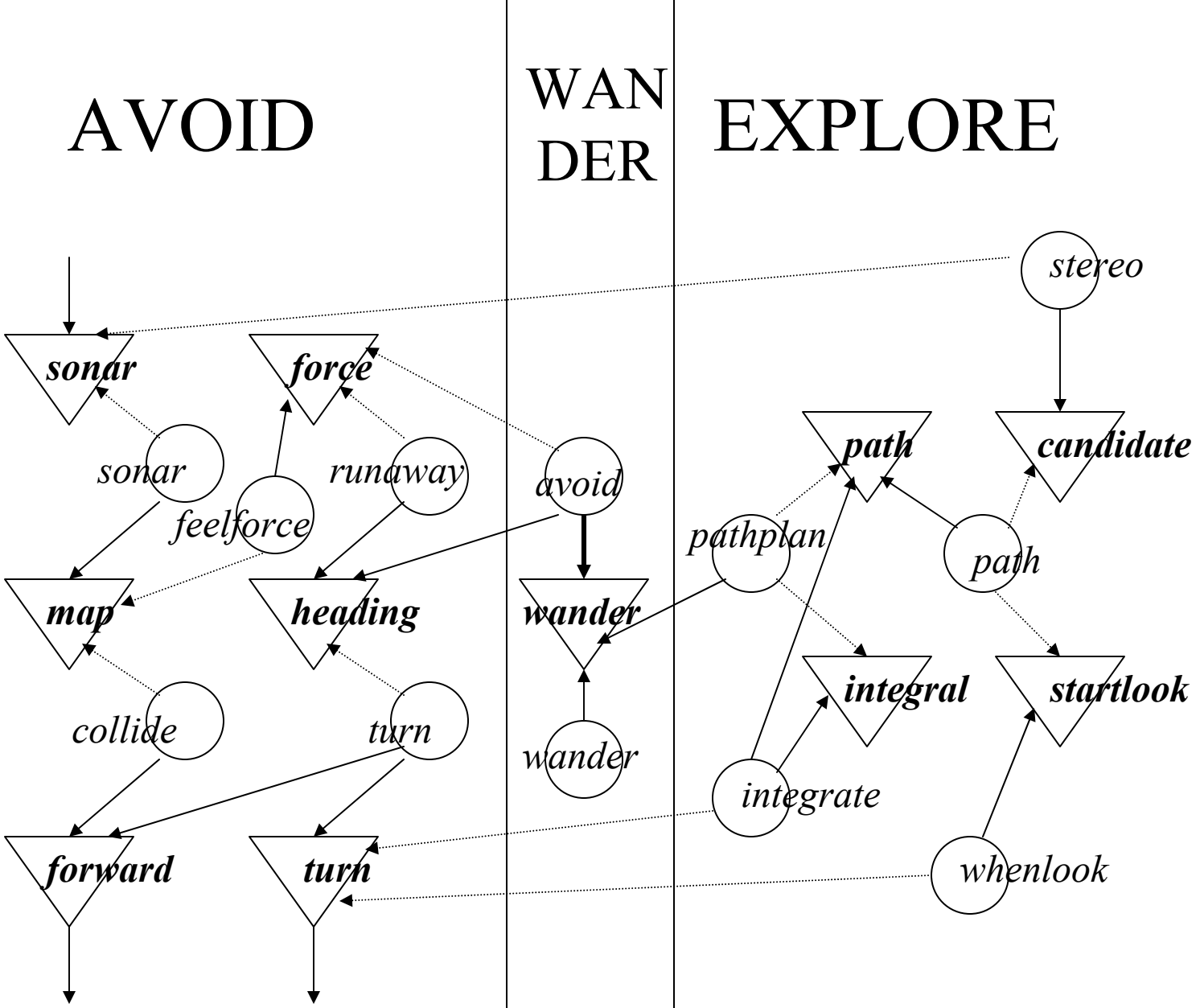
Path vytýči ktorým smerom sa presunie-me do inej oblasti



Je to ale relatívny smer, takže na to aby sme ho mohli absolutne chápať potrebuje-me sumárnu odchýľku

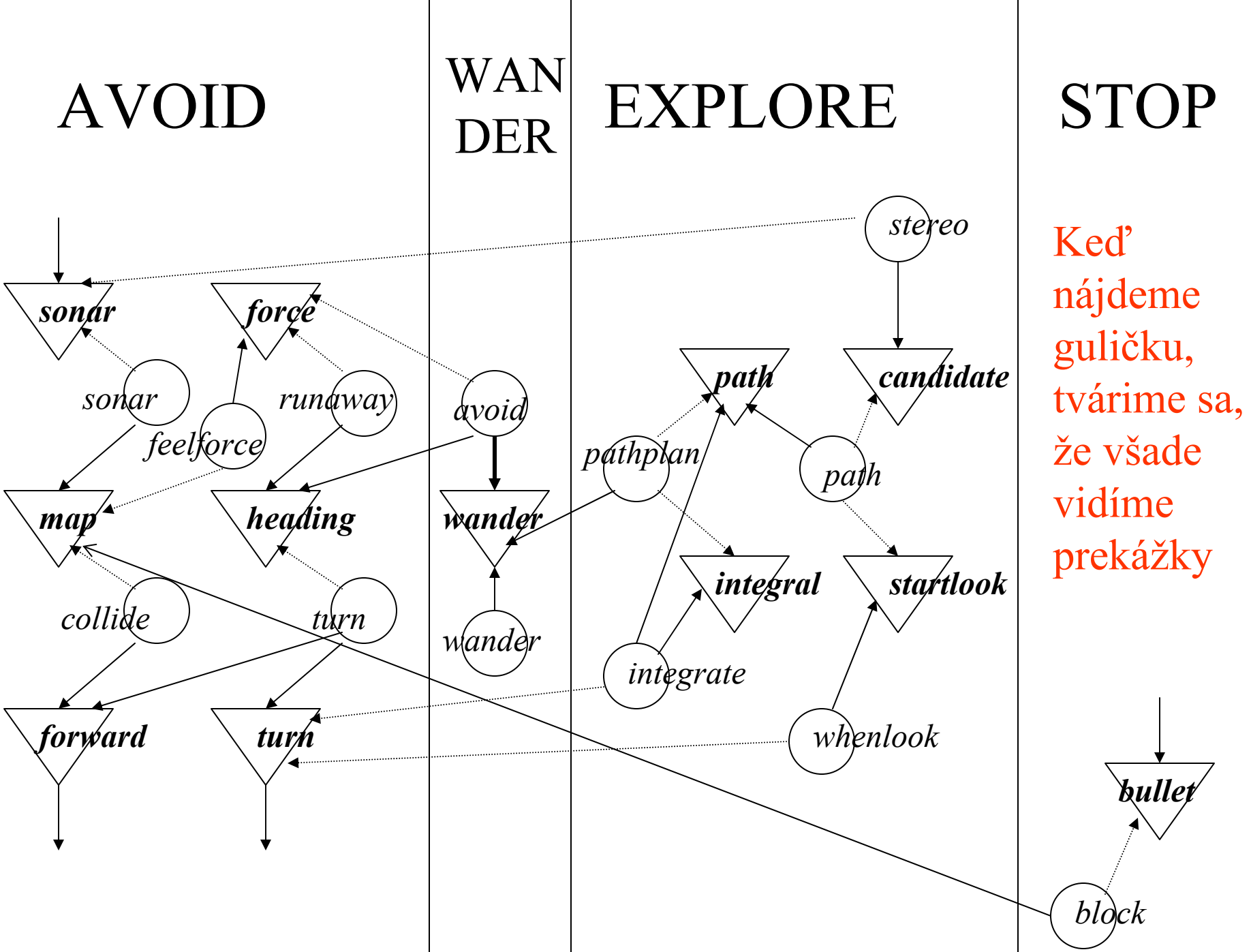


Pathplan v každej chvíli vie o aký uhol sa treba odchýli aby sme dodržali zvolený uhol na presun



Už vieme prechádzať priestorom.

Hoci neuplatňujeme žiadne logicky správne prehľadávanie.



Keď nájdeme guličku, tvárime sa, že všade vidíme prekážky