

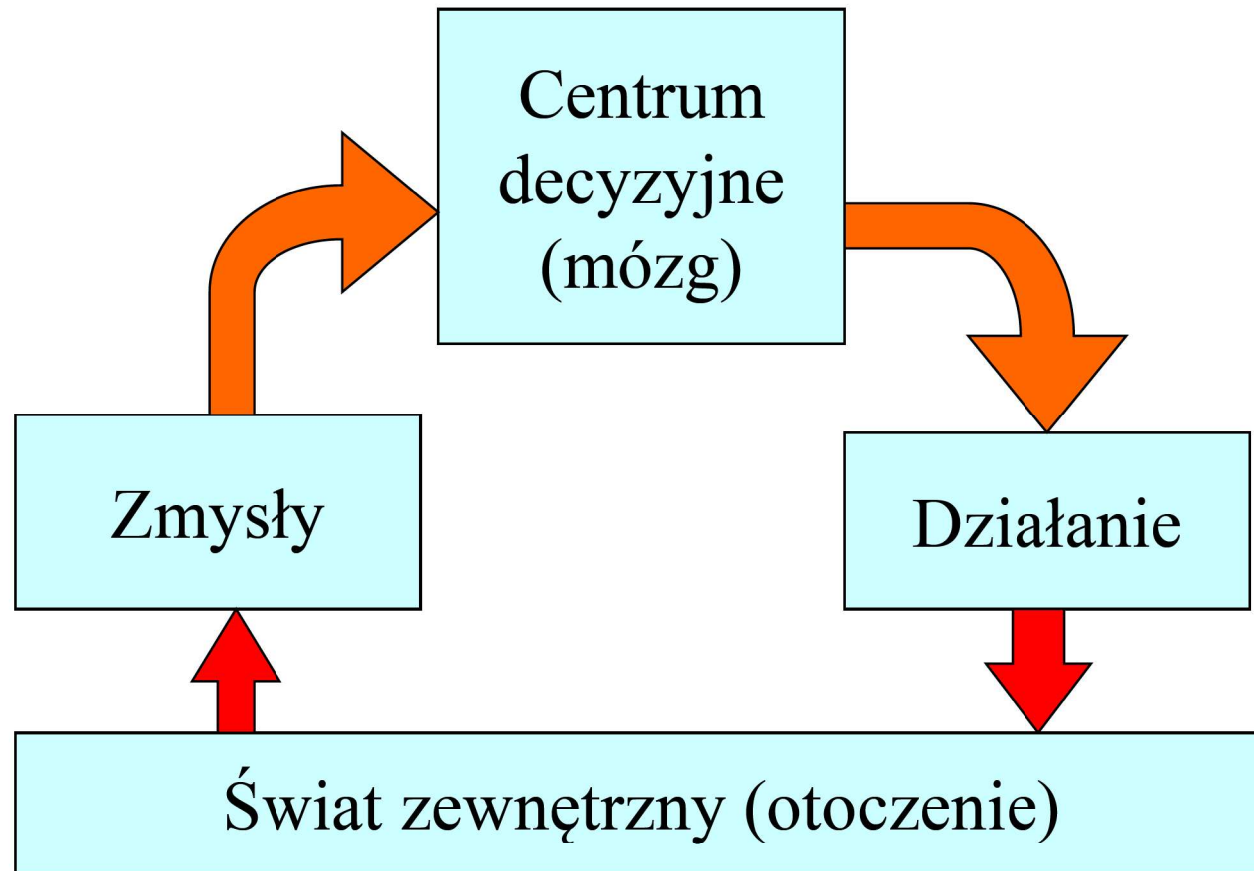
Roboty mobilne

Systemy sterowania

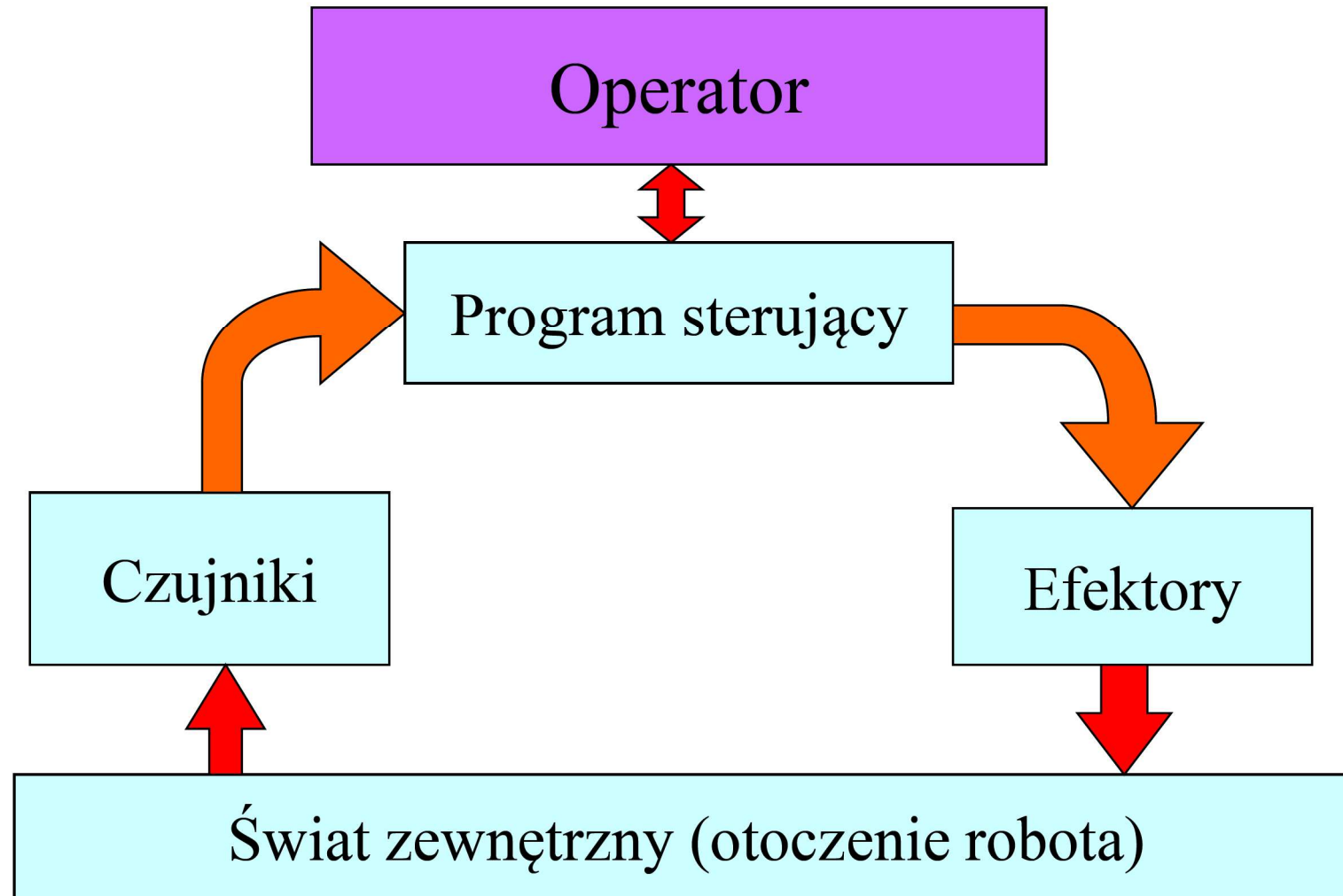
Systemy sterowania

- Zadania systemu sterowania
- Stosowany sprzęt
- Sterowanie *klasyczne*
- Sterowanie *behavioralne*

Sposób funkcjonowania człowieka



System sterowania robotem mobilnym



Zadania systemu sterowania

- pobieranie i przetwarzanie danych z czujników
- gromadzenie informacji
- podejmowanie decyzji odnośnie ruchu i innych działań
- sterowanie urządzeniami wykonawczymi (efektorami)
- współpraca z operatorem
- współpraca z urządzeniami zewnętrznymi

Stosowany sprzęt

- Podział zadań na komputer pokładowy i stacjonarny
- Komputer stacjonarny – zadania bardzo obciążające obliczeniowo i niekoniecznie pilne
- Komputery pokładowe:
 - dedykowany układ logiczny – bardzo proste roboty
 - mikrokontrolery
 - zazwyczaj sterowanie na niskim poziomie
 - stosowane w niedużych robotach jako podstawowe
 - ze wzrostem wydajności coraz chętniej stosowane
 - komputery klasy PC:
 - energooszczędne do zastosowań wbudowanych, np. PC-104 i inne rozwiązania

Sprzęg z otoczeniem

- specjalizowane układy we/wy
- magistrale do obsługi czujników oraz efektorów
- przetworniki analogowo-cyfrowe i cyfrowo-analogowe
- urządzenia do komunikacji z komputerami stacjonarnymi (WiFi, Bluetooth, radiowe zwiększonego zasięgu, GSM, podczerwień)
- urządzenia do współpracy z operatorem:
 - rozpoznawanie i synteza mowy
 - joysticki, pokrętła, przełączniki, kontrolki, wyświetlacze, panele operatorskie, ...

System operacyjny

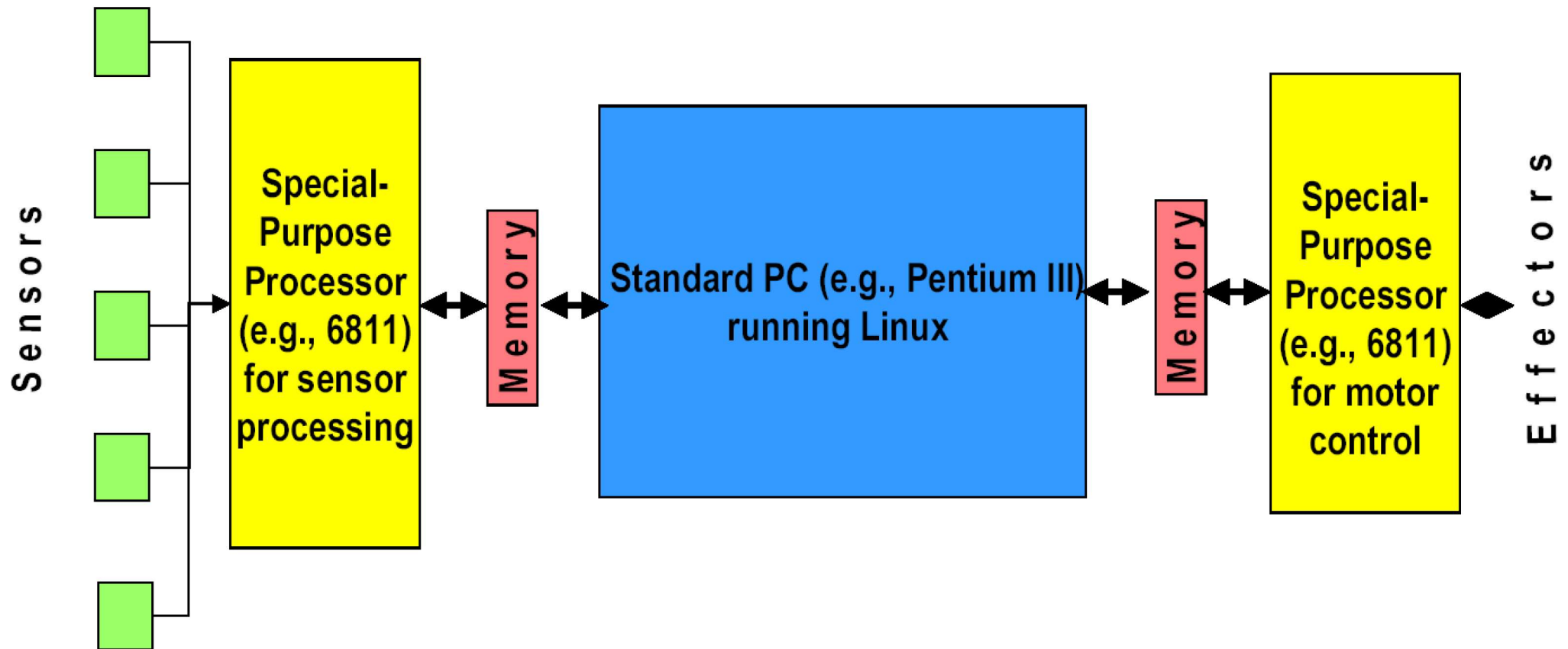
Wymagania **systemu czasu rzeczywistego**:

- łatwość obsługi nietypowych urządzeń zewnętrznych (obsługa tzw. zdarzeń zewnętrznych)
- terminowe wykonywanie zadań sterowania
- łatwość programowania wielowątkowego
- możliwość komunikowania się między zadaniami (wątkami)
- dokładny pomiar czasu
- kontrola nad kolejnością wykonywania zadań
- przewidywalne zachowanie w każdych warunkach (stabilność)
- wyeliminowanie wszelkich elementów przypadkowości działania

Przykładowa implementacja

Sprzętowo dość leciwa, ale zasady działania się nie zmieniły

Zamiast pamięci wspólnej stosuje się rozwiązania sieciowe



Sterowanie *klasyczne* (hierarchiczne)

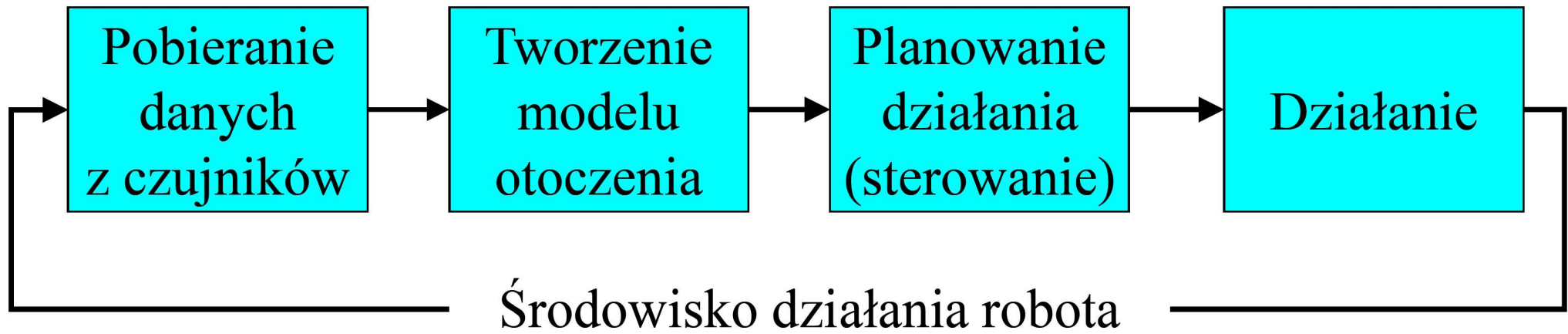
Ogólna charakterystyka:

- Struktura hierarchiczna (jasno określone zależności)
 - warstwy niższe pracują pod kontrolą wyższych
 - warstwy wyższe wykorzystują do wnioskowania efekty pracy warstw niższych
 - wyższe warstwy generują tzw. **podcele** dla niższych
- Czytelny podział funkcjonalny na poziomy
- Przewidywalna komunikacja i sterowanie
- Cel planowania zmienia się zależnie od poziomu
- Istotna rola sposobu reprezentacji wiedzy (reguły poruszania się, planowanie ruchu, ...)

Sterowanie *klasyczne* (hierarchiczne)

- system wzorowany na sposobie myślenia i działania człowieka
- zakłada się posiadanie określonej reprezentacji otoczenia (mapy)
- działanie robota jest zaplanowane i poparte dość skomplikowanym procesem decyzyjnym

Sterowanie hierarchiczne



- Przydatne dla wstępnie znanego środowiska o charakterze regularnym (zwykle wnętrza budynków: magazyny, biura, ...)
- Potrzeba posiadania modelu otoczenia
- Problem z niepewnymi danymi o otoczeniu

Przykłady architektury hierarchicznej (1)

Planer misji:

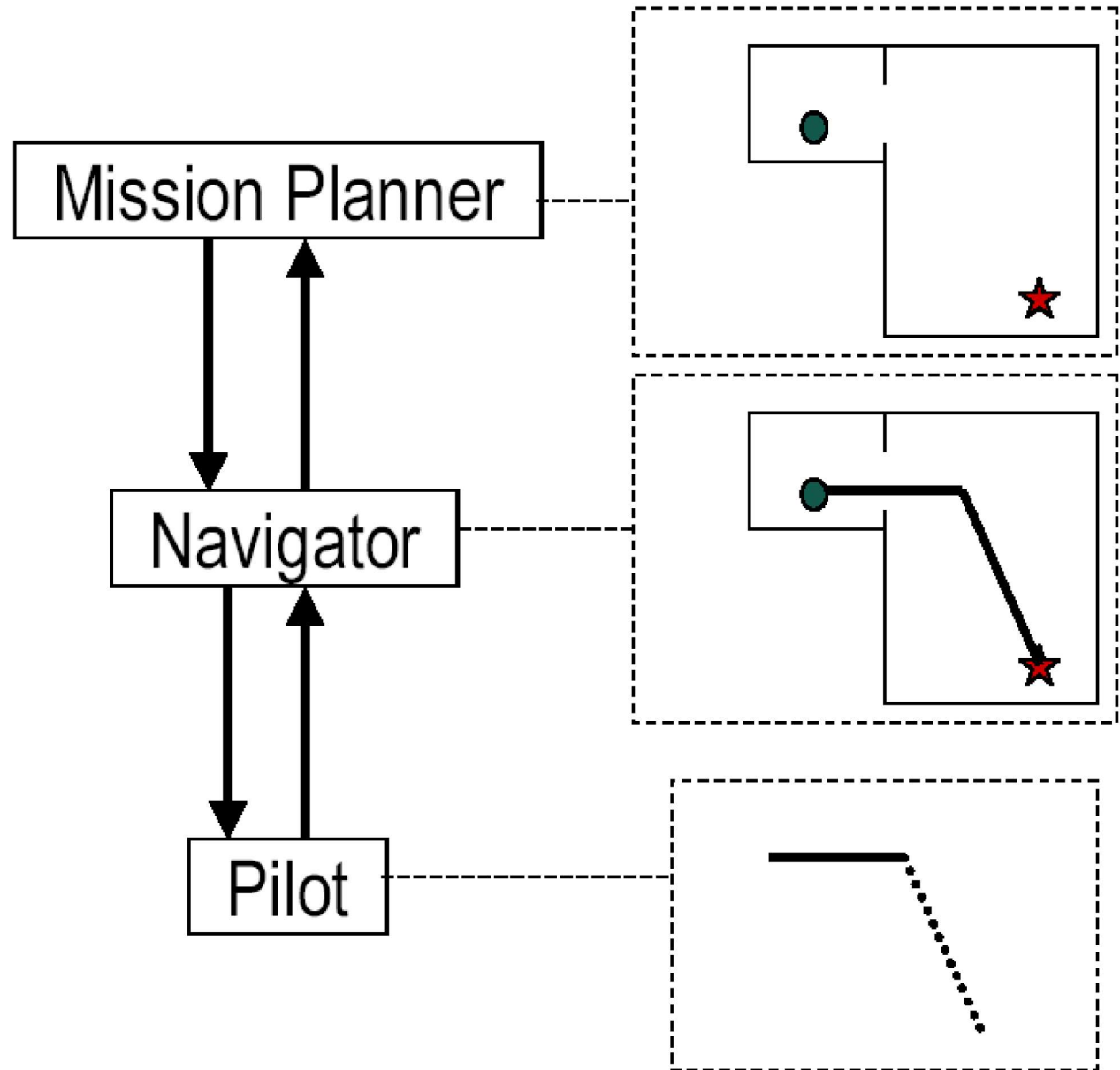
używa mapy do samolokalizacji i określenia celu

Nawigator:

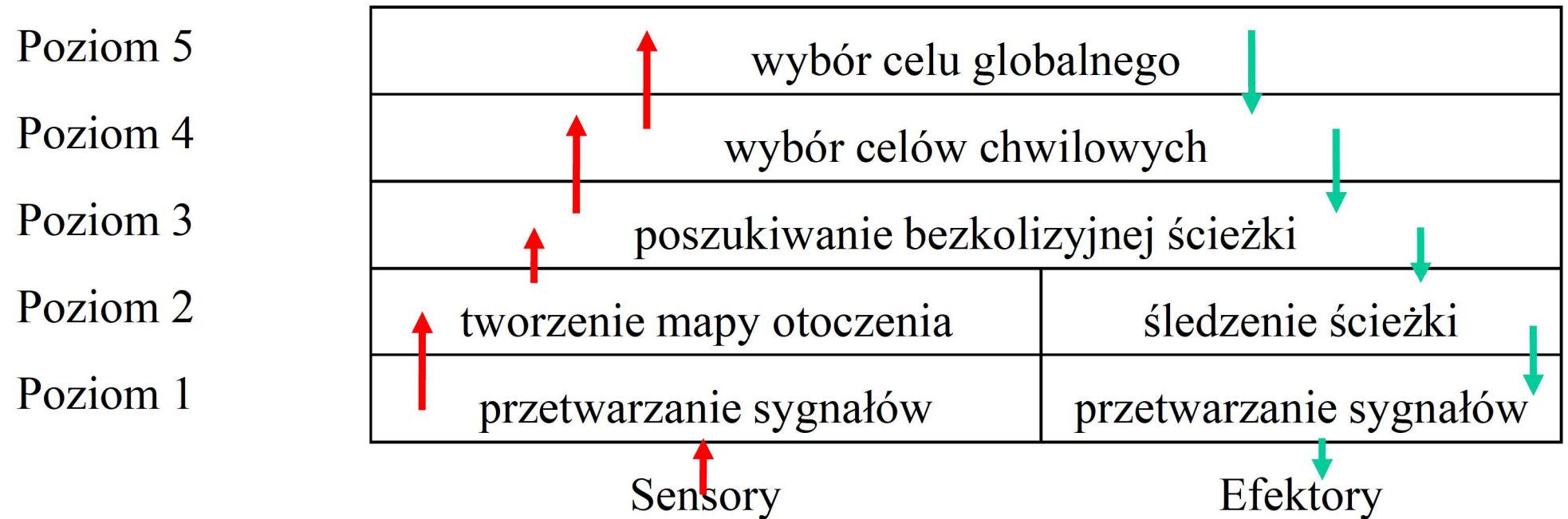
planuje ścieżkę od aktualnej pozycji do celu

Pilot:

określa, co robot musi zrobić, by zrealizować fragment ścieżki



Przykłady architektury hierarchicznej (2)



Sterowanie hierarchiczne

Zalety

- dość łatwa do realizacji budowa modułowa
- sprawna praca w dobrze znanym środowisku

Wady

- powolne działanie (przetwarzanie *w górę* i *w dół*)
- kłopotliwa modyfikacja
- kiepskie działanie w warunkach niepełnej wiedzy

Sterowanie hierarchiczne

Zastosowania:

- systemy zamknięte (przemysł, usługi)
- systemy przewidywalne
- realizacja zadań powtarzalnych
- gdy potrzeba zawsze takich samych zachowań