Systemy wbudowane

Wykład 2 – System Wbudowany: specyfikacja i inżynieria

Przemek Błaśkiewicz

13 marca 2019

Zastosowania

- Księżyc, Mars i inne kosmosy;
- automotive, air industry;
- klima, sterowanie, pompowanie, przelewanie, ...
- dźwięk, wideo, sieci;
- AGD, RTV, ...

- +90% wynalazków związanych jest z elektroniką/komputerami;

- +90% wynalazków związanych jest z elektroniką/komputerami;
- automotive – zestawy „entertainment” (!!), samodzielne parkowanie...

- +90% wynalazków związanych jest z elektroniką/komputerami;
- automotive – zestawy „entertainment” (!!), samodzielne parkowanie...
- fly-by-wire, radary, radio, GPS,

- +90% wynalazków związanych jest z elektroniką/komputerami;
- automotive – zestawy „entertainment” (!!), samodzielne parkowanie...
- fly-by-wire, radary, radio, GPS,
- 30-45% kosztów auta to elektronika;

- +90% wynalazków związanych jest z elektroniką/komputerami;
- automotive – zestawy „entertainment” (!!), samodzielne parkowanie...
- fly-by-wire, radary, radio, GPS,
- 30-45% kosztów auta to elektronika;
- szacunkowo – liczba SW ogółem na świecie $> 10^{10}$;

Trudna definicja

System wbudowany

To system (komputerowy), będący częścią większego systemu, nim sterujący lub przekazujący dane.

Trudna definicja

LUB

Realizuje specyficzne i jednoznacznie określone funkcje.

Trudna definicja

LUB

System gdzie "przenikają się" *software* i *hardware*.

Trudna definicja

LUB

Działa jak komputer, ale nie wygląda jak komputer.

Charakterystyki SW

- Efektywność: energetyczna, $$$, ciężar, objętość kodu, ciepło, ...

Charakterystyki SW

- Efektywność: energetyczna, $$$, ciężar, objętość kodu, ciepło, ...
- Dedykowane: konkretne zastosowania/środkowisko/problem;

Charakterystyki SW

- Efektywność: energetyczna, $$$, ciężar, objętość kodu, ciepło, ...
- Dedykowane: konkretne zastosowania/środkowisko/problem;
- Specjalny UI: bez klawiatury/ekranu, (problem z debugowaniem: trudnodostępne miejsce wdrożenia, złożoność testowania, nikłe narzędzia CAD);

Charakterystyki SW

- Efektywność: energetyczna, $$$, ciężar, objętość kodu, ciepło, ...
- Dedykowane: konkretne zastosowania/środkowisko/problem;
- Specjalny UI: bez klawiatury/ekranu, (problem z debugowaniem: trudnodostępne miejsce wdrożenia, złożoność testowania, nikłe narzędzia CAD);
- Ograczniczenia czasu rzeczywistego: reakcje na (a)synchroniczne zdarzenia o różnych priorytetach i koszcie niedotrzymania czasu wykonania; *W systemach czasu rzeczywistego dobra odpowiedź wygenerowana za późno jest złą odpowiedzią.*

Charakterystyki SW

- Efektywność: energetyczna, $$$, ciężar, objętość kodu, ciepło, ...
- Dedykowane: konkretne zastosowania/środkowisko/problem;
- Specjalny UI: bez klawiatury/ekranu, (problem z debugowaniem: trudnodostępne miejsce wdrożenia, złożoność testowania, nikłe narzędzia CAD);
- Ograczniczenia czasu rzeczywistego: reakcje na (a)synchroniczne zdarzenia o różnych priorytetach i koszcie niedotrzymania czasu wykonania; *W systemach czasu rzeczywistego dobra odpowiedź wygenerowana za późno jest złą odpowiedzią.*
- Hybrydowość i reaktywność: cyfrowo-analogowe, oddziałują i odbierają bodźce od otoczenia;

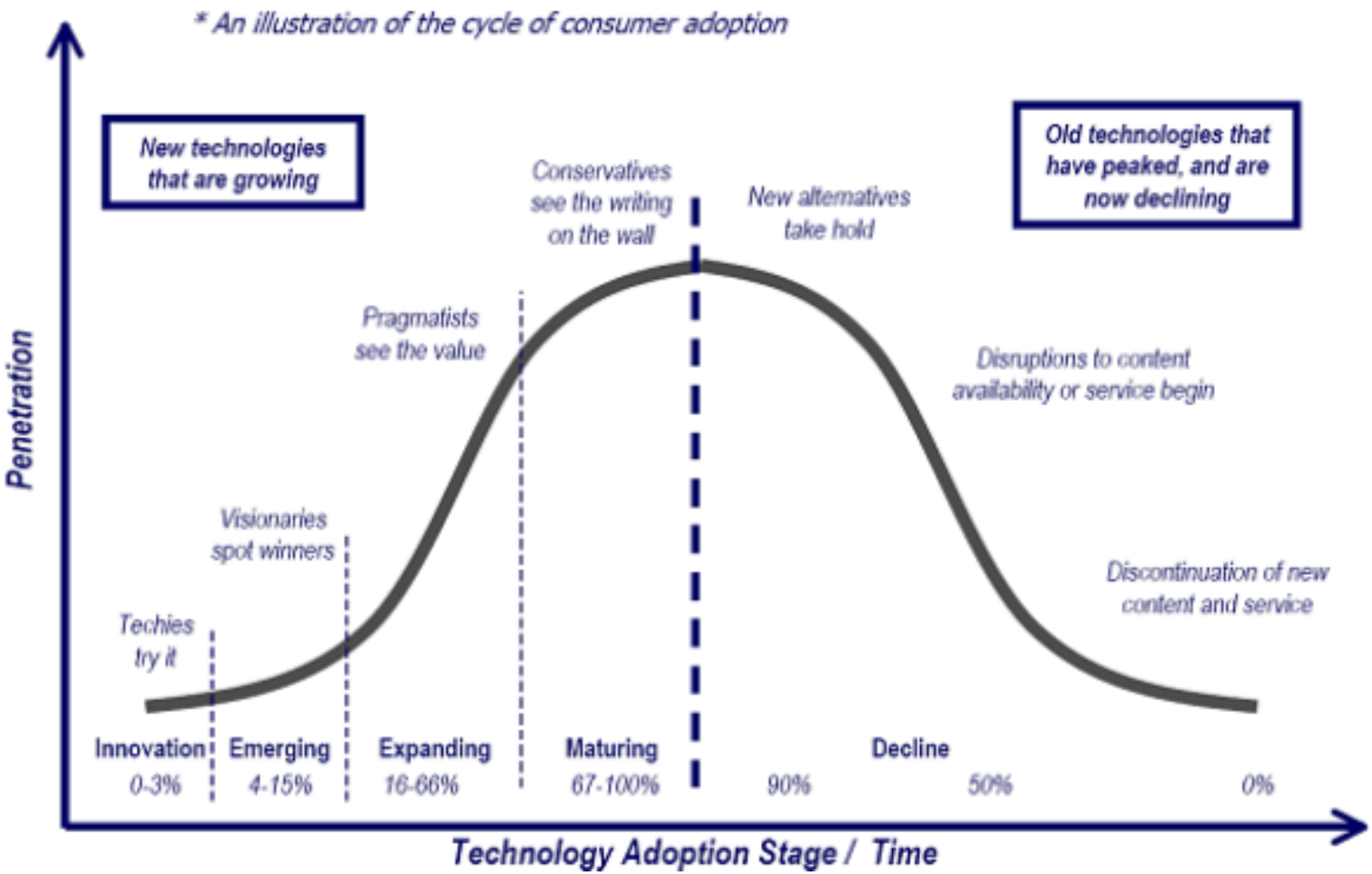
Charakterystyki SW

- Efektywność: energetyczna, $$$, ciężar, objętość kodu, ciepło, ...
- Dedykowane: konkretne zastosowania/środkowisko/problem;
- Specjalny UI: bez klawiatury/ekranu, (problem z debugowaniem: trudnodostępne miejsce wdrożenia, złożoność testowania, nikłe narzędzia CAD);
- Ograczniczenia czasu rzeczywistego: reakcje na (a)synchroniczne zdarzenia o różnych priorytetach i koszcie niedotrzymania czasu wykonania; *W systemach czasu rzeczywistego dobra odpowiedź wygenerowana za późno jest złą odpowiedzią.*
- Hybrydowość i reaktywność: cyfrowo-analogowe, oddziałują i odbierają bodźce od otoczenia;
- Na pograniczu dwóch światów (częstotliwościowych, świetlnych, ruchomych, ...).

Charakterystyki SW

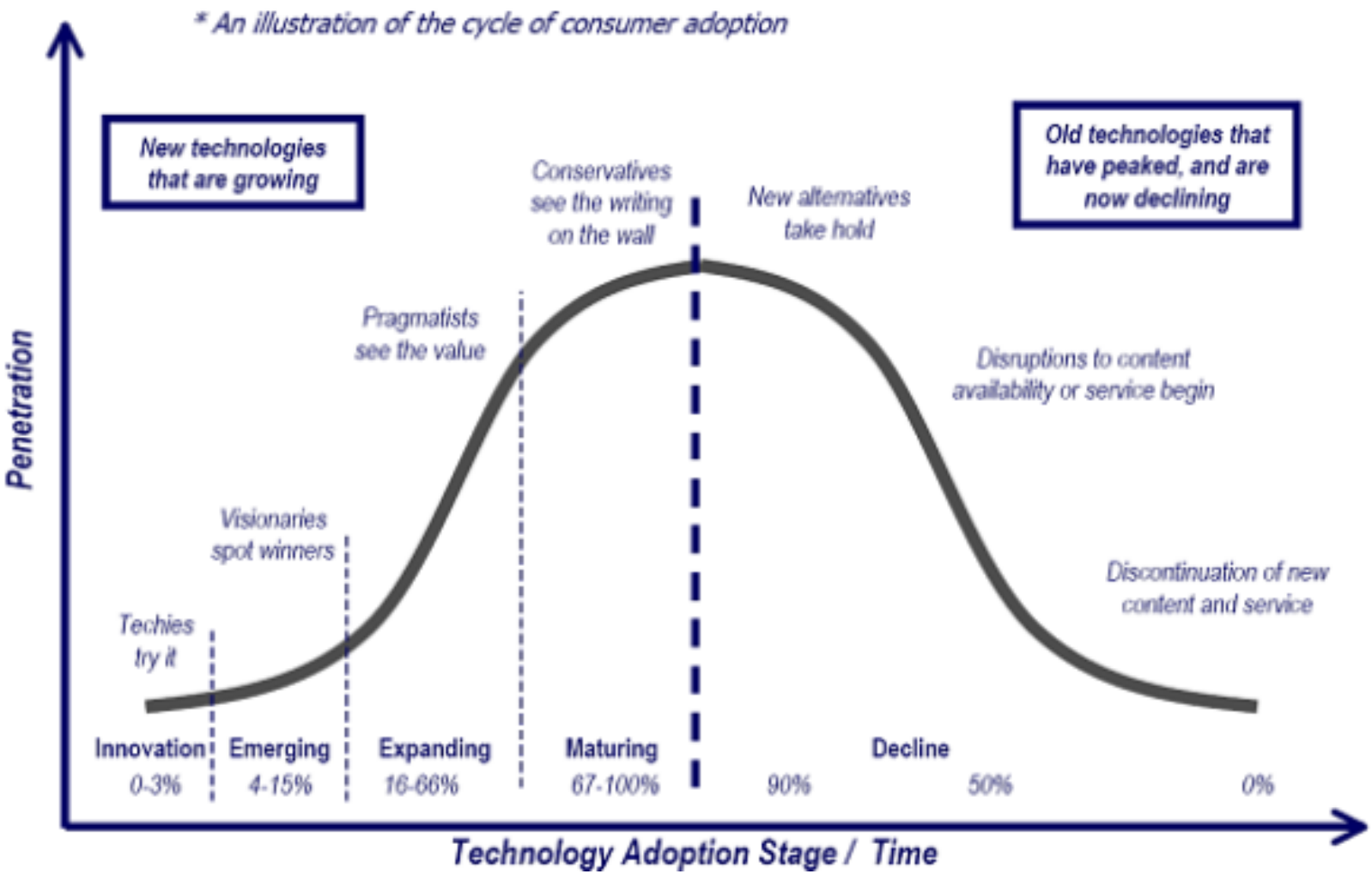
- Efektywność: energetyczna, $$$, ciężar, objętość kodu, ciepło, ...
- Dedykowane: konkretne zastosowania/środkowisko/problem;
- Specjalny UI: bez klawiatury/ekranu, (problem z debugowaniem: trudnodostępne miejsce wdrożenia, złożoność testowania, nikłe narzędzia CAD);
- Ograczniczenia czasu rzeczywistego: reakcje na (a)synchroniczne zdarzenia o różnych priorytetach i koszcie niedotrzymania czasu wykonania; *W systemach czasu rzeczywistego dobra odpowiedź wygenerowana za późno jest złą odpowiedzią.*
- Hybrydowość i reaktywność: cyfrowo-analogowe, oddziałują i odbierają bodźce od otoczenia;
- Na pograniczu dwóch światów (częstotliwościowych, świetlnych, ruchomych, ...).
- Zoptymalizowane: cenowo, pod względem złożoności, późniejszego rozbudowywania, użycia gotowych elementów;

Trochę marketingu



http://www.crtc.gc.ca/eng/publications/reports/policymonitoring/2013/cmr6.htm

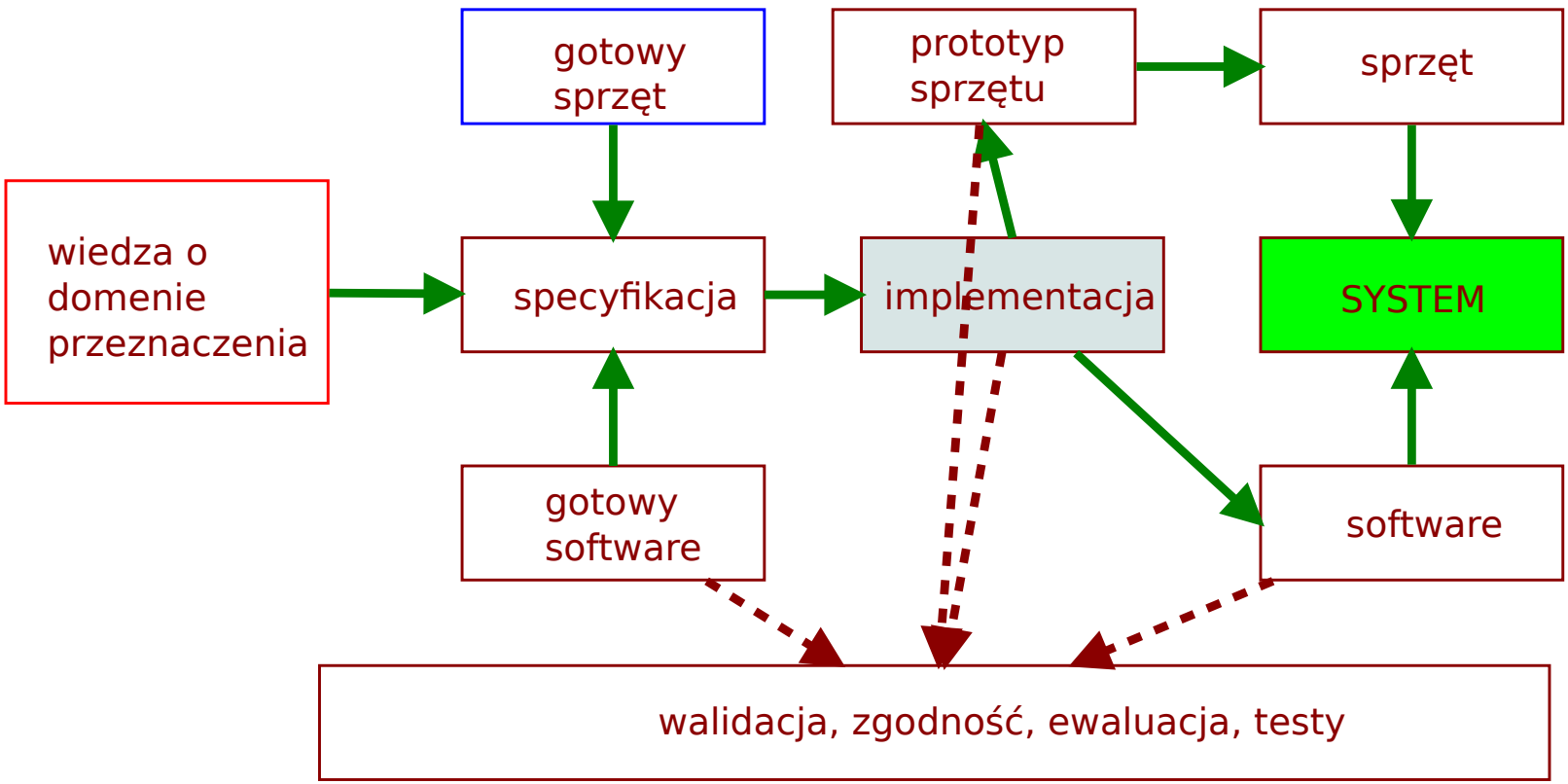
Trochę marketingu



http://www.crtc.gc.ca/eng/publications/reports/policymonitoring/2013/cmr6.htm

- koszt jednostkowy K_u;
- koszt jednorazowy wytworzenia K_e (NRE – *non-recurring engineering cost*
- koszt całkowity $K_t = K_u \times n + K_e \rightarrow$ koszt na sztukę: $K_t/n = K_u + K_e/n$.

Tworzenie systemu



Określenie wymagań

- co jest budowane

Określenie wymagań

- co jest budowane
- za ile;

Określenie wymagań

- co jest budowane
- za ile;
- jak [szybko] ma działać (*performance*);

Określenie wymagań

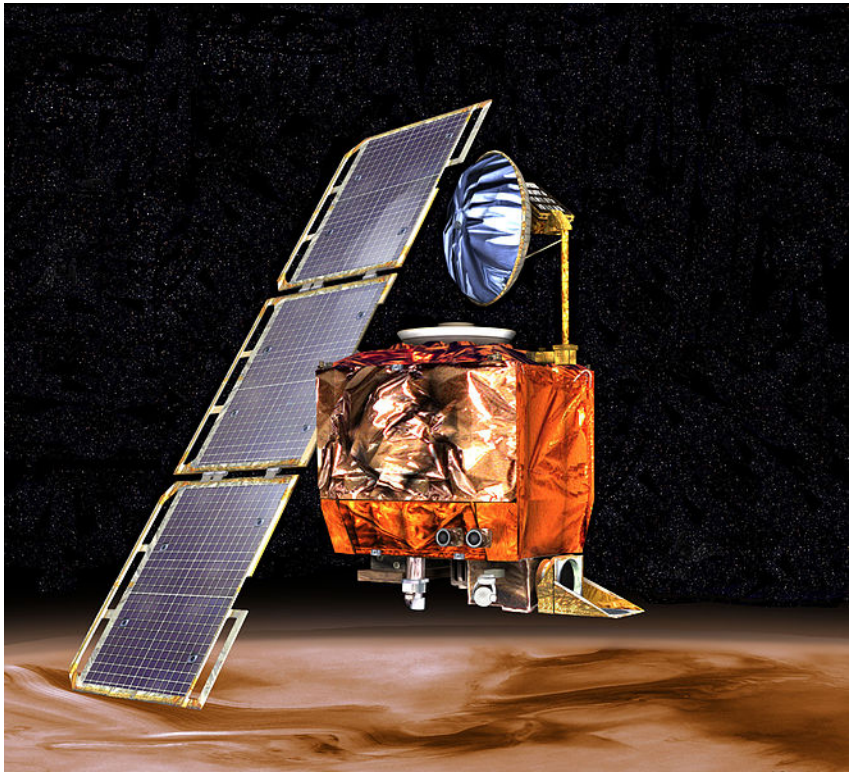
- co jest budowane
- za ile;
- jak [szybko] ma działać (*performance*);
- fizyczność (wymiary, waga, pobór/wydalanie energii);

Określenie wymagań

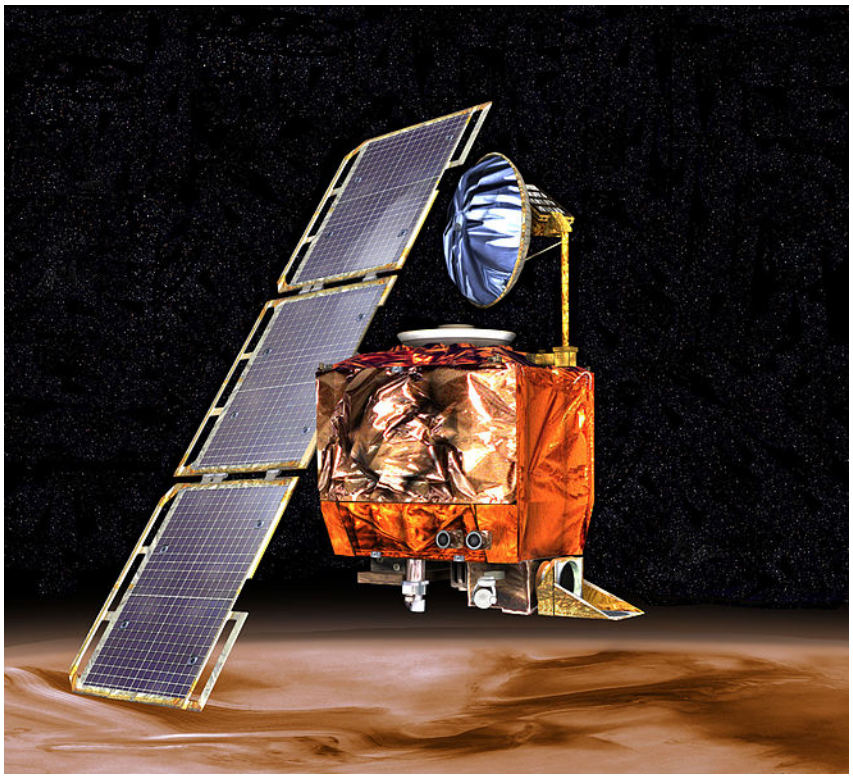
- co jest budowane
- za ile;
- jak [szybko] ma działać (*performance*);
- fizyczność (wymiary, waga, pobór/wydalanie energii);

Nazwa	żeby nazwać
Cel	żeby wiedzieć po co
Wejścia	typy danych, częstotl.,
Wyjścia	komu/jak wyśw.
Funkcjonalności	jak przetwarza We na Wy?
Wydajność/efektywność	tj. ograniczenia na czas, etc.
Koszty produkcji	ile max $?
Zasilanie/pobór prądu	...
Rozmiary fizyczne	...

Mars Climate Observer – 1999



Mars Climate Observer – 1999



"There might have been some overconfidence, inadequate robustness in our processes, designs, or operations, inadequate modeling and simulation of the operations, and failure to heed early warnings."
Zob. James Oberg,"Why the Mars probe went off course", IEEE Spectrum 36(12) December (1999): 34–39.

Określenie funkcjonalności

Specyfikacja

Nieco dokładniejszy opis niż ten w wymaganiach. Zawiera odpowiedzi na takie pytania:

Określenie funkcjonalności

Specyfikacja

Nieco dokładniejszy opis niż ten w wymaganiach. Zawiera odpowiedzi na takie pytania:

- jakie dane? (rozmiar, częstotliwość...)

Określenie funkcjonalności

Specyfikacja

Nieco dokładniejszy opis niż ten w wymaganiach. Zawiera odpowiedzi na takie pytania:

- jakie dane? (rozmiar, częstotliwość...)
- skąd? (interfejsy? standardowe schematy?)

Określenie funkcjonalności

Specyfikacja

Nieco dokładniejszy opis niż ten w wymaganiach. Zawiera odpowiedzi na takie pytania:

- jakie dane? (rozmiar, częstotliwość...)
- skąd? (interfejsy? standardowe schematy?)
- dokąd? (LCD? głośnik?)

Określenie funkcjonalności

Specyfikacja

Nieco dokładniejszy opis niż ten w wymaganiach. Zawiera odpowiedzi na takie pytania:

- jakie dane? (rozmiar, częstotliwość...)
- skąd? (interfejsy? standardowe schematy?)
- dokąd? (LCD? głośnik?)
- w jaki sposób? (w tle?, równolegle?)

Projekt

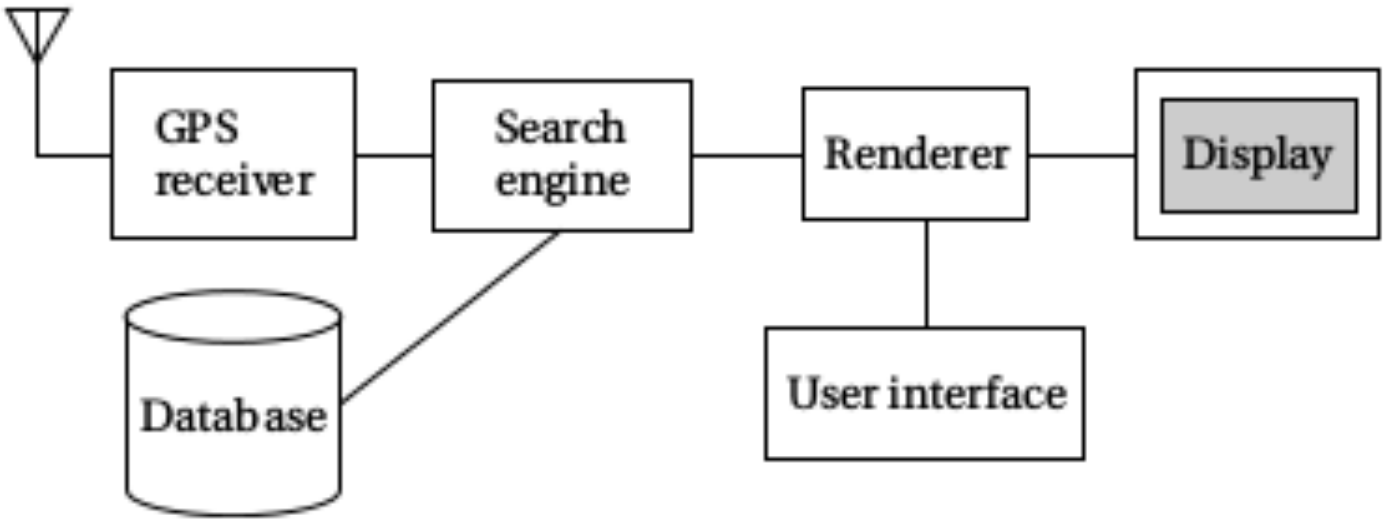
Projekt architektury

Opis na wysokim poziomie *komponentów* systemu. Diagramy blokowe opisujące (1) całość systemu, (2) komponenty sprzętowe, (3) komponenty programowe.

Projekt

Projekt architektury

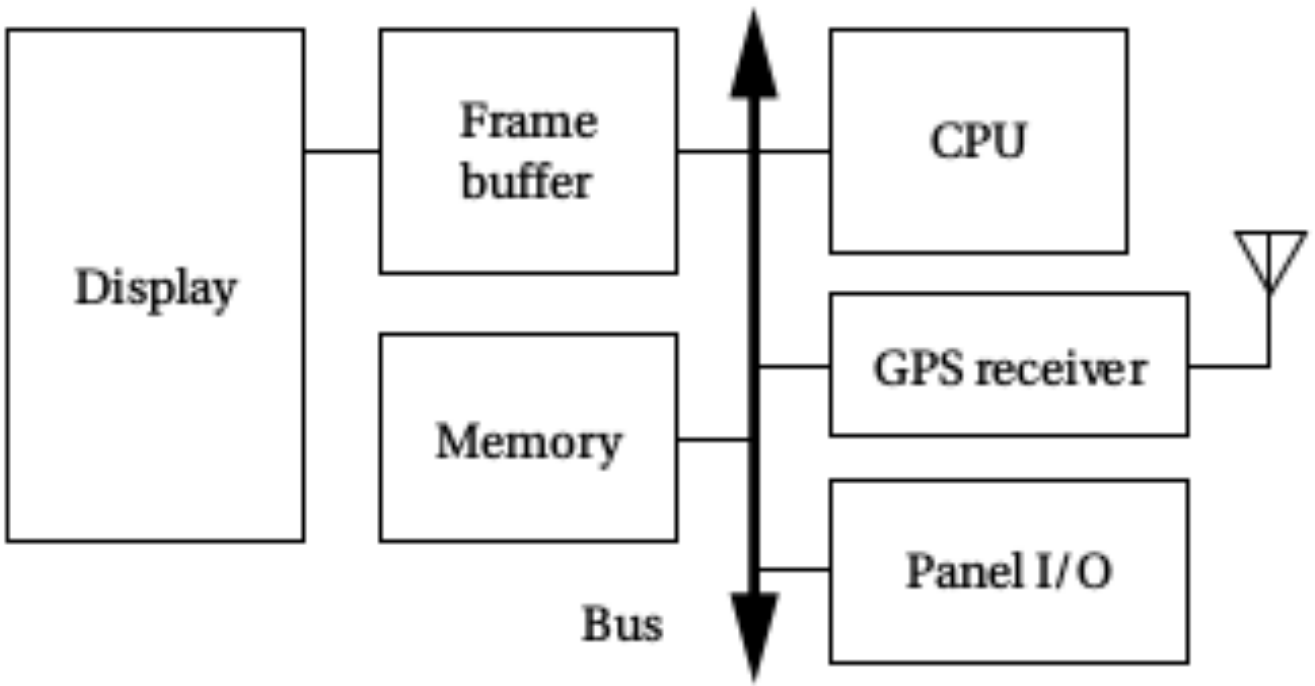
Opis na wysokim poziomie *komponentów* systemu. Diagramy blokowe opisujące (1) całość systemu, (2) komponenty sprzętowe, (3) komponenty programowe.



Projekt

Projekt architektury

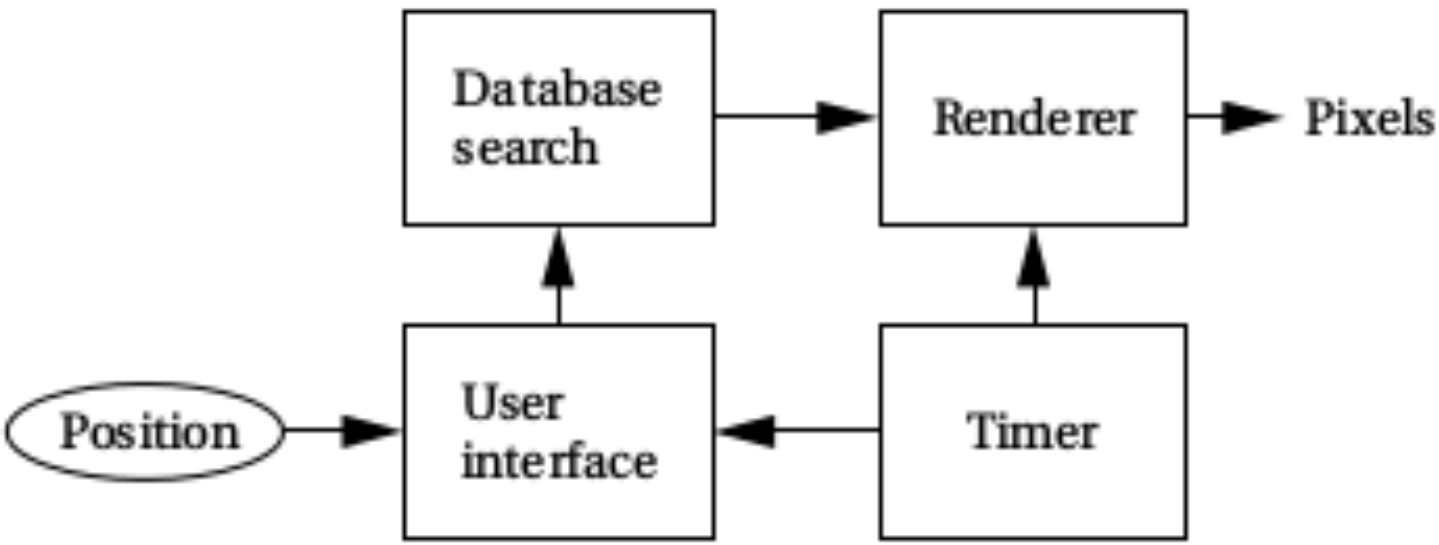
Opis na wysokim poziomie *komponentów* systemu. Diagramy blokowe opisujące (1) całość systemu, (2) komponenty sprzętowe, (3) komponenty programowe.



Projekt

Projekt architektury

Opis na wysokim poziomie *komponentów* systemu. Diagramy blokowe opisujące (1) całość systemu, (2) komponenty sprzętowe, (3) komponenty programowe.



Proces tworzenia

Wymagania

Proces tworzenia

Wymagania

Specyfikacja

Proces tworzenia

Wymagania

Specyfikacja

Architektura

Proces tworzenia

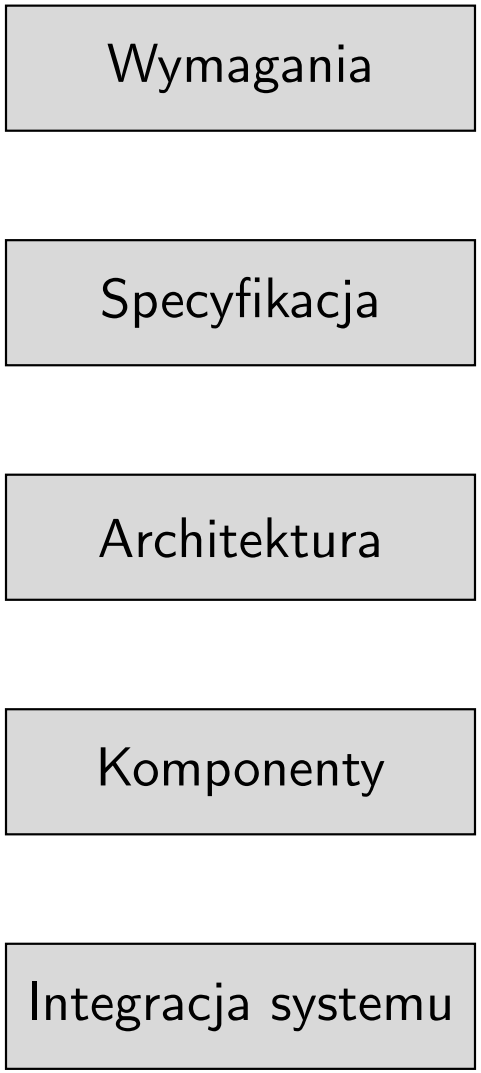
Wymagania

Specyfikacja

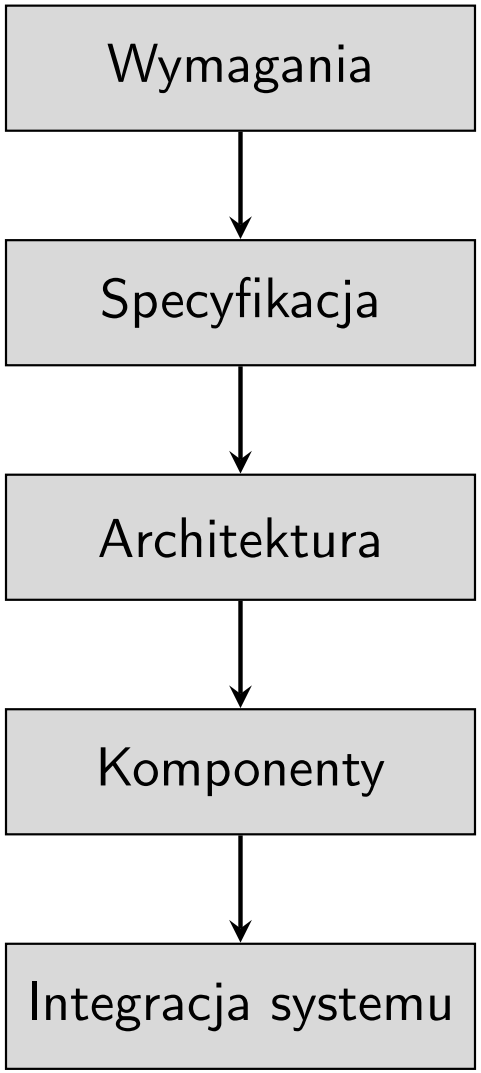
Architektura

Komponenty

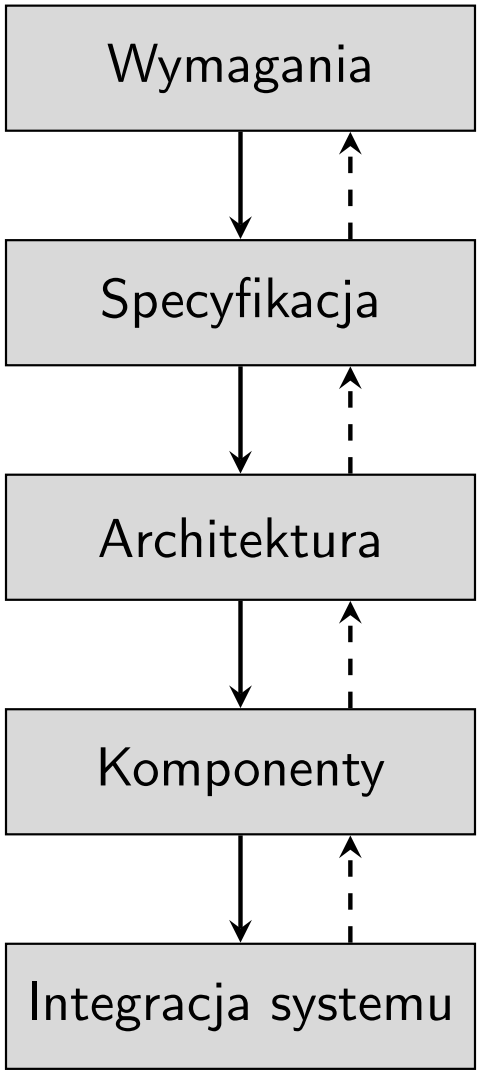
Proces tworzenia



Proces tworzenia



Proces tworzenia



Wymagania dla specyfikacji – opis

Wymagania dla specyfikacji – opis

- hierarchiczność ($\rightarrow$ czytelność): behawioralna (jak się zachowuje, jakie *procesy* zachodzą) oraz strukturalna (z czego się składa);

Wymagania dla specyfikacji – opis

- hierarchiczność ($\rightarrow$ czytelność): behawioralna (jak się zachowuje, jakie *procesy* zachodzą) oraz strukturalna (z czego się składa);
- zachowanie w czasie i/lub zachowanie pod wpływem zdarzeń (dynamika i/lub reaktywność);

Wymagania dla specyfikacji – opis

- hierarchiczność ($\rightarrow$ czytelność): behawioralna (jak się zachowuje, jakie *procesy* zachodzą) oraz strukturalna (z czego się składa);
- zachowanie w czasie i/lub zachowanie pod wpływem zdarzeń (dynamika i/lub reaktywność);
- zarządzanie zdarzeniami (wewnętrzne, zewnętrzne), wyjątki;

Wymagania dla specyfikacji – opis

- hierarchiczność ($\rightarrow$ czytelność): behawioralna (jak się zachowuje, jakie *procesy* zachodzą) oraz strukturalna (z czego się składa);
- zachowanie w czasie i/lub zachowanie pod wpływem zdarzeń (dynamika i/lub reaktywność);
- zarządzanie zdarzeniami (wewnętrzne, zewnętrzne), wyjątki;
- współbieżność czy sekwencyjność;

Wymagania dla specyfikacji – opis

- hierarchiczność ($\rightarrow$ czytelność): behawioralna (jak się zachowuje, jakie *procesy* zachodzą) oraz strukturalna (z czego się składa);
- zachowanie w czasie i/lub zachowanie pod wpływem zdarzeń (dynamika i/lub reaktywność);
- zarządzanie zdarzeniami (wewnętrzne, zewnętrzne), wyjątki;
- współbieżność czy sekwencyjność;
- synchronizacja i komunikacja między częściami składowymi;

Wymagania dla specyfikacji – praktyczność

Wymagania dla specyfikacji – praktyczność

- czytelność, przenaszalność i elastyczność;

Wymagania dla specyfikacji – praktyczność

- czytelność, przenaszalność i elastyczność;
- jednoznaczność ($\rightarrow$ powtarzalność implementacji, *dependability*);

Wymagania dla specyfikacji – praktyczność

- czytelność, przenaszalność i elastyczność;
- jednoznaczność ($\rightarrow$ powtarzalność implementacji, *dependability*);
- pozafunkcjonalne dane (waga, żywotność, kolor...);

Wymagania dla specyfikacji – praktyczność

- czytelność, przenaszalność i elastyczność;
- jednoznaczność ($\rightarrow$ powtarzalność implementacji, *dependability*);
- pozafunkcjonalne dane (waga, żywotność, kolor...);
- wiedza na temat okoliczności działania;

Wymagania dla specyfikacji – praktyczność

- czytelność, przenaszalność i elastyczność;
- jednoznaczność ($\rightarrow$ powtarzalność implementacji, *dependability*);
- pozafunkcjonalne dane (waga, żywotność, kolor...);
- wiedza na temat okoliczności działania;
- opis modelu obliczeniowego.

Modele obliczeniowe

Model obliczeniowy

Model opisujący interakcje między komponentami systemu, sposób wykonywania obliczeń przez komponenty, charakteryzujący komponenty, definiujący rodzaj przetwarzanych danych.

Modele obliczeniowe

Model obliczeniowy

Model opisujący interakcje między komponentami systemu, sposób wykonywania obliczeń przez komponenty, charakteryzujący komponenty, definiujący rodzaj przetwarzanych danych.

- CFSMs – Communicating Finite State Machines,

Modele obliczeniowe

Model obliczeniowy

Model opisujący interakcje między komponentami systemu, sposób wykonywania obliczeń przez komponenty, charakteryzujący komponenty, definiujący rodzaj przetwarzanych danych.

- CFSMs – Communicating Finite State Machines,
- opis dyskretny,

Modele obliczeniowe

Model obliczeniowy

Model opisujący interakcje między komponentami systemu, sposób wykonywania obliczeń przez komponenty, charakteryzujący komponenty, definiujący rodzaj przetwarzanych danych.

- CFSMs – Communicating Finite State Machines,
- opis dyskretny,
- opis matematyczny,

Modele obliczeniowe

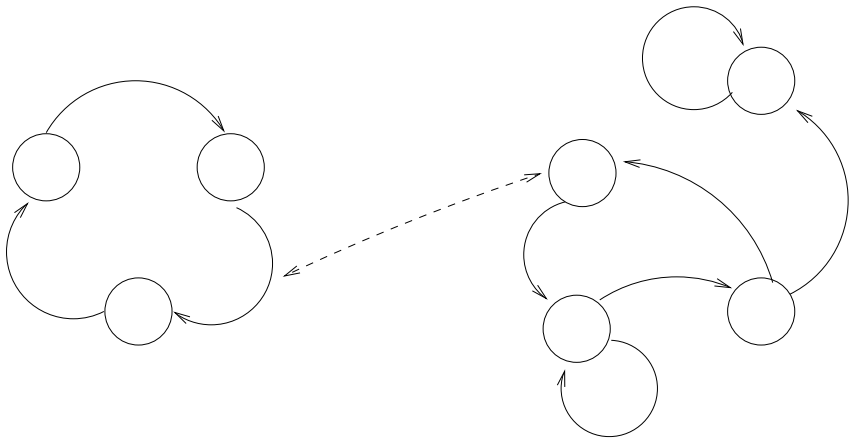
Model obliczeniowy

Model opisujący interakcje między komponentami systemu, sposób wykonywania obliczeń przez komponenty, charakteryzujący komponenty, definiujący rodzaj przetwarzanych danych.

- CFSMs – Communicating Finite State Machines,
- opis dyskretny,
- opis matematyczny,
- (a)synchroniczne przesyłanie komunikatów:

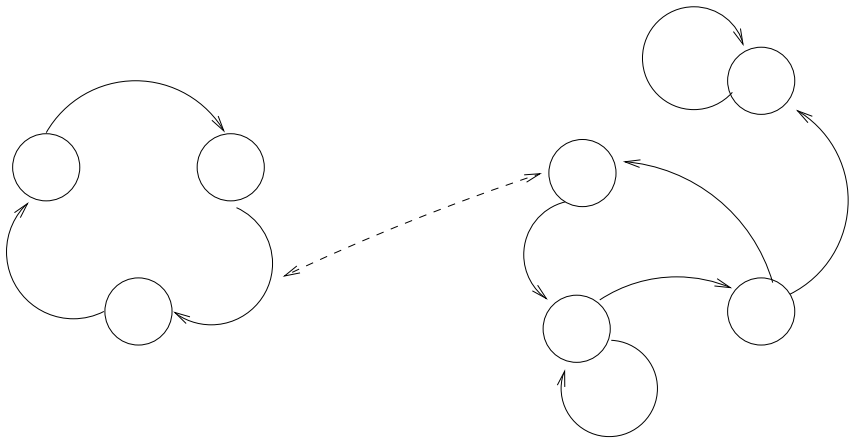
Communicating FSMs

- CFSMs – Communicating Finite State Machines



Communicating FSMs

- CFSMs – Communicating Finite State Machines



- opis dyskretny

czas	1	2	3	4
WY	a:=5	b:=3	a:=0, b:=0	a:= 5

Opis matematyczny, komunikaty

- opis matematyczny, dla systemów analogowych (przypomnienie - systemy wbudowane działają na granicach analog-cyfra, środowisko naturalne-komputer....) – równania różniczkowe, modelujące:

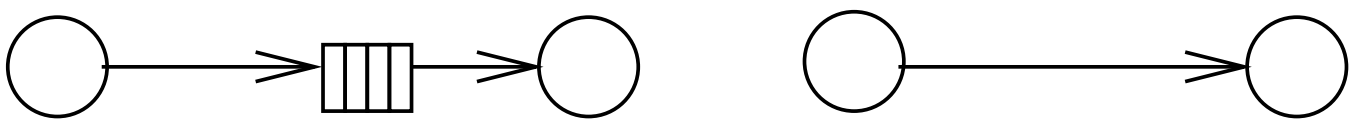
$$\frac{dp}{dT} = \frac{L}{T\Delta V} \rightarrow p(t) = p_0 e^{\frac{17.5043 \cdot x}{241.2 + t}}$$

Opis matematyczny, komunikaty

- opis matematyczny, dla systemów analogowych (przypomnienie - systemy wbudowane działają na granicach analog-cyfra, środowisko naturalne-komputer....) – równania różniczkowe, modelujące:

$$\frac{dp}{dT} = \frac{L}{T\Delta V} \rightarrow p(t) = p_0 e^{\frac{17.5043 \cdot x}{241.2+t}}$$

- (a)synchroniczne przesyłanie komunikatów:



State Charts

State Charts

- rozszerzają FSM o hierarchiczność procesów, współbieżność, złożone zmienne, niedeterminizm i rozgłaszanie informacji;

State Charts

- rozszerzają FSM o hierarchiczność procesów, współbieżność, złożone zmienne, niedeterminizm i rozgłaszanie informacji;
- jako rozszerzenie automatów Mealy i Moore'a;

State Charts

- rozszerzają FSM o hierarchiczność procesów, współbieżność, złożone zmienne, niedeterminizm i rozgłaszanie informacji;
- jako rozszerzenie automatów Mealy i Moore'a;
- pozwalają na:

State Charts

- rozszerzają FSM o hierarchiczność procesów, współbieżność, złożone zmienne, niedeterminizm i rozgłaszanie informacji;
- jako rozszerzenie automatów Mealy i Moore'a;
- pozwalają na:
  - modelowanie stanów systemu;

State Charts

- rozszerzają FSM o hierarchiczność procesów, współbieżność, złożone zmienne, niedeterminizm i rozgłaszanie informacji;
- jako rozszerzenie automatów Mealy i Moore'a;
- pozwalają na:
  - modelowanie stanów systemu;
  - modelowanie systemów reaktywnych;

State Charts

- rozszerzają FSM o hierarchiczność procesów, współbieżność, złożone zmienne, niedeterminizm i rozgłaszanie informacji;
- jako rozszerzenie automatów Mealy i Moore'a;
- pozwalają na:
  - modelowanie stanów systemu;
  - modelowanie systemów reaktywnych;
  - identyfikację zdarzeń wpływających na zmianę stanu;

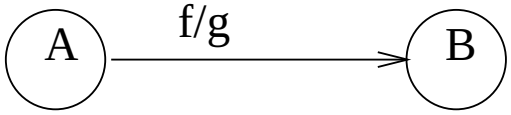
State Charts

- rozszerzają FSM o hierarchiczność procesów, współbieżność, złożone zmienne, niedeterminizm i rozgłaszanie informacji;
- jako rozszerzenie automatów Mealy i Moore'a;
- pozwalają na:
  - modelowanie stanów systemu;
  - modelowanie systemów reaktywnych;
  - identyfikację zdarzeń wpływających na zmianę stanu;
  - inżynierię twórczą i odtwórczą.

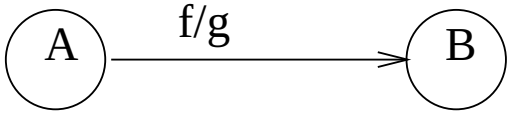
State Charts – przejścia

State Charts – przejścia

- deterministyczne:



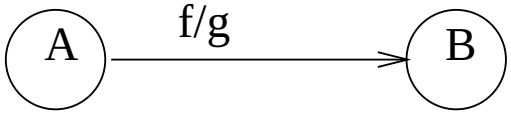
State Charts – przejścia

- deterministyczne:
  

  przejście z A do B jeśli zaistnieje f; jako efekt powstaje g;

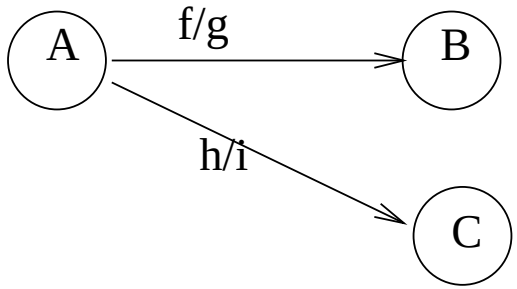
State Charts – przejścia

- deterministyczne:



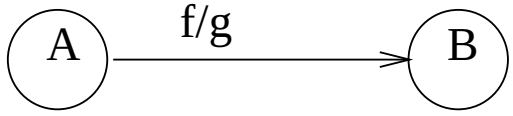
przejście z A do B jeśli zaistnieje f; jako efekt powstaje g;

- niedeterministyczne:



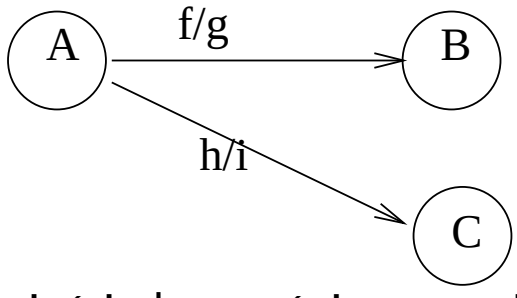
State Charts – przejścia

- deterministyczne:



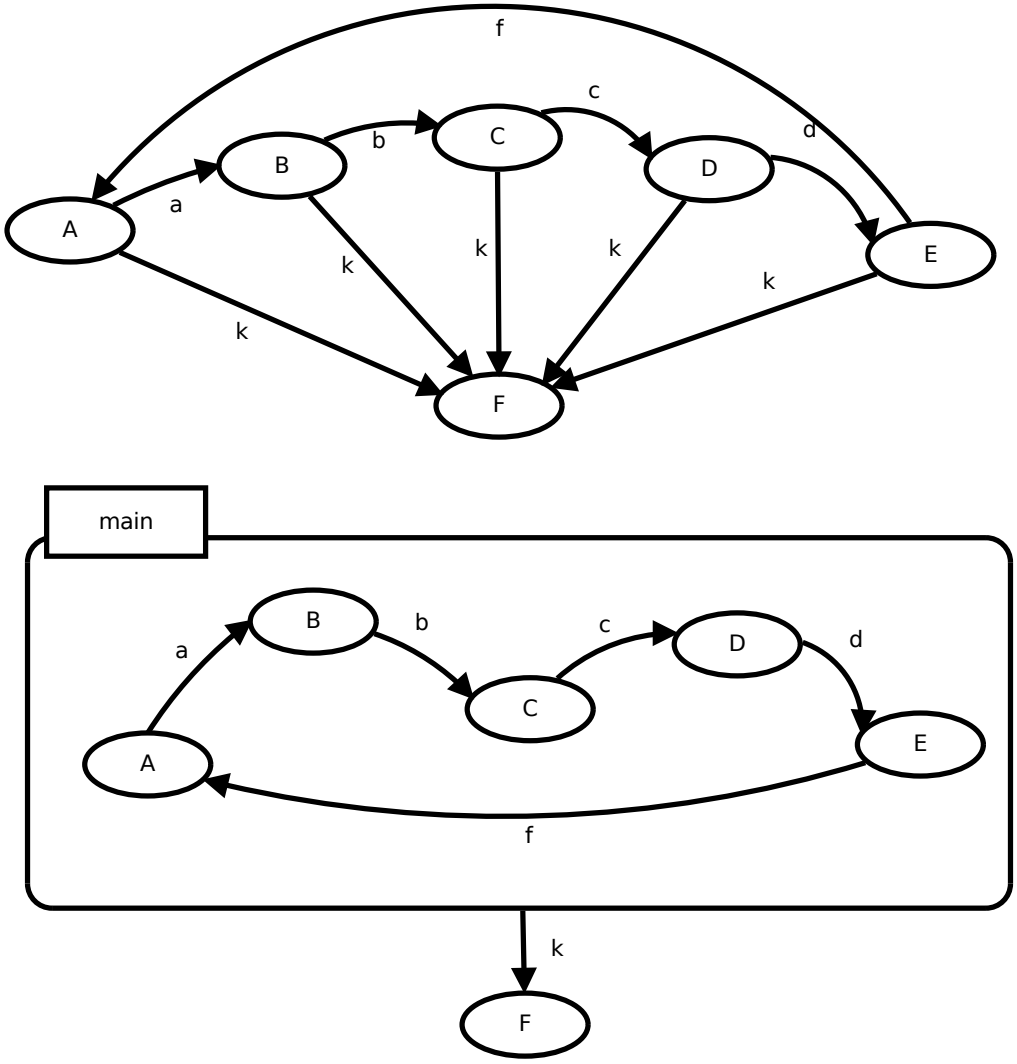
przejście z A do B jeśli zaistnieje f; jako efekt powstaje g;

- niedeterministyczne:

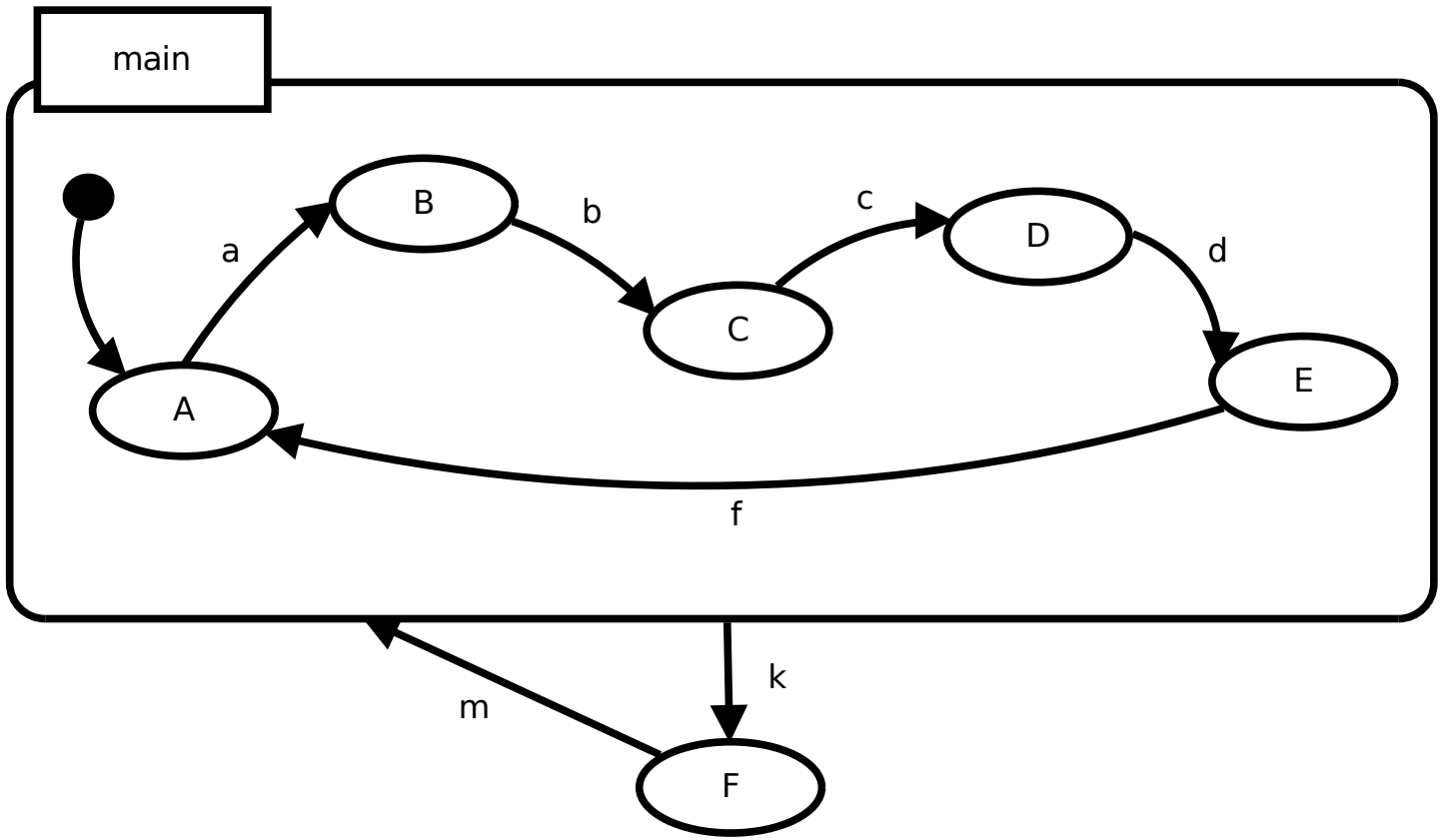


zdarzenia f i h mogą istnieć jednocześnie – przejście nieokreślone

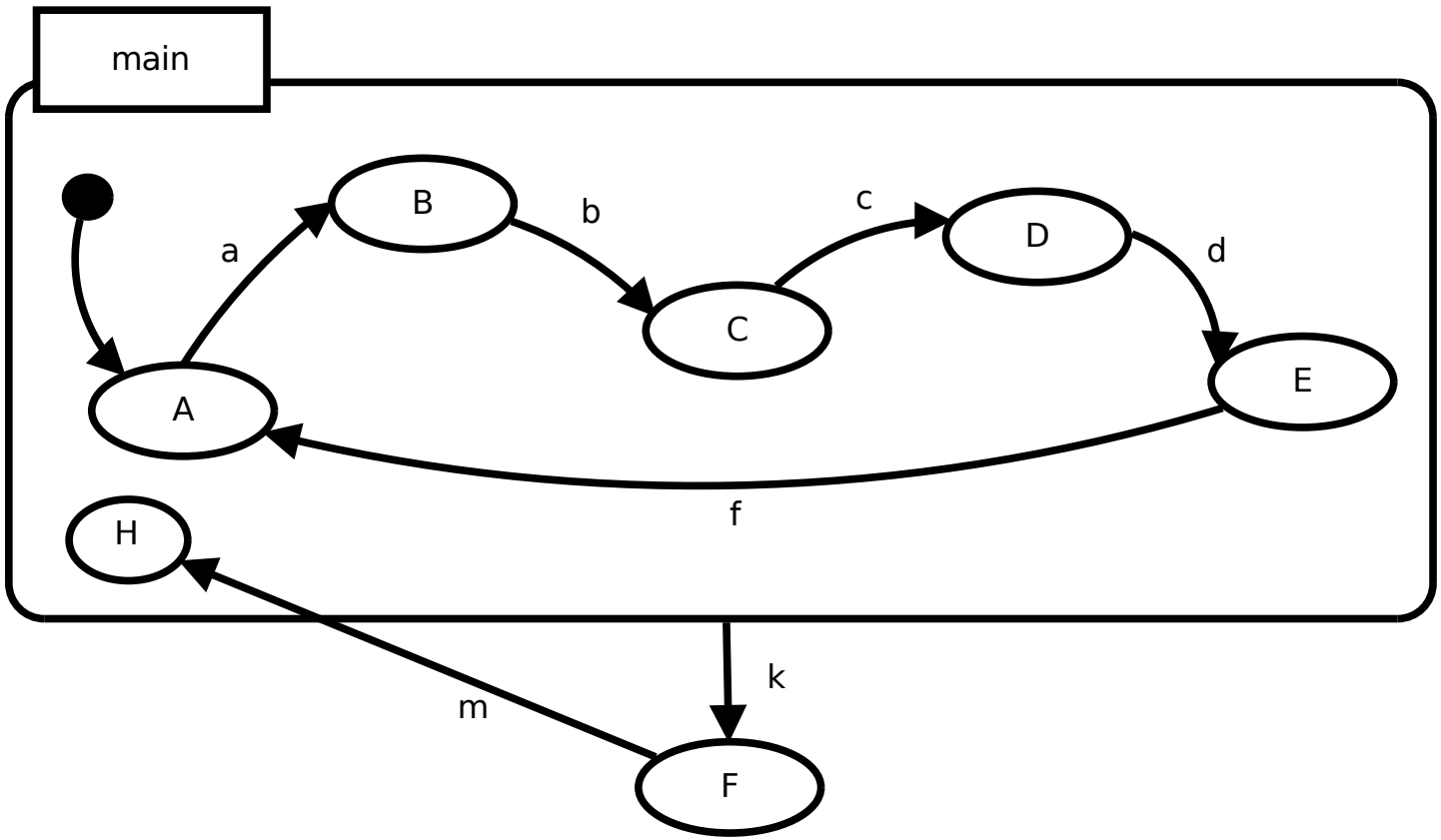
State Charts - hierarchiczność



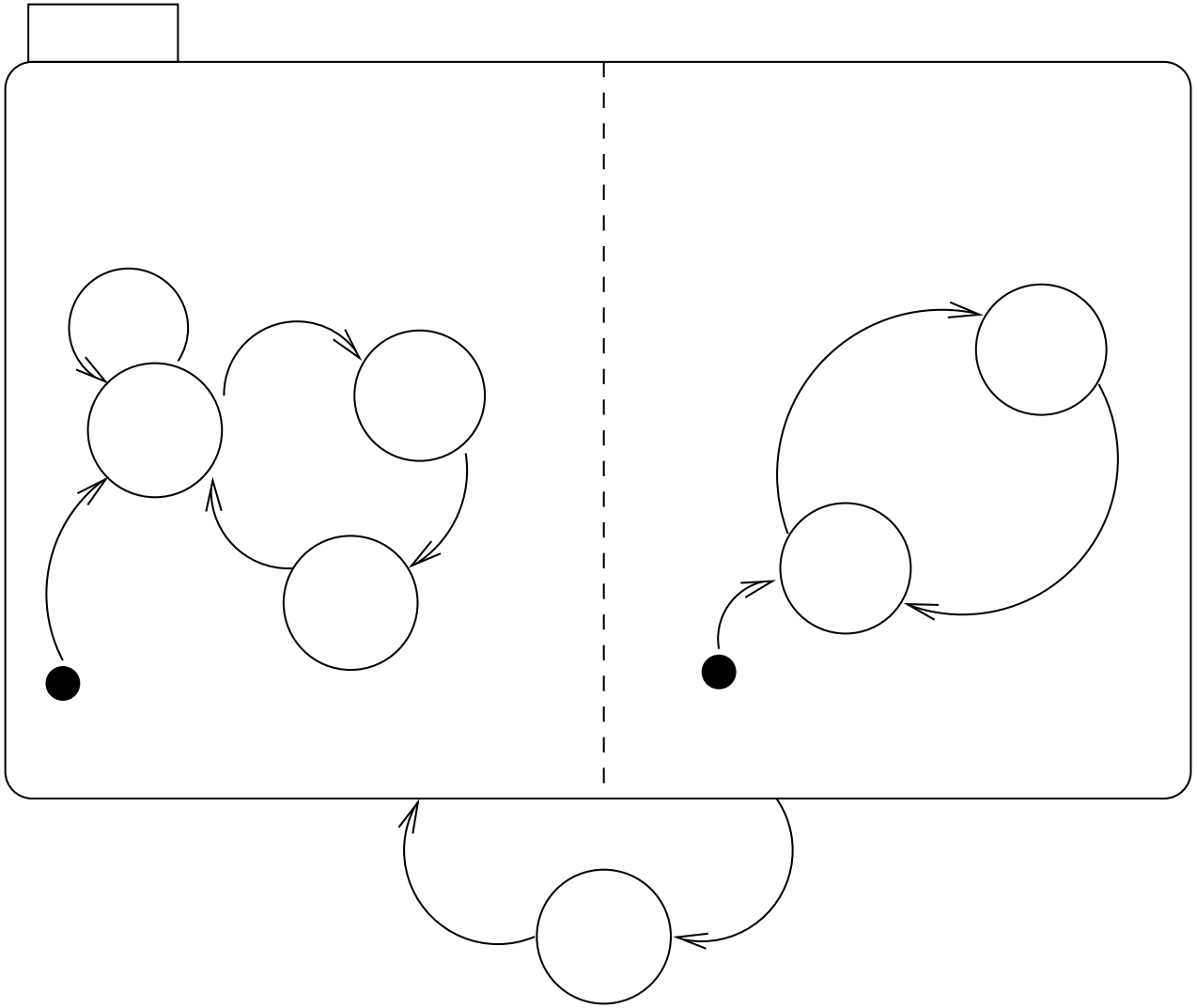
State Charts - stany domyślne i historia



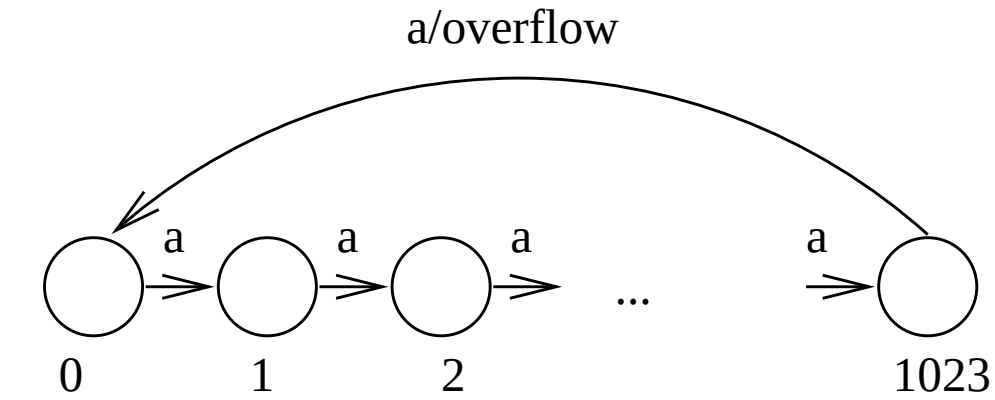
State Charts - stany domyślne i historia

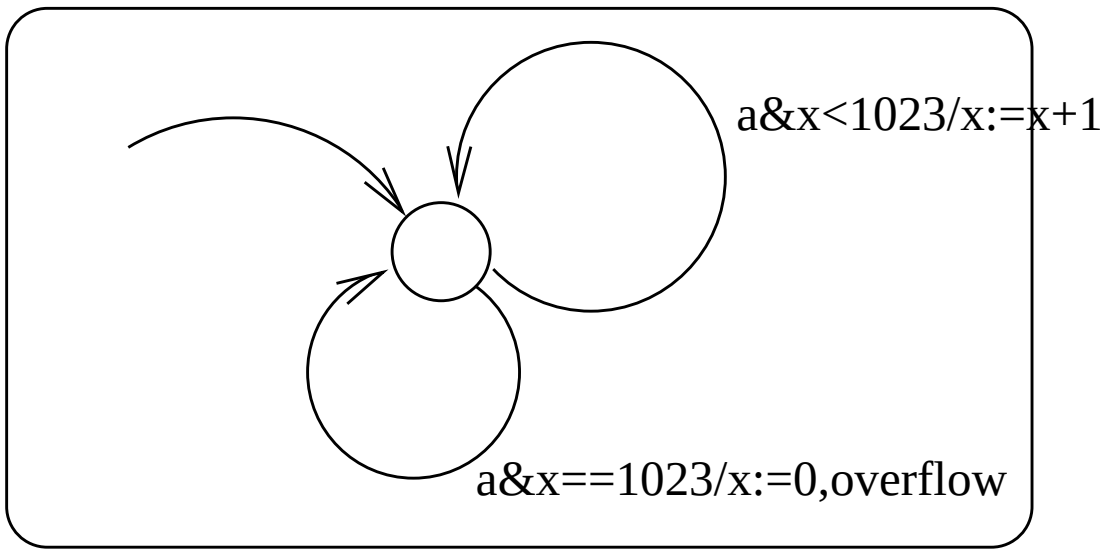


State Charts - stany AND, współbieżność

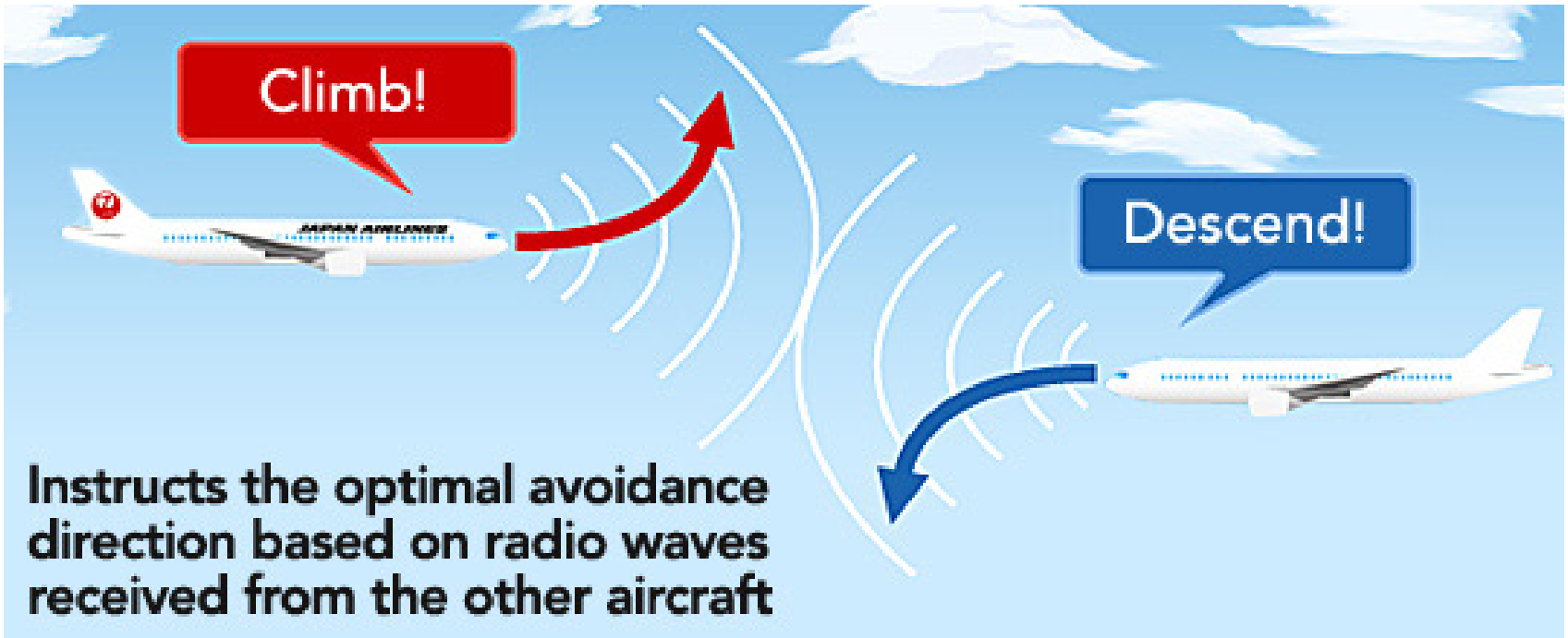


State Charts - zmienne





TCAS – Traffic and Collision Avoidance System



www.jal.com/en/flight/safety/equipment/tcas.html

TCAS – Traffic and Collision Avoidance System

TCAS

TCAS - system lotniczy o wysokim znaczeniu dla bezpieczeństwa:

TCAS

TCAS - system lotniczy o wysokim znaczeniu dla bezpieczeństwa:

- śledzi pozycję samolotu oraz innych okolicznych;

TCAS

TCAS - system lotniczy o wysokim znaczeniu dla bezpieczeństwa:

- śledzi pozycję samolotu oraz innych okolicznych;
- podejmuje decyzję o niebezpieczeństwie kolizji;

TCAS

TCAS - system lotniczy o wysokim znaczeniu dla bezpieczeństwa:

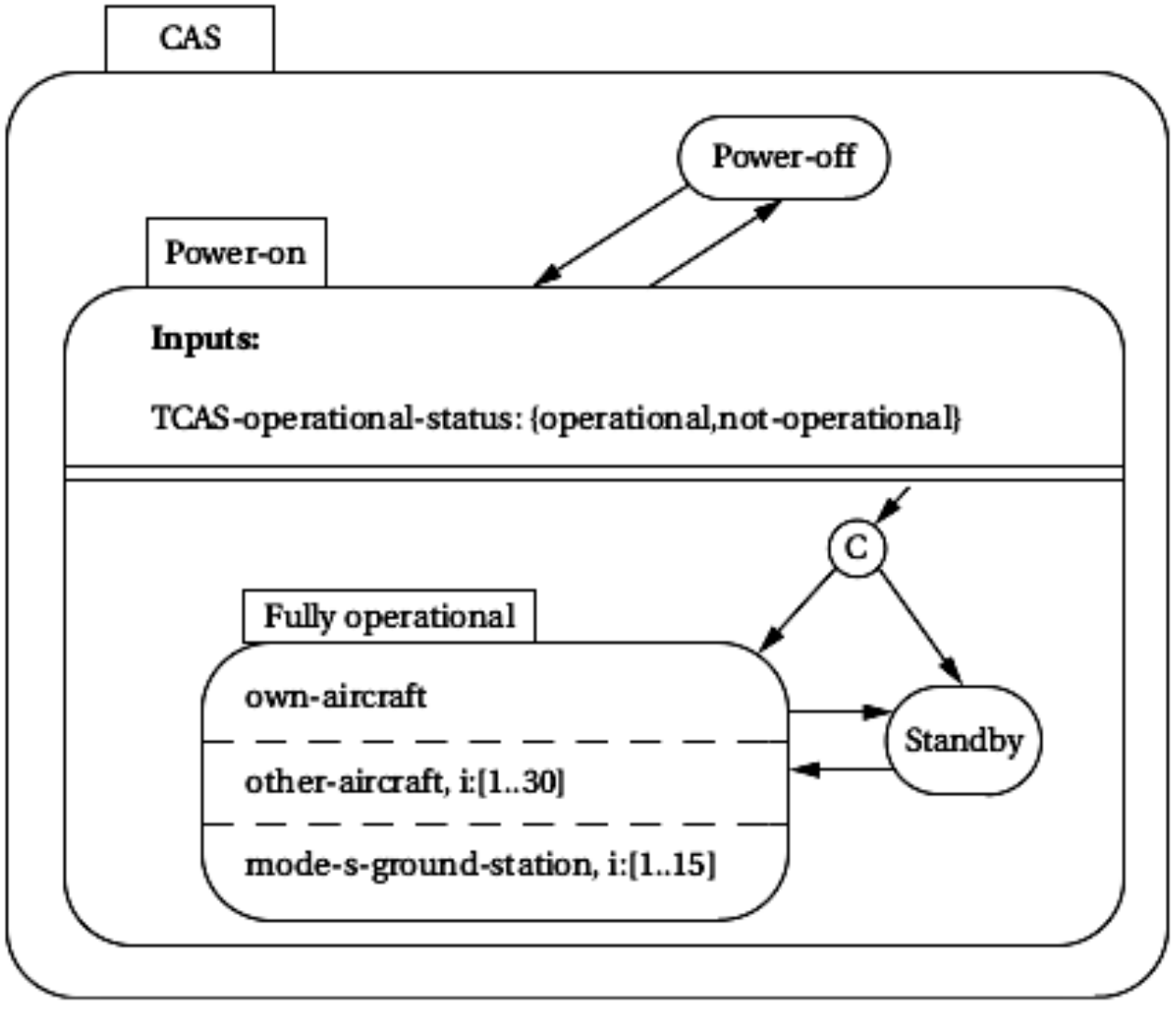
- śledzi pozycję samolotu oraz innych okolicznych;
- podejmuje decyzję o niebezpieczeństwie kolizji;
- podpowiada co zrobić w danej sytuacji (góra, dół, ...);

TCAS

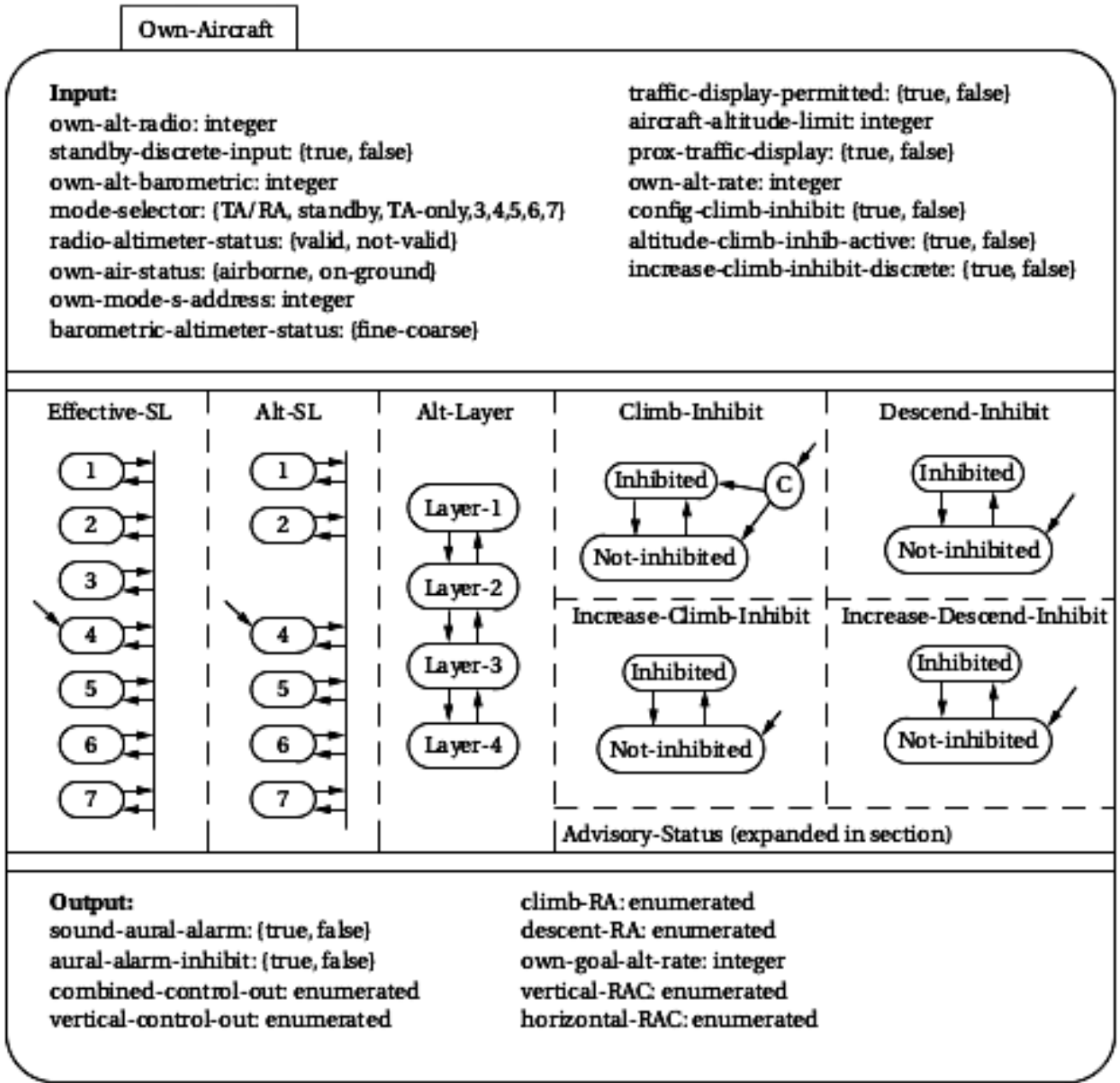
TCAS - system lotniczy o wysokim znaczeniu dla bezpieczeństwa:

- śledzi pozycję samolotu oraz innych okolicznych;
- podejmuje decyzję o niebezpieczeństwie kolizji;
- podpowiada co zrobić w danej sytuacji (góra, dół, ...);
- powinien wykrywać wszystkie przypadki, bez błędów pierwszego rodzaju;

TCAS (1)


CAS
Power-off
Power-on
Inputs:
TCAS-operational-status: {operational,not-operational}
C
Fully operational
own-aircraft
other-aircraft, i:[1..30]
mode-s-ground-station, i:[1..15]
Standby

TCAS (2)


Own-Aircraft
Input:
own-alt-radio: integer
standby-discrete-input: {true, false}
own-alt-barometric: integer
mode-selector: {TA/RA, standby, TA-only,3,4,5,6,7}
radio-altimeter-status: {valid, not-valid}
own-air-status: {airborne, on-ground}
own-mode-s-address: integer
barometric-altimeter-status: {fine-coarse}
traffic-display-permitted: {true, false}
aircraft-altitude-limit: integer
prox-traffic-display: {true, false}
own-alt-rate: integer
config-climb-inhibit: {true, false}
altitude-climb-inhib-active: {true, false}
increase-climb-inhibit-discrete: {true, false}
Effective-SL
Alt-SL
Alt-Layer
Climb-Inhibit
Descend-Inhibit
1
2
3
4
5
6
7
Layer-1
Layer-2
Layer-3
Layer-4
Inhibited
C
Not-inhibited
Increase-Climb-Inhibit
Increase-Descend-Inhibit
Advisory-Status (expanded in section)
Output:
sound-aural-alarm: {true, false}
aural-alarm-inhibit: {true, false}
combined-control-out: enumerated
vertical-control-out: enumerated
climb-RA: enumerated
descent-RA: enumerated
own-goal-alt-rate: integer
vertical-RAC: enumerated
horizontal-RAC: enumerated

Skąd wziąć obiekty?

Truizmy:

Skąd wziąć obiekty?

Truizmy:

- systemy wbudowane składają się z komponentów;

Skąd wziąć obiekty?

Truizmy:

- systemy wbudowane składają się z komponentów;
- komponenty współdziałają między sobą i użytkownikiem;

Skąd wziąć obiekty?

Truizmy:

- systemy wbudowane składają się z komponentów;
- komponenty współdziałają między sobą i użytkownikiem;
- opis na podstawie komponentów jest łatwiejszy do zrozumienia;

Skąd wziąć obiekty?

Truizmy:

- systemy wbudowane składają się z komponentów;
- komponenty współdziałają między sobą i użytkownikiem;
- opis na podstawie komponentów jest łatwiejszy do zrozumienia;
- opis *top-to-bottom* vs. *bottom-to-top*.

Model CRC

Model CRC

Classes – klasy: logiczne składy danych i funkcjonalności;

Model CRC

Classes – klasy: logiczne składy danych i funkcjonalności;

Responsibilities – odpowiedzialność: co robią klasy i po co;

Model CRC

Classes – klasy: logiczne składy danych i funkcjonalności;

Responsibilities – odpowiedzialność: co robią klasy i po co;

Collaborators – współpracownicy; inne klasy, współgrające z daną.

1. stwórz pierwotną listę klas: nazwa, co robi, czy reprezentuje fizyczny obiekt, czy teoretyczny;

1. stwórz pierwotną listę klas: nazwa, co robi, czy reprezentuje fizyczny obiekt, czy teoretyczny;
2. dopisz do każdej klasy listę odpowiedzialności i współpracowników;

1. stwórz pierwotną listę klas: nazwa, co robi, czy reprezentuje fizyczny obiekt, czy teoretyczny;
2. dopisz do każdej klasy listę odpowiedzialności i współpracowników;
3. stwórz scenariusze wykorzystania klas, rozpocznij od podstawowych, reagujących na zdarzenia zewnętrzne;

1. stwórz pierwotną listę klas: nazwa, co robi, czy reprezentuje fizyczny obiekt, czy teoretyczny;
2. dopisz do każdej klasy listę odpowiedzialności i współpracowników;
3. stwórz scenariusze wykorzystania klas, rozpocznij od podstawowych, reagujących na zdarzenia zewnętrzne;
4. użyj opisu klas na karteczkach jako *opisu roli* – spróbuj sprawdzić, czy obecny opis (atrybuty, działania) umożliwia wypełnienie wszystkich ról dla danej klasy;

1. stwórz pierwotną listę klas: nazwa, co robi, czy reprezentuje fizyczny obiekt, czy teoretyczny;
2. dopisz do każdej klasy listę odpowiedzialności i współpracowników;
3. stwórz scenariusze wykorzystania klas, rozpocznij od podstawowych, reagujących na zdarzenia zewnętrzne;
4. użyj opisu klas na karteczkach jako *opisu roli* – spróbuj sprawdzić, czy obecny opis (atrybuty, działania) umożliwia wypełnienie wszystkich ról dla danej klasy;
5. zaktualizuj opisy klas (dodaj nowe?), scenariusze, zgodnie z wykrytymi problemami;

1. stwórz pierwotną listę klas: nazwa, co robi, czy reprezentuje fizyczny obiekt, czy teoretyczny;
2. dopisz do każdej klasy listę odpowiedzialności i współpracowników;
3. stwórz scenariusze wykorzystania klas, rozpocznij od podstawowych, reagujących na zdarzenia zewnętrzne;
4. użyj opisu klas na karteczkach jako *opisu roli* – spróbuj sprawdzić, czy obecny opis (atrybuty, działania) umożliwia wypełnienie wszystkich ról dla danej klasy;
5. zaktualizuj opisy klas (dodaj nowe?), scenariusze, zgodnie z wykrytymi problemami;
6. zaktualizuj relacje między klasami;

1. stwórz pierwotną listę klas: nazwa, co robi, czy reprezentuje fizyczny obiekt, czy teoretyczny;
2. dopisz do każdej klasy listę odpowiedzialności i współpracowników;
3. stwórz scenariusze wykorzystania klas, rozpocznij od podstawowych, reagujących na zdarzenia zewnętrzne;
4. użyj opisu klas na karteczkach jako *opisu roli* – spróbuj sprawdzić, czy obecny opis (atrybuty, działania) umożliwia wypełnienie wszystkich ról dla danej klasy;
5. zaktualizuj opisy klas (dodaj nowe?), scenariusze, zgodnie z wykrytymi problemami;
6. zaktualizuj relacje między klasami;
7. kontynuuj 4-6 aż nie będzie dziur.

Therac-25 – zła historia

Podstawowe założenia zadań krytycznych:

Therac-25 – zła historia

Podstawowe założenia zadań krytycznych:

- 8-fazowy proces dostarczania leczniczych dawek radiacyjnych;

Therac-25 – zła historia

Podstawowe założenia zadań krytycznych:

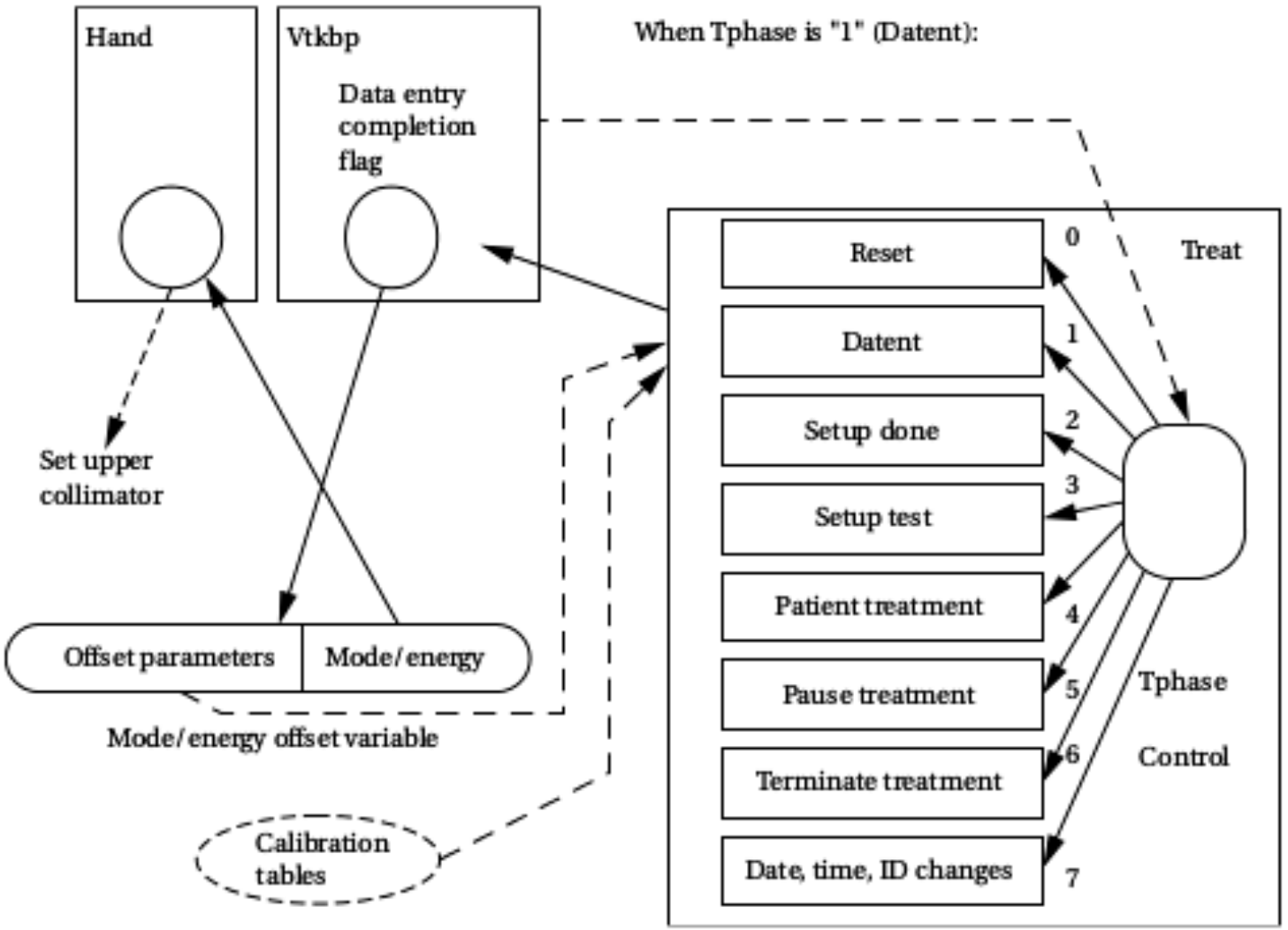
- 8-fazowy proces dostarczania leczniczych dawek radiacyjnych;
- kontroluje serwomechanizm sterujący działkiem promieniotwórczym;

Therac-25 – zła historia

Podstawowe założenia zadań krytycznych:

- 8-fazowy proces dostarczania leczniczych dawek radiacyjnych;
- kontroluje serwomechanizm sterujący działkiem promieniotwórczym;
- nadzorca kontroluje stan systemu i sprawdza ograniczniki.

Therac-25 – zła historia



Nancy G. Leveson and Clark S. Turner "*An investigation of the Therac-25 accidents*" IEEE Computer July (1993): 18–41.

Therac-25 – zła historia

- ograniczona analiza bezpieczeństwa;
- niskie p-stwo przypisane zdarzeniom w sposób arbitralny;
- nadmiernie złożony kod (asembler dla PDP-11) wykorzystujący niepewne schematy kodowania;
- *jeden* twórca kodu na przestrzeni kilku lat;
- *the most serious computer-related accidents to date (at least nonmilitary and admitted)*.

Przypadki użycia

Nazwa PU:	**Numer PU:**	**Priorytet:**
Aktor podstawowy:	**Typ:**	
Udziałowcy i cele:		
Krótki opis:		
Wyzwalacz:	**Typ:**	
Powiązania: **Asocjacja:** **Zawieranie:** **Rozszerzanie:** **Generalizacja:**		
Zwykły przepływ zdarzeń: 1.		
Przepływy poboczne:		
Przepływy alternatywne/wyjątkowe:		

Rzeczy do zapamiętania

- gdzie w świecie rozlokowały się SW;
- wymagania vs. specyfikacja.
- *State Charts*, opisy przypadków użycia
- (do odświeżenia) - diagramy UML

Rzeczy do zapamiętania

- gdzie w świecie rozlokowały się SW;
- wymagania vs. specyfikacja.
- *State Charts*, opisy przypadków użycia
- (do odświeżenia) - diagramy UML

Do przeczytania

- Nancy G. Leveson and Clark S. Turner "*An investigation of the Therac-25 accidents*" IEEE Computer July (1993): 18–41.
- Thomas M. Whitney, France Rode, Chung C. Tung, "*The 'powerful pocketful': an electronic calculator challenges the slide rule*" Hewlett-Packard Journal (1972): 2–9.
- James Oberg,"Why the Mars probe went off course", IEEE Spectrum 36(12) December (1999): 34–39.
- Nancy G. Leveson, Mats Per Erik Heimdahl, Holly Hildreth, and Jon Damon Reese, "Requirements specification for process-control systems" IEEE Transactions on Software Engineering 20(9) (1994): 684–707.
- J.Kinner, "F-35 Mission Systems Software, Case Study"