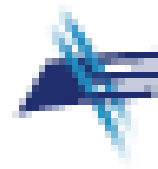




POLITECHNIKA WARSZAWSKA



Serwer obliczeniowy IDOS dla zadań sterowania optymalnego

Wiele osób z Wydziału Mechatroniki (Tarnawski, Fajdek Stachura), Elektroniki i Technik Informacyjnych (Malinowski, Karbowski) PW, IBS PAN (Myśliński), IPPT PAN (Bajer), NASK (Błaszczuk)

oraz Radosław Pytlak

r.pytlak@mchtr.pw.edu.pl

Plan wystąpienia

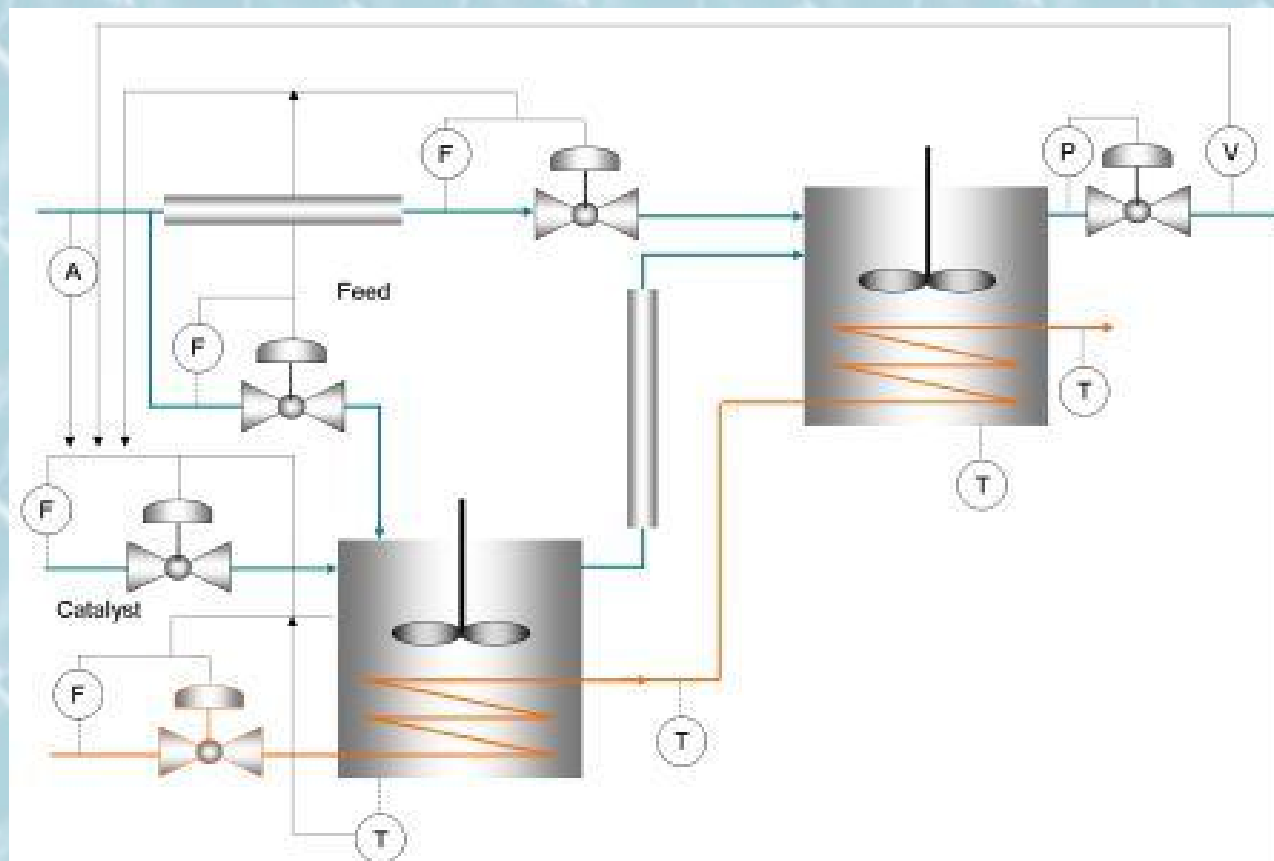
- Dlaczego serwer dla zadań sterowania optymalnego?
 - zastosowania sterowania optymalnego (SO), trudności w stosowaniu SO
- Krótki przegląd metod SO
 - metody pośrednie oparte na różnych wersjach Zasady Maksimum
 - metody bezpośrednie, dyskretyzacja a’priori równań różniczkowych

Plan wystąpienia

- Zasady działania serwera IDOS (**Interactive Dynamic Optimization Server**)
 - IDOS jako odpowiednik serwera NEOS (serwera dedykowanego dla zadań optymalizacji statycznej)
- DOML – Dynamic Optimization Modeling Language
 - język modelowania zadań SO będący rozwinięciem języka Modelica
 - jeden język wiele solverów do niego podpiętych
 - solwery dostępne poprzez serwer
- Grant NR02-0009-06

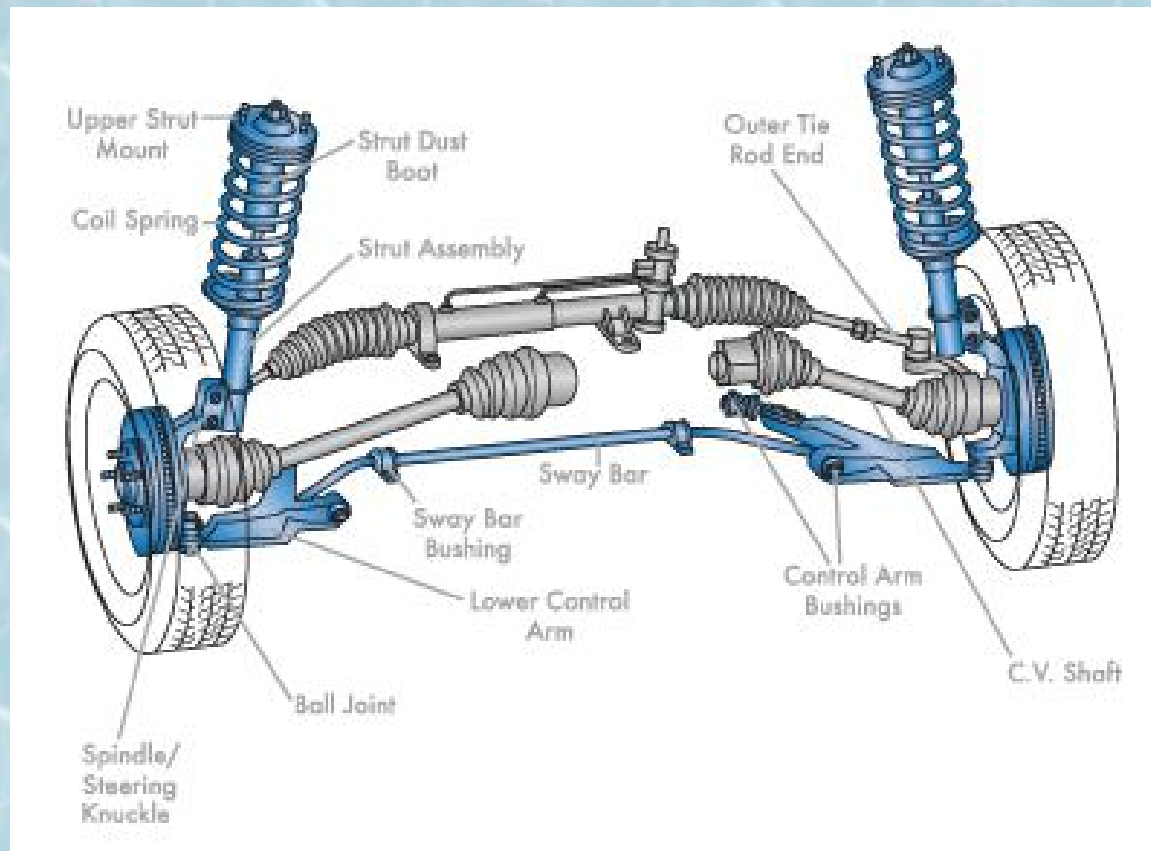
Zastosowania sterowania optymalnego

- inżynieria procesowa



Zastosowania sterowania optymalnego

- projektowanie układów mechanicznych



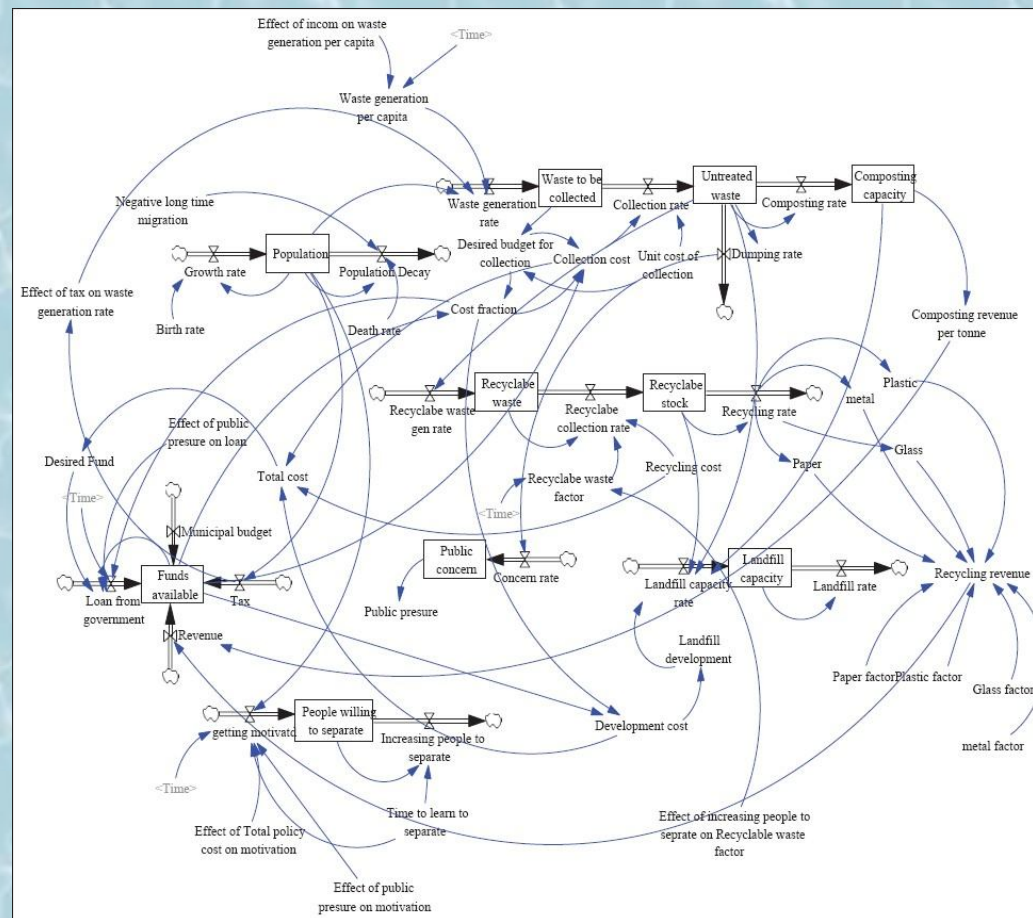
Zastosowania sterowania optymalnego

- optymalne sterowanie statkami kosmicznymi



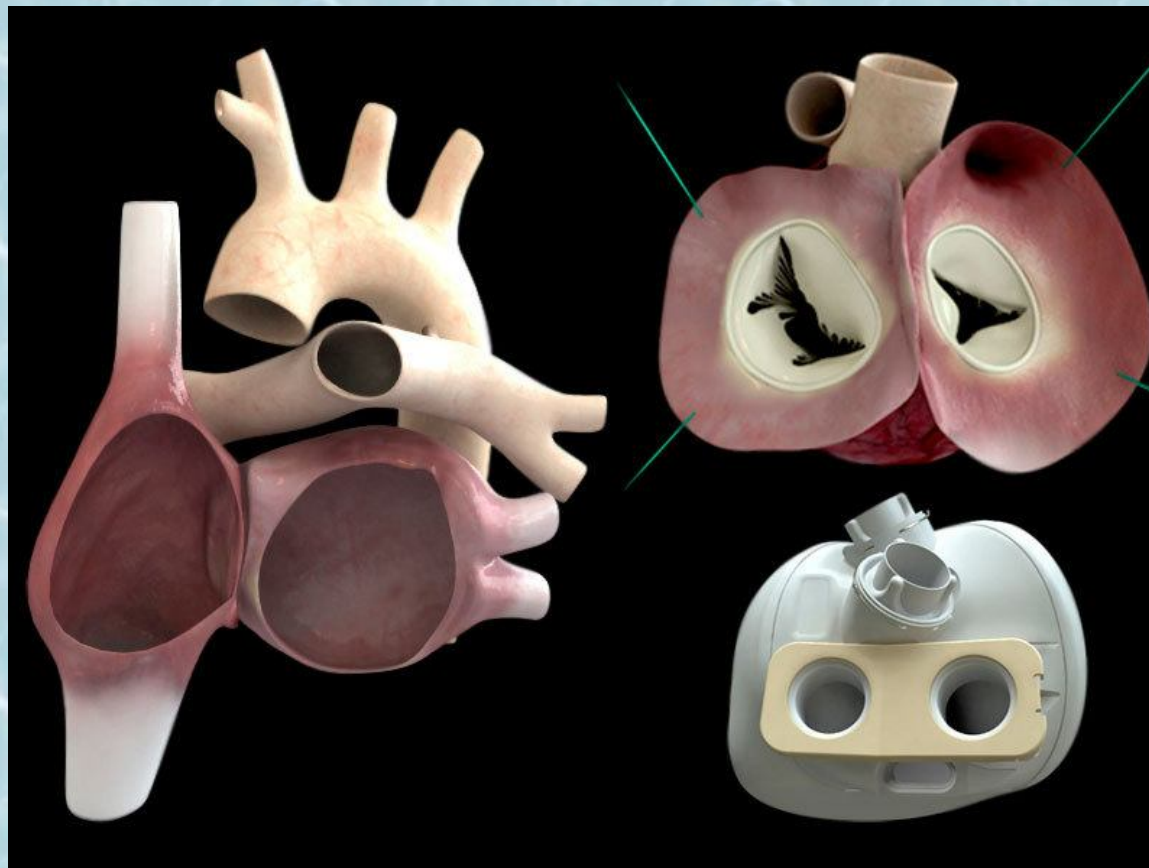
Zastosowania sterowania optymalnego

- wybór optymalnych reguł decyzyjnych (Systems Dynamics)



Zastosowania sterowania optymalnego

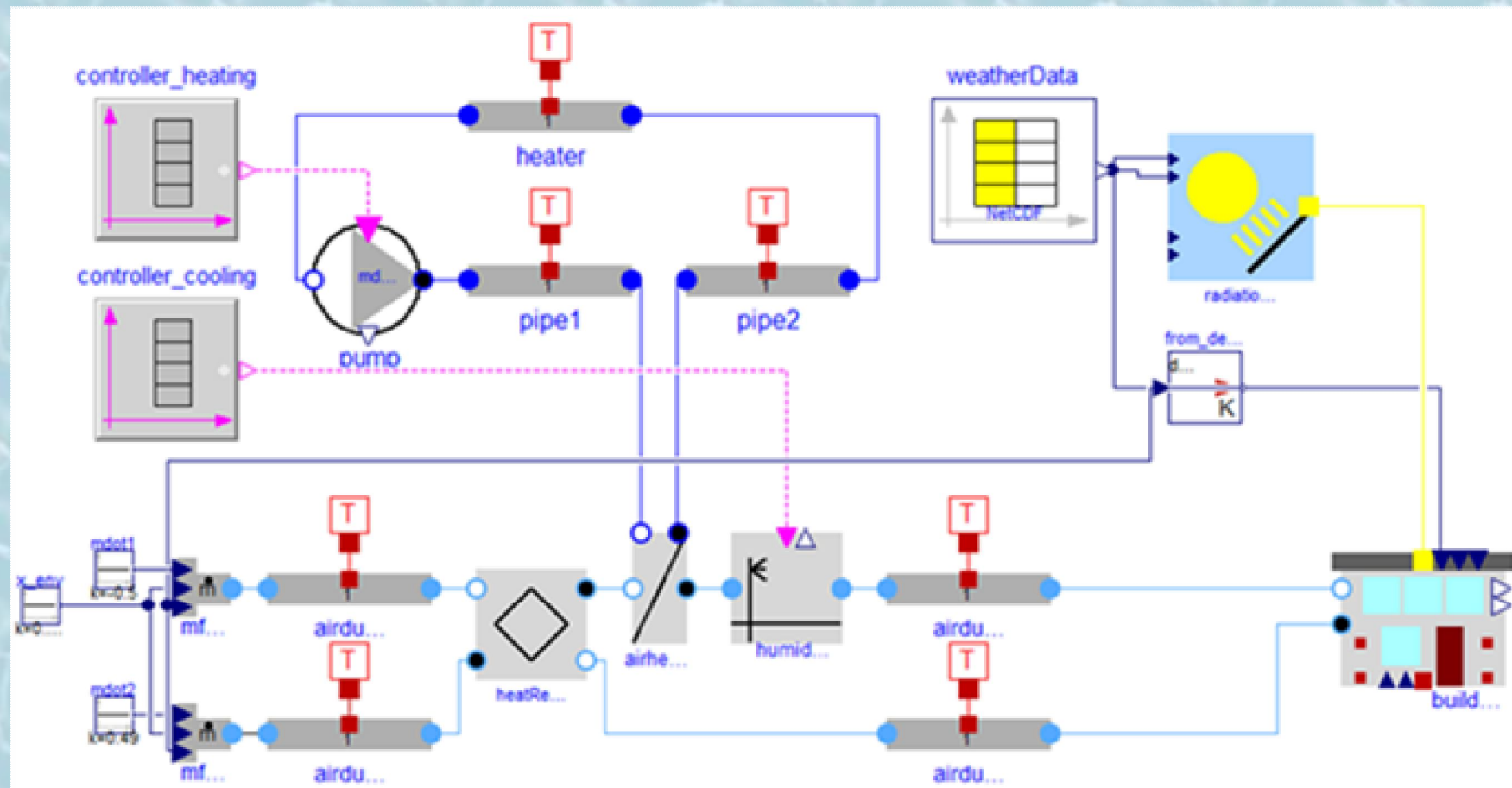
- nauki medyczne – optymalne terapie



Zastosowania sterowania optymalnego

- sterowanie samolotami bezzałogowymi w szyku
- określanie optymalnych strategii inwestycyjnych
-
- i inne

Modele dynamiczne mogą być bardzo złożone



tysiące równań różniczkowo-algebraicznych

Sformułowanie formalne

$$\min_u \left[\int_0^{t_f} f^0(x(t), u(t), t) dt + \Phi(x(t_f)) = \hat{F}^0(u) \right]$$

$$F(\dot{x}(t), x(t), u(t), t) = 0, t \in T$$

$$q(x(t), t) \leq 0, t \in T$$

$$h_i^1(x(t_f)) = 0, i \in E$$

$$h_j^2(x(t_f)) \leq 0, j \in I$$

$$u \in \mathcal{U}$$

Główne przeszkody szerokiego stosowania SO

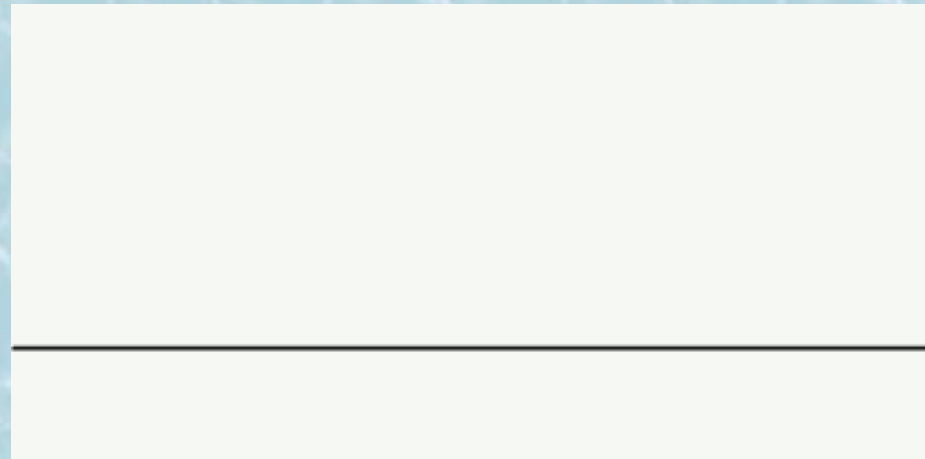
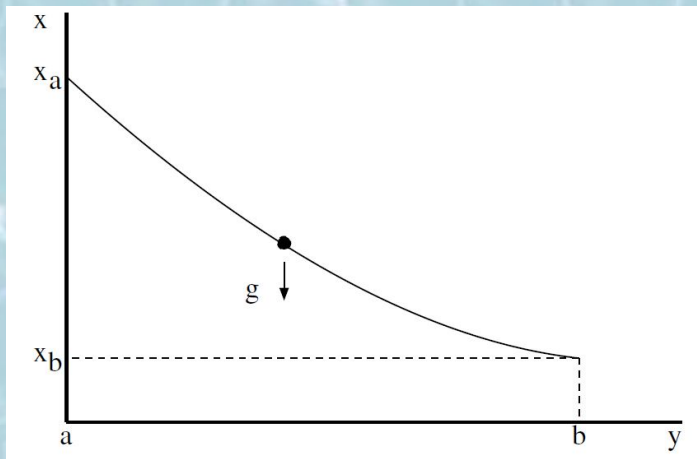
- zadania SO wymagają modeli, w ostatnich latach coraz szerzej stosowane: przemysł samochodowy, chemiczny, energetyka, lotnictwo, budownictwo, procesy biznesowe;
- procedury numeryczne dla zadań SO bardzo rozbudowane:
 - **procedury numerycznego całkowania**
 - **procedury optymalizacji**
 - **procedury algebry liniowej dla zadań wielkiej skali,**
 - **procedury automatycznego różniczkowania;**

Główne przeszkody szerokiego stosowania SO

- sterowanie optymalne to ‘trudna’ część matematyki;
- brak języka modelowania dla zadań SO zdolnego do obsługi większości zasadniczo różnych solwerów.

Krótką historia teorii SO

- Zadanie brachistochrony postawione przez **Galileo Galilei** w 1636, nieprawidłowo rozwiązane
 - Postawione повторно i rozwiązane przez **Johanna Bernoulliego** w 1696, również przez **Jacoba Bernoulliego**, **Leibnitza**, **l'Hopitala** oraz **Newtona** (pod zmienionym nazwiskiem);



Krótką historia teorii SO

- **L. Euler**, student Bernoulliego, opublikował w 1744 *Method of finding plane curves that show property of maximum or minimum*
- **J.L. Lagrange** uprościł w 1755 warunki Eulera – powstały warunki konieczne rachunku wariacyjnego Eulera-Lagrange'a.
- Równanie **Hamiltona** (1834) – **Jacobiego** (1842), znalezienie rozwiązania r. wariacyjnego poprzez rozwiązanie równania cząstkowego
- **R. Bellman** (1957), **L. Pontryagin** (1956) rozszerzenie wyników na zadania SO.

Warunki optymalności

$$\min_{(x,u)} \left[F_0(x, u) = \int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt \right]$$

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t)), \quad t \in [t_0, t_f], \\ x(t_0) &= x_0, \quad x(t_f) = x_1. \end{aligned}$$

$$H(t, x, u, p) = p^T f(t, x, u) + f^0(t, x, u).$$

$$\begin{aligned} \dot{x}(t) &= H_p(t, x(t), u(t), p(t))^T \\ \dot{p}(t) &= -H_x(t, x(t), u(t), p(t))^T \end{aligned}$$

$$H(t, x(t), u(t), p(t)) = \max_{\tilde{u} \in \Omega} H(t, x(t), \tilde{u}, p(t))$$

Metody rozwiązywania zadań SO - pokrótce

- Metody pośrednie – oparte na warunkach koniecznych optymalności, trudne w stosowaniu, gdy występują ograniczenia na stan: konieczność podania funkcji przełączeń dla sterowań, przedziałów dla aktywnych ograniczeń, reguły maksymalizacji hamiltonianu.
- Metoda strzału, lokalna, ale dająca bardzo dokładny wynik

Metody rozwiązywania zadań SO - pokrótce

- Metody bezpośrednie – wymagają dyskretyzacji *a priori* równań różniczkowych

$$\min_{(x,u)} \left[\int_{t_0}^{t_f} f^0(t, x(t), u(t)) dt + \Phi(x(t_f)) \right]$$

$$\dot{x}(t) = f(t, x(t), u(t)), \quad t \in [t_0, t_f], \quad x(t_0) = x_0.$$

$$\min_{\{(x_k, u_k)\}_{k=0}^N} \left[\sum_{k=0}^{N-1} f_k^0(x_k, u_k) + \Phi(x_N) \right]$$

$$x_1 - f_0(x_0, u_0) = 0$$

$$x_2 - f_1(x_1, u_1) = 0$$

$$\dots \quad \dots \quad \dots$$

$$x_N - f_{N-1}(x_{N-1}, u_{N-1}) = 0$$

Metody rozwiązywania zadań SO - pokrótce

- Metody bezpośrednie – wymagają dyskretyzacji *a priori* równań różniczkowych
- Bardzo duża liczba zmiennych decyzyjnych oraz graniczeń w wynikowym zadaniu optymalizacji statycznej
- Można stosować bez głębszej wiedzy z teorii SO (podejście inżynierskie)
- Niezbyt wysoka dokładność rozwiązania (równania różniczkowe dyskretyzowane *a priori*)

Interactive Dynamic Optimization Server

- Pielęgnacja głównych metod dla zadań SO w jednym miejscu
- Dostęp do metod SO poprzez język modelowania dla zadań sterowania optymalnego – **DOML**
(Dynamic Optimization Modeling Language)
- IDOS opracowany na wzór serwera **NEOS**
(Networked Enabled Optimization System)
opracowany w latach dziewięćdziesiątych w Argonne National Laboratory pod Chicago

Interactive Dynamic Optimization Server

- Serwer NEOS jest dla zadań optymalizacji statycznej (obecnie 2 mln zgłoszeń rocznie)
- NEOS bazuje na językach dla zadań optymalizacji statycznej (zadań NLP): AMPL, GAMS, które istniały przed jego powstaniem
- IDOS, za pośrednictwem języka DOML umożliwia stosowanie sekwencji solverów dla danego zadania

IDOS – dlaczego na PW?

- PW (Wydział Elektroniki – Instytut Automatyki) to przez wiele lat wiodący na świecie ośrodek w dziedzinie optymalizacji (w tym dynamicznej) – prof. Findeisen, prof. Malinowski, prof. Wierzbicki
- R. Pytlak przez szereg lat uczestniczył w pracach nad optymalizacją dla języka gPROMS ([Pytlak, *Numerical methods for optimal control problems with state constraints*, Lecture Notes in Mathematics 1707, Springer-Verlag, 1999])
- **gPROMS** (IC) oraz **Modelica** (Lund) to wiodące języki symulacyjne dla tworzenia bardzo dużych modeli (równania DAEs)
- **DOML** stanowi rozwinięcie języka Modelica na optymalizację dynamiczną



Interactive Dynamic Optimization Server

System equations	Method type	Solvers
ODEs	<i>a priori</i> discretization	OLADO: SQP (Powell's type) + Euler IPOPT + Euler HQP + Euler HQP + OMUSES: HQP + Euler HQP + DOPRI5 HQP + GRK4 HQP + IMP HQP + ODETS HQP + SDIRK HQP + RK4
ODEs	adjoint equations	RKCON + Radau IIA IPOPT + SUNDIALS (cvodes)
ODEs	shooting method	BNDSCO + RK4

metody bezpośrednie

metody pośrednie

Metody pośrednie – metoda strzału

Interactive Dynamic Optimization Server

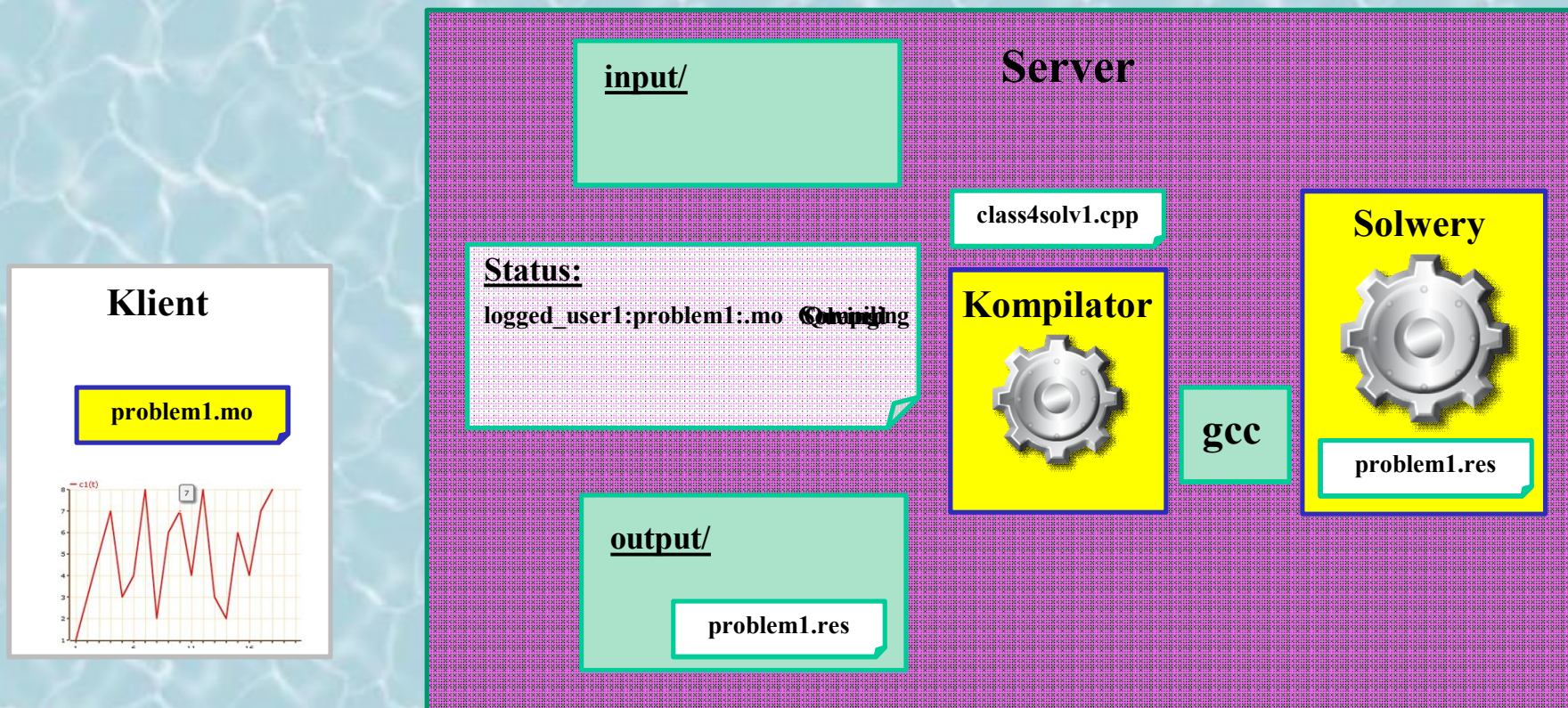
System equations	Method type	Solvers
ODEs (int. controls)	<i>a priori</i> discretization	BONMIN + Euler
ODEs (int. controls)	adjoint equations	BONMIN + SUNDIALS (cvodes)
DAEs (index = 1)	adjoint equations	RKCON + Radau IIA IPOPT + SUNDIALS (idas) - u. dev.
DAEs (index ≤ 3)	adjoint equations	RKCON + RADAU5 + SUNDIALS (kinsol) + MAXIMA
PDEs	adjoint equations	IPOPT + FeLT - u. dev.

metody bezpośrednie –
sterowania całkowite

metody pośrednie – sterowania
całkowite

metody pośrednie – równania
różniczkowo-algebraiczne

IDOS – jak działa



- Zagadnienia techniczne: konta użytkowników, kolejkovanie zadań i szeregowanie, obsługa błędów, etc.

DOML

- Języki do modelowania dla optymalizacji statycznej
 - GAMS, AMPL, AIMMS, ..., NEOS(!);
- Języki do modelowania i symulacji układów dynamicznych
 - **gPROMS**, MatLAB (Simulink, Simscape), **Modelica**;
- Języki do modelowania dla optymalizacji dynamicznej
 - **Optimica** (JModelica.org);
 - TACO (A Toolkit for AMPL Control Optimization);
 - Bocop;
 - PROPT (within TomLab, an extension to MatLAB).

Porównanie

	Optimica	Bocop	TACO	PROPT	DOML 
Programming-language-independent	+	-	+	+	+
Modern (o-o, modular, declarative)	+	0	-	-	+
Open standard (available open-source)	+	+	0	-	+
Based on a popular standard	+	-	+	0	+
May use existing libraries of models	+	-	0	-	+
Allows script programming (eg Python)	+	-	0	+	-
Set up as a server available over internet	-	-	0	-	+
Fosters use of multiple solvers	-	-	+	+	+
Emphasizes chaining of solvers	-	-	-	-	+
Incorporates PDE problems	0	-	-	-	+

DOML (Dynamic Optimization Modeling Language)

zadanie1.mo

```
package Example
  optimization Problem1 (startTime = 0, finalTime = 180)
    minimize Real objective = cost(finalTime) + y3(finalTime);
    Real cost(start=0.0);
    constant Real c1 = 0.09;
    parameter Real p1(free=true, initialGuess=18.7, min=15.0, max=20.0);
    parameter Real y1_init(free=true, initialGuess=137.0);
    ...
    Real y1(start=y1_init, min=100.0, max=200.0);
    Real y2(start=0.00123);
    ...
    input Real u(min=100.0, max=200.0);
  equation
    der(cost) = y2*y2 + y4*y4;
    der(y1) = p1 * y2 + ((p2/K) * u;
    der(y2) + 0.5*(k1 * y1*y1*y1*y1 * sqrt(y2)) + u = 0;
    ...
    Ks*y1*y4 = y6;
  constraint
    y2(finalTime) = 0.0;
    y1(finalTime) <= 0.002;
    y1 + y3 <= 250.0;
  end Problem1;
end Example;
```

DOML – oryginalne rozwiązania

zadanie2.mo

```
package Example

import Modelica.DOML.Inputs.Spline;

optimization Problem2 (startTime = 0, finalTime = 10)
  annotation(solver="idas_ipopt", steps="10");
  minimize Real y3AtEnd = y3(finalTime);
  Spline u_guess(degree=2, table=[0,0;1,0.2;2,0.6;3,0.5, ... ]
  Spline const1_lagrange(degree=0, table=[0,1;1,3,0;2,1; ... ]
  ...
  input Real u(initialGuess=u_guess);
  ...
  Formula eq1(lagrange=eq1_lagrange);
  Formula const1(lagrange=const1_lagrange);
  equation
    eq1: der(y1) = p1 * y2 + ((p2/K) * u;
    ...
  constraint
    const1: y1*y1 + y2*y2 <= 3.0;

  end Problem2;
end Example;
```

Określanie pośrednich rozwiązań

zadanie3.mo

```
package Example

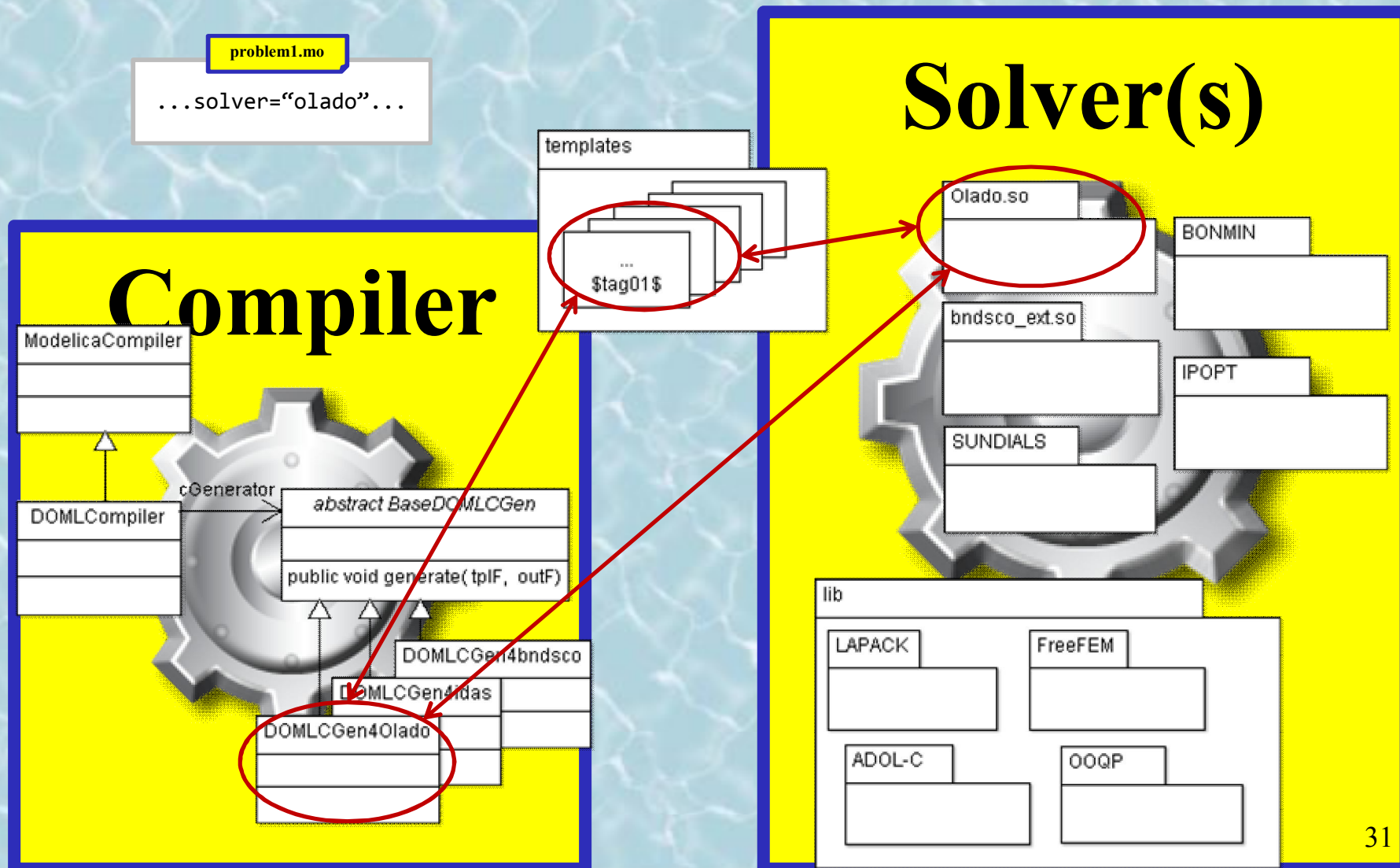
import Modelica.DOML.Inputs.Spline;

optimization Problem3 (startTime = 0, finalTime = 10)
  minimize Real y3AtEnd = y3(finalTime);
  Spline u_guess(degree=0, table=[0,0;1,0.2;2,0.6;3,0.5, ... ]);
  Spline eq1_lagrange(degree=0, table=[0,1;1,0;2,1; ... ]);
  Spline const1_lagrange(degree=0, table=[0,0;1,0.5;2,1.1; ... ]);
  ...
  input Real u(initialGuess=u_guess);
  ...
  Formula eq1(lagrange=eq1_lagrange);
  Formula const1(lagrange=const1_lagrange);

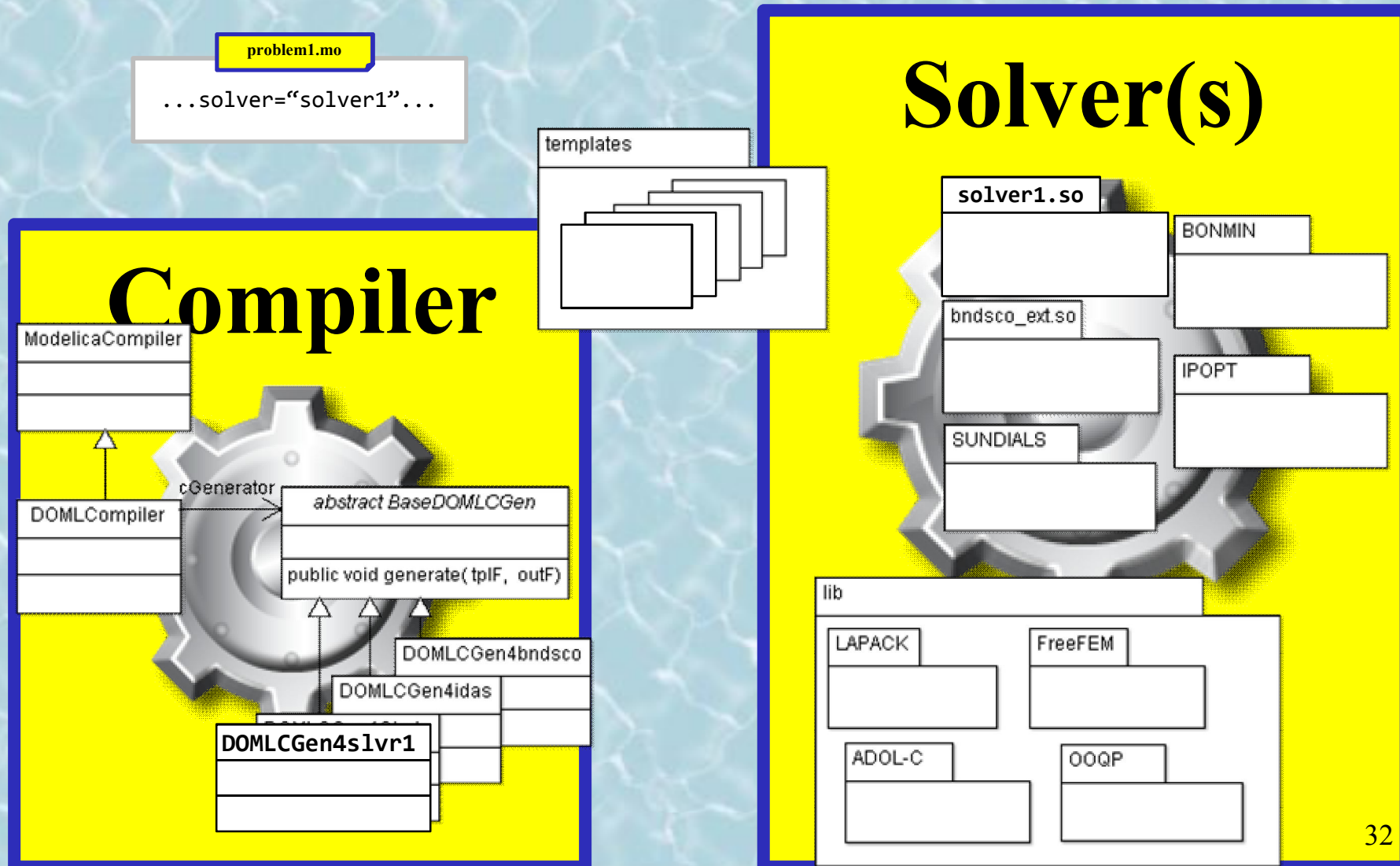
equation
  eq1: der(y1) = p1 * y2 + ((p2/K) * u;
  ...
constraint
  const1: y1*y1 + y2*y2 <= 3.0;

end Problem3;
end Example;
```

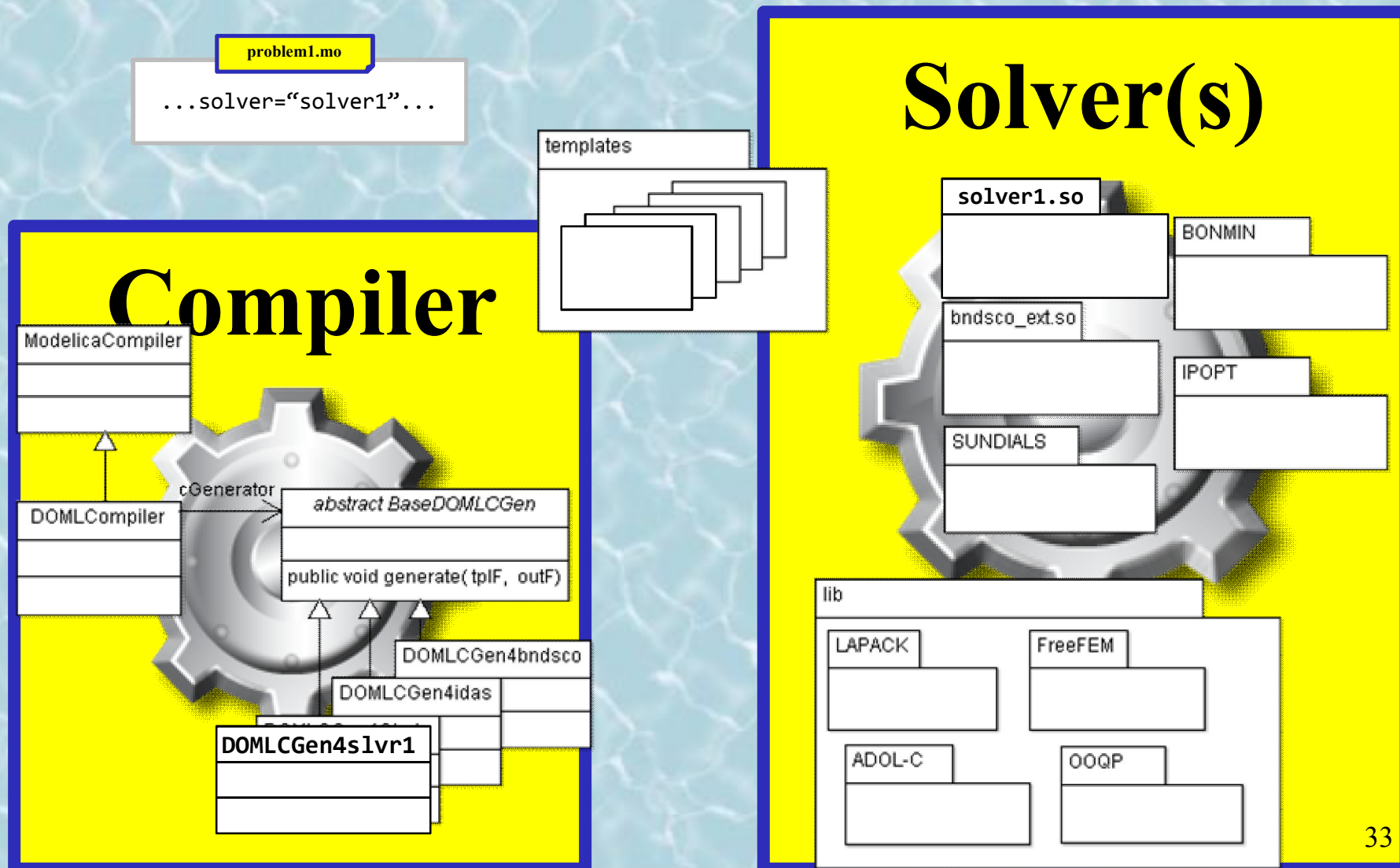
IDOS – stosowanie wielu solverów



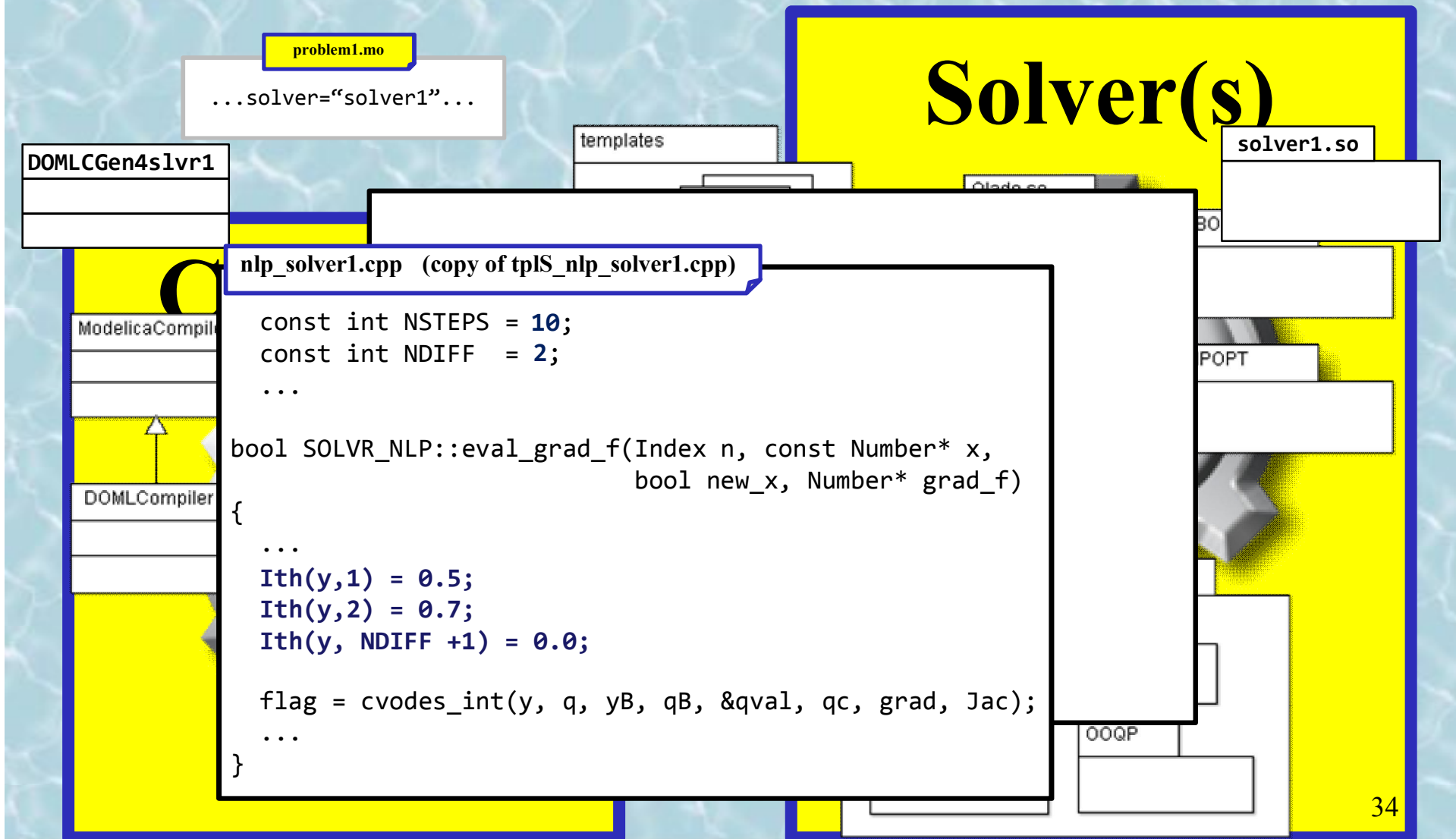
IDOS – stosowanie wielu solwerów



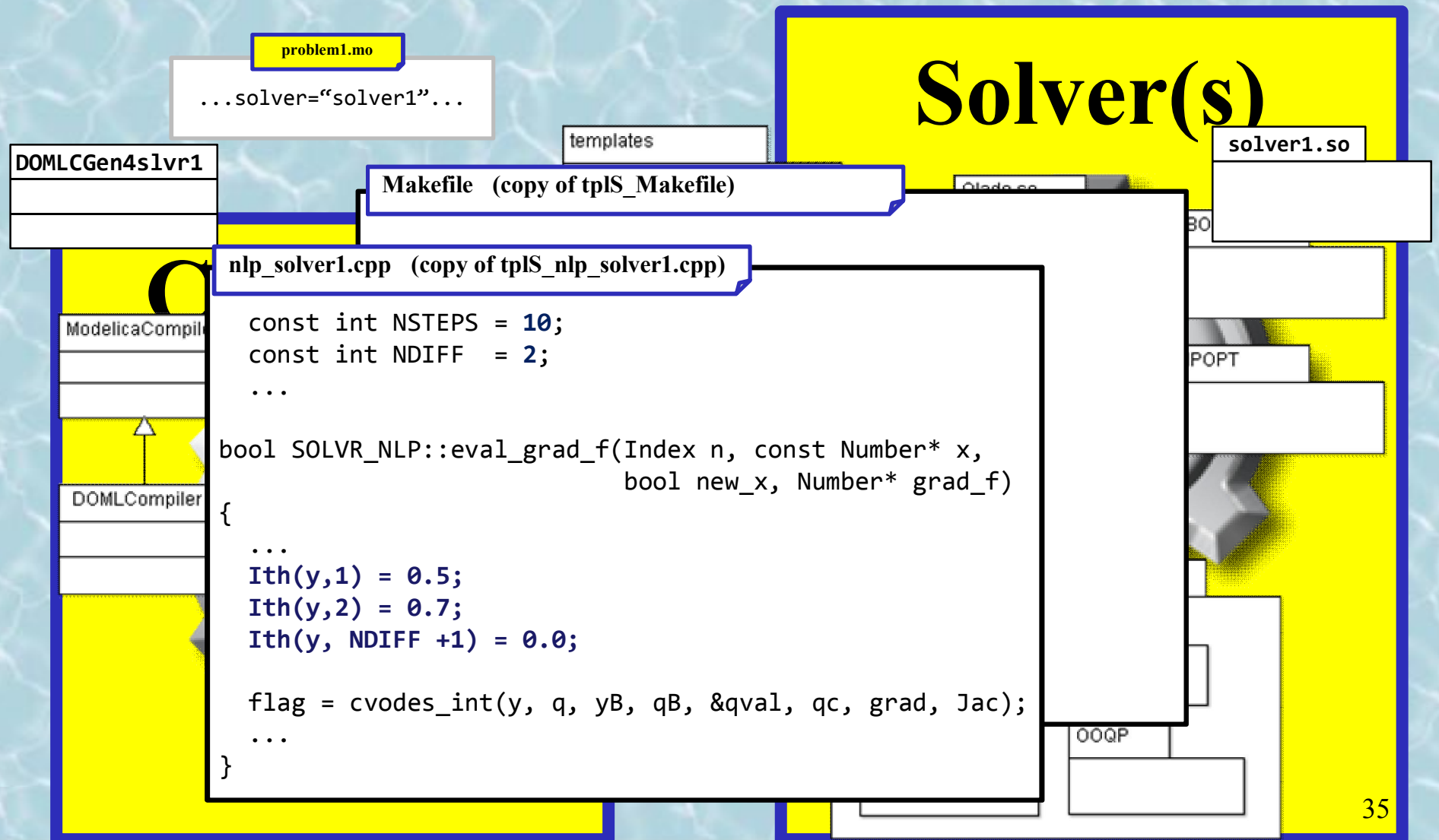
IDOS – stosowanie wielu solverów



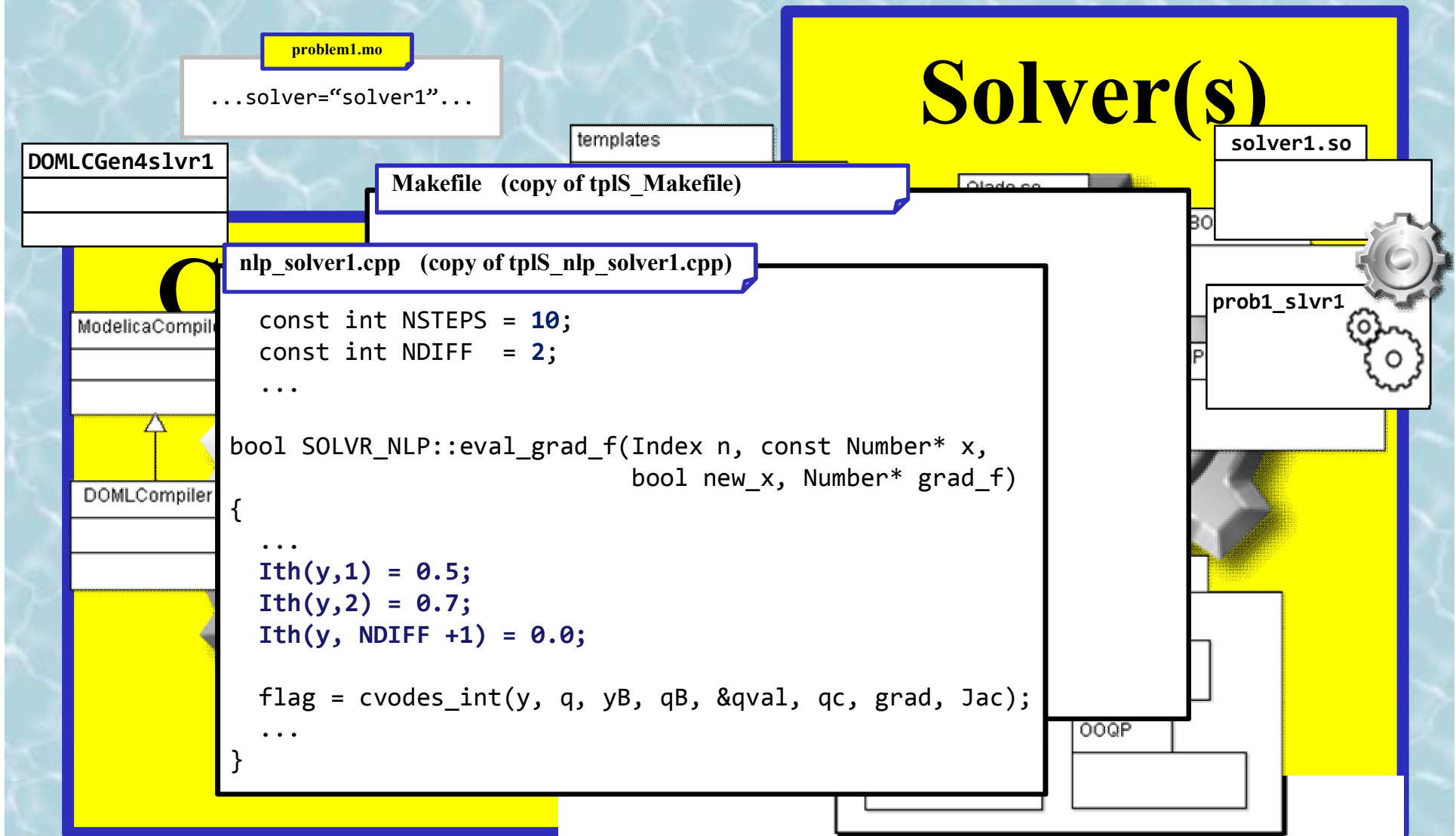
IDOS – stosowanie wielu solverów



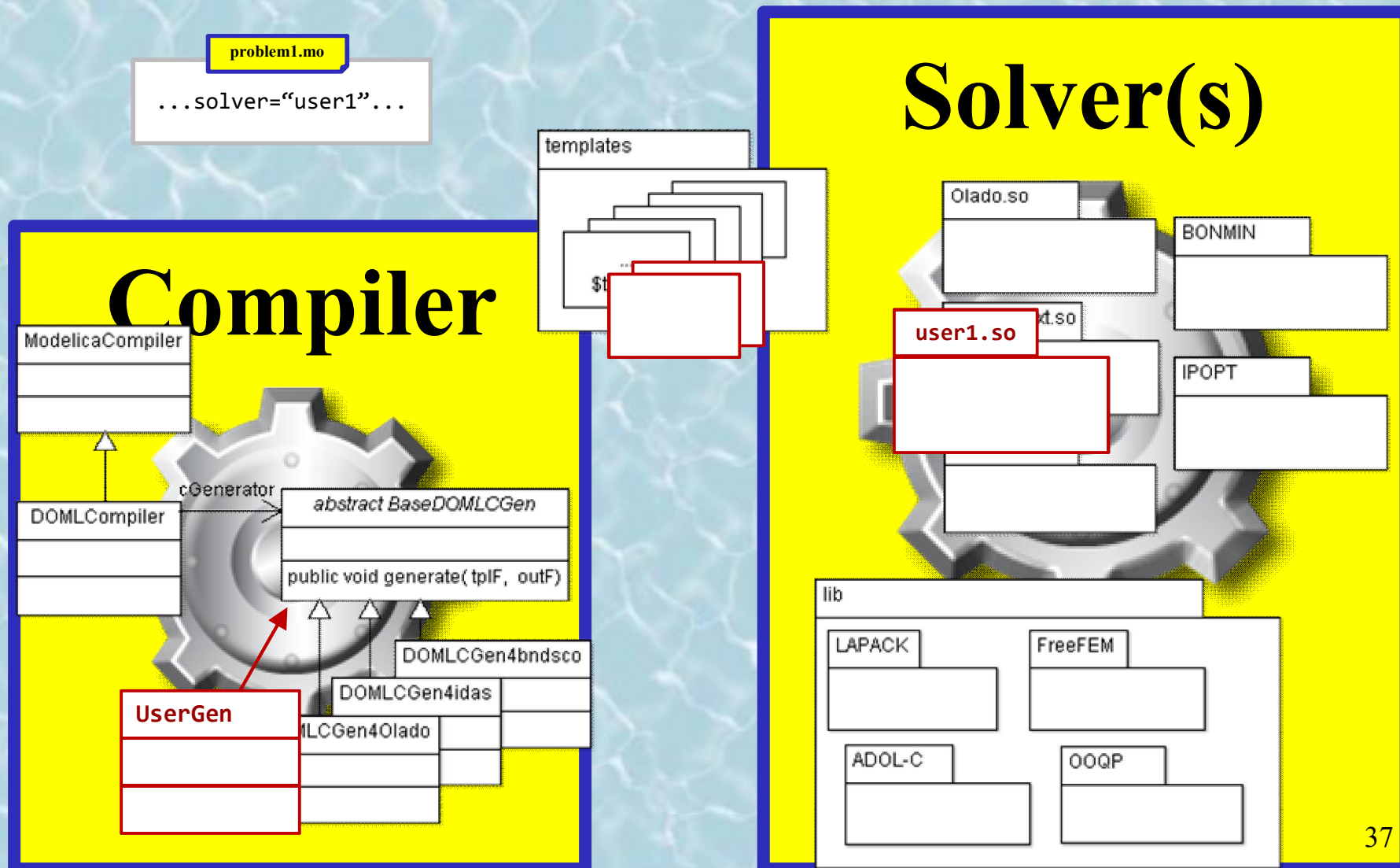
IDOS – stosowanie wielu solverów



IDOS – stosowanie wielu solverów



Umieszczanie solverów na IDOS



Aktualny stan

- Testowanie i rozwój – dostępny szeroki zestaw solwerów;
- Serwer uruchomiony ze swoją stroną internetową;
otwieranie konta na życzenie
(względy bezpieczeństwa; [mailto: b.fajdek@mchtr.pw.edu.pl](mailto:b.fajdek@mchtr.pw.edu.pl))
- Obsługa zadań z równaniami cząstkowymi (PDEs) w ograniczonym zakresie

Zamierzenia na przyszłość

- Całościowe testowanie (obsługa błędów)
- “Reverse compiler” automatyczne ‘ładowanie’ wyników poprzez plik DOML (dla realizacji sekwencjonowania solwerów).
- Inteligentny edytor plików DOML
- Obliczenia równoległe i rozproszone dla obsługi zadań o wielkiej ilości zmiennych
- Automatyczny wybór i sekwencjonowanie solwerów

Podsumowanie

- IDOS - pierwszy serwer dla zadań optymalizacji dynamicznej.
- Serwer oparty na języku modelowania dla zadań optymalizacji dynamicznej DOML umożliwiającym stosowanie zarówno metod bezpośrednich jak i zaawansowanych metod pośrednich.
- DOML stanowi rozwinięcie języka Modelica co zapewnia stosowanie opracowanych wcześniej bibliotek układów dynamicznych.

Dziękuję za uwagę