

5.4. MODELICA

Istotą podejścia zastosowanego w Modelice jest **modelowanie fizyczne [146]**. Oznacza to, że połączenie ikon elementów na schemacie strukturalnym realizuje prawa fizyki obowiązujące w miejscu połączenia. Jest to pewna analogia do grafu wiązań (Bond grafu), a zarazem istotna różnica w porównaniu z SIMULINKIEM. Połączenia bloków służą tam do jednokierunkowego przekazywania informacji, a nie energii). Przy omawianiu własności Modeliki, autor wykorzystuje pakiet Dymola, który dostarcza narzędzia do tworzenia tych modeli i wykorzystuje modele z Modeliki do symulacji.

Istotnym usprawnieniem projektowania interdyscyplinarnego w Modelice jest ujednoczenie sposobów tworzenia i zapisu modeli. Umożliwia to łatwe tworzenie modeli złożonych zawierające elementy i podsystemy o różnej naturze fizycznej, pochodzące z różnych pakietów symulacyjnych.

Propozycje standaryzacji w zakresie modelowania przez długi czas nie miały wsparcia ze strony wiodących producentów. Powodem mogła być obawa o utratę części dotychczasowych rynków zbytu i zmniejszenia dochodów. Sytuacja zmieniła się, gdy realizację tego problemu podjęto w ramach projektu ESPRIT "*Simulation in Europe Basic Research Working Group (SiE-WG)*" [61], [164], finansowanej przez Wspólnotę Europejską. Autor tej pracy uczestniczył we wczesnym etapie prac SiE-WG podczas odbywania stażu na uniwersytecie w Gandawie (Belgia).

Rezultatem prac grupy utworzonej w ramach SiE-WG było ogłoszenie we wrześniu 1997 specyfikacji języka Modelica. Język ten jest dostępny nieodpłatnie, wraz z bibliotekami i wybranym oprogramowaniem na serwerze www.modelica.org

Język Modelica został stworzony zbiorowym wysiłkiem twórców kilku wcześniejszych, obiektowo-zorientowanych języków: Allan, Dymola [37], NMF [134], ObjectMat [40], Omola, SIDOPS+ [23], Smile [63] oraz przez naukowców i praktyków z przemysłu. Prócz wymienionych wyżej, wykorzystano w różnym stopniu idee zawarte w językach ASCEND [123], gPROMS, U.L.M. [58] i VHDL-AMS [56]. Na podstawie powyższych wzorów przyjęto obiektowo zorientowaną syntaktykę i semantykę dla Modeliki [36].

Aktualnie rozwój Modeliki przebiega następująco:

1. Powołano niedochodową organizację: *Modelica Association*. Jej zadaniem jest propagowanie i promocja języka Modelica. Jest ona właścicielem praw do znaku towarowego i logo Modeliki, specyfikacji języka i bibliotek standardowych.
2. Specyfikacja języka, biblioteki standardowe i narzędzia są udostępnione w Internecie i mogą być wykorzystane i rozwijane przez wszystkich zainteresowanych. Są one dostępne nieodpłatnie dla celów badawczych,

- rozwojowych i przemysłowych pod adresem (<http://www.modelica.org/>),
3. Wspólnota Europejska finansuje projekt RealSim: (*Realtime Simulation for Multi-Physics Systems*) [128]. Celem projektu jest stworzenie narzędzi wykorzystujących Modelikę, które przyczynią się do skrócenia czasu i kosztów modelowania złożonych systemów interdyscyplinarnych oraz poprawa efektywności ich symulacji w czasie rzeczywistym i w trybie HiL.
 4. Oprogramowanie symulacyjne oraz biblioteki i translatory oparte o Modelikę są rozwijane przez niezależne firmy komercyjne, które mogą się zrzeszać w *Modelica Association* [100].

Standardowa postać modeli z bibliotek Modeliki może być wykorzystana przez dowolny pakiet symulacyjny pod warunkiem zastosowania odpowiedniego translatora. Powinien on dokonać przekształceń symbolicznych równań różniczkowych (ODE) i różniczkowo-algebraicznych (DAE). Potrzebna jest też analiza struktur dziedziczenia klas modeli i połączeń bloków w schematach graficznych oraz wiele innych operacji. Nie jest to problem banalny, gdyż Modelika akceptuje równania źle uwarunkowane i w postaci uwikłanej. Problemy te zostały rozwiązane przez niektórych producentów oprogramowania:

- Dymola z firmy Dynasim AB, w wersji 4 jest w pełni zintegrowana z Modeliką. Ponadto Dymola generuje M-pliki, Cmx-pliki i S- funkcji, które mogą być eksportowane do MATLABA i SIMULINKA, a w postaci S-funkcji są akceptowane nawet przez RTW. Język ten opisano w dalszej części pracy.
- MathModelica z firmy MathCore wraz z Modeliką tworzy środowisko symulacyjne, które może być zintegrowane z pakietami Mathematica oraz Microsoft Visio.

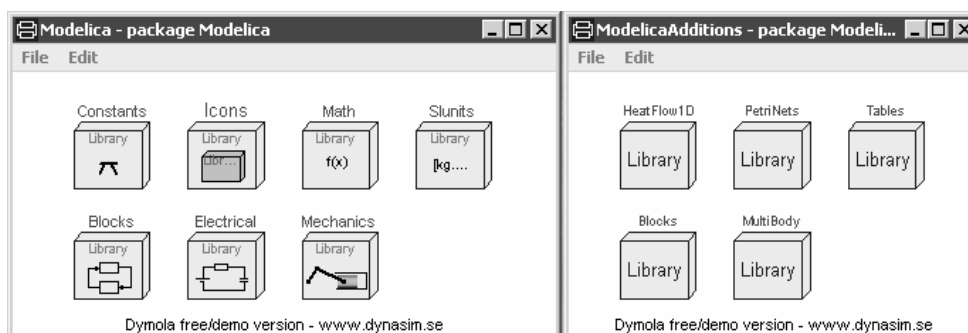
Możliwe jest przenoszenie modeli zgodnych ze standardem Modeliki pomiędzy pakietami symulacyjnymi różnych producentów, jeśli tylko zaadoptują oni odpowiednie standardy. Ułatwi to wymianę i weryfikację modeli przygotowanych przez różne grupy użytkowników, przy modelowaniu urządzeń i procesów o różnej naturze.

MATLAB i Dymola uzupełniają się wzajemnie. Mogą one współpracować ze sobą dzięki narzędziom opracowanym przez Dynasim AB (producent Dymoli i uczestnik zespołu opracowującego Modelikę). Pozwala to na wykorzystanie bogatych bibliotek obu pakietów do modelowania systemów o różnej naturze fizycznej oraz zastosowanie ich do prototypowania w czasie rzeczywistym.

Autor zdecydował się na zamieszczenie w tej pracy opisu pakietów Modelika i Dymola, kierując się przekonaniem, że modelowanie fizyczne zyska w ciągu kilku lat powszechne uznanie zarówno wśród naukowców, jak i praktyków w przemyśle.

5.4.3. Biblioteka standardowa i biblioteka dodatkowa modeli

Modelica jest dostępna w Internecie bezpłatnie wraz z dwoma pakietami bibliotecznymi: standardowym i dodatkowym (*additions*). Biblioteka ThermoFluid jest dostępna nieodpłatnie w www.SourceForgeProject/Modelica.htm. Dodatkowo można dokupić biblioteki oferowane jako opcje pakietu Dymola (rozdział 5.5). Są to: HyLib (*Hydraulics library*) i Powertrain (układy przekazywania mocy w pojazdach). Biblioteki Modeliki mają wielopoziomową strukturę hierarchiczną. Poniżej pokazano okna obu bezpłatnych bibliotek, a w nich ikony z nazwami dołączonych bibliotek specjalistycznych.



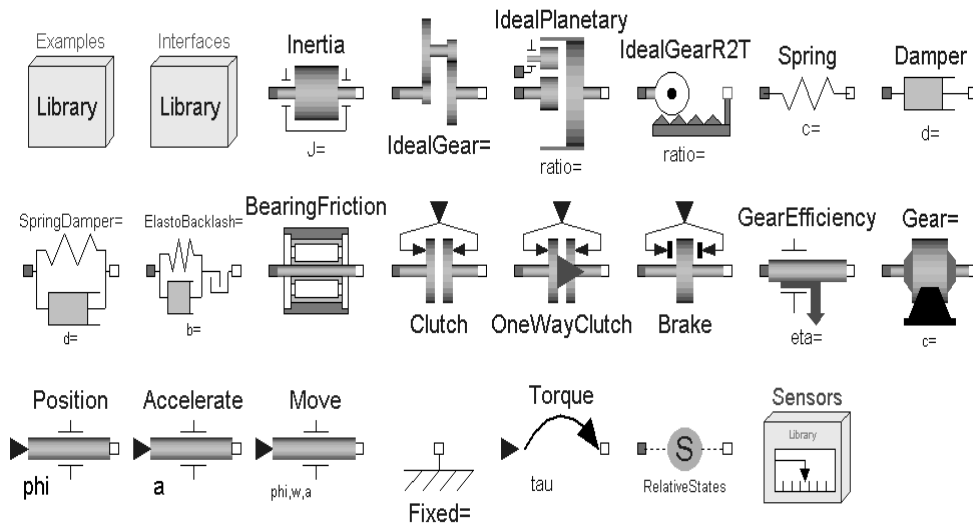
Rysunek 5.4-5. Biblioteki pakietu Modelica: standardowa i dodatkowa

W zespole inicjującym powstanie modeliki byli nie tylko naukowcy, ale też i przedstawiciele przemysłu (m.in. z zakładów Solvay z Belgii). Zapewne obecność przedstawicieli przemysłu jest powodem zwrócenia szczególnej uwagi na różnorodne jednostki miar i utworzenia odrębnej biblioteki dla jednostek stosowanych w Modelice. Są one dostępne po kliknięciu w ikonę *Slunits* (jednostki SI). Fragment zawartości tej biblioteki pokazano poniżej.

```
// Mechanics (chapter 3 of ISO 31-1992)
type Mass = Real (
  final quantity="Mass",
  final unit="kg",
  min=0);
type Density = Real (
  final quantity="Density",
  final unit="kg/m3",
  displayUnit="g/cm3",
  min=0);
type RelativeDensity = Real (
  final quantity="RelativeDensity",
  final unit="1",
```

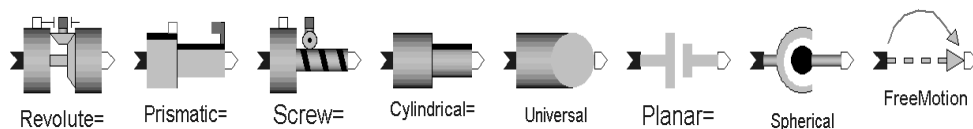
min=0) ;

Dalej pokazano przykłady zawartości biblioteki mechanicznej i elektrycznej z pakietu podstawowego Modeliki. Biblioteka mechaniczna zawiera: *Translational* Sublibrary (*elementy przesuwne*) oraz *Rotational* Sublibrary (*elementy obrotowe*), pokazano na rysunku 5.4-6.



Rysunek 5.4-6. Biblioteka podstawowa: bloki elementów obrotowych

Z kolei na rysunku 5.4-7 pokazano modele złączy z bezpłatnej biblioteki dodatkowej Multibody.



Rysunek 5.4-7. Biblioteka dodatkowa: złącza

Modelika jest obiektowo-zorientowana i każdy jej element ma pola, w których przechowywane są wszystkie istotne, z punktu widzenia celu symulacji, parametry wybranego bloku. Przypisanie parametrom potrzebnych wartości jest wykonywane w sposób dogodny dla użytkownika, na podstawie typowych danych katalogowych dla wybranego urządzenia. Pokazano to na przykładzie modelu przekładni:

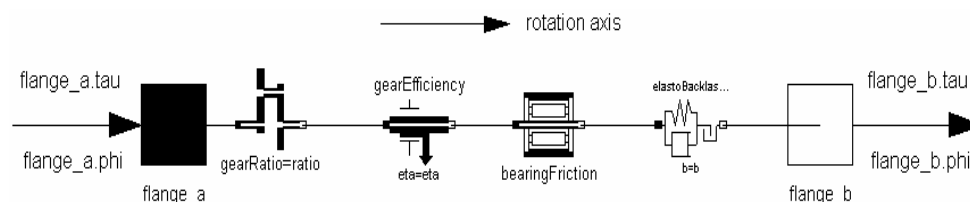
Parameter	Default	Value	Description
ratio	1		transmission ratio (flange_a.phi/flange_b.phi)
eta	1		Gear efficiency
friction_pos[.:.]	[0, 1]		[w.tau] positive sliding friction characteristic (w>=0)
peak	1		peak*friction_pos[1,1] = maximum friction torque at zero velocity
c	1.e5		Gear elasticity (spring constant) [N.m/rad]
d	0		(relative) gear damping [N.m.s/rad]
b	0		Total backlash [rad]

Rysunek 5.4-8. Tabela z danymi przekładni. Użytkownik wypełnia kolumnę Value

W modelu przekładni (rys. 5.4-9) wejściem jest moment i kąt na **wale_a** (*flange_a*), a wyjściem jest kąt i moment przyłożony na **wale_b** (*flange_b*). Prócz tego określa się wartości następujących parametrów:

- Przełożenie (*ratio*),
- Sprawność (*gear efficiency*) uwzględniająca straty na tarcie w zębach przekładni,
- tarcie w łożyskach, mokre (*friction_pos*) i suche (*peak*),
- elastyczność, tłumienie i luzy.

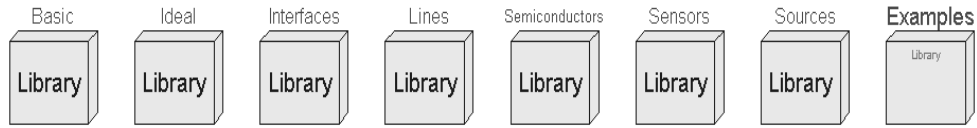
Jeśli niektórym parametrom nie nadano wartości, to przyjęte zostaną wartości domyślne (*default*), podane w tabeli na rysunku 5.4-8. W razie potrzeby po dowolnej stronie przekładni, do wału *a* lub do wału *b*, może być dołączony model inercji z bezpłatnej biblioteki z elementami obrotowymi (rys. 5.4-6).



Rysunek 5.4-9. Model przekładni

W omawianym modelu uwagę zwraca rozdzielanie tarcia suchego i mokrego w łożyskach oraz tarcia w zębach przekładni. Umożliwia poprawną symulację przy uruchamianiu i zatrzymaniu przekładni oraz przy pracy z dowolną prędkością.

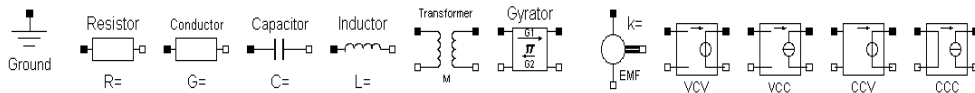
Biblioteka elektryczna ma jeszcze więcej zestawów modeli. Pokazano je na rysunku 5.4-10.



Dymola free/demo version - www.dynasim.se

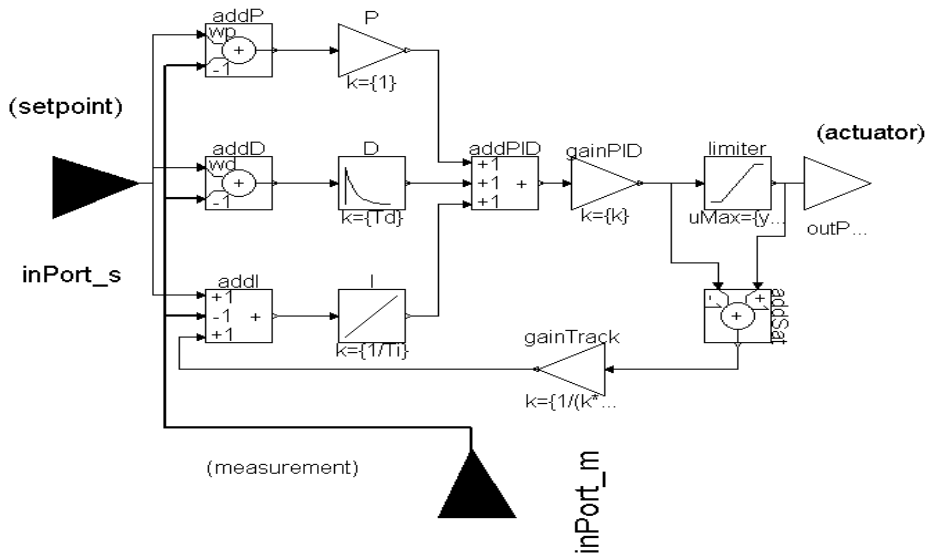
Rysunek 5.4-10. Biblioteka elektryczna

Pokazanie wszystkich modeli wymagałoby sporo miejsca. Skłoniło to autora do zaprezentowania na rysunku 5.4-11 jedynie podstawowych modeli elementów elektrycznych, oznaczonych jako *Basic*.



Rysunek 5.4-11. Modele elementów elektrycznych

Omawiany w rozdziale 5.4.2.1 regulator PID znajduje się w bibliotece *Modelica.Blocks.Continuous*. Zgodnie z podaną w opisie tego bloku informacją, zamiast standardowego bloku PID (rys. 5.4-4) zaleca się stosowanie bloku *LimPID*, z ograniczeniem wartości sygnału wyjściowego. Topologię połączeń regulatora LimPID podano poniżej na rysunku 5.4-12.



Rysunek 5.4-12. Zalecana wersja regulatora PID (z nasyceniem)