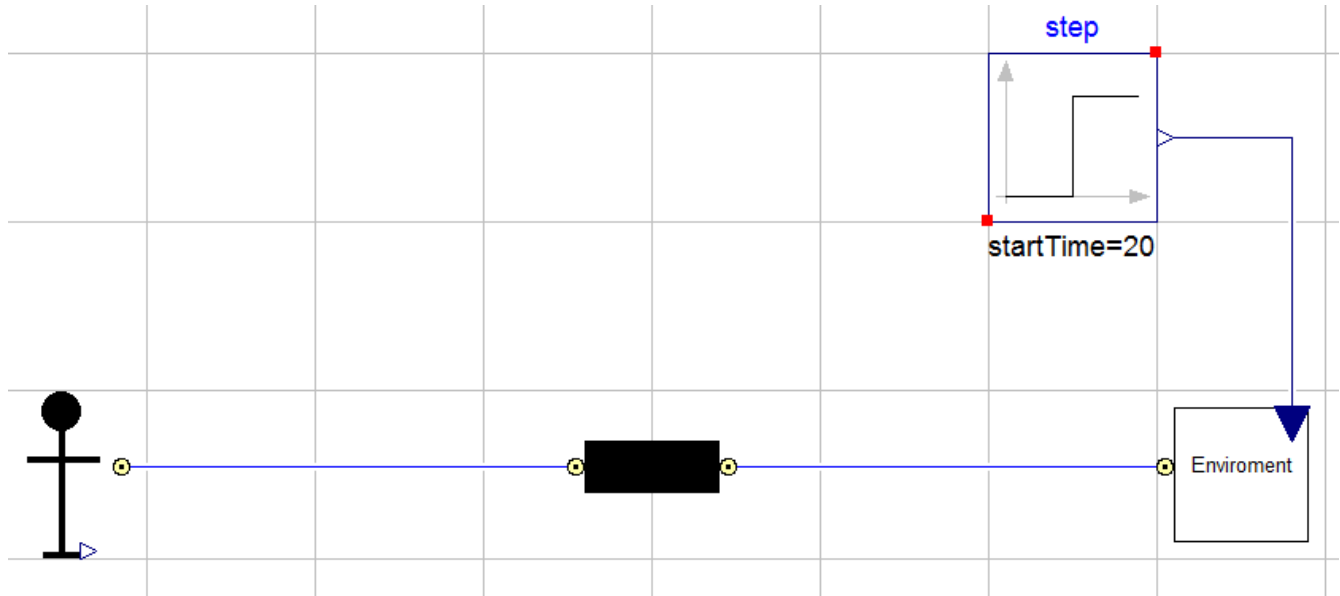


MOS 13 - Expandable connectors

Zadání

Vydeme z úlohy Regulace tělesné teploty, konkrétně **první části bez regulace**, který může vypadat třeba takto.



Obrázek 1: Model jednoduchých ztrát

Zaměříme se na extrakci parametrů z bločků *Tělo*, *Kondukční_ztráty* a *Prostředí* a soustředíme je do nového bločku *ControlPanel*, kde budeme přepočítávat teplotu pomocí funkce na kelviny.

V bloku *těla* nastavujeme parametry:

- hmotnost m
- Vlastní teplo vzniklé bazálním metabolismem $vlastniTeplo$

a budeme chtít číst

- aktuální teplotu těla (v [°C])

V bloku *kondukčních ztrát* nastavujeme:

- koeficient tepelné vodivosti (konduktanci) k .

V bloku *prostředí* chceme nastavovat stejné parametry, jako v bloku Step. Tedy:

- offset,
- `startTime`,
- `height` (změna signálu po uplynutí `startTime`). Viz `Modelica.Blocks.Sources.Step`.

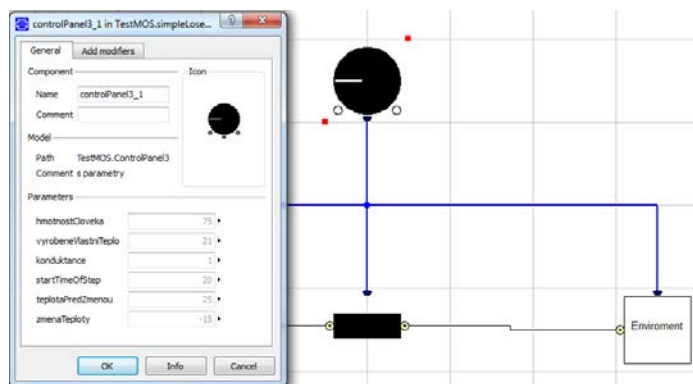
Prvek *ControlPanel* tedy obsahuje parametry *m*, *vlastní teplo*, *k*, *offset*, *startTime*, *height* a přepočítává aktuální teplotu těla na Kelviny. Propojení *ControlPanelu* s ostatními bločky přes expandable connector¹, který bude dědit svou ikonku z jiného konektoru.

Doporučujeme vyrobit nejdříve *ControlPanel* s parametry *m*, *vlastní teplo* a *k*. Otestovat a v případě úspěchu implementovat *Step*, znova ověřit a nakonec implementujte převod teploty.

V Dymole se vám to bude napojovat snadněji, v OpenModelice musíte manuálně napsat rovnice Conner. Pozor na zmatení názvů, na které napojujete, jinak vám to bude hlásit nevyvážený počet rovnic!

Doporučený postup

1. Vytvořte si nový konektor (např. *konektor_ikonka*), kterému nakreslíte ikonku sběrnice nebo čeho chcete.
2. Vytvořte si nový expandable connector, který zdědí ikonku z konektoru výše.
expandable connector MujExpik
extends konektor_ikonka;
3. Upravte bločky Tělo a Kondukční_ztráty tak, aby parametry, které nastavujete v *ControlPanelu* byly vyjádřeny jako *RealInput* a ty budou připojeny do *busConnectoru*.
4. Jednotlivé bločky pak spojte pomocí *expandable connectoru*. Akauzální konektory budou zvlášť.
5. Vyroberte bloček *ControlPanel* a vložte do něj *bus*. Bloček pak vložte ho do celkového modelu a propojte s jednotlivými *busConnectory*.
6. V bločku *ControlPanel* nastavte proměnné, využít můžete například bloku *Constant* z knihovny *Modelica.Blocks.Sources*. Nebo místo nich použijte *RealOutput* a implementací jako parametry umožníte jejich nastavování přímo z vlastností².



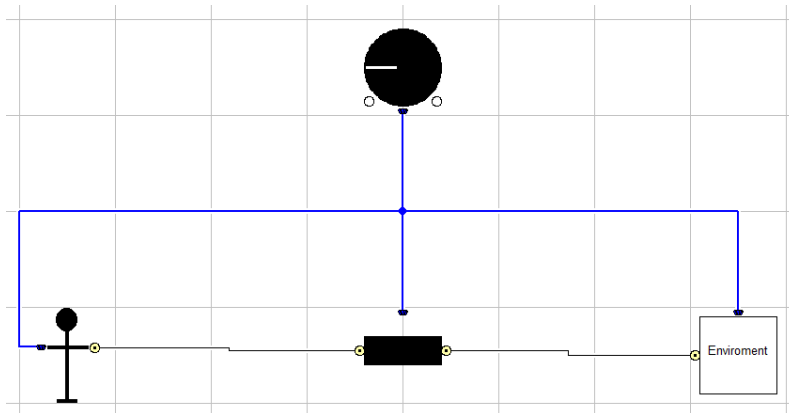
Obrázek 2: *controlPanel* s parametry

¹ Na našem pracovišti používáme termín *bus connector*, volně přeloženo jako sběrnice, neplést tedy s MHD.

² *Expandable konektor* se může napojit jen a pouze na konektor. Pokud tedy do něj chceme vložit nějakou hodnotu, pak musíme vytvořit konektor (*realOutput*) a v rovnici mu přiřadit hodnotu

7. V bločku ControlPanel přepočtete teplotu těla z °C na °Kelviny pomocí funkce CToK.

Schéma



Obrázek 3: Konečné schéma modelu

Černě jsou akauzální vodiče (vedou teplo a tepelný tok). Modře jsou “řídící” signály z expandable connectoru. Na této úrovni stačí graficky propojit bločky. Expandable connector se o dosazení co a kam postará sám.

Poznámky

Když mluvíme o bločkách, většinou nemáme na mysli doslovně abyste použili *block*. Naopak, používejte *model*. *Block* znamená, že třída má jasné kauzální proměnné – všechny musí být definované jako *input*, *output*, nebo *protected* (soukromé). Většinou nám to je jedno, proto použijeme obecný *model*.

Modelica podporuje vícenásobnou dědičnost. To jest, že můžeme dědit z libovolného množství tříd najednou. Ne vždy je to však praktické. Dědičnost implementujeme klíčovými slovy *extends* (dědí) a *partial* („chovná“ třída – určena ke zdědění, sama o sobě nemá smysl). Abychom mohli z nějaké třídy (modelu, konektoru...) dědit, nemusí být nutně *partial*.

Dědí se vše – deklarace proměnných, rovnice, dokonce i anotace. Toho využíváme v této úloze.

Expandable konektor (=bus konektor) implementujeme klíčovým slovem *expandable* před *connector*. Konektor nemusí obsahovat žádné proměnné (naopak se to nedoporučuje), zato se dokáže sám rozšířit o připojené konektory.

Expandable konektor můžeme napojovat pouze na libovolné (RealInput, PressureFlow...) další konektory „rovnicí“ *connect()*. Nemůžeme z něj ale vytahávat proměnné stylem, jakým jsme byli zvyklí u klasických konektorů, tj. např. *konektor1.teplota* nám nebude povoleno. Místo toho ho musíme napojit na existující konektor, například RealInput, RealOutput, nebo jakýkoli jiný. Z toho již můžeme vyčítat a nastavovat proměnné klasicky.

Expandable connector je určen pro kauzální přenos hodnot, to jest všechny jeho proměnné musí být na jedné straně vstupem a na druhé výstupem. Pro další podrobnosti o expandable konektoru se podívejte do specifikace jazyka modelica na www.Modelica.org.

Pokud v části deklarace proměnných napíšeme klíčové slovo *protected*, pak všechny proměnné pod ním až do začátku bloku *equation* (nebo do dalšího klíčového slova *public*, které definuje opak), jsou deklarovány jako *protected*. Není pak možno k ním zvnějšku přistupovat.

Odevzdání

Odevzdávejte do **16. 12. 2014 21:00** do systému.

Nápověda

Krátkou nápovědu si poptáte tradičně na fóru.