Úvod

1.1. Tepelná pohoda

Aby mohl člověk žít zdravě, efektivně a šťastně, potřebuje k tomu především kvalitní pomníky a prostředí ve kterém bezprostředně žije. Většinu života prožijeme ve svých obyblích a jeden ze základních faktorů spokojenosti je teplota vzduchu v oklí. Jinou teplotu potřebujeme ráno když vstáváme jinou přes den, když odpočíváme, pracujeme nebo se vracíme z práce a jinou zase, když jdeme spát.

Na náš vnitřní pocit má však možná větší vliv zdroj tepla než samotná teplota. Jinak totiž vnímáme teplo od slunce, které prostupuje okny a stěnami, jinak vnímáme teplo z podlahového vytápění jinak ze stěnového sálavého vytápění, jinak z obyčejného radiátoru a jinak vnímáme příjemně ohřátý jonizovaný vzduch z krbu.

Jistě tak tušíte, že aby se každý z nás dostal do té správné tepelné pohody, potřebuje jiné teploty v různých místnostech v různou denní dobu.

1.2. Způsoby regulace teploty v obytných objektech

Existuje několik způsobů, kterými se dnes reguluje teplota v obytných prostorech. Ty základní si nyní uvedeme a k nim i jejich výhody a nevýhody.

1.2.1. Evitermní regulace

Tepelné ztráty každého domu se zvyšují s klesající venkovní telotou. Pokud tuto závislost vyneseme do grafu, můžeme toho využít při ekvitermní regulaci.

Ekvitermní křivka je závistost venkovní teploty na teplotě média, kterou musí zdroj do otopné soustavy přivést, aby se ve vytápěném objektu udržela konstantní teplota. Teploměr je potřeba umístnit do stínu, mimo dosah oken a nejlépe na severní stranu budovy.

Při ekvitermní regulaci tedy měříme venkovní teplotu a podle ekvitermní křivky

ovládáme nejčastěji trojcestný ventil a tím systém regulujeme. Jemné doregulování každé místnosti zajišťují termoregulační hlavice nastavené na příslušnou teplotu.

Výhodou této regulace je jendoduchost, spolehlivost, ekonomičnost a skutečnost, že již v mnoha rozličných realizacích funguje bez problémů.

1.2.2. Regulace na referenční místnost

Tato velice jednoduchá regulace je založena na měření teploty v jediné místnosti domu, která je pokud možno častěji obývána, není vystavena přímému slunenčímu svitu a je vybavena stejnými tepelnými zdroji jako většina ostatních místností v objektu.

V této místnosti je umístěn termostat, který dává centrálnímu zdroji tepla impulsy pro sepnutí a rozpnutí.

Výhodou této regulace je jednoduchost a přehled vytíženosti zdroje. Dále je možné zapojit inteligentní termostat, který v sobě má časové teplotní programy a tak topit v různé hodiny na různé teploty.

I když však bude nastaven časový teplotní program, bude pro všechny místnosti stejný a to ještě pouze v ideálním případě, když je topná soustava vyvážená. Tato regulace se vyznačuje překmity a tím i menší tepelnou pohodou a ekonomickou efektivností. V neposlední řadě je nepříjemné, když je termostat nevhodně umístněn a vystaven slunečním paprskům, vyhodnotí tak teplotu jako dostačující, nesepne zdroj tepla a v ostatních místnostech je zbytečně nízká teplota.

1.2.3. IRC – Individula Room Control

Tato regulace je založena na principu autonomního přístupu ke každé místnosti. V každé místnosti musí být nezávislé ovládání zdroje tepla a senzor teploty. Řídící jednotka potom reguluje teplotu v každé místnost zvlášť podle příslušného časově teplotního programu. Komunikace mezi řídící jednotkou a jednotlivými komponenty (moduly) systému je možné realizovat:

a) Po vodičové sběrnici
   - U stávajících budov (rekonstrukcí) nutnost vedení kabeláže ke každému modulu.
   + Stabilita a trvalý provoz bez nutnosti výměny baterií

+ Odstíněno od cizího rušení signálu

b) Bezdrátové

- Výměna baterií
- Rušení cizím signálem
+ Libovolné umístění modulů a řídící jednotky (v dosahu signálu)

Pokud u IRC pomineme problémy s propojením jednotlivých komponent systému s řídící jednotkou, jedná se o univerzální systém regulace, kde jediné omezení představuje inteligence a kapacita řídící jednotky. Přináší maximální tepelnou pohodu v každém pokoji, šitou na míru jak konkrétnímu využití místnosti tak i konkrétnímu obyvateli této místnosti. S tímto systémem regulace se váží i nemalé energetické a tím i finanční úspory, které se pohybují kolem 30%.

2. Zadání

2.1. Cíle

Cílem této práce je navrhnout univerzální řešení konfigurace IRC regulátoru vytápění. Systém by měl umět pracovat s libovolnými komponenty, jejichž vlastnosti a parametry budou popsány pomocí definičního XML souboru. Pokud tedy výrobce přiloží k modulu příslušný definiční soubor, může se takovýto modul (teploměr, hlavice, okení senzor, čerpadlo, ventil, rele) stát plnohodnotnou součástí systému. Poté, co se z těchto modulů pomocí konfiguračního softwaru sestaví regulační okruhy pro jednotlivé místnosti, provedeme export do rozšířeného souboru. Tento soubor je již určen pro řídící jednotku, která podle něho bude schopná celý systém odregulovat.

2.2. Požadavky

- import definičních souborů jednotlivých modulů
- každý modul může mít libovolné množstvý parametrů
- parametry je možné rozdělovat do logických skupin
- uživatel má možnost použít pro moduly i parametry vlastní symbolická jména
- parametry je možné dotatečně kalibrovat
- parametry je možné průměrovat, sčítat, odečítat a násobit
- software zobrazí uživateli pro konkrétní operaci pouze smysluplné parametry
- uživatal má možnost programově měnit adresu modulu v rámci systému
- možnost definovat podmínky a podmiňovat s nimi akce
- jednoduché nastavení časových teplotních programů
- export rozšířeného souboru obsahujícího veškerá nastavení regulovaného systémů

3. Použitá techologie

3.1. Značkovací jazyk XML

XML (*eXtensible Markup Language*, česky *rozšiřitelný značkovací jazyk*) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem *W3C (World Wide Web Consortium)*. Umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat. Jazyk je určen především pro výměnu dat mezi aplikacemi, publikování dokumentů a uchovávání dat aplikací.

Tento formát je otevřený,tedy zdarma přístupný pro všechny uživatele, a není úzce svázán s nějakou platformou nebo proprietární technologií, což v důsledku znamená, že je, v případě potřeby zpracovatelný libovolným textovým editorem. Filozofií jazyka je popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, přičemž se ovšem sám o sobě nezabývá vzhledem dokumentu nebo jeho součástí. Prezentaci dokumentu (vzhled) je možné definovat připojeným stylem. Další, s tímto související možností poskytovanou tímto formátem, je pomocí různých stylů provést transformaci do jiného typu dokumentu, nebo do jiné struktury XML.

Hlavní výhodou jazyka XML je, že nemá žádné předdefinované značky (tagy, názvy jednotlivých elementů) a také jeho syntaxe je podstatně přísnější, než je u jeho předchůdců i konkurentů obvyklé. Pomocí XML značek (*tagů*) vyznačujeme v dokumentu význam jednotlivých částí textu. Dokumenty tak obsahují více informací, než kdyby se používalo značkovaní zaměřené na prezentaci (vzhled) – definice písma, odsazení a podobně. XML dokumenty jsou informačně bohatší. To lze samozřejmě s výhodou využít v mnoha oblastech. Největší přínos bude samozřejmě pro prohledávání, kdy můžeme určit i jaký význam má mít hledaný text.

Další výhodnou vlastností je fakt, že jako znaková sada se implicitně používá ISO 10646 (také *Unicode*). V XML proto můžeme snadno vytvářet dokumenty, které obsahují znaky národních abeced. Problémy s konverzí z jednoho kódování do druhého odpadají díky možnosti současně použít i jiné kódování dle konkrétních požadavků, ovšem takové kódování musí být v každém dokumentu přesně definováno.

XML neobsahuje předdefinované značky, je tedy třeba definovat v dokumentu

vlastní značky, které budeme používat. Tyto značky je možné (nepovinně) definovat v souboru DTD (*Document Type Definition*). Potom je možné automaticky kontrolovat, zda vytvářený XML dokument odpovídá této definici. Tato kontrola je prováděna programem nazvaným parser. Při vývoji aplikací můžeme parser použít, a ten za nás detekuje většinu chyb v datech.

Další vlastností XML je, že v jednom dokumentu můžeme používat najednou nezávisle na sobě několik druhů značkovaní pomocí jmenných prostorů (namespaces). To umožňuje kombinovat v jednom dokumentů několik různých definic ve formě DTD nebo schémat bez konfliktů v pojmenování elementů.

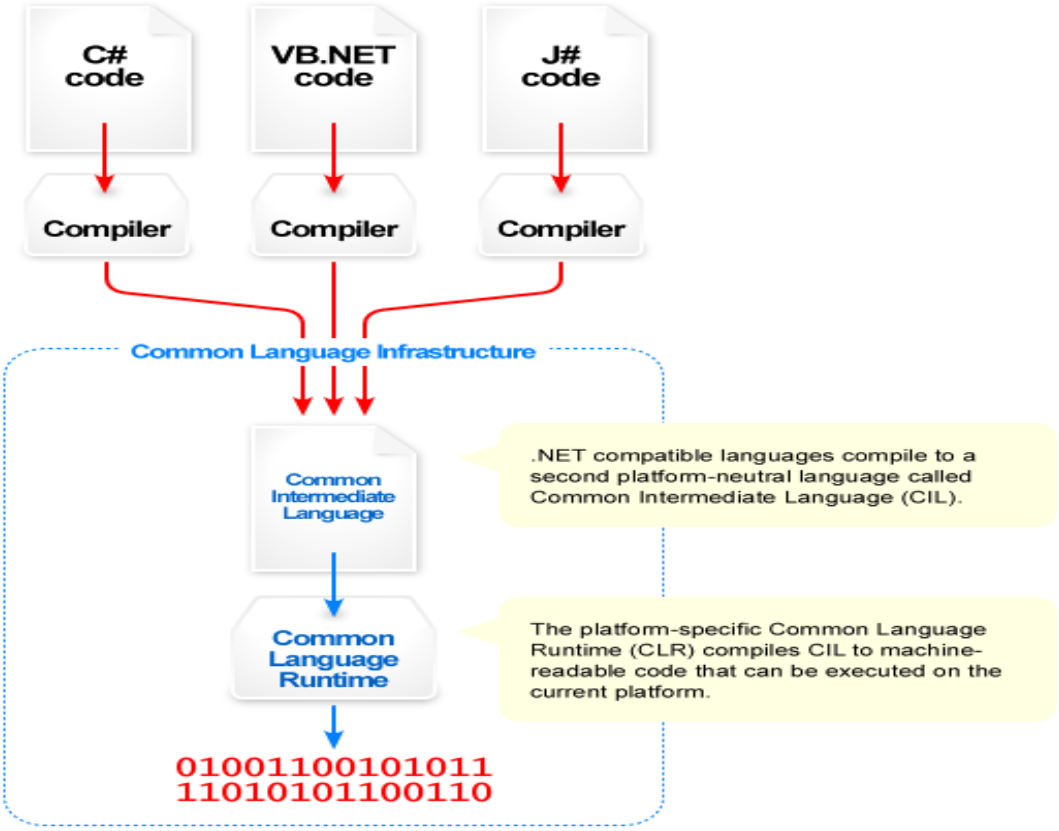
3.2. Microsoft .NET Framework

Pod tímto názvem se skrývá komponenta operačního systému Microsoft Windows, poskytující množství již zpracovaných programových řešení pro běžně nastávající programové požadavky (například numerické algoritmy, uživatelská rozhraní, síťová komunikace) a obsahující běhové prostředí pro vykonávání programů napsaných pro toto prostředí jedním z podporovaných jazyků (základními jsou Visual Basic, J#, C# a C++) nazvané *Common Language Runtime*. Velkou výhodou je, že při dodržení jstých omezení lze přecházet od jednoho jazyka k druhému bez nutnosti manuálně upravovat zdrojový kód. Toto prostředí se v poslední době stalo hlavním běhovým prostředím pro produkty společnosti Microsoft (například nový operační systém Microsoft Vista je postaven na jeho základě). Tato komponenta je od roku 2002 součástí některých distribucí operačních systémů Microsoft Windows. V případě její absence ji lze zdarma doinstalovat například s využitím služby Windows Update, a to i do starších verzí systému(od Windows 98 výše). Struktura Microsoft .NET Framework je uvedena na obrázku 1.

Common Language Runtime poskytuje virtuální stroj pro vykonávání programů, umožňující programovat bez konkrétních znalostí o CPU na kterém výsledný program poběží. Dalšími funkcemi jsou například správa paměti, obhospodařování vyjímek a programová bezpečnost. Architektura *Common Language Runtime* obsahuje následující hlavní prvky:

- *Common Type System* je základním prvkem pro spolupráci mezi podporovanými jazyky. Definuje ve formě objektů základní datové typy, které jsou použitelné ve všech podporovaných jazycích (každý z nich však může definovat navíc své vlastní). Pokud tedy v programu používáme jen tyto základní typy, lze libovolně přecházet od jednoho jazyka k jinému.

- *Common Language Specification* je sada pravidel a vlastností, kterými se musí vyznačovat jazyk pdporovaný v rámci platformy .NET.
- Common Intermediate Language představuje nejnižší člověku srozumitelný jazyk v infrastruktuře .NET Framework, do něhož jsou překládány programy napsané v jakémkliv základním podporovaném jazyce. Tento jazyk je čistě zásobníkově orientovaný a teprve z něj je dalším překladem získám bytekód.
- *Just In Time Compiler* je překladač využívající pro zvýšení výkonu systému techniku dynamického překladu, při níž jsou části bytekódu programu překládány do strojového jazyka daného procesoru až v rámci běhu programu.
- *Virtual Execution System* definuje prostředí pro vykonávání kódu v prostředí .NET. Hlavním úkolem je poskytovat podporu pro vykonávání instrukcí Common Intermediate Language.



obr.1

4. Návrh konkrétního řešení

4.1. Základní pojmy

V následujícím přehledu jsou vypsány pojmy, které jsou používány v dalším textu a jejichž mylné pochopení by mohlo vést ke zkreslenému pochopení výkladu.

ENTITA

- součást systému mající vlastní jméno a ID
- modul, parametr, konstanta, časový program, jednoduchá podmínka, sdružená promenná

ŘÍDÍCÍ JEDNOTKA

- inteligentní řídící mikrokontrolér
- importuje nastavení celého systému
- komunikuje s moduly po sběrnici (nebo jinak)
- načítá parametry z modulů a ukládá je do modulů
- nutnost vnitřní podobnosti s návrhovým programem

MODUL

- obecně jakákoliv jednotka či jiná atomická součást systému
- musí být z pohledu řídící jednotky jednoznačně přímo adresovatelný

př.: teploměr, radiátorová hlavice, čerpadlo, okenní senzor

PARAMETR

- vnitřní proměná modulu
- parametr musí být z pohledu modulu jednoznačne přímo adresovatelný
- má vstupní (in,set) nebo výstupní (out) charakter

PROMĚNÁ

- kombinace parametrů uvnitř řídící jednotky

KONSTANTA

- vnitřní proměné s pevnou hodnotou ID a názvem

ČASOVÝ PROGRAM

- proměná teplota závislá na čase, datu a dnu v týdnu

VLASTNÍ JMÉNO - jméno parametru či modulu, které je této entitě přiděleno implicitně výrobcem či dodavatelem XML definice.

SYMBOLICKÉ JMÉNO

- jméno parametru či modulu, které je této entitě přiděleno uživatelem pro lepší orientaci, nejčastěji vystihující konkrétní použití entity (modulu či parametru modulu).

SYSTÉM - v našem případě budeme toto oblíbené slovo používat pro soubor všech použitých entit a jejich vazeb

SDRUŽENÁ PROMĚNÁ

- kombinace dvou a více proměných a konstant pomocí znamének +(plus) a *(krát)
- má vlastní jméno a jde opět dále sdružovat

JEDNODUCHÁ PODMÍNKA

- dva číselné parametry spojené operátorem: <,>,=,!= nebo samotný parametr typu „BOOL“
- má vlastní jméno a nabývá pouza hodnot TRUE, FALSE

KOMBINACE PODMÍNEK

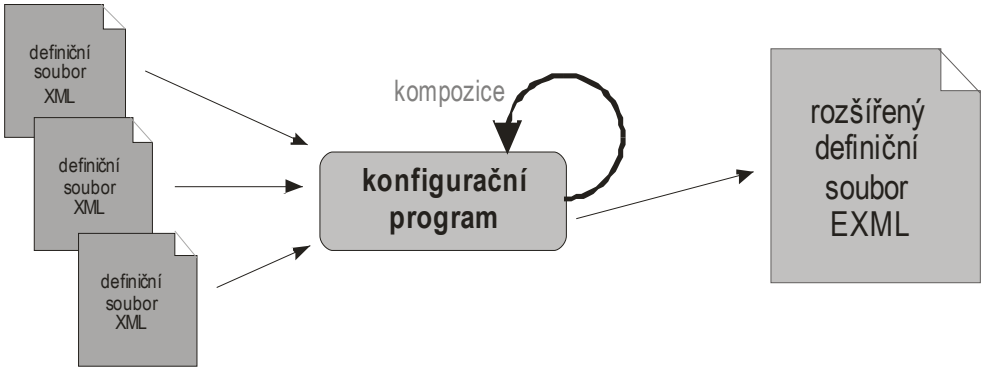
- dvě a více jednoduchých podmínek spojených operárory AND,NAND, OR, NOR nebo XOR
- má vlastní jméno a nabývá pouza hodnot TRUE, FALSE

DEFINIČNÍ SOUBOR

- xml soubor obsahující deterministický popis modulu
- slouží k načtení do konfiguračního programu
- může obsahovat i popis více modulů

ROZŠÍŘENÝ DEFINIČNÍ SOUBOR

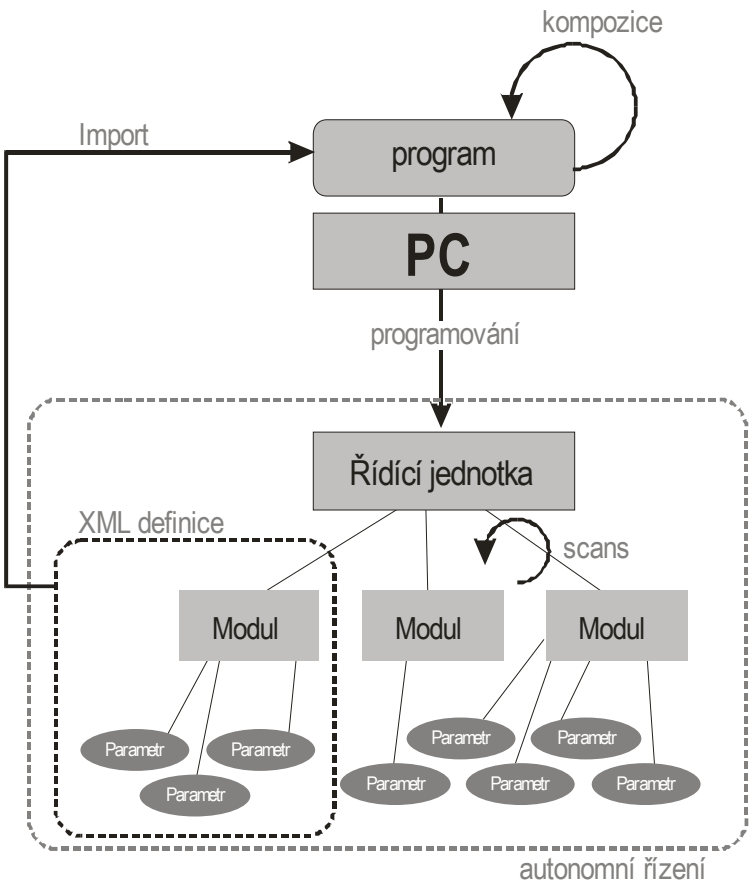
- výstup z konfiguračního souboru
- je v něm popsáno nastavení celého regulovaného systému
- pomocí něho se programuje řídící jednotka



PODMÍNĚNÁ AKCE

- ulož(pošli) do parametru konstantu.
- ulož do parametru jinou hodnotu paramertu
- změň odkaz reference Regulačního okruhu na jinou referenci
- změň odkaz akčního členu Regulačního okruhu

4.2. Princip



Na tomto obrázku je znázorněn celý proces, který začíná načtením definičních souborů, pokračuje sestavením regulačního systému a vrcholí naprogramováním řídící jednotky a samotným řízením resp. exportem regulačního systému do ExtendedXML (rozšířeného XML definičního souboru)

4.3. Vlastnosti parametrů

---------- Povinné vlastnosti ----------

TYPE - datový typ parametru, může nabývat hodnot

BOOL – jednobitová hodnota

CHAR - symbol, znam

NUMBER – číslo (porovnávatelné a řaditelné)

STRING – textový řetězec

type může mít ještě atribut:

ENUM–znamená, že parametr může nabývat pouze výčtových hodnot

ITEM - počet řádků s touto vlastností určuje počet výčtových hodnot
- smysluplné jsou min dva řádky
- jeho obsah je název výčtové hodnoty

item má atribut:

VAL – je samotná výčtová hodnota

SIZE - kolik bitů zabere parametr v paměti řídící jednotky

CODE - způsob jakým bude binární reprezentace parametru převáděna na daný typ (kódování) a zpět. nabívá hodnot HEX,BCD,NEG,DEC

u typu NUMBER má atribut

DEC_POINT - číslo - posun desetiné tečky doleva

ACCESS - tato vlastnost omezuje přístup k parametru vzhledem k samotnému modulu směrem k řídící jednotce.

IN – vstupní parametr (např nastavení radiátorové hlavice)

OUT – výstupní parametr (např. Teploměr)

SET – Nastanení modulu

Hodnotu takového parametru nastaví uživatel v konfiguračním programu a touto hodnotou je modul nastaven při prvním kontaktu s řídící jednotkou (např. kalibrační konstanty či zpoždění výstupů)

NAMEP - jméno, které výrobce implicitně parametru přidělil
- slouží k lepší orientaci v parametrech
- uživatel má možnost přiřadit parametru symbolické jméno

IDP - číselný identifikátor parametru v rámci modulu
- v rámci modulu musí být jedinečný
- jedná se vlastně o adresu parametru v rámci modulu

---------- Nepovinné parametry ----------

MIN - minimální hodnota parametru určená výrobcem modulu

MAX - maximální hodnota parametru určená výrobcem modulu

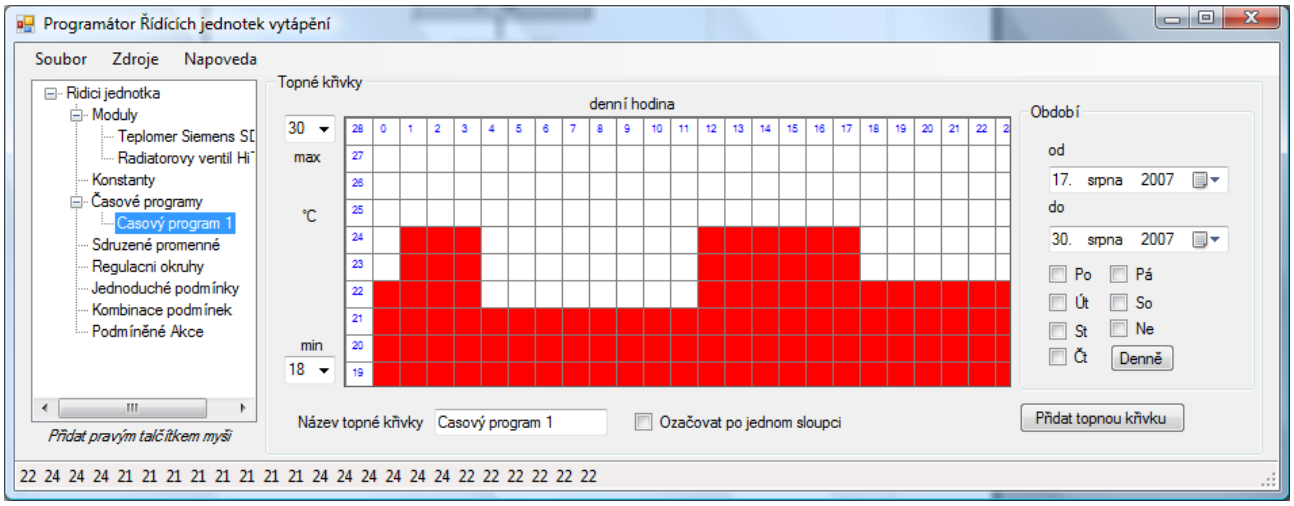
DEFAULT - doporučená hodnota (obvyklá, standardní, ve středu lineární oblasti, nastavit při restartu či výpadku elektřiny)

CATEGORY - textová hodnota shodná pro parametry podobného významu.
- parametry se stejnou vlastností category budou zařazeny společně do podsložky s tímto názvem.

STYLE - číslo - další možnost rozlišení různých typů parametrů.

5. Ovládání

Na levé straně aplikace se nachází stromová strukrura entit, která se postupně s budováním systému rozrůstá. Můžeme se v ní rychle a pohdlně pohybovat mezi jedlotlivími položkami.



Rozsáhlejší prostor napravo je určen pro nastavování hodnot jednotlivých částí systému, pro vytváření nových a editaci stávajících vazeb.

5.1. Cesty ke definičním souborům:

Nejprve je zapotřebí ukázat aplikaci uložiště XML definičních souboru jednotlivých modulů, které budeme v systémů používat. V nich jsou podrobně popsány vlastnosti jednotlivých modulů a jejich parametrů. Zdroje -> import

5.2. Vkládání modulů

Před přidáním každého modulu je uživateli dovoleno použít symbolické jméno a tak si příslušný modul pro další práci pojmenovat dle aktuální potřeby a účelu dané aplikace. Uživatel si může také zvolit adresu pod kterou bude modul v systému adresovatelný. Stiskem klávesy „přidat“ se modul objeví ve stromové struktuře pod svým symbolickým jménem.

Po kliknutí na konkrétní modul se zobrazí parametry modulu přehledně

rozřazené do kategorií. Parametry mohou být primárně dvojího typu.

1. parametry „in“ nebo „out“ které modul odesílá, či přijímá ve standardním regulačním cyklu.
   - takovým může uživatel zadat jako hodnotu symbolické jméno
2. parametry typu „set“
   - takovým nastavuje uživatel konkrétní hodnotu definující stav (nastavení) modulu

Hodnoty je možné změnit poklepáním na přílsušné políčko. Pro uložení změn do systému nestačí ovšem pole opustit, ale je nuté editaci potvrdit tlačítkem.

5.3. Konstanty

K tomu, aby mohla řídící jednotka správně fungovat je nutné zadefinovat všechny kostanty, které budeme v programu dále používat a ktrerým tak můžeme přiřadit i symbolický název.

Pro názornost zavedeme například konstantu, kterou můžeme nazvat „Zámrazová teplota“ a přiřadit jí hodnotu 5 a jednotku °C. Ostatní nastavení na této kartě jsou pro vnitřní reprezentaci konstanty v řídící jednotce a proto jim není nutné v tomto textu věnovat zvláštní pozornost.
U takovýchto jednoduchých dekadických konstant nastavujteme kódování na „decimal“.

Pokud budeme chtít v budoucnu například kalibrovat nějaké vstupy vynásobením K1 a přičtením K2, je třeba tyto před kalibrací zadefinovat v této sekci.

5.4. Časové teplotní programy

Pro maximální teplotní pohodu a úsporu energie je možné si nastavit vlastní časový teplotní program, který figuruje dále v regulaci jako referenční hodnota.
Jedná se vlastně o variaci na konstantu s tím rozdílem, že její hodnota se v čase může měnit.

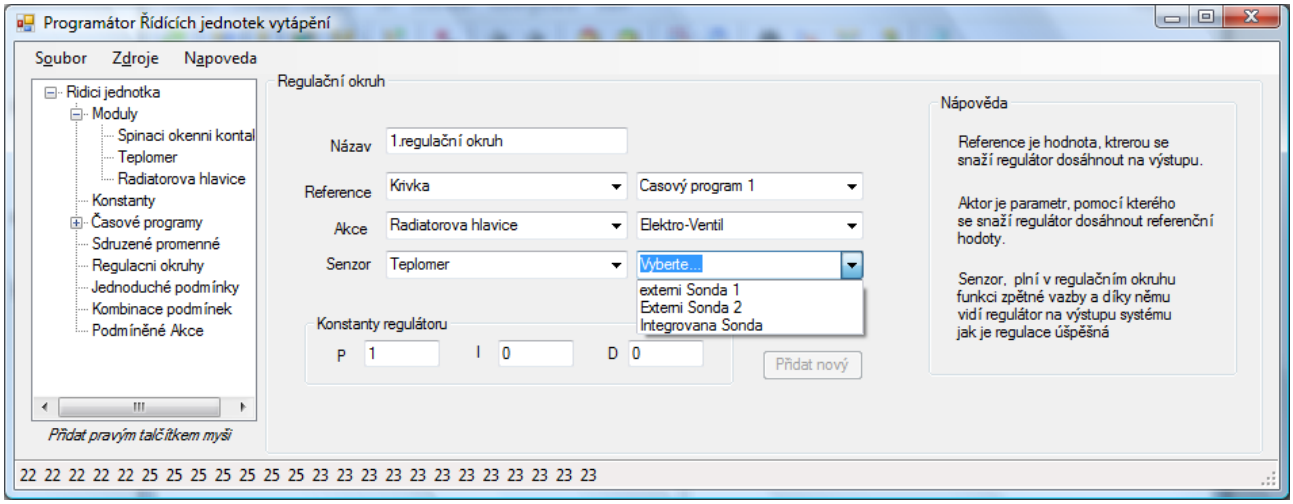
Prostor pro definování časového teplotního programu je rozdělen pro každý den do 24 sloupců přičemž teplotní rozlišení je 1°C. Jedná se o kompromis mezi paměťovou náročností pro řídící jednotku a rozlišovací schopností člověka kombinovanou se setrvačností tepelných systémů.

Pro každý program je ještě třeba zvlolit datumový interval po který bude křivka aktivní a podobně i výčet aktivních dnů v týdnu, které vytvoří s intervalem průnik.

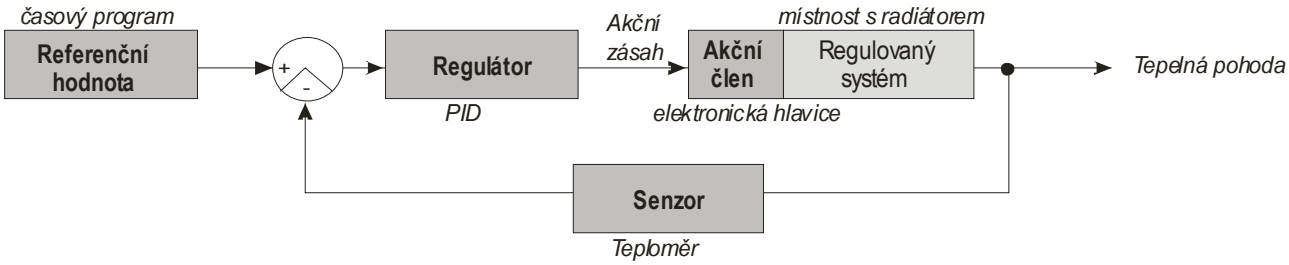
Po přidání se časový program pod svým jménem objeví ve stromové struktuře na levé straně aplikace.

5.5. Regulační okruhy

Jedna z nejdůležitějších součástí konfiguračního programu. Zde se nastavuje regulační vazba mezi jednotlivými parametry modulů a časovými programy.



Každý regulační okruh představuje uzavřenou regulační smyčku, která reguluje teplotu v jedné místnosti, nebo v několika propojených místnostech. Ke správné činnosti potřebuje každý takový okruh 3 základní komponenty zpětnovazební regulace, které jsou vnitřně realizovány jako odkazy na parametry (proměnné)



- REFERENČNÍ HODNOTA (REFERENCE)

Mělo by se jednat o časovou křivku, konstantu avšak při obecnějším využití vlastně jakýkoliv výstupní parametr modulu, či sdruženou proměnnou. Pokud je regulátor správně nastaven měla by se tato hodnota po odeznění regulačních přechodových dějů objevit na výstupu regulovaného systému.

- AKČNÍ ČLEN (AKTOR)

Vykonavatel akčního zásahu. V této kolonce nabídne program uživateli parametry s vlastnosní ACCESS=IN čili vstupní parametry, které mohou mít obecně roli akčního členu.

- SENZOR

Prvek regulační smyčky zajišťující zpětnovazební kontrolu nad tím, jaké výsledky mají dané regulační zásahy na výstupu systému. Bez tohoto prvku by nebylo možné regulovat.

Při vkládání je třeba nejdříve vybrat modul resp. jeho symbolické jméne a poté symbolické jméno parametru.

Tyto tři základní součásti regulačního řetězce jsou sice nezbytné, ale rozhodně nepostučují k popisu chování regulátoru. V nastavení každé regulační smyčky je proto možnost nastavení specifických konstant PID regulátoru.

5.5.1. Nastavení konstant regulátoru

Každý regulační okruh funguje jako nezávislý regulátor. Načítá tedy v určitém časovém intervalu vstupní hodnoty a podle vzorce PID regulace a příslušných konstant nastavuje hodnoty na výstupu. Výstup regulátoru je současně vstupem pro akční člen a vsupem regulátoru je zase rozdíl reference s hodnotou na výstupu změřenou senzorem. V našem případě je tedy vstupem regulátoru „požadovaná teplota“-“skutečná“. Vzorec PID regulace sestává ze tří částí. Každá z nich přispívá k výsledku svým dílem a ten je ještě přenásoben vlastní konstantou.

- P (proporcional)

Konstanta určující kolikrát má být akční zásah větší než rozdíl referenční hodnoty a hodnoty změřené senzorem na výstupu(skutečné hodnoty). Jedná se o proporcionální zesílení regulátoru a jde o stavovou hodnotu, na kterou nemají vliv časové změny statních veličin. Zvyšováním této konstanty se zrychluje regulace ovšem také zvětšuje překmit.

- I (integral) pro diskrétní případy (S-suma)

Tato konstanta multiplikuje časovou sumaci nebo chceteli integraci vstupu regulátoru. Pokud je na vstupu regulátoru konstantní kladná hodnota K, zvyšuje se hodnota na výstupu regulátoru podle předpisu K*I*t, kde t je čas.

- D (diferencial)

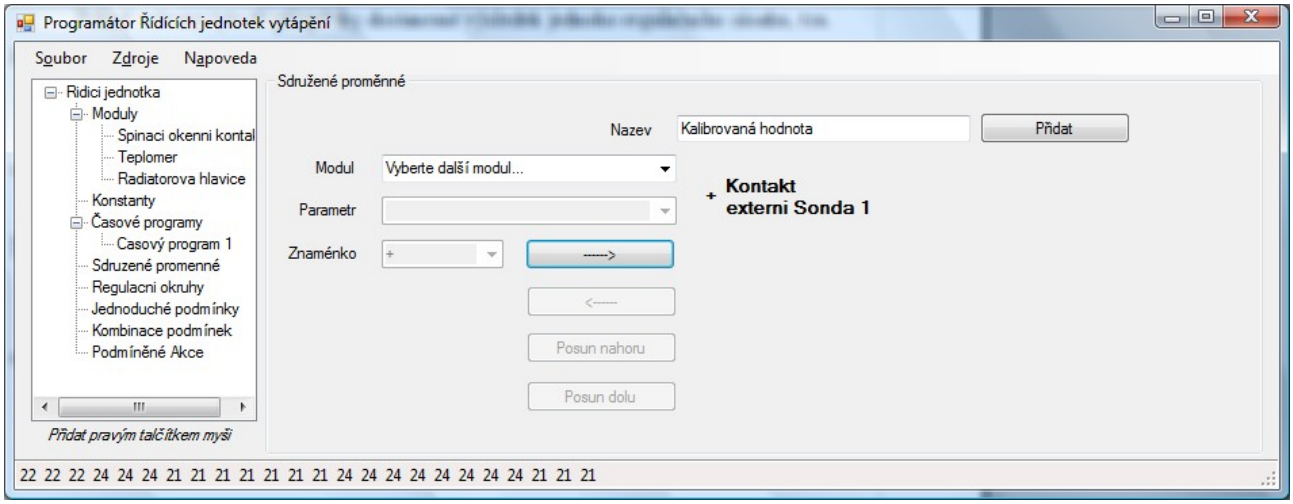
Konstanta určující kolikrát má být vynásoben rozdíl poslední a předposlední hodnoty vsupující do regulátoru. Přítomnost této konstanty omezí překmity způsobené zvětšením proporcionální konstanty .

Pokud sečteme tyto tři příspěvky, dostaneme výsledek jednoho regulačního zásahu, tzn. Pokud necháme P=1, I=0, D=0 získáme obyčejný proporcionální regulátor se zesílením 1.

Konfigurační program dovolí uživateli vytvořit nový regulační okruh až v případě, že jsou všechny parametry správně nastaveny.

5.6. Sdružené proměnné

Tato část programu je navržena hned kvůli několika specifickým aplikacím a její



univerzánost můžeme využít jistě i v jiných specifických případech.

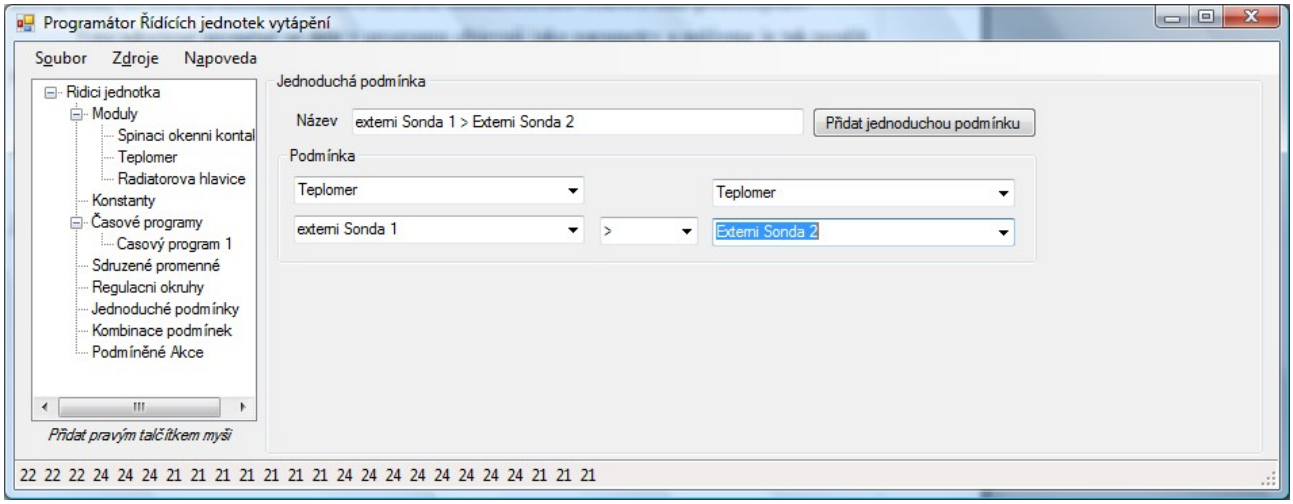
Můžeme zde vytvářet nové parametry (proměnné) pomocí matematické kombinace základních parametrů modulů nebo parametrů a konstant. Je dokonce možné kombinovat jednoduché paramery s již vytvořenými sdruženými proměnnými. Jednotlivé komponenty sdružené proměnné lze pohodlně přesouvat pomocí příslušných tlačítek, které jsou aktivní pouze máli to smysl.

Sdružené proměné se dají uplatnit například při dokalibrování parametrů, kde můžeme multiplikativně aditivní metodou nastavit lineární charakteristiku senzoru kam potřebujeme.

Tyto sdružené proměné se dále v programu objevují jako parametry a můžeme je tak použít například jako nový senzor pro regulační okruh.

Další uplatnění můžou sdružené parametry nalézt např. průměrování hodnot z dvou či více senzorů nebo pro nastavení různých vah více parametrům.

5.7. Jednoduché podmínky



Jednoduché podmínky jsou výrazy spojující dva libovolné parametry vyskytující se ve stávajícím nastavení systému a kombinující je pomocí operátorů <,>,=,!=. Vzniká nová vnitřní proměnná mající vlastní symbolický název a hodnotu TRUE nebo FALSE. Takováto jednoduchá podmínka může dále figurovat v kombinaci podmínek, nebo se již může stát přímou podmínkou pro podmíněnou akci.

5.8. Kombinace podmínek

Proměnná obsahující výraz kombinující jednoduché podmínky nebo přímo parametry typu

BOOL pomocí binárních operátoru AND, OR, NAND, NOR nebo XOR. Vzniká opět proměnná mající vlastní název a nabývající hodnoty TRUE nebo FALSE.

5.9. Podmíněné akce

Operace přiřazení, která se provede na základě splněné podmínky. Může se jednat například o nastavení parametru na nějakou konstantu nebo dokonce nějakou jeho vlastní extrémní hodotu (min,max,def). Následující seznam obsahuje možné využití podmíněných akcí v praxi.

Podmínka: Ovevřené okno (okenní senzor) OR Senzor pohybu = FALSE (nikdo tam není)
Akce: Změn Referenci regulačního okruhu na konstantu „zámrazová teplota“

Podmínka: Teplota podlahy přesáhla (29°C)
Akce: Změna Akčního členu regulačního okruhu na „radiátor“

Podmínka: Teplota podlahy klesla pod (28°C)
Akce: Změna Akčního členu regulačního okruhu na „hlavice podlahového vytápění“

Tyto podmínky nemusí být nutně aplikovány přímo na parametry, pro které jsou implicitně určeny, ale dají se kombinovat i do abstraktních struktur. Například bychom mohli programově spojit rele spínacní světlo s okením senzorem a při otevření okna by v místnosti zhaslo světlo. Možná by tento zlepšovák pomohl od dotěrného hmyzu, ale je to jen jedno z mnoha dalších aplikací, jejichž množství a složitost zaleží pouze na fantazii uživatele.

6. Struktura XML

6.1. Struktura XML definičního souboru

Každý správně napsaný XML soubor musí mít pouze jeden kořenový element. Abychom mohli v jednom souboru definovat více modulů, určili jsme si kořenový element <UNITS>. Každý modul bude poté ohraničen elementy <UNIT>. Každá jednotka(modul) má své jméno udávané výrobcem <NAME> a je v systému na nějaké adrese <ADRESS>, kterou volí uživatel. Vzhledem k tomu, že každý modul může mít libovolné množství parametrů, budou parametry ohraničeny elementem <PARAMETRS> a každý parametr bude v elementu <PARAMETR>.

```
<units>
        <unit>
                <name>
                <adress>
                <parametrs>
                        <parametr>
                                <nameP></nameP>
                                <adress></adress>
                                <type></type>
                                <access></access>
                                <category></category>
                                <min></min>
                                <max></max>
                                <default></dafault>
                                <item val=""></item>
                        </parametr>
                        .
                        .
                        .
                <parametrs>
        </unit>
        .
        .
        .
</units>
```

6.2. Rozšířený definiční soubor XML

Po nastavení všech výše uvedených kroků a definování celého regulačního systému včetně

symbolických jmen modulů a parametrů, jejich adres, vazeb, podmínek a v neposladní řadě i časových programů a sdružených proměnných, můžeme přistoupit k exportu dat pro řídící jednotku.

Exportujeme opět do souboru formátovaného podle standardů XML stanovených koncorsiem W3C. Tento soubor obsahuje kompletní popis nastavení celého systému, včetně všech úprav, které uživatel v programu provedl. Soubor je četelný v běžném textovém editoru a dá se tak po vytvoření bez problémů zkontrolovat.

Pro programování řídící jednotky je však soubor zbytečně velký a tak bude třeba jej ještě upravit nějakou komprimační aplikací. Taková je již však nad rámec této práce.

6.2.1. Struktura rozšířeného XML definičního souboru

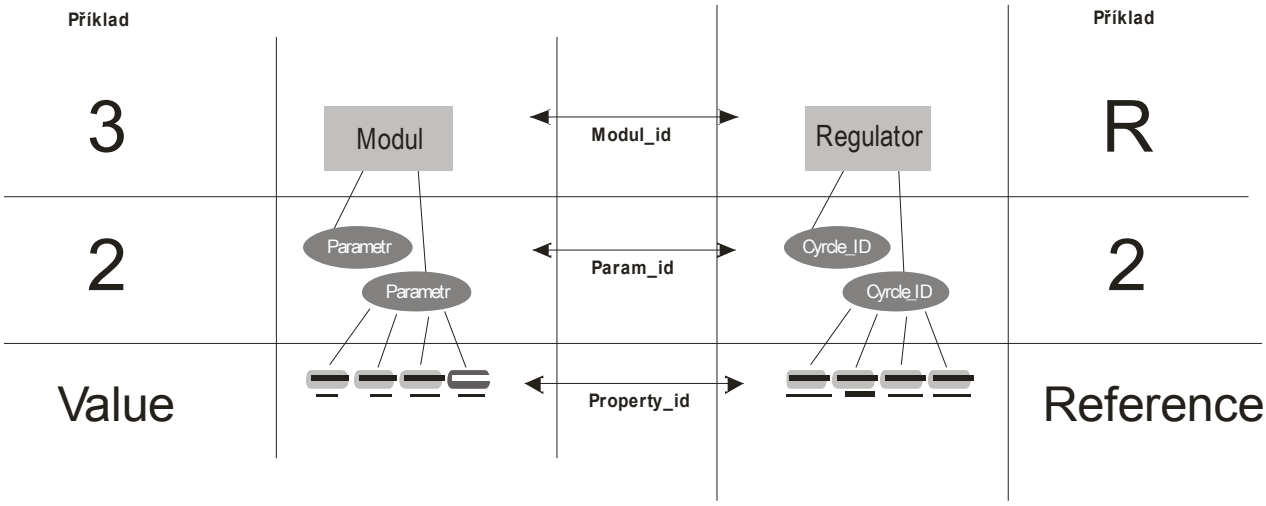
Rozšířený souboru musí mít jediný kořenový element <SYSTEM>. V něm může být až osm dalších elementů <UNITS>, <KONSTANTS>, <KALIBRATIONS>, <TIME_PROGRAMS>, <REGULATION>, <CONDITIONS>, <KOMBINATIONS>, <ACTIONS>. V elementu <REGULATION> snacházejí elementy <CYRCLE>. Každý tento element specifikuje jeden regulační orkuh (smyčku) obsahující elementy <REFERENCE>, <ACTOR>, <SENZOR> s odkazy na jendotlivé komponenty regulátoru.

Každá konkrétní hodnota v systému je dohledatelná pomocí trojice identifikátorů. První identifikátor nejčastěji označován jako „Modul_id“ druhý identifikátor je „Param_id“ a třetí „Property_id“. Pokud „Modul_id“ nabývá číselné hodnoty jedná se o adresu modulu v rámci systému a „Param_id“ je potom adresa parametru v rámci modulu. „Modul_id“ však může nabývat ještě znakových hodnot:

K - konstanta
S - sdružená proměnná
P - podmínka
C - kombinovaná podmínka (Coupled Condition)
T - teplotní program

„Param_id“ je potom ukazatel na jednu konkrétní komponentu z množiny a „Property_id“ doplňuje informaci o jakou vlastnost dané komponenty se skutečně jedná. U parametrů to je

nejčastějí jejich hodnota (u teploměru naměřená teplota atd.), ale pomocí propety se dá ukázat i na minimum, maximum, nebo defaultní hodnotu parametru. Pokud není property_id uvedeno je implicitně nastaveno na VALUE. Pokud ukazují první dva identifikátory na regulační okruh můžeme pomocí „Property_id“ vybrat přímo komponentu tohoto regulačního okruhu (reference, actor, senzor nebo některou z konstant P,I,D)




```
<SYSTEM>
        <UNITS>
                <UNIT>                          // modul
                        <NAME>
                        <S_NAME>                // symbolicke jmeno
                        <ADRESS>
                        <PARAMETRS>             // parametry
                </UNIT>
        </UNITS>
        <KONSTANTS>                             // Konstanty
                <KONSTANT>
                        <NAME>
                        <ID>
                        <VALUE>
                </KONSTANT>
        </KONSTANTS>
        <TIME_PROGRAMS>                         // časové teplotní programy
                <TIME_PROGRAM>
                        <NAME>
                        <ID>
                        <FROM>
                        <TO>
                        <VALUES>                // 24 hodnot za sebou oddělených středníkem
                <TIME_PROGRAM>
        <TIME_PROGRAMS>
        <REGULATION>                            // regulační okruhy
                <CYRCLE>                        // jedna regulační smyčka
                        <ID>
                        <REFERENCE>
                                <MODUL_ID>K     // příklady použití
```

```
                <PARAM_ID> 1
            </REFERENCE>
            <ACTOR>
                <MODUL_ID>1
                <PARAM_ID> 2
            </ACTOR>
            <SENZOR>
                <MODUL_ID>3
                <PARAM_ID> 2
            </SENZOR>
        <CYRLCE>
    </REGULATION>
    <SIMPLE_CONDITIONS>                 // jednoduché podmínky
        <CONDITION>
            <ID>
            <LEFT>
                <MODUL_ID>4             // levá strana výrazu
                <PARAM_ID> 2
            </LEFT>
            <MARK> = </MARK>            // znaménko
            <RIGHT>
                <MODUL_ID>2             // pravá strana výrazu
                <PARAM_ID> 1
                <PROPERTY_ID>DEFAULT
            </RIGHT>

        </CONDITION>
    </SIMPLE_CONDITIONS>
    <COUPLED_CONDITIONS>                // kombinované podmínky
        <COUPLED_CONDITION>
            <OPERATOR>OR
            <CONDITIONS>
                <CONDITION>
                    <MODUL_ID>2         // pravá strana výrazu
                    <PARAM_ID> 1
                </CONDITION>
            </CONDITIONS>
        </COUPLED_CONDITION>
    </COUPLED_CONDITIONS>
    <ACTIONS>                           // akce
        <ACTION>
            <CONDITION_ID>              // ukazatel na podmínku pro vykonání akce
            <SOURCE>
                <MODUL_ID>P             // jaká hodnota se má přiřadit
                <PARAM_ID> 2
            </SOURCE>
            <DESTINY>
                <MODUL_ID>T             // kam se má tato hodnota přiřadit
                <PARAM_ID> 1
                <PROPERTY>REFERENCE
            </DESTINY>

        </ACTION>
    </ACTIONS>
</SYSTEM>
```

7. Závěr

7.1. Možnosti využití a dalšího rozšíření

Tento systém je navržen jako otevřená koncepce, jejíž možnosti nekončí pouze u regulace vytápění, ale jednoduchou variací programu můžeme docílit konfiguraci libovolného strukturovaného systému s podobnými požadavky.

Aplikace je navržena tak, aby postihla co možná největší variabilitu komponent užívaných pro moderní regulaci vytápění budov. Například u podlahového vytápění je nutné šikovně kombinovat tepelné příspěvky od podlahy a radiátoru tak, aby byla místnost vytápěna co možná nejvíce z podlahy, ale přitom aby teplota podlahy nepřekročila určitou teplotu. Takovéto konfigurace je program schopen díky podmíněným akcím, které ovlivňují regulační okruhy.

Dalším krokem ke zprovoznění celého systému by mělo být navržení modulů, řídící jednotky a komunikačního protokolu mezi nimi.

7.2. Hodnocení

Cíle, ktereré jsme si určili na začátku práce se nám povedli realizovat v uspokojivém rozsahu. Funkčnost programu, jeho rozsah a ovládání tak odpovídá jeho využití a uživatel by měl být schopen po krátkém seznámení s programem a s filosofií navržené architektury zvládnout konfiguraci regulace vytápění v jakémkoliv objektu.

7.3. Závěr

Při zpracovávání této práce jsem využil znalosti nabyté během uplynulého tříletého studia na Elektrotechnické fakultě ČVUT, katedře kybernetiky a to především z předmětů Systémy a modely, Systémy a řízení, Programovací jazyky pro řízení, Manažerské informační systémy a další.

Pracoval jsme s literaturou uvedenou v seznamu 7.4. a další informace jsem hledal na internetu s podporou vyhledávače Google. Inspirací mi byly též podobné produkty známých světových, ale i českých firem. Vzhledem k tomu, že si velké společnosti koncepční know-how archkitektury systémů dobře střeží, je výsledkem této práce koncepce má vlastní.

7.4. Seznam použité literatury a zdrojů

[1] C# Programujeme profesionálně *(Robinson, Allen,Cornes,Glynn-2003)*

[2] www.interval.cz

[3] www.builder.cz

[4] Řešené příklady v C# *(Josef Pirkl-2005)*

[5] www.etatherm.cz

[6] Základy XML *(Grusová-2002)*

[7] Začníináme programovat v C# *(Eric Gnuuerson-2001)*

[8] Windows Forms v jazyce C# *(Charles Petzold-2006)*

[9] Vytápění - pro učební obory *(Stanislav Tajbr-1998)*