ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

Ústav letecké dopravy

B3710 – LED – Letecká doprava

Konceptualizace vybraných částí modelu bezpečnosti STAMP

Conceptualization of Selected Parts of STAMP Safety Model

Bakalářská práce

Natalia Guskova

Vedoucí práce Ing. Andrej Lališ Ph.D

Praha 2018

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
d ě k a n
Konviktská 20, 110 00 Praha 1

K621..... **Ústav letecké dopravy**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Natalia Guskova

Kód studijního programu a studijní obor studenta:

B 3710 – LED – Letecká doprava

Název tématu (česky): **Konceptualizace vybraných částí modelu bezpečnosti STAMP**

Název tématu (anglicky): Conceptualization of Selected Parts of STAMP Safety Model

Zásady pro vypracování

Při zpracování bakalářské práce se řiďte osnovou uvedenou v následujících bodech:

- Analýza přístupu k modelování bezpečnosti pomocí systémové teorie
- Identifikace vhodných nástrojů konceptualizaci domény
- Selekce klíčových částí modelu STAMP pro konceptualizaci
- Návrh konceptuálního modelu s využitím vybraných nástrojů
- Vyhodnocení využitelnosti navrženého modelu

ČVUT FD

Rozsah grafických prací: Dle pokynů vedoucího bakalářské práce

Rozsah průvodní zprávy: minimálně 35 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: Leveson, N. Engineering a Safer World: Systems Thinking Applied to Safety. 2011.

Guizzardi, G. Ontological foundations for structural conceptual models. 2005.

Vedoucí bakalářské práce: **Ing. Andrej Lališ, Ph.D.**

Datum zadání bakalářské práce: **20. října 2017**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání bakalářské práce: **27. srpna 2018**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE Fakulta dopravní L. S.

Ing. Jakub Kraus, Ph.D.
vedoucí
Ústavu letecké dopravy

prof. Dr. Ing. Miroslav Svítek, dr. h. c.
děkan fakulty

Potvrzuji převzetí zadání bakalářské práce.

Natalia Guskova
jméno a podpis studenta

V Praze dne 20. října 2017

Čestné prohlášení

Čestně prohlašuji, že tuto bakalářskou práci jsem vypracovávala samostatně, s použitím odborné literatury a uvedením veškerých použitých informačních zdrojů.

Souhlasím s užíváním tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze, srpen 2018

Natalia Guskova

Poděkování

Tímto bych chtěla velice poděkovat svému vedoucímu bakalářské práce Ing. Andrejovi Lališovi Ph.D. za jeho odborné rady, trpělivost a podporu v průběhu psani této práce. Taktéž bych chtěla poděkovat vědecké skupině z fakulty elektrotechnické ČVUT za odborné konzultace a poskytování učebních materiálů.

A nakonec bych chtěla poděkovat své rodině za cennou podporu a víru v mé síly během vysokoškolského zahraničního studia a psaní této bakalářské práce.

Abstrakt

Cílem předložené bakalářské práce „Konceptualizace vybraných částí modelu bezpečnosti STAMP“ je vytvoření konceptuálního modelu s využitím ontologií. Model je tvořen ze dvou částí, které jsou identifikovány jako klíčové v teorii STAMP: řídící smyčka (Control loop) a analýza STPA (System-Theoretic Process Analysis). Pro modelování byla zvolena ontologie UFO (Unified Foundational Ontology), koncepty které jsou využívané v modelovacím jazyce OntoUML a proto také výsledné modely jsou v tomto jazyce. Pro modelování konceptů, které modelovací jazyk OntoUML nepodporuje, byl použitý grafický editor yEd a samotná ontologie UFO. Model byl úspěšně vytvořen, validován a tím připraven pro jeho případné využití v řízení bezpečnosti. Model může být také využit pro následující rozvoj problematiky řízení bezpečnosti a vytvoření kompletního modelu STAMP.

Klíčová slova

Konceptuální modelování, ontologie, UFO, OntoUML, STAMP, STPA, Teorie řízení se zpětnou vazbou

Abstract

The goal of this bachelor thesis "Conceptualization of Selected Parts of STAMP Safety Model" is the creation of conceptual model using ontology engineering. The model is made of two parts, which were identified as the core concepts in STAMP theory: a control loop and a System-Theoretic Process Analysis (STPA). For modeling, UFO ontology was chosen (Unified Foundational Ontology) which concepts are used in OntoUML language and so the resulting models are also in OntoUML. To model concepts out of support of the OntoUML language, yEd graphic editor was used with the very UFO ontology. The model was created in the Menthor Editor application and represented in this work by the yEd editor. The model has been successfully created, validated and now it is ready for its potential use in safety management. The model can also be used to develop solutions to safety issues and to create a complete STAMP model.

Keywords

Conceptual modeling, ontology, UFO, OntoUML, STAMP, STPA, feedback control theory

Obsah

Seznam příloh

Seznam obrázků

Seznam tabulek

Seznam zkratek

ICAO	International Civil Aviation Organization	Mezinárodní organizace pro civilní letectví
IATA	International Air Transport Association	Mezinárodní asociace leteckých dopravců
ECAC	European Civil Aviation Conference	Evropská konference civilního letectví
EU	European Union	Evropská Unie
AVSEC	Aviation Security	Letecká bezpečnost
STAMP	Systems-Theoretic Accident Model and Process	Systémově-teoretický model nehod a procesů
STPA	System-Theoretic Process Analysis	Systémově-teoretická analýza procesů
CAST	Causal Analysis based on STAMP	Analýza příčin, která je založená na STAMP
UFO	Unified Foundational Ontology	Sjednocená základní ontologie
ER	Entity-Relationship model	Model objektů a vztahů mezi nimi
IT	Information technologies	Informační technologie

Úvod

Rychlý rozvoj letecké dopravy za sebou nese nejenom vytvoření nových technologii, také i zvýšení počtu cestujících a letadel ve světě. Vzdušný prostor začíná být více obsazen, a proto můžeme mluvit i o přetížení jeho kapacity a kapacity letišť. Kvůli tomu se zvyšuje i pravděpodobnost vzniku rizika, a jako následek nehoda. V této situaci vzniká otázka, jaké opatření by mohli vypomoci v zajištění bezpečného provozu. Proto bylo zapotřebí vytvoření provozních oblastí, které budou schopné zajišťovat bezpečí. Jedním z nich je řízení bezpečnosti, které je nově vzniklou provozní oblastí, která se zabývá analýzou nehod. Do nedávné doby se bezpečnostní inženýři zabývali analýzou vzniklých nehod, teď je hlavním cílem predikce možných nebezpečných stavů a jejich eliminace před vznikem. Například eliminovat vznik nebezpečného stavu můžete pomoci zavádění rámce pravidel pro řízení procesu, které budou dodržované, anebo předem navrhnout možné variace postupu zabraňujících rozvoji nebezpečného stavu.

Tato problematika je důležitá pro letectví, protože ve 21. století je hlavním cílem letectví bezpečný provoz. A nejvyšších ukazatelů bezpečí se dá dosáhnout jen při snížení výskytu nebezpečných stavů v provozu. Proto bylo vytvořeno nové odvětví v letectví – safety. Safety inženýři se zabývají provozní bezpečností, zavadění pravidel, analýz a nástrojů pro podporu bezpečného provozu v letectví.

Existuje řada nástrojů pro analýzu vzniku nebezpečí, ale neexistuje technická stránka, díky které by se dalo rychle a automatizovaně provádět analyzování procesu. Jsou vytvořené různe analýzy, technologie, modely procesů Proto bude v této práci představená snaha o zavádění nástrojů pro automatizaci některých instrumentů používaných v analýze bezpečnosti.

Zajímavým modelem, který by mohl být namodelován, je model bezpečnosti STAMP. Důvodem je, že zkoumá jádro řízeného procesu – řídící smyčku, a umožňuje zjistit chyby, které zde mohou nastat, anebo zjistit příčiny vzniku těchto chyb. Hlavními nastrojí STAMP jsou analýzy CAST a STPA. Kde první zjišťuje příčiny vzniklých nehod a tá druha zkoumá potenciální příčiny nehod během jejich vývoje.

Zaváděním technické stránky v safety inženýrství se dá zvýšit efektivitu zjišťování nehod a predikce nebezpečí, která mohou nastat a vyvolat tím velké peněžní i lidské ztráty, které mohou ovlivnit rozvoj letecké dopravy a tím i snahu o zlepšování dopravní infrastruktury. Člověk jakož to potenciální cestující si zpravidla volí takovou společnost, která mu nabízí nejlepší a nejbezpečnější přepravu popřípadě nejlepší ekvivalent ekvivalent přepravy.

1. Bezpečnost

Bezpečnost je důležitou součástí života každé živé bytosti na Zemi. Každá bytost se snaží dosáhnout stavu bezpečí, ať už primitivní organizmus anebo člověk. Například v roce 1943 Abraham H. Maslow ve svém článku „Teorie motivace člověka“ poprvé představil svou teorii, ve které vysvětlil, proč má každý člověk určité důvody, motivaci k tomu či jinému činu. Základem této teorie je struktura potřeb člověka, ve které mají některé potřeby určitou prioritu před ostatními. Maslow tvrdil, že na prvním místě jsou potřeby základní, ať už jsou to potřeby fyziologické nebo pocit bezpečí. Tyto potřeby zajištují přežití člověka, a vždy budou hnací silou pro jakékoliv lidské chovaní. Až po překonání těchto potřeb je člověk schopen zabývat se sociálním životem a dále i sebezdokonalováním. Pro uspokojení potřeby bezpečí a jistoty, člověk potřebuje existenci zákonů a pořádků, potřebuje nemít strach, chtít vědět, že nic neohrožuje jeho život. [1] Důležité je podotknout, že s rozvojem lidstva a vědy se měnili i představy člověka o věcech, které vlastní nebo o prostředí, ve kterém žije. Tím pádem některé obavy s časem zmizely, některé zase naopak vznikly a některé přebývají až doteď.

Dnes v době informačních technologií člověk dosáhl vysoké úrovně svého rozvoje. Existuje medicína, vysoké technologie, sociálně regulovaný život. Ale stále jak i před mnoha lety, člověk potřebuje uspokojit své základní potřeby. Proto i dnes existuji různé firmy, úřady, mezinárodni organizace, které se zabývají otázkou bezpečnosti.

1.1. Bezpečnost v letectví

Letectví jako jedním z nejdůležitějších odvětví technického života lidstva, má též své nástroje pro regulování bezpečnosti. Leteckou legislativu můžeme rozdělit do tří obecných skupin. První skupinou, která koriguje legislativu v oblasti civilního letectví je skupina mezinárodních organizací, do které patří ICAO, IATA a ECAC. Další skupinou je regionální legislativa, například pro státy Evropské Unie (EU) platí samostatná legislativa, která je tvořena nařízeními Evropského parlamentu a Evropské rady. Třetí skupinou je pak legislativa na úrovni státu. Například v České republice je vytvářena Úřadem pro civilní letectví, zákonem č. 49/1997 Sb. o civilním letectví a v neposlední řade jsou všechny tyto dokumenty upraveny a doplněny do finální podoby v leteckých předpisech. Je nutné zdůraznit, že tento systém platí pro členy jednotlivých organizací, které jsou uvedeny výše. V jiných případech se zajištění bezpečnostních standardů řídí vlastní legislativou.

Letecká bezpečnost se dělí na dvě části, kde jedna se zabývá protiprávními činy, a ta druhá provozní bezpečností.

1.2. Security aneb ochrana civilního letectví před protiprávními činy

V Annex 17, který je vydán ICAO a dále v českém znění v leteckém předpisu L17, je pojem security definován jako ochrana civilního letectví před protiprávními činy. Tohoto cíle se dosáhne pomocí kombinace bezpečnostních opatření, lidských a materiálních prostředků. [2]

Brzy po vzniku civilního letectví se objevila problematika nezákonných činů ohrožujících bezpečnost civilního letectví. Nejprve se jednalo o činy, například poškození letecké techniky nebo jiných zařízení, způsobené leteckým personálem nebo jinou osobou. Proto v roce 1963 byla přijata Tokijská úmluva o trestných činech na palubě letadla, která se vztahovala na přestupky způsobené posádkou, nebo cestujícím s násilným fyzickým činem.

Kvůli stálému zvyšování protiprávních činů byly přijaty další úmluvy o potlačování protiprávních činů, v roce 1970 Haagská úmluva tykají se zmocnění letounu a v roce 1971 Montrealská úmluva pojednávající o protiprávných činech ohrožujících bezpečnost civilního letectví.

Mezinárodní organizace reagovaly na rychle se rozvíjející letecký terorismus a následné požadavky leteckých velmocí. Proto se v roce 1974 stal pojem “security v civilním letectví“ plnohodnotným oborem, a byl vložen do přílohy Chicagské úmluvy pod číslem 17.

Datum 11.9.2001 však změnil pohled na security od kořenů. Bezpečnost se stala prioritou a boj s terorismem se začal brát jako veliký problém, proti kterému se musí bojovat každodenními technologickými postupy. Kvůli tomu byl zpracován panel ICAO pro ochranu letectví AVSEC.

1.3. Safety anebo provozní bezpečnost

Pojem safety se jinak označuje jako provozní bezpečnost. Hlavním nástrojem pro regulaci safety je legislativa. Například do oblastní bezpečnosti patří dozor nad provozuschopností a způsobilostí leteckých výrobků nebo zařízení používaných na letištích, licencování leteckého personálu, pracovníků letišť, dodržování bezpečnostních letových norem, regulace výstavby a provozu letišť.

S postupem času se provozní bezpečnost zdokonaluje. Dříve se lidé zabývali hlavně technickou stránkou. Později se safety management[1] zabýval analýzou vzniklých leteckých nehod a podle toho sepisoval pravidla a doporučení, jak předejít podobným nebezpečným situacím. Díky technologickému pokroku se dnes safety zabývá hlavně predikcí možných leteckých nehod a tím se jim snaží včas předejít.

Provozní bezpečnost se nezabývá jen nehodami letounů, ale také predikcí nehod, incidenty, které mohou této nehody vyvolat, legislativou v letectví. Aby celý „letecký systém“ fungoval bezpečně, musí být propojeny všechny složky a fungovat jako celek.

[1] Safety management - je nástrojem pro sledování bezpečnostních rizik a zlepšování výkonnosti v bezpečnosti

2. Systémové inženýrství

Každá válka za sebou nese nejenom lidské ztráty ale i prudký technický rozvoj. Druhá Světová válka nebyla výjimkou. Vědci a konstruktéři vytvářeli nové stroje a technologie, které byli mnohem komplikovanější než předchozí. A s postupem času se zjišťovalo, že systémy[2] , které vytvářely, nesplňovaly navrhované cíle. To vedlo k přestavbě systémů, a tým pádem k peněžním ztrátám a zpoždění výroby produktů.

Řešením tohoto problému byl vznik systémového inženýrství. Byl to pokus o zavedení struktury navržení projektů a poté pravidel realizace těchto projektů. Hlavním cílem bylo zlepšit výsledky inženýrské práce během vytvoření složitých systémů. Pro jednoduché systémy, tento přístup není zapotřebí. [3]

Základní myšlenka systémového inženýrství spočívá v tom, že použitím strukturovaného tvoření procesu, se dá vyvinout koncept, který má definovaný základní cíl systému a má vyznačené hranice[3] systému, omezení, která jsou následované na každém kroku, vytvoření výrobku, modelu. Technika systémového inženýrství umožňuje zaznamenat a upravit nežádoucí výsledek na každém kroku tvorby, případně se umožňuje vrátit na předchozí krok, a předejít vzniklé chybě. Toto před vznikem systémového inženýrství nebylo možné. Konstruktér měl jen možnost vyrobit produkt a až poté zjistit, zda že splňuje svůj cíl.

Nicméně požadavky (základní cíl a omezení), určené na začátku, mohou být ne vždy zcela přesně specifikované. Proto existuje možnost, že během tvorby celého konceptu v jednotlivých krocích nejenom že budou upravené vzniklé chyby, ale budou i navržené možné změny počátečních podmínek, které poté budou uskutečněné.

Do nástrojů systémového inženýrství patří také specifikace. Je zapotřebí specifikovat vlastnosti systému, jeho vstupy a výstupy, hranice, základní cíl, strukturu systému, fyzické a logické komponenty a tak dále. Nakonec je důležité stanovit sledovanost všech vazeb, vztahů mezi částmi systému. Umožňuje to zjistit, kde nastal určitý stav nebo děj. [3] [4]
Existuje spousta různých analýz, které pomáhají ke zjišťování možných problém v systému. Jedna z nich je STPA (System-Theoretic Process Analysis), která bude podrobně popsána v následujících kapitolách.

---

[2] Systém – Soubor prvků (označených jako součást systému), které společně působí jako celek, aby dosáhli společného cíle.

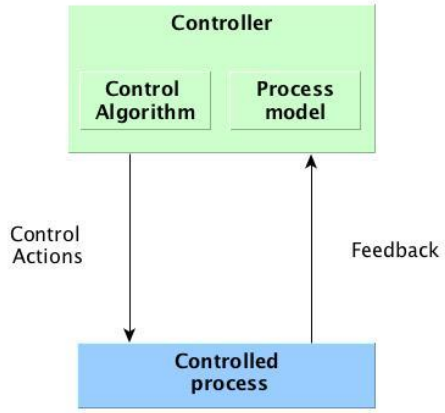
[3] V této práci pod pojmem *hranice* je myšlen anglický pojem *constraint*. Hranice ukazuje, jakým způsobem se dá dosáhnout základní cíl, popisuje, co se striktně musí udělat a co se naopak nesmí.

2.1. Koncepty „Control“[4] a „Control loop“[5] v systémovém inženýrství

Koncept řízení (control) je velmi důležitou složkou při navrhováni konstrukce a provozu jakéhokoliv systému. Pokud nebude existovat žádné řízení, lehce muže dojít k porušení hranic a omezení cíle daného systému. Typicky se toto týká komplexních systémů. Například letadlo, je sociotechnicky komplexní systém, který nesplní svůj cíl „let“, pokud ho pilot a počítač nebudou řídit. Kdežto naopak „stůl“ je jednoduchým systémem, který splňuje svůj cíl bez řídícího prvku.

Pod pojmem řízení by se nemělo chápat prosazování svých vlastních cílů a projevování své autority. Tento pojem by měl být brán v širokém smyslu. Pod pojmem „řízení“ je myšlena kontrola a podpora splňovaní základního cíle systému, údržba toho systému, podpora fail - safe[6] a jiné. Nemají být vyloučeny sociální stránky řízení, například motivace, kultura řízení. Bez zavádění takového řízení není možné zaručit spolehlivou práci systému.

Existuje velké množství řídicích smyček, které se dají použít v řízení různých systémů. V této práci je pro nás zajímavá řídící smyčka, která pochází z Feedback Control Theory (Teorie řízení se zpětnou vazbou), kterou profesorka Nancy G. Leveson, působící v Massachusettském technologickém institutu na katedře letectví a kosmonautiky, používá ve své knize „Engineering a safer world: systems thinking applied to safety.” [4][5]



Obrázek 1: Schéma základní řídicí smyčky [4]

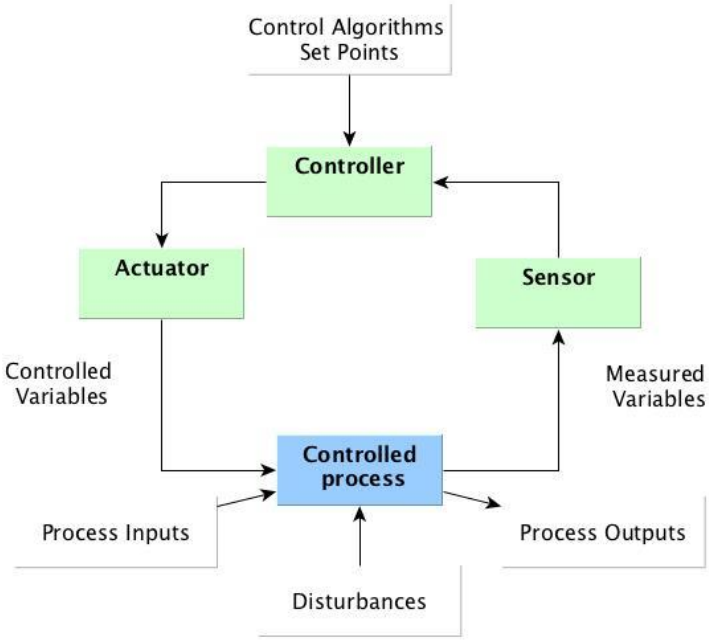
(Controller – řídicí, Controlled process – řízeny proces, Control Algorithm – algoritmus řízení, Process model – model procesu, Control Actions – řídicí čin, Feedback – zpětná informace)

[4] Control(en) – řízeni, řízeni systému

[5] Control loop(en) - řídicí smyčka

[6] Fail-safe – vlastnost technického systému, který umožňuje, že v případě poruchy některých částí systému, systém přejde do nového režimu provozu, který nepředstavuje nebezpečí pro lidi nebo životní prostředí.

Na obrázku 1 je představena schéma základní řídicí smyčky, která se dá použít jak v jednoduchém systému, tak i v komplexním. Koncept řídícího má v sobě dva významné koncepty: model procesu a algoritmus řízení. Model procesu obsahuje cíl a omezení systému. Model procesu je vždy jeden pro každý proces, jedinou výjimkou je komplexní řídící – počítač a člověk. V tom to případě člověk a počítač používají ten samý model, který je u každého změněn o malou část pro potřebu interakce dvou řídících. Takže člověk je schopen v určitých případech na základě daného modelu se chovat jinak, čímž se bude se snažit eliminovat nebezpečí. Naproti tomu počítač není tohoto schopen, je schopen jen postupovat podle algoritmu jemu předem zadaného. Algoritmus řízení obsahuje postupy, jak má proces probíhat a jaká je jeho skladba. Může v sobě obsahovat rozhodovací prvky. Pomocí algoritmu systém bude řízen. Na obrázku číslo 2 je představena standartní řídící smyčka, která má stejný princip fungování, jako základní smyčka.



Obrázek 2: Schéma standartní řídící smyčky [4]

(Controller – řídicí, Actuator – aktivní prvek řízení, Controlled process – řízeny proces, Sensor – senzor, Controlled Variables – proměnné, závislé řízeni, Measured Variables – zpětná informace, Process Inputs – vstupní informace, Process Outputs – výstupní informace, Disturbances – šum, náhodná složka)

Obrázek 2 popisuje standartní řídící smyčku, která bude používaná v této práci. Řídící dostane zpětnou informaci o stavu procesu, na základě této informace určí následující řízení, pošle informaci na aktivní prvek řízení a tím zahájí nové chovaní řízeného procesu, a poté řídicí opět dostane zpětnou informace o tom, jak probíhá řízený proces. V této smyčce je důležitým prvkem náhodná složka. Úkolem řídícího je dosažení cíle, pro který je navržen řízený proces, a pro úspěšné dosaženi tohoto cíle jsou stanovená omezení. Proto ve zpětné vazbě řídící

dostane informaci o existujícím šumu, rozhodne, jaký má šum vliv a pošle pokyny, jak ho omezit.

Důvodem, proč jsou v této kapitole podrobně vysvětleny koncepty řízení a řídící smyčky, jsou nástroje používané pro bezpečnostní inženýrství. Ve 21. století se bezpečnostní inženýři zabývají predikcí nehod. Zkoumají, zda jsme schopni zkoumat a řídit systém při každém jeho kroku, zda jsme schopni určit k jakému nebezpečí, a kde může dojít a jak jej předem eliminovat, anebo navrhnou možné či jiné chování systému, díky kterému bude nebezpečný stav eliminován. Nebezpečný stav bude předem namodelován do systému i s navrženými způsoby řešení této situace.

3. STAMP - Systems-Theoretic Accident Model and Process

STAMP (Systems-Theoretic Accident Model and Process) je model bezpečnosti založený na systémové teorii. Zabývá se příčinou nehod. Koncepce STAMP tvrdí že nehody vznikají na základě vnějšího rušení systémů, chyb komponentů systémů nebo špatné vzájemné komunikace a interakce mezi částmi systémů. K těmto problémům dochází kvůli nevyhovujícímu řízení řídicím. Příkladem je nedostačující řízení (anglicky inadequate control) anebo nedodržení bezpečnostních omezení (anglicky safety constraints) v návrhu, vývoje a v provozu systému. Proto cíl bezpečného řízení může být dosažen za pomoci zesílení omezení ještě během řízení. [6]

Model STAMP představuje rozšířený model kauzalit, ve kterém interagují mezi sebou systémové chyby, nebezpečné události, které následně vytvářejí jeden komplexní model vzniku nebezpečí. Výhodou STAMP je možnost omezit nebezpečí u kořene procesu, prozkoumat systém od základu. Dovoluje pracovat s komplexními systémy, které se skládají například ze softwaru, člověka, organizace. STAMP je model, který představuje teoretický základ pro analýzy STPA (Systems Theoretic Process Analysis) a CAST (Causal Analysis based on STAMP). [3]

3.1. CAST

CAST je analýzou příčiny nehod založenou na STAMPu. Tato metoda umožňuje zkoumat vznik nehody[7] a identifikovat příčinné faktory, které nežádoucí stavy způsobily. Pro CAST jsou charakteristické následující body definované prof. Leveson [7]:

1. Nehoda je komplexní událost, nemá žádný anebo několik základních kořenových příčin. Stejně tak i ve STAMPu, který tvrdí, že k nehoda nastane jen díky komplexu několika událostí.
2. Obviňování je hlavním nepřítelem bezpečnosti. Znalosti o tom, kdo nebo co způsobilo nehodu není dostačujícím k tomu, aby nebezpečí nebylo opakované. Vždy je zapotřebí zjistit důvod proč nebezpečny stav nastal a poté najít řešení, jak tomu předejít.
3. Lidská chyba je příznakem toho, že systém potřebuje přestavbu. Nelze změnit lidské chování bez změny samotného systému, který toto chování vyvolává.
4. Zpětné posuzováni zabraňuje studiu nehody. Pokud safety manažer bude řešit nehody z pohledu „kdyby to neudělal, tak by se to nestalo“, nenajde základ události. Zapotřebí

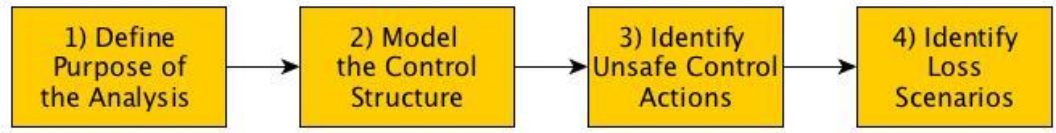
[7] Accident – nehoda, při které došlo k lidskému zranění, poškození nebo zničení techniky
Incident - událost jiná než letecká nehoda, spojená s provozem letadla, která ovlivňuje nebo by mohla ovlivnit bezpečnost provozu. [2]

zapřemýšlet proč k této situace došlo a proč řídící nenásledoval bezpečnostní omezení, nebo následované omezení nebyli dostačujícími.

3.2. STPA

STPA (System Theoretic Process Analysis) je systémová teoretická analýza procesů. Je proaktivní metodou, která zkoumá potenciální příčiny nehod během jejich vývoje, čímž umožňuje této příčiny řídit anebo eliminovat. Je to metodologie pro identifikaci a řízení nebezpečí. [4]

Analýza STPA má dva základní kroky. Pro jejich použití je zapotřebí provést dva doplňujících před kroky, které vyplívají ze systémového inženýrství a STAMPu.



Obrázek 3: Schéma základních kroků STPA [3]

(Define Purpose of the Analysis – stanovit cil analýzy, Model the Control Structure – namodelovat strukturu řízení, Identify Unsafe Control Actions – identifikovat nebezpečny řídicí čin, Identify Loss Scenarios – identifikovat možný vývoj ztrát)

Na obrázku 3 představena základní metoda STPA, která je popsaná prof. Leveson v STPA handbook viz [3].

1. Prvním krokem je **stanovit cíl analýzy**. Určit k čemu daná analýza bude použita, jaké typy ztrát touto analýzou budou řízené. Jsou stanovené takzvané základní otázky, které nám omezují rozsah analýzy. První krok můžeme rozdělit na čtyři pod kroky: identifikovat ztráty, identifikovat nebezpečí na úrovni systému, identifikovat jaké systém má omezení a protřídit nebezpečí podle toho, zda jsou vhodné pro analýzu anebo existuje možnost je zanedbat a použít v jiné analýze.
2. Druhým krokem je **namodelovat strukturu řízení zkoumaného systému**. U STPA je používaná standartní řídící smyčka, která je představena na obrázku 2.

Třetí a čtvrtý kroky jsou základní kroky používané pro analýzu STPA

3. **Identifikovat nebezpečný řídící čin**. Nebezpečný řídicí čin je řídící čin, který je řídicím v určitém kontextu a je příčinou nebezpečí. Řídicí čin se může stát nebezpečným jestli:
   3.1. Nebylo provedeno řízení potřebné pro bezpečny průběh procesu
   3.2. Bylo provedeno vědomě nebezpečné řízení
   3.3. Řízení potřebné pro bezpečný průběh procesu bylo provedené příliš brzo nebo příliš pozdě, anebo ve špatném pořadí
   3.4. Řídicí čin trval příliš dlouho nebo moc krátce
4. **Identifikovat možný vývoj ztrát**. Dovoluje to mít scénář možných ztrát, zjistit, jak nebezpečný stav může nastat. Pomocí toho dostaneme příčinné faktory, které za sebou

vedou k nebezpečí nebo ohrožení. Nástroje, které můžeme použít pro zkoumaní příčinných faktoru je několika. Jsou to:

4.1. Průzkum části řídící smyčky za účelem zjištění, zda tyto části mohou vyvolat nebezpečné řízení

4.2. Pokud existuje omezení v systému, tak je nutné ho prozkoumat, pokud neexistuje, je zapotřebí je ho vytvořit

4.3. Identifikace možných potenciálních konfliktu mezi řídicími v komplexním řízení

4.4. Pozorováni degradace navrženého řízení v průběhu času. Pro to je zapotřebí použít management změn, audit neplánovaných změn a také analýzu možných nehod.

4. Ontologie

Ontologie, jako část filosofie vznikla zásluhou řeckého filosofa Aristotela[8]. Aristoteles se ve své „první filosofii“ zabýval otázkou jsoucna a bytí. Zajímali ho otázky důvodů existence a počátky existence světa. Tyto úvahy myslitele Aristotela jsou nazývané metafyzikou. Metafyzika objasňuje dvě příčiny existence jsoucna. Prvním je existence hmoty. Mnohé věci existuji z důvodu existence hmoty, která je konečná, věčná, pasivní, inertní a nezničitelná. Počáteční hmoty jsou vzduch, voda, země, oheň a éter. Druhým důvodem je existence tvaru. Tvarem muže být nějaká podstata, stimul, cíl a příčina bytí rozmanitých objektů v jednotvárné hmotě. Aristoteles tvrdil že jsoucným objektem je spojení hmoty a tvaru. Jsou dvě příčiny vzniku spojení. Jednou je síla, která vyvolá spojení v určitou chvíli a druhou je cíl, kvůli kterému toto spojení nastane. Vyšším cílem poté je blahobyt. [8][10]

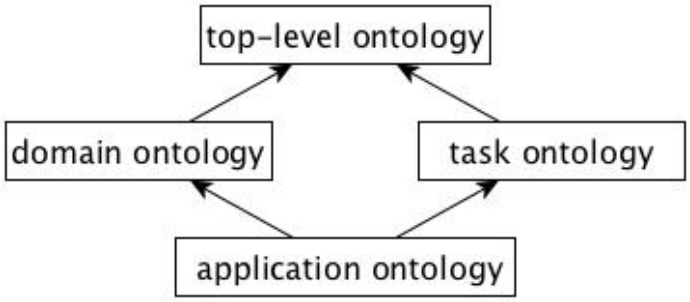
Ontologie stejně jako metafyzika je filosofická nauka o jsoucnu a bytí. Termín ontologie byl zaveden v 2. polovině 15. století německými filosofy, označili jím nauku o bytí. Velmi se rozšířil v 18. století díky německému filosofovi Christianu Wolfovi. [11] Ontologie se liší od metafyziky hlavně tím, že zkoumání jsoucna entit[9] se rozšiřuje na zkoumání vztahů mezi těmito objekty, které navzájem patří do odlišných světů. Ontologie pokládá komplexní otázky, které poté specifikují objekt nebo vztah a také vlastnosti objektů. Ontologické principy jsou využívány ve vědeckých výzkumech, například ve výběru konceptů, hypotézy, při rekonstrukci axiomů, při navrhování techniky zpracováváni informací.

Důležitým je že ontologie jako obor filosofie,která využívá takové koncepty jako část, systém, stav, vztah, proces, změna, čas a jine koncepty, které jsou navzájem shodné s těmi, co jsou využívané v informačních technologiích, zejména v konceptuálním modelování. Například, v konceptuálním modelování, jsou využívané koncepce part and whole[10], instance, klasifikace, vlastnosti, vztahy a mnohé jiné. Tato myšlenka přivedla ke vzniku aplikované ontologie, která je obecně využívána v konceptuálních jazycích. Taky jedním z pojmů ontologie je že je to teorie, která se týká druhů entit a konkrétně abstraktů entit, které mohou být zařazeny do jazykového systému. To znamená že ontologie může být použita stejně jako sémantika, a tím pádem může být aplikovaná v konceptuálních modelovacích jazycích. [12]

[8] Aristotelés ze Stageiry (384–322 př. Kr.)– řecky filosof, následník Platona, zakladatel soustavné filosofie a různých věd, například anatomie, astronomie, biologie, matematiky, fyziky a jine. [9]

[9] Entita = objekt

[10] Part and whole = část a celek



Obrázek 4: Schéma závislosti různých druhů ontologie [12]

(Top level ontology - ontologie vyšší úrovně, Domain ontology - doménová ontologie, Task ontology - ontologie úkolu, application ontology - aplikovaná ontologie)

Existuje řada klasifikace ontologií. Na obrázku 4 jsou představené některé z nich: [12]

Top level ontology (ontologie vyšší úrovně) – zabývá se obecnými koncepty, například čas, entita, událost, děj. Tyto koncepty jsou závislé na doméně modelu.

Domain ontology (doménová ontologie) – popisuje slovník obecné domény, například auto.

Task ontology (ontologie úkolu) – popisuje úkol obecné domény, například jízda.

Doménová ontologie a ontologie úkolu specifikuje prvky představené v ontologii vyšší úrovni.

Application ontology (aplikovaná ontologie) – popisuje koncepty využívané v předchozích dvou ontologiích. Ve většině případů tato ontologie používá role entit domény.

4.1. Ontologie v informačních technologiích

Zájem využívat ontologie v informatice poprvé projevil ve své práci (1967) S.H. Mealy, kde popsal základy modelování dat a definoval tři oblasti zpracovávaní informaci. To jsou: skutečný svět sám o sobě, idea, existující jen v myšleni člověka a symboly na papíru nebo paměťovém nosiči. [12] Mealy diskutoval o existenci objektu nezávisle na mnoho různých reprezentacích těchto entit.

Později ke konci 20. století vědecká veřejnost se vrátila ke studiu aplikace ontologii do informačních technologií (IT). Byla vytvořena řada konferencí na toto téma a poté definované tři oblasti k využívání ontologie v počítačových vědách. Jsou to databázové a informační systémy, softwarové inženýrství a uměla inteligence.

4.2. Ontologie v informačních systémech

Existuji tři základní vrstvy architektury pro vytvoření datového modelu: implementační, prezentační/technologická a konceptuální úroveň.

Konceptuální úroveň - Konceptuální modelování je prvním stupněm ve tvorbě datového modelu. Je návrhem systému, ve kterém jsou popsané vlastnosti prvku, které jsou součásti světa navrhovaného modelu.
Technologická úroveň - Tyká se určení hranic pro uživatele budoucích dat, ukazuje, jakým způsobem a proč data budou využívané
Implementační úroveň - Na této úrovni se řeší fyzické využíváni dat, procedury ukládáni dat

Pro vytvoření těchto úrovní byli vytvořené různé modelovací koncepce. Příkladem je logické modelování, sémantické modelování a entity-relationship model (vztahy mezi subjekty). [12]

4.3. Ontologie a doménové inženýrství

Hlavním důvodem vzniku doménového inženýrství je snížení nákladů při tvorbě softwaru[11] a pokus o vytvoření vyšší úrovně abstrakce základny každého programu. Z důvodu rychlého pokroku informačních technologií často nastává situace, že je zapotřebí použít existující program pro tvorbu nových technologií. Doménové inženýrství navrhuje vytvoření doménového modelu, do kterého budou zahrnuté základní informace, objekty a vztahy mezi nimi. Existují různé nástroje tvorby základny. Jedna z nich je ontologie. Výhodou ontologie jsou jejich koncepty entit a vztahů, ze kterých každý má definované své vlastnosti. Model vytvořený pomocí ontologie se dá snadno převést do programovacích nástrojů. [12]

4.4. Ontologie a sémanticky web.

V této kapitole je pojem ontologie je vyznačen jako formální znázornění komplexních oblastních znalostí entit, které se podílejí s ostatními entitami na zajištění interoperability[12] v inteligentních systémech. [14] Sémantickým webem je web, ve kterém jsou veškeré informace poskytované s dobře definovaným významem, pro zlepšení spolupráce člověka a počítače. [14][15]

Ontologie je výhodou pro sémantický web z pohledu předání informace počítači. Počítač není schopen porozumět dokumentu, pokud inženýr předem nedefinuje základní znalosti o pojmech využívaných v dokumentu do počítače pomoci konceptuálních jazyků.

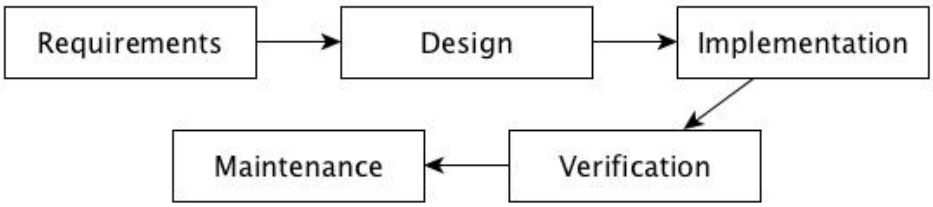
[11] Software – programové vybaveni, počítačový program

[12] Interoperabilita je schopnost různých systémů vzájemně interagovat

5. Ontologické inženýrství

Ontologické Inženýrství je soubor činností, které jsou používané při konceptualizaci, návrhu, implementaci a zavádění ontologie. Zahrnuje filozofii, metafyziku, reprezentace znalostí, metodologie vývoje, sdílení a opětovné využití znalostí, řízení znalostí, systemalizaci doménových znalostí, standardizaci, hodnocení a jiné. Soubor činnosti ontologického inženýrství je velice rozšířené. [16]

Existuje několik modelů tvorby projektu v softwarovém inženýrství, které jsou používané i při tvorbě ontologie. Jednou z nich je Waterfall Model[13] [17]



Obrázek 5: Schéma vodopádového modelu tvorby ontologie [17]

(Requirements – podmínky/požadavky, Design - návrh, Implementation - realizace, Verification – validace, Maintenance - údržba)

Na obrázku 5 je představen jeden z modelů tvorby ontologie. Ke kroku požadavky můžeme například zařadit nástroje, které budou určovat hranice ontologie. K nim patří Competency questions (otázky způsobilosti) [23]. Toto jsou jednoduché otázky z pohledu uživatele budoucí ontologie. Například *je auto schopné jezdit?,* kdybychom chtěli vytvořit ontologicky model řízeni auta. Je vhodné použití axiomů, například *toto auto má maximálně jednoho řidiče*, a také je vhodné uvést vstupy a výstupy modelu, například *řidič musí mít licenci*.

Dále jsou kroky navrhované a realizované ontologií, pro ně jsou používané různé jazyky, ve kterých jsou určené koncepce entit a vztahů. Pro validaci syntaktické části ontologie se dá využít aplikace, ve kterých jsou ontologie tvořené. Takže pro ověření ontologického modelu se dají použít různé nástroje, algoritmy a techniky. Toto je možné například porovnáním vytvořené ontologie a axiomů, které se k té ontologii vztahují. [18][19] Poté je zapotřebí hotovou ontologii udržovat, kontrolovat zda odpovídá současnému stavu světa modelu. Případně je zapotřebí daný model opravit nebo doplnit.

[13] Waterfall model – vodopádový model je sekvenční vývojový proces. Podle vodopádového modelu k následujícímu kroku muže dojit jen v případě kompletního splnění předchozího kroku.

5.1. Ontologie a konceptuální modelování

V sedmdesátých letech 20. století vznikla myšlenka použití ontologii v databázových systémech, konkrétně v datovém modelování na konceptuální úrovni. Důvodem této myšlenky je, že konceptuální modely jsou často tvořeny ER modely[14]. ER model je založen na chápáni světa jako množina entit (objektů) a vztahů mezi nimi. Nepopisuje, jaké operace budou s daty probíhat, jen poukazuje na vztahy. Ontologie má stejný cíl.

Konceptuální modelování používá koncepty entita, vztah a atributy. Entita je objekt reálného světa, odlišitelný od jiných objektů. Můžeme definovat množinu entit, je to množina entit, které sdílí stejné vlastnosti, jinými slovy řečeno – atributy. Vztah je asociací mezi několika entitami. A taktéž jako u entit, existuje i vztahová množina, kde množina vztahů téhož typu sdílí tytéž vlastnosti.

5.2. UFO a jeho použití v konceptuálním modelování

Unified Foundational Ontology anebo UFO je upper-level ontologie[15], ve které jsou nadefinované různé koncepty, které se dají využít v různých jazycích založených na ontologii. Je to univerzální ontologie nejvyšší úrovni abstrakce, která se hodí na jakoukoliv aplikaci. UFO dovoluje reprezentovat reálný svět pro následující použití v konceptuálních jazycích.

Základními koncepty UFO jsou enduranty[16] a perduranty[17]. Ontologie endurantů popisuje jen objekty, kde naopak ontologie perdurantů v sobě zahrnuje koncepty události a procesů. Můžeme říct, že ontologie vyšší úrovně se zabývá obecnými pojmy typů a jejich součástí, objekty a jejich vlastnostmi, vztahy mezi identitami a klasifikací objektů, rozdíly mezi typy konceptů a jejich vztahy, rozdíly mezi relacemi a part-whole vztahy.

UFO rozebírá každou doménu do hloubky a je schopna každý objekt v ní popsat.
UFO se rozšiřuje na pět části, každá má v sobě určité koncepty: [12]

---

[14] ER model - entity-relationship model (model objektů a vztahů mezi nimi)

[15] Upper-level ontologie = a top-level ontologie = foundation ontologie – je to ontologie, která se skládá s mnoha obecných konceptu, typu „object", „property", „relation" které jsou stejné ve všech doménách

[16] Enduranty nebo continuants, jinak řečeno objekty, substance. To jsou objekty, které mají striktně stanovené vlastnosti a jsou neměnné v čase. Například endurantem je pojem *měsíc*. V jakýkoliv čas a v jakékoliv situaci pojem měsíc je stejný, označuje zemský satelit, tvořen z určité hmoty a nemá atmosféru.

[17] Perdurant nebo occurrents, jsou to události, procesy. Perduranty existují v průběhu času. Jestli bude pozastaven čas, bude možným dostat informace jen o části celého procesu, celou koncepci určitého perdurantu se dá pochopit jen jestli se podíváme na děj v celku. Příklad je koncept konverzace. Jestli pozastavíme čas, tak můžeme vidět jen určité slovo, ale neposkytuje nám to identitu celého tématu, který se řeší v této konverzaci.

První tři jsou závislé na sobě. Patří do ní UFO – A (Endurants) – entity, UFO – B (Perdurants) – události a procesy, a UFO – C (Social Reality) – sféra mezinárodních a sociálních věci, včetně jazyků. Čtvrtá a pátá jsou UFO – S (Services) – ontologie služeb a UFO – L (Legal Relations) – ontologie zákonů.
UFO – A poskytuje konceptuálnímu modelovaní některé koncepce entit, strukturu taxonomií, part-whole vazby, zahrnuje vnitřní vlastnosti, atributy, a vlastnosti obecného modelu, takže zahrnuje vlastností a charakteristiky vazeb, vztahů a role objektů.
UFO – B vlastni koncepce událostí a procesů, popisuje jaké objekty se zúčastnili toho procesu nebo děje, důvody existence dějů, jejich změny a jak jsou spojeny enduranty a perduranty přes jejich povahu.
UFO – C nakonec řeší sociální specializované aspekty jako víra, naděje, touha, cíl, akce, nároky a jiné.

5.3. Aplikace UFO v modelovacím jazyce OntoUML

Doktor Guizzardi [20] ve své disertační práci popsal možnost aplikace UFO v modelovacím jazyce UML 2.0[18]. Tato idea zaujala vědce, a proto vznikl nový modelovací jazyk – OntoUML. Důvodem volby UML pro aplikaci UFO se stala komplexnost metamodelu[19] jazyku UML a jeho časté používání k modelování komplexních systémů komunitou informačních technologií. UML umožnuje rychlý rozvoj ontologii v IT, protože má hodně uživatelů, to znamená že jsou lidi s tímto jazykem seznámené, má mnoha základních modelů pro modelování, a téyo modely jsou podobné ontologii.

Samostatnou výhodou je fakt, že na podporu využívání jazyka OntoUML jsou vytvořené různé aplikace, ve kterých uživatel může namodelovat různé ontologické modely, poté zkontrolovat, zda jsou syntakticky korektní a v integrovaném nástroji pro vizualizaci, resp. tvorbu instancí, Alloy Analyzer model validovat po obsahové stránce. Příkladem takových aplikací je Menthor a OLED (OntoUML lightweight editor). Alloy je obecný modelovací jazyk, který dovoluje vyjádřit syntaxe a sémantiku informace, což umožňuje jeho kompatibilitu s UML (OntoUML). [22] Takovéto funkcionality jsou v oblasti modelovacích jazyků, resp. ontologií zcela jedinečné.

OntoUML je v teto chvíli schopen využívat jen koncepty z UFO – A. V jazyce OntoUML je každá třida reprezentovaná pomoci boxů, kde každý box má svoje jméno (název třídy) a stereotyp (koncept z ontologie UFO). Syntax OntoUML bude podrobněji popsána dále.

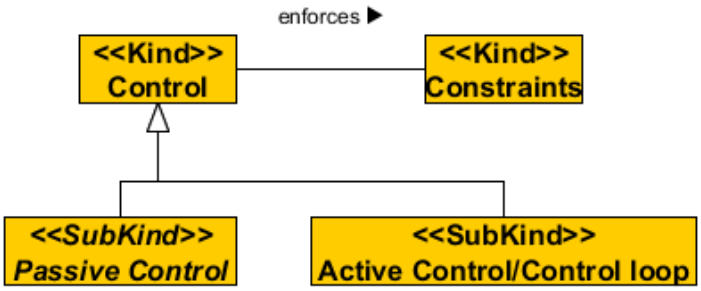
---

[18] UML (Unified Modeling Language) je grafický jazyk používaný ve softwarovém inženýrství pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů. Používá objektově orientovanou metodiku. [21]

[19] Metamodel – je souhrn omezení a pravidel tvorby modelů.

6. Popis modelu

Cílem této práce bylo zvolit klíčové části modelu STAMP a modelovat je pomocí ontologie.



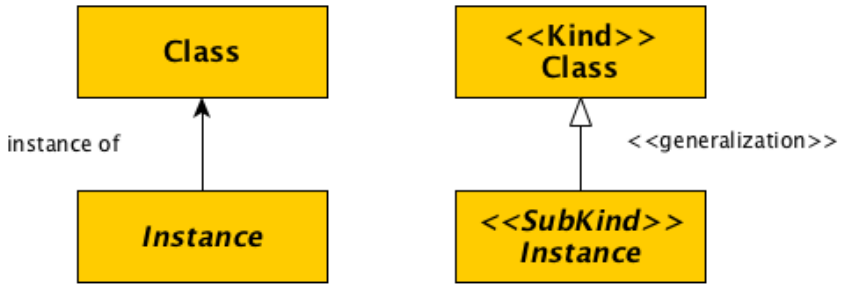
Obrázek 6: Schéma obecného modelu řízení s použitím ontologií

(Control – řízení, Constraints – omezení, Passive Control – pasivní řízení, Active Control/Control loop – aktivní řízení/řídící smyčka, enforce – prosadit, vynutit)

Na obrázku 6 je znázorněno obecné schéma modelu řízení. Rozlišuje, že řízení (control) se dělí na aktivní a pasivní. Pasivní řízení je to, které zajištuje bezpečnost pomocí své přítomnosti, jsou to pevné hmotné objekty, například svodidla na dálnici, které zabraňují vyjetí auta ze silnice. K pasivním prvkům řízení taktéž patří gravitace a jiné fyzikální zákony. Aktivní řízení je shodné s řídící smyčkou. Aktivní řídicí prvky vyžadují některá opatření k zajištění bezpečnosti, například detekce nebezpečné události nebo stavu (sledování), měření některých proměnných, interpretace měření (diagnóza) a odezva (postupy pro obnovení procesu nebo provádění procedur pro podporu bezpečného stavu), které musí být provedeny před vznikem nebezpečí. Jak bylo uvedeno v kapitole 3 a označeno na obrázku 6, každé řízení by mělo prosazovat (enforce) požadavky na systém (contraints). Například během řízení konkrétního procesu jsou sledovány konkrétní parametry a typicky je daná jejich limitní přípustná hodnota (např. bezpečná vzdálenost vozidla od okraje vozovky), dodržení které je ale v kompetenci konkrétního řízení. S ohledem na zmíněné, v kontextu této práce chápeme prvek Active Control jako ekvivalentní s konceptem Control loop.

Model STAMP je základem pro analýzy CAST a STPA. CAST a STPA používají pro své kroky prvky řídicí smyčky, proto se samotná smyčka jeví jako klíčová a vhodná pro modelování pomocí ontologií. Analýzy CAST a STPA jsou ve mnohém velice podobné, pro potřeby této práce bude proto modelována jenom jedna z nich, a to konkrétně analýza STPA.

6.1. Koncepty definované v UFO – A a UFO – B



Obrázek 7: Schéma použiti tříd a instance v UML do a po aplikace UFO – A

(Class – třida, Instance – objekt/instance, Instance of – je instancí, Generalization – specifikace, Kind a SubKind[20] - stereotypy z UFO - A)

Jazyk UML používá při modelování třídy (class), které sdružují množinu společných vlastností konkrétních objektů. Třídy můžeme specifikovat pomocí instancí (objektů). Toto je graficky znázorněno na obrázku 7 vlevo. Na obrázku 7 vpravo je ukázáno použití konceptů z UFO v OntoUML. OntoUML taktéž jako UML je založen na třídách, ale navíc k nim přiřazuje stereotypy s vlastním, doplňujícím významem.

Před vysvětlením značeni stereotypů ukázaných na obrázku 7 je zapotřebí probrat základní vlastnosti, podle kterých se dají třídit stereotypy použitých konceptů z UFO – A.

OntoUML zatím používá jenom koncepty z UFO – A. Každý stereotyp má své vlastnosti. První vlastnost, podle které se dají rozdělit stereotypy je Rigidita. Třida je **rigidní**, jestli v jakémkoliv světě a v jakémkoliv čase je schopna zachovat všechny své vlastnosti neměnné, to znamená, že v každém světě klasifikovaný objekt bude ten samý objekt. Do rigidních stereotypů patří: „Category“, „Collective“, „Kind“, „Mode“, „Quality“, „Quantity“, „Relator“, „SubKind“. **Anti-Rigidní** stereotypy jsou ty, které se mění v čase anebo se mění v závislosti na situaci napříč všemi různými světy. Například člověk je rigidní typ, protože kdykoliv bude použita třída *člověk*, každý čtenář bude mít na mysli stejné vlastnosti a *člověk* nemůže v jiném světě mít vlastnosti třídy *auto*. Člověk může být nějakou dobu student, nějakou dobu mít světlé vlasy, anebo nějakou dobu být vdaný. Tedy třídy *Student, Světlé vlasy* a *Vdaný* jsou anti-rigidní, protože člověk je studentem, pokud má uzavřenou smlouvu se školou, má světlé vlasy, pokud je neobarví do jiné barvy a je vdaný, pokud není rozvedený. Anti-Rigidní stereotypy jsou „Role“, „Phase“ a „RoleMixin“. [12]

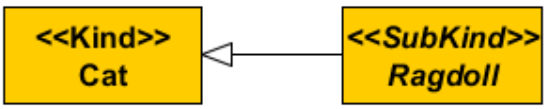
[20] Názvy stereotypů použité v této práci budou použité v anglické terminologii. Většina z terminů UFO nemá česky ekvivalent a jejich překlad muže být matoucí.

Druhou vlastností, podle které se dá rozdělit stereotypy, je identita. Identita je to, co liší jeden objekt od jiného. Pokud objekt poskytuje princip identity, můžeme určit, zda je to přesně ten objekt, o kterém se mluví. Například každý člověk má svůj otisk prstu, nebo duhovku v očích, podle kterých se dá identifikovat, že je to ten konkrétní člověk. Nebo pojem Kočka taky poskytuje identitu pro pojem domácí mazlíček. Pak existující stereotypy z UFO – A můžeme rozdělit na ty, které poskytují identitu pro své instance, jako jsou „Kind“, „Collective“, „Quantity“, „Relator“, „Mode“, „Quantity“. Pak jsou ty, které neposkytuji identitu, ale používají společný princip identity od svého rodiče (nadřazené třídy) a jsou to „Subkind“, „Role“ a „Phase“. A nakonec jsou ty, které neposkytují identitu a jejich instance následují různé principy identity: takové jsou „RoleMixin“, „Mixin“ a „Category“.

V této práci je také použit koncept „Event“, který má snahu popsat události z reálného světa a je popsán v UFO – B ale není definován v OntoUML. „Event“ je ústřední koncept UFO-B a může být použit jak samostatný (atomic), tak i komplexní (complex), dle toho, zdali může být redukován na menší „pod-události“. „Complex event“ se dle podmínek skládá minimálně ze dvou „Atomic event“. „Event“ je vždy závislý na objektech, které se v něm účastní a je specifický tím, že během jeho realizace dojde k nějaké změně (ve stavech objektů nebo v situaci atp.). Tedy běžné třídy popisující procesy nebo události se dají modelovat s pomocí stereotypu „Event“. Důležité je zmínit, že aby bylo možné vázat třídy se stereotypem „Event“ v konceptuálním modelování s některými UFO – A koncepty, je zapotřebí koncept „Event“ v této práci chápat jako „*Event type“*, jelikož UFO-A pracuje jenom s enduranty a typ události je ze své povahy endurant. Pod pojmem typ (type) se chápe, že třídy nepopisují konkrétní událostí a objekty, ale jen sdružují objekty nebo události, které mají stejné vlastnosti. [24] Pod pojmem typ události („Event type“) tedy nechápeme konkrétní událost v čase a prostoru, nýbrž pouze vzorec (tzv. pattern v angličtině) události, který může mít libovolný počet instancí (pak již konkrétních událostí). Takovýto endurant je možné modelovat s využitím běžných OntoUML stereotypů a dle pravidel OntoUML jej vázat na ostatní koncepty v tomto jazyce. Tato práce pracuje pouze s typy objektů (tzv. univerzály dle terminologie UFO) a tedy nepracuje se žádnými instancemi. Tato skutečnost nebude v samotných modelech z praktických důvodů zdůrazňována.

Jedním ze základních stereotypů tříd v OntoUML je „Kind“ (použit na obrázku 7), který je specifický tím, že poskytuje svým instancím identitu a že je to rigidní koncept. Tento stereotyp přiřazujeme tedy třídám, kterých instance (objekty) splňují takové vlastnosti. Přiklad použití stereotypu „Kind“ je představen na obrázku 8. „Kind“ může být třída *kočk*a (obr. 8), která zachovává identitu všem svým instancím a ve všech světech. Kočka se nemůže během své existence stát ničím jiným a každou kočku jsme teoreticky schopni odlišit od jiných koček. Tedy

kromě standardního sdružení vlastností koček (jejich velikost, barva, hmotnost, vzhled či vydávání specifických zvuků), které jsou obsaženy v použití třídy Cat (obrázku 8).Je zde také použit stereotyp „Kind“, který nám blíže specifikuje další vlastnosti této třídy (poskytování identity a rigiditu konceptu).



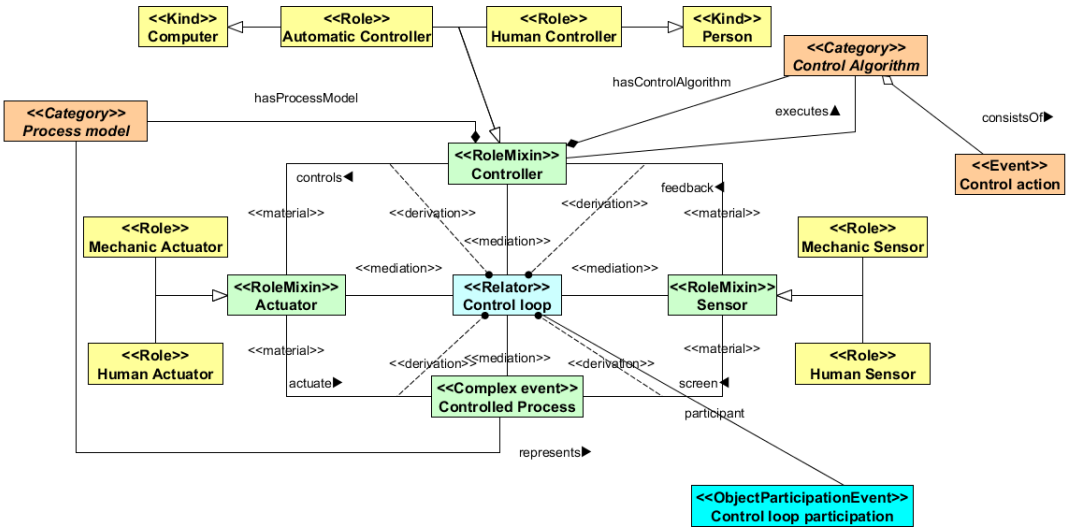
Obrázek 8: Příklad použití stereotypů „Kind“ a „SubKind“

(Cat - kočka

Dále každou třídu, která dědí všechny vlastnosti rodičovské třídy, i má několik vlastností navíc, můžeme klasifikovat pomoci stereotypu „SubKind“. Například vezmeme druh koček Ragdoll a můžeme tvrdit, že Ragdoll je poddruhem typu Kočka, nebo jak je vyznačeno na obrázku 8, rodičovskou třídou pro Ragdoll se stereotypem „SubKind“ je třida Cat se stereotypem „Kind“. [13] Platnou vazbou, kterou se tyto koncepty spojují již z UML je specifikace/generalizace zobrazená prázdnou šipkou při rodičovském konceptu. Steretoyp „SubKind“ z pohledu ontologie UFO říká, že princip identity (tedy na základě čeho od sebe odlišíme instance zmíněné třídy) se dědí od rodičovského konceptu (kočka).

6.2. Model Control loop

Control loop vznikl jako první konceptuální model na zaklade ontologie UFO a s využitím jazyka OntoUML. Prvky pro modelování byli použity z obrázku 2 a informace pro identifikaci vlastností těchto prvků byli použity z teorie modelu STAMP.

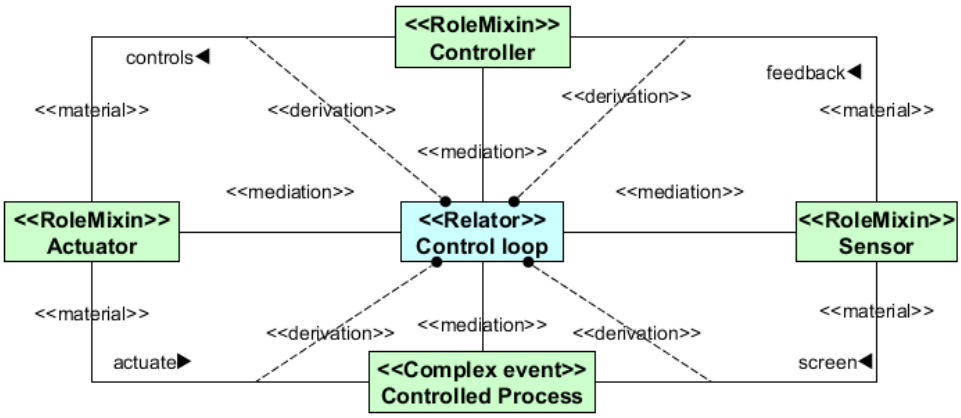


Obrázek 9: Schéma ontologického modelu Control loop

(Computer – počítač, Automatic Controller – automatický řídicí, Human Controller – lidský řídicí, Person – člověk, Control action – řídicí akce, Mechanic Actuator – mechanický aktivní řídicí prvek, Human Actuator – lidský aktivní řídicí prvek, Mechanic Sensor – mechanický senzor, Human Sensor – lidský senzor, Control loop participation – účast řídicí smyčky v dalších procesech, HasProcessModel – má model procesu, HasControlAlgorithm – má algoritmus řízení, ConsistsOf – skládá se z, Execute – provádí, Feedback – zpetna vazba, Control – řídit, Actuate – aktivovat, Screen – zkoumat, Participant – účastník, Represent - představovat)

Na obrázku je 9 představen ontologický model Control loop, který vznikl v této práci. Tyrkysovou barvou uprostřed modelu je ukázaná třida Control loop[21]. Zelenou barvou jsou znázorněny čtyři hlavní účastnici řídicí smyčky. Žlutou barvou jsou vyznačeny třídy objektů, které specifikuji třídy řídící, senzor a aktivní prvek řízení. Oranžovou barvou jsou vyznačeny řídicí algoritmus a model procesu jako součásti řídicího, díky kterým řídí řízený proces. Modrou barvou jsou znázorněny třídy událostí popsané pomocí UFO – B stereotypů a spojující model Control loop s STPA.

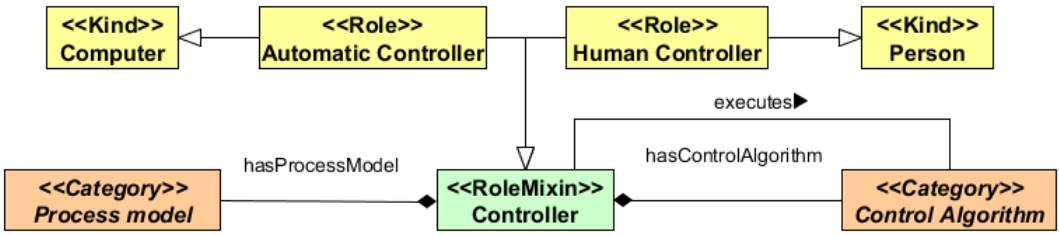
[21] Všechny třídy použité v modelu Control loop a modelu STPA zdržují obecné vlastnosti objektů této třídy. Třídy v této práci nemají modelované žádné instance.



Obrázek 10: Schéma ontologického modelu základní řídicí smyčky (Control loop)

(material – materiální vazba, derivation – derivace, mediation - mediace)

Na obrázku 10 je představen základní model řídicí smyčky. Ve středu modelu se nachází třída reprezentující Control loop, se stereotypem „Relator“. Stereotyp „Relator“ je rigidní a poskytuje identitu, ukazuje tvůrce vztahů mezi dvěma a více objekty. V případě Control loop, je prvkem, který vytváří vlastnosti, díky kterým se dá spojit dohromady řídicí, aktivní prvek řízení, senzor a řízený proces. Jinak řečeno, Control loop je základním elementem tvořícím vztah mezi těmito prvky. Často se s „Relatorem“ pomocí vazby mediace spojují stereotypy „Role“. „Role“ se používá jako reprezentant anti-rigidní specializace stereotypů, které poskytují identitu. Například třídě *člověk* můžeme přiradit stereotyp „Kind“ a spojit generalizaci s třídou *student*. Toto nám ukáže, že každý student je člověkem a také to, že člověk může být v nějakém časovém rozmezí studentem. Vazby použité v tomto modelu jsou tři: derivace, mediace a materiální vazba. Vazba mediace se dle ontologie UFO-A vždy používá pro spojení „Relatora“ a objektu, který je součástí modelu díky vztahům, které vytváří „Relator“. Materiální vazba popisuje materiální vztah mezi entity spojené „Relatorem“. Derivace poskytuje materiální vazbě definovanou informaci, hranice, vlastnosti, kterých se dá použít v materiální vazbě na základě identity poskytované „Relatorem“.



Obrázek 11: Schéma ontologického modelu řídicího (Controller)

Pro prvky Actuator, Sensor a Controller byli vybrány stereotypy RoleMixin z toho důvodu že do sebe zahrnují instance s různými principy identity, ale nejsou rigidní. Například člověk,

který je v čase t_1 řídicím, nemusí byt řídicím v čase t_2. „RoleMixin“ je definována pomocí „Role“, přebírající identitu od stereotypů „Kind“:

Na obrázku 11 stereotypy „Rolemixin“ a „Role“ přebírají princip identity u konceptů se stereotypem „Kind“. „RoleMixin“ *Controller* je definována dvěmi „Role“, které přebírají různé identity u svých rodičovských tříd. Jsou to třídy *Automatic Contoller* a *Human Controller*. Automaticky řídící je v této práci definován jako počítač, v případě aplikace tohoto modelu na reálnou situaci počítačem může být jakýkoliv mechanický prvek schopný řídit proces, nebo souhrn těchto prvků. Stejným principem na obrázku 9 jsou specifikované „RoleMixin“ stereotypy tříd aktivního řídicího prvku a senzoru. V případě třídy *Human Actuator* a třídy *Human Senzor* je myšlená verbální komunikace člověka. Například zrak jako senzor a hlas jako aktivní řídicí prvek.

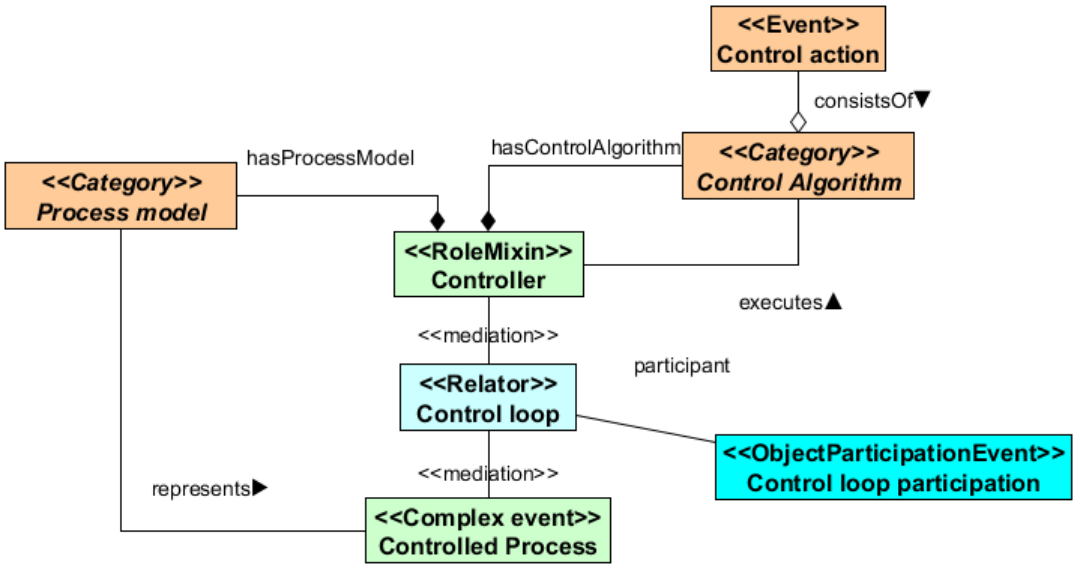
Zajímavou časti obrázku 11 jsou třídy *Process model* a *Control Algorithm* se stereotypem „Category“. Stereotyp „Category“ se používá k agregaci vlastnosti individuálů, kteří sledují různé principy identity a jsou rigidní. V případě modelu procesu, který obsahuje různé proměnné a může být implementován jako software, nebo se může jednat o model procesu, který je integrální součástí řídícího člověka (*Human Controller*). Algoritmus řízení obsahuje postupy průběhu procesu a jeho skladbu. Může taky obsahovat rozhodovací prvky, a jeho implementace se také může v různých světech lišit podobně jako u modelu procesu.

Controller je endurantem, který má různé principy identity, které mu poskytuje *Computer* a *Human*. Podle teorie systémového inženýrství, každý řídicí, buď člověk, nebo počítač, obsahuje model řízeného procesu i algoritmus řízení. Proto pomocí pevné (kosočtverec je zabarven černou barvou) *component of* vazby identifikujeme, že řídicí, (počítač, člověk nebo komplexní řízení pomocí počítače a člověka), vždy obsahuje model procesu a algoritmus. A navzájem model procesu a algoritmus jsou rigidní, protože budou vždy stejné a budou se vztahovat k určitému řízenému procesu bezezměnný v závislosti na čase.

Na obrázku 12 jsou znázorněné dvě důležitý vazby. První je spojení vazbou asociace stereotypu „Category“ třídy *Process model* se stereotypem „Complex event“ třídy *Controlled proces*. Vazbou pomocí orientované asociace ***představuje(represents)*** je myšleno, že model procesu představuje řízený proces a že je jeho zjednodušenou reprezentací.

Mezi třídou *Control loop participation* a *Control action*, je můstek pro spojení modelu Control loop a STPA. *Control loop participation* je účastník (participant), vznik kterého je podmíněn

potřebou navázání *Control loop* do objektu, který je účastníkem událostí, v které se řídicí smyčka vyskytuje.

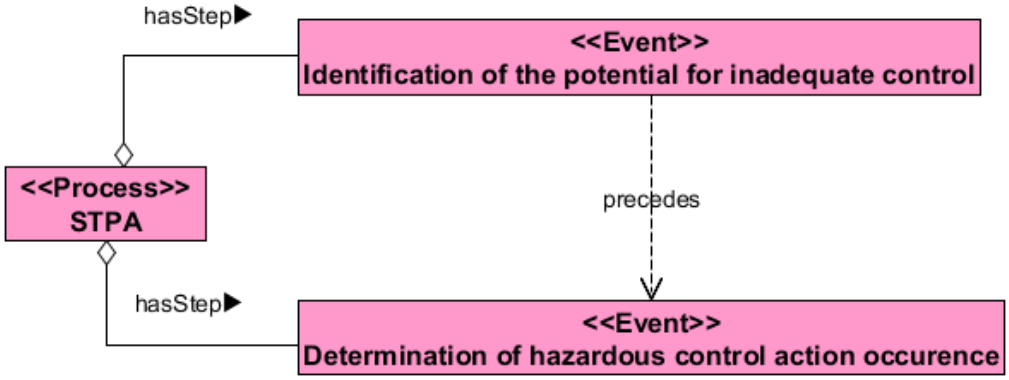


Obrázek 12: Vztah modelu procesu vůči řízenému procesu a rozhraní řídicí smyčky na analýzu STPA

6.3. Model STPA

Podobným způsobem jako u řídící smyčky byl vytvořen ontologicky model STPA, který je ale z praktických důvodů představen jako celek v příloze 3. Tato kapitola pojednává pouze o jeho klíčových částech, které objasňuje. Taktéž pro alternativní zobrazení STPA, byl vytvořen diagram aktivit představený na obrázku 17, jelikož se jedná o proces a ten je vhodné v jazyce UML reprezentovat pomocí diagramu aktivit spíše než pomocí diagramu tříd. Konceptuální modelování v OntoUML umožňuje jen tvorbu diagramů tříd ale v případě analýzy, která je dynamická, je vhodné použít alternativní zobrazení dle UML, které tvorbu dynamických modelů umožňuje. [21]

V příloze 3, růžovou barvou jsou vyznačeny hlavní kroky analýzy STPA. Modrou barvou následující kroky, kterými se tato analýza provádí a její komponenty. Červenou barvou jsou vyznačeny nebezpečné prvky, které si tato analýza klade za cíl identifikovat. Tato barevná anotace také odpovídá obrázkům v této kapitole.



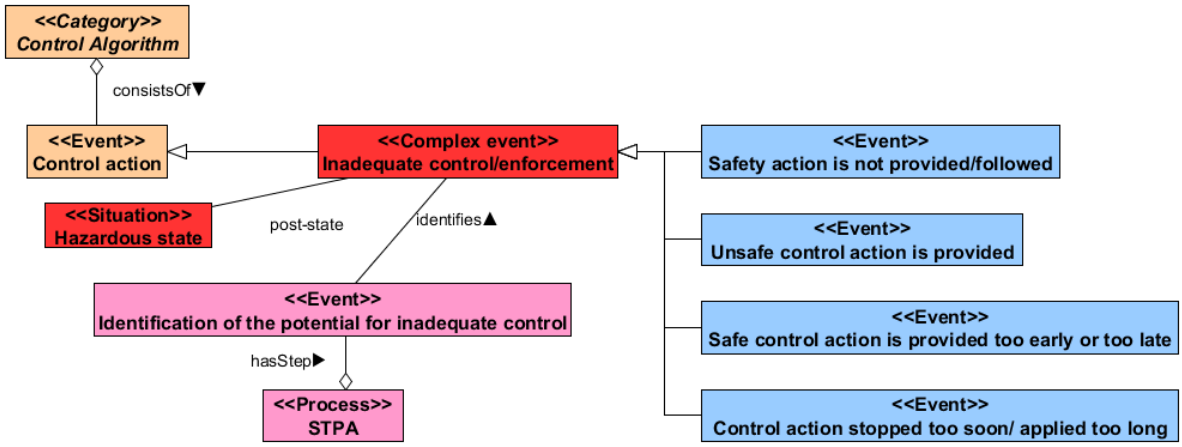
Obrázek 13: Schéma ontologického modelu základních kroku analýzy STPA

(Identification of the potential for inadequate control – identifikace potenciálního nevhodného řízeni, Determination of hazardous control action occurrence – určení vzniku nebezpečných řídicích akcí, HasStep – má krok, Precedes - předchází)

Na obrázku 13 je znázorněn model základních kroku analýzy STPA. Třida *STPA* využívá stereotyp „Process“ protože analýza je posloupnost kroků pro dosažení konkrétního cíle. Pomocí *component of* vztahů s orientovaným popisem pomocí vazeb *hasStep,* je v této práci ukázáno, že třídy kroku identifikace a kroku určení jsou součástí třídy analýzy STPA, což znamená, že mají identitu třídy STPA. Zároveň to ukazuje, jakým způsobem interagují tyto třídy, že jsou to uspořádané kroky této analýzy. Čerchovanou směrovanou čarou je vyznačena logika následování těchto kroků. Krok identifikace vždy předchází kroku určení.

Ze dvou kroků STPA, bude podrobně popsán pouze krok 1., protože využití stereotypů a vazeb v těchto dvou krocích je velmi podobné. V příloze 3 je model STPA, včetně kroku 2.

Jak je to vidět na obrázku 14, krok 1. analýzy STPA je závislý na třidě nebezpečný stav. Tato třida má stereotyp „Situation“, a ukazuje nám pomocí vazby s jménem post-state (stav posle). Stereotyp „Situation“ se používá pro označení situace tak, jak ji chápeme v přirozeném jazyce a která je v ontologiích specifická tím, že je statická: participují v ní objekty, které mají stálý stav. Jedná se tedy o jakousi momentku (snapshot) části světa. Stereotyp „Event“ reprezentuje událost schopnou změnit tuto momentku z jednoho stavu do druhého, a tím od sebe oddělit dvě situace. UFO – B umožňuje pomocí specifikované vazby označit stav, který byl před průběhem událostí a po skončení událostí. Na obrázku 14 vidíme že situace s nebezpečným stavem nastane jako následek po existenci události nevhodného řízení.



Obrázek 14: Schéma ontologického modelu kroku 1. analýzy STPA

(Control algorithm – algoritmus řízení, Control action – akce řízení, Hazardous state – stav nebezpečí, Inadequate control/enforcement – nevhodné řízení, Safety action is not provided/followed -bezpečná řídicí akce nebyla provedena, Unsafe control action is provided – nebezpečná řídicí akce je provedena, Safe control action is provided too early or too late – bezpečná akce řízení byla provedena příliš brzy nebo příliš pozdě, Control action stopped too soon/applied too long – Akce řízení byla zastavená příliš brzy nebo byla aplikovaná příliš dlouho)

Třida *Inadequate control* má stereotyp „Complex event“, protože nevhodné řízení je souhrn několika typů událostí, v případě zobrazeného modelu, souhrn čtyř typů událostí v pravé části obrázku se stereotypem „Event“. Vazba generalizace ukazuje, že všechny tyto události mají identitu nevhodného řízení, resp. že jsou jeho podmnožinou (specifikací). Podobně tak třída nevhodného řízení má identitu třídy *Control action* se stereotypem „Event“ (tedy nevhodné řízení je podmnožinou řízení, resp. jeho akcí), kde ten je součástí algoritmu řízení. Nakonec mezi růžovým boxem identifikace a červeným boxem nevhodného řízení existuje vazba s názvem identifikuje co znamená, že během kroku identifikace se provádí identifikováni nevhodného řízení.

Stejně je to s krokem 2. V modelu v příloze 3 jsou vyznačeny kroky, které jsou součástí kroku určení důvodu vzniku stavu nebezpečí a je ukázáno, jaký krok, kterému předchází a který krok se kterým probíhá společně. Důležitou částí je rozhodovací logika u kroku průzkumu existujících omezení, kde je určeno, zda se provádí proces dle třídy navrhování omezení anebo třidy průzkum existujících omezení.

6.4. Aplikace modelu STPA na model Control loop

Modely STPA a Control loop jsou navzájem spojeny ve dvou místech. První je třida *Control action* se stereotypem „Event“ a třída *Control Algorithm* se stereotypem „Category“, kde *compoent of* relace ukazuje, že řídicí akce je součástí algoritmu řízení. A druhou je vazba mezi

třídou *Control loop* se stereotypem „Relator“ a třídou *Control loop participant* se stereotypem „ObjectParticipationEvent“ napojených vazbou specifikovanou jako účastník (participant). Vznik třídy *Control loop participant* je podmíněn potřebou navázání *Control loop* do objektu, který je účastníkem událostí, ve které se řídicí smyčka vyskytuje.

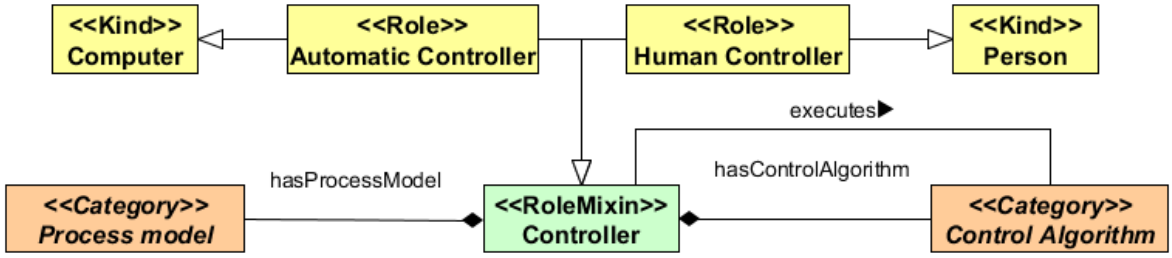
7. Validace modelu

7.1. Competency question

Jak již bylo uvedeno v kapitole 5, otázky způsobilosti jsou používané v prvním kroku tvorby ontologického modelu. V této práci budou tyto otázky použity jako jeden z nástrojů vyhodnoceni modelu. Byli navrhnuté následující otázky, které jsou představeny v tabulce 1.

Tabulka 1: Seznam otázek způsobilostí v anglickém jazyce a jejich český ekvivalent

anglicky		česky
Competency Questions STPA		**Otázky způsobilostí STPA**
1.	Which parts does the Control loop contain?	Jaké části sdružuje řídicí smyčka?
2.	What kinds of the Controller could be?	Jaké druhy řídicího mohou být?
3.	What does the Process model represent?	Co reprezentuje model procesu?
4.	Which Control action can lead to the Hazardous state?	Jaké akce řízení mohou vyvolat stav nebezpečí?
5.	What are the main steps od STPA?	Jaké jsou hlavní kroky STPA?
6.	How can we considerate of degradation of control actions?	Jak můžeme uvažovat o degradaci akce řízení?
7.	How can we determine occurrence of Hazardous state?	Jak můžeme určit příčiny vzniku stavu nebezpečí?



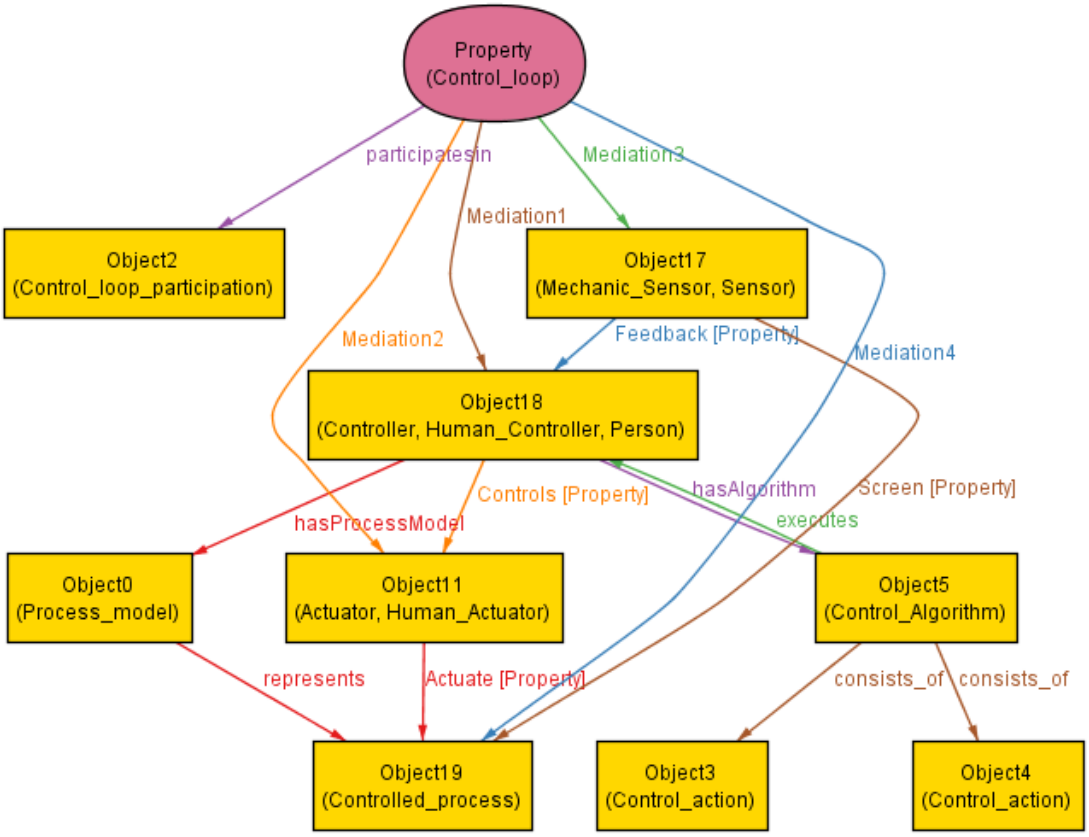
Obrázek 15: Schéma ontologického modelu odpovědí na otázku způsobilostí číslo 2

Odpovědi na představené otázky byli modelovány pomocí ontologie a zkontrolované syntakticky a za pomocí vizualizace Alloy v aplikaci Menthor. Pro příklad je na obrázku 15 zobrazen model otázky číslo 2. Validace byla provedena pomocí konceptuálního modelování odpovědí na tyto otázky s využitím ontologií. Poté byly porovnávány modely odpovědí na tyto otázky s celým modelem vzniklým v této práci. Validace v tomto ohledu dopadla pozitivně,

všechny modely byli identické s částmi vzniklého modelu. Model můžeme z tohoto pohledu tedy považovat za validní.

7.1. Vizualizace Alloy

Druhým způsobem validace je vizualizace Alloy. Alloy vygeneruje instance světa v určitém čase *t.* Tyto instance generuje z modelu tříd, který je vytvořen. Úkolem je vygenerovat pomoci Alloy několik světů a určit, zda taková situace může v reálném světě nastat nebo ne.



Obrázek 16: Schéma vygenerovaného světa s instancemi ontologického modelu Control loop

Na obrázku 16 je znázorněn jeden z vygenerovaných světů instance modelu Control loop. Na tomto obrázku žlutými boxy jsou vyznačeny objekty (instance), které jsou unikátní a existují jen v tomto světě. Růžovým oválem je znázorněna vlastnost. Barevnými šipky jsou vyznačeny vazby mezi objekty.

Contol loop je v Alloy zobrazen jako vlastnost (property) a je to tím, že Control loop ukazuje důvod nebo vlastnost, díky které jsou objekty s vazbou mediace spojeny. Alloy vygeneroval instance Control loop s vazbou *je účastníkem* (participates in) vůči objektu *Control loop participation.* Toto je pravda dle kapitoly 6.4. Dále je znázorněno jak jsou za pomocí vazeb mediace (na obrázku 16 „Mediation1“, „Mediation2“ a „Mediation3“) spojeny objekty: osoba, která je člověk a která je řídicí, lidský aktivní prvek řízení, mechanicky sensor a řízený proces.

Je to taktéž správně, protože je známo, že v modelované řídicí smyčce musejí tyto čtyři objekty existovat současně.

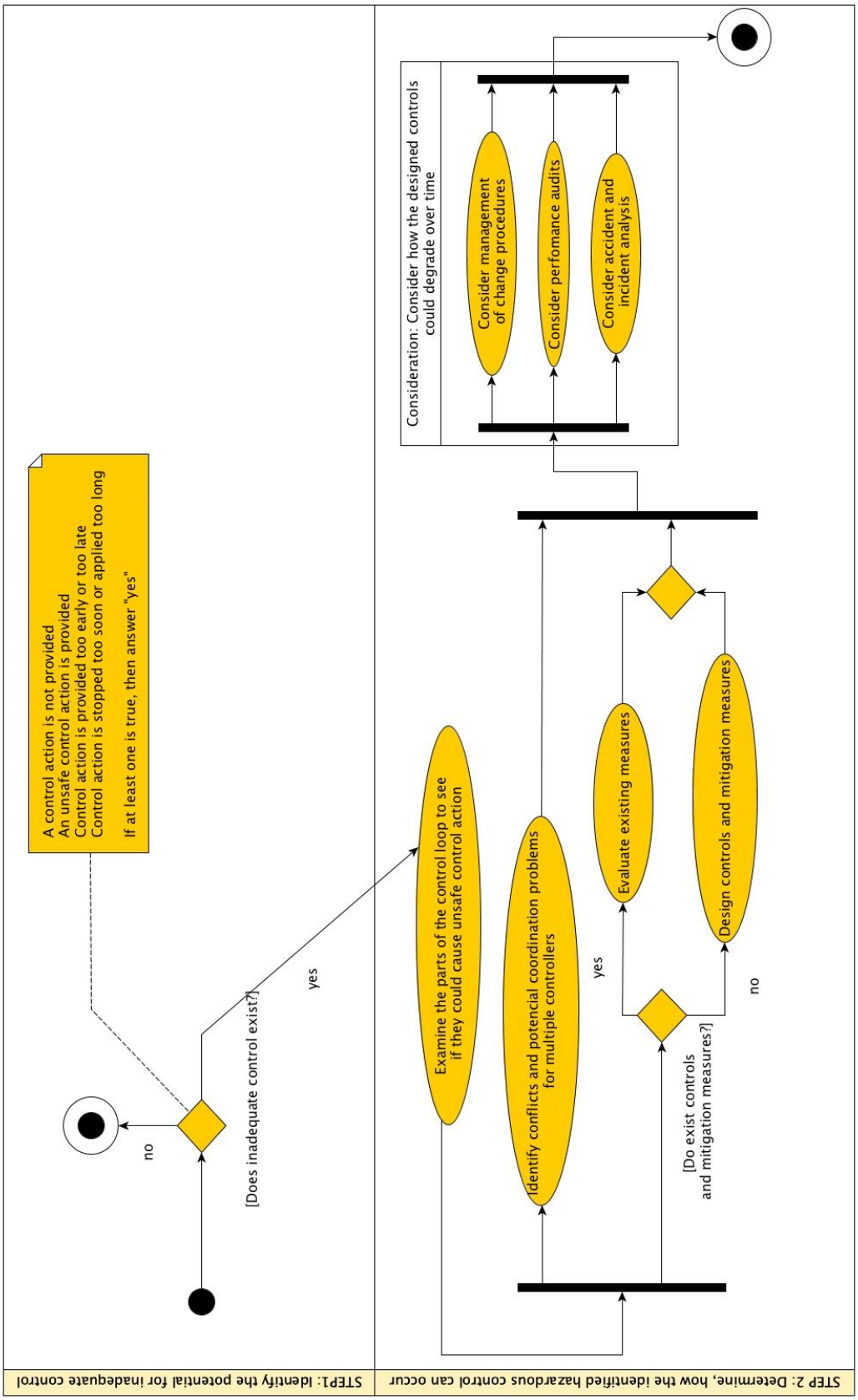
Objekt 18 – člověk řídicí, má nejvíc vazeb. Vidíme, že řídicí má algoritmus objekt 5, který v sobě zahrnuje dvě akce řízení (objekt 3 a objekt 4), a který tento řídicí uplatňuje. Řídicí taktéž má model procesu, který reprezentuje řízený proces (objekt 0). Řízený proces je monitorován senzorem (objekt 17) a usměrňován (actuate) aktivním prvkem řízení (objekt 11, Human actuator myšlený jako verbální komunikace), který je řízen řídícím, který pracuje na základě zpětné vazby od senzoru.

Podobným způsobem byl zkontrolován model STPA. V příloze 5 se nachází jeden z vygenerovaných světů. Tato instance je zejména zajímavá proto, že obsahuje dvě instance řídící smyčky. V tomto světě STPA identifikuje nebezpečný stav, který nastal kvůli akci řízení, které je nevhodné, ale nenabízí varianty nevhodného řízení. Byl však kompletně vygenerován krok 2, podle kterého se dá identifikovat důvod vzniku takového řízení.

V obou případech validace proběhla úspěšně, všechny objekty a vazby ve všech světech bylo možné pro existenci v reálnem světě, proto model můžeme považovat i z pohledu validace v jazyce Alloy za validní.

7.2. Diagram aktivit v UML

Jako třetí způsob validace jsme použily diagram aktivit UML. Vizualizace Alloy dovoluje vygenerovat tři světy najednou. Proto byli vygenerovány současně tři světy STPA, a zkoumané pomoci porovnávání diagramu aktivit UML STPA a modelu Alloy, zda vygenerované vazby Alloy mají stejné pořadí jako vazby v diagramu představeném na obrázku 17, resp. zda tyto instance zobrazují možný stav světa, kam se můžeme dostat v průběhu analýzy STPA. Validace tímto způsobem proběhla úspěšně, model je možné považovat za validní.



Obrázek 17: Diagram aktivit UML popisující průběh STPA

8. Diskuze

Problematika řízení bezpečnosti je docela složitým oborem. Důvodem je nedávný vznik této vědecké oblasti a s tím související existence chybějících nástrojů pro aplikace v teorii bezpečnosti. Řízení bezpečnosti je nástrojem pro sledování bezpečnostních rizik a zlepšování výkonnosti v bezpečnosti. Zajištění existence bezpečnosti je vždy vyžadované pro rozvoj člověka samotného a procesů, které člověk vytváří.

Teorie safety je v celku složitá, a proto je zapotřebí vytvářet nástroje pro její usnadnění a automatizaci. Jedním z nástrojů je využití informačních technologií v řízení bezpečnosti. V současnosti se velký rozvoj v IT technologiích zaměřuje také na využití ontologie v konceptuálním modelování, které umožnuje snížit náklady při tvorbě software a taktéž umožňuje vytvoření vyšší úrovně abstrakce základny každého programu s možnou následující aplikací v reálním světě.

Jedním z nástrojů pro sledování rizik a jejich zkoumání je bezpečnostní model STAMP, současně využívaný i americkou vládní agenturou zodpovědnou za americký kosmický program a všeobecný výzkum v oblasti letectví NASA (National Aeronautics and Space Administration). Výhodou STAMP je použití řídící smyčky se zpětnou vazbou, která je často využívaná v systémovém inženýrství, pro aplikaci analýz CAST a STPA pro průzkum procesů v rámci teorie STAMP, určovaní nevhodných řízení, které mohou vyvolat nebezpečí a průzkum existujících nebezpečných stavů.

Cílem této práce byla konceptualizace vybraných částí modelu STAMP, a to řídicí smyčky a analýzy STPA. Je to pokus o vytvoření části abstraktní základny pro následující vytvoření softwaru celkového modelu STAMP. Vytvořená základna je připravená pro aplikaci dat z procesů existujících v reálném světě.

Vzniklý ontologický model na této úrovni je schopen pomoci leteckým (ale i jiným) organizacím v aplikaci výše uvedené analýzy STAMP. Podle tohoto modelu budou organizace schopny automaticky vygenerovat možné varianty průběhu řízeného procesu a ukázat možné variace nevhodných řízení. Organizace tak nebudou ohraničeny jen chápáním vlastních procesů odborníky z praxe, ale jako podpora jim bude sloužit tento model, který vygeneruje konečný možný počet variací situací, čímž sníží riziko přehlédnutí důležité informace nebo nebezpečí.

Nedostatkem tohoto modelu je, že zatím nebyl aplikován v reálném prostředí leteckých organizací a model není v této chvíli určen pro aplikaci na složité procesy. Proto je před

používáním tohoto modelu zapotřebí model rozšířit podle požadavků zkoumaného procesu, resp. specifikací prostředí, kde by měl být aplikován.

Expert pracující s ontologií může zkoumat konkrétní situaci tím, že specifikuje instance entit z konkrétního prostředí a pro jeho propojení určí odpovídající třídy z abstraktního modelu této práce. Takovéto propojení může sloužit základním případům užití (use-case), kde cílem je analyzovat konkrétní systém z pohledu bezpečnosti a stanovit v něm aktuální nebezpečí (co je primárním cílem analýzy STPA). Obecně je možno říct, že ontologický model podporuje analytické případy užití, potřebuje však model instancí konkrétní události nebo části řídící struktury konkrétní organizace.

Ve vzniklém konceptuálním modelu je v této práci namodelované selhání řídicího. Využitím tohoto modelu jsme po propojení s konkrétní instancí schopni vytvořit seznam řídicích a analyzovat např. jejich algoritmy řízení s pomocí vygenerování všech chyb, které mohou nastat. To znamená, že tímto existuje podpora zkoumání všech řídicích smyček. Model vygeneruje všechny řídící smyčky a poté expert na základě vygenerovaných výsledků určí, zda tam může nastat chyba a jak jí předcházet.

Jako vize do budoucna je navrhováno rozšířeni konceptuálního modelu pomoci identifikování všech možných chyb řídící smyčky, specifikovaných v teorii modelu STAMP. Toto umožní pro letecké organizace vytvořit seznam všech objektů řídící smyčky a seznam odpovídajících chyb, které mohou v těchto krocích vzniknout. Tímto se dají namodelovat všechny kroky STPA a poté i rozšířit model o kompletní teorii STAMP jako i podporu pro jiné nástroje řízení bezpečnosti. V konečném výsledku by měl vzniknout abstraktní základní model, který by mohl být použit jako celek nebo jenom z části pro konkrétní analýzu. Taktéž se i hodí pro přípravu budoucích pracovníků v oblasti bezpečnosti, kterým může být prezentován jako přehledná vizualizace analytických procesů.

Využití této práce v budoucnu může být rozsáhlé. Tento model je v nejvyšší úrovni abstrakce, proto může být použit i mimo letectví, např. ve zdravotnictví, vojenské oblasti, oblasti stavební, strojní a jiné. Avšak vznik tohoto konceptuálního modelu s použitím ontologii byl primárně zamýšlen s jeho využitím v letecké oblasti, v oblasti řízení leteckého bezpečnosti.

Předpokládané je také použití této práce pro vznik kompletního konceptuálního modelu STAMP, ale je důležité si uvědomit, že tento model byl vytvořen na základě představ o světě letecké bezpečnosti autorem této práce, proto je přípustná i varianta vzniku jiného modelu na základě jiného chápáni teorie modelu STAMP v jiných oblastech a doménách.

Závěr

Cílem této práce bylo vytvořit konceptuální model vybraných částí modelu STAMP. Pro dosažení tohoto cíle bylo zapotřebí určit části modelu STAMP, které budou modelované a nástroje pro jejich modelování.

Ze studia problematiky modelu STAMP bylo zjištěno, že jeho základem je řídící smyčka (Control loop). Také bylo zjištěno, že řídící smyčka je využívána analýzami CAST a STPA, z nichž jedna (STPA) byla vybrána pro modelování.

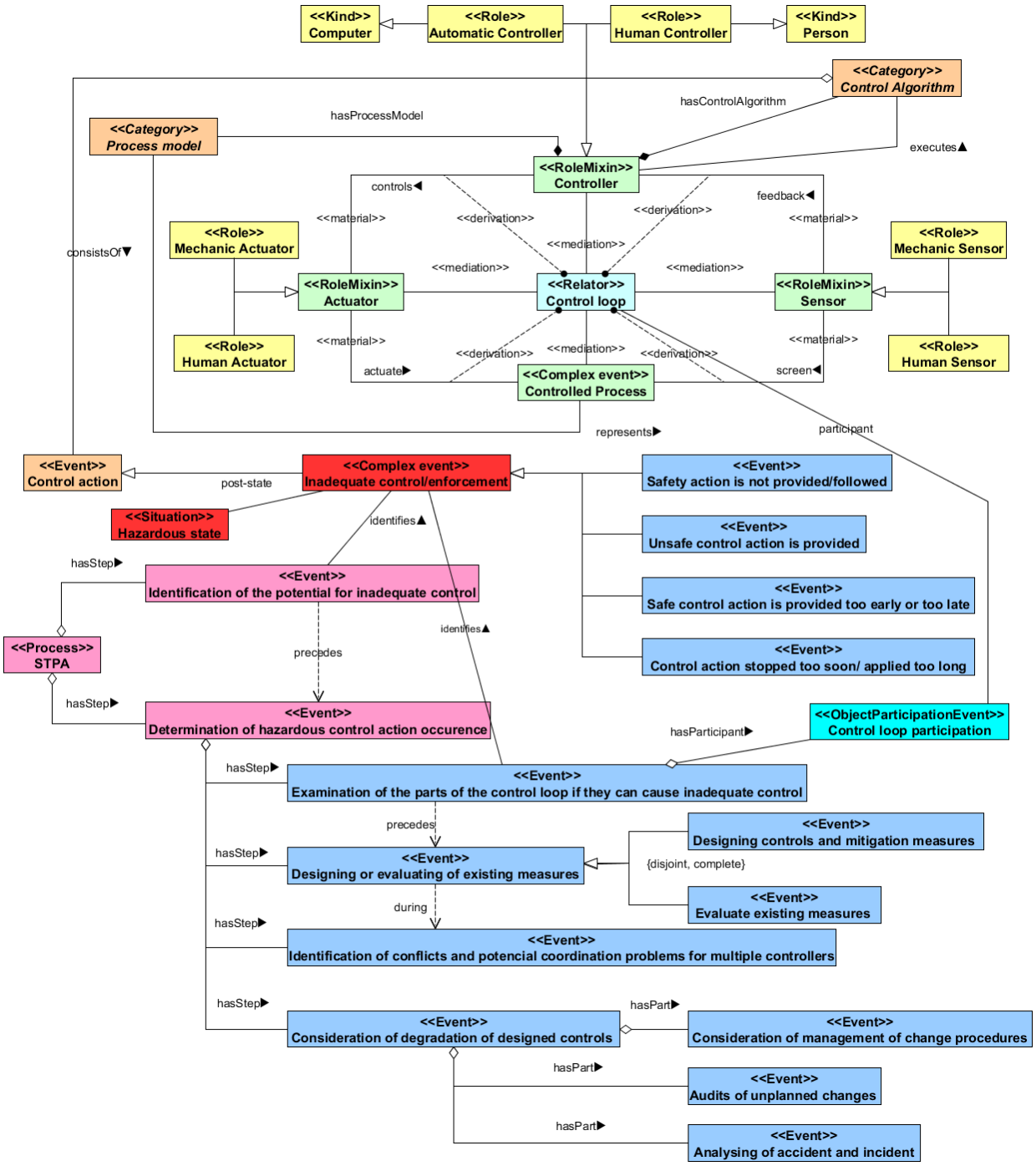
Jako nástroj pro vytvoření konceptuálního modelu byl vybrán jazyk OntoUML, používající koncepty z UFO (Unified Foundational Ontology). Důvodem výběru OntoUML byl fakt, že používá koncepty z ontologie a to dovoluje více specifikovat vlastnosti entit modelu. Na podporu využívání jazyka OntoUML jsou vytvořeny různé aplikace, ve kterých uživatel může namodelovat různé ontologické modely a poté zkontrolovat, zda jsou syntakticky korektní a validovat je po obsahové stránce. Aplikace Menthor byla použitá v této práci pro technickou stránku tvorby modelu. Také byl zvolen grafický editor yEd pro vytvoření obrázků pro tuto práci. Hlavním důvodem použití editoru bylo to, že ani zvolená aplikace Menthor nepodporuje všechny koncepty z UFO použité v této práci, takže jenom s tímto nástrojem by nebylo možné reprezentovat celou obsahovou stránku modelu. Pro alternativní zobrazení průběhu analýzy STPA byl využít diagram aktivit z jazyku UML. Tento diagram dovolil ukázat dynamickou část analýzy.

Po vytvoření konceptuálního modelu byli použity tři způsoby validace. Validace na základě otázek způsobilostí (Competency question), vyhodnocení vizualizace Alloy a porovnávání diagramu aktivit UML a instance vygenerovaných Alloy. Všechny validace proběhly úspěšně, a proto model považujeme v této fázi vývoje za validní.

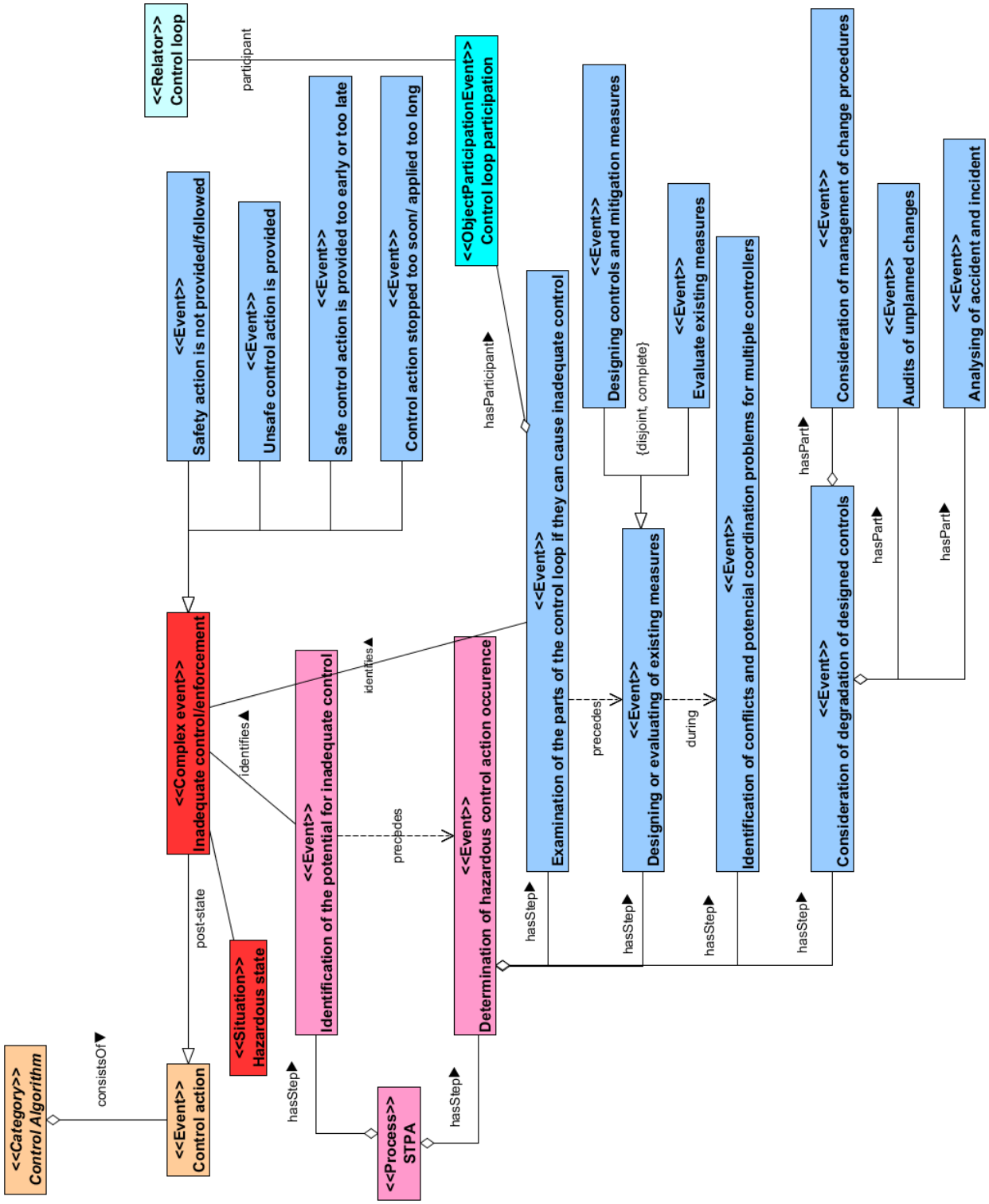
Cíl práce můžeme považovat za dosažený. Vznikl konceptuální model řídící smyčky s aplikaci STPA, který je syntakticky a obsahově validní.

Jako vize do budoucna je vhodné aplikovat tento model na konkrétních situacích s použitím instancí těchto situací. Taky je možné využít tohoto modelu v řízení bezpečnosti pro jeho usnadnění a automatizaci v některých aktivitách. Je také vhodné pokračovat ve vývoji samotného ontologického modelu a tak pokrýt celou teorii modelu STAMP, čímž by se zvětšil pozitivní dopad a synergie z jeho využití nejenom v leteckém průmyslu.

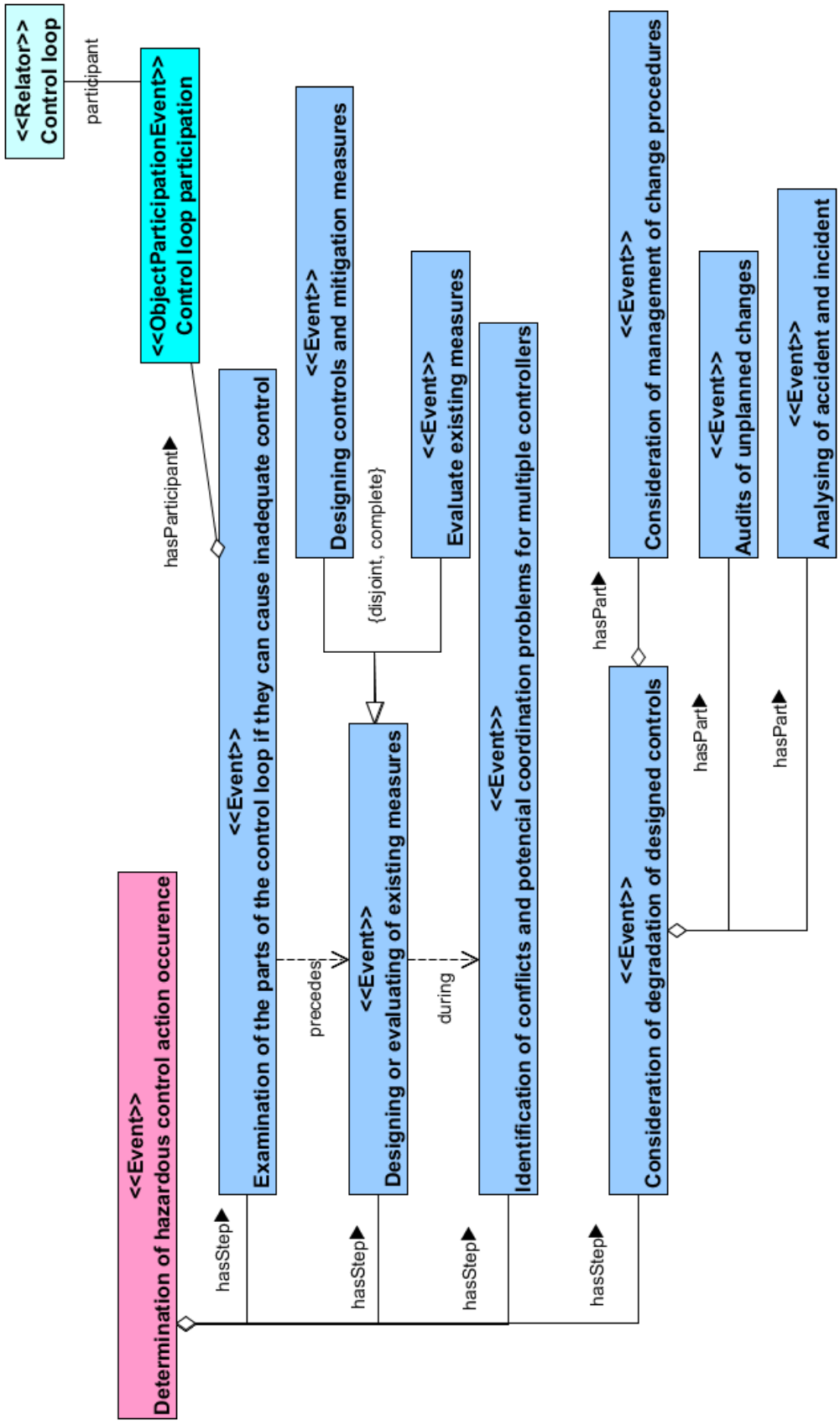
Příloha 1: Konceptuální model vybraných části modelu STAMP (Řídicí smyčka a aplikace STPA)



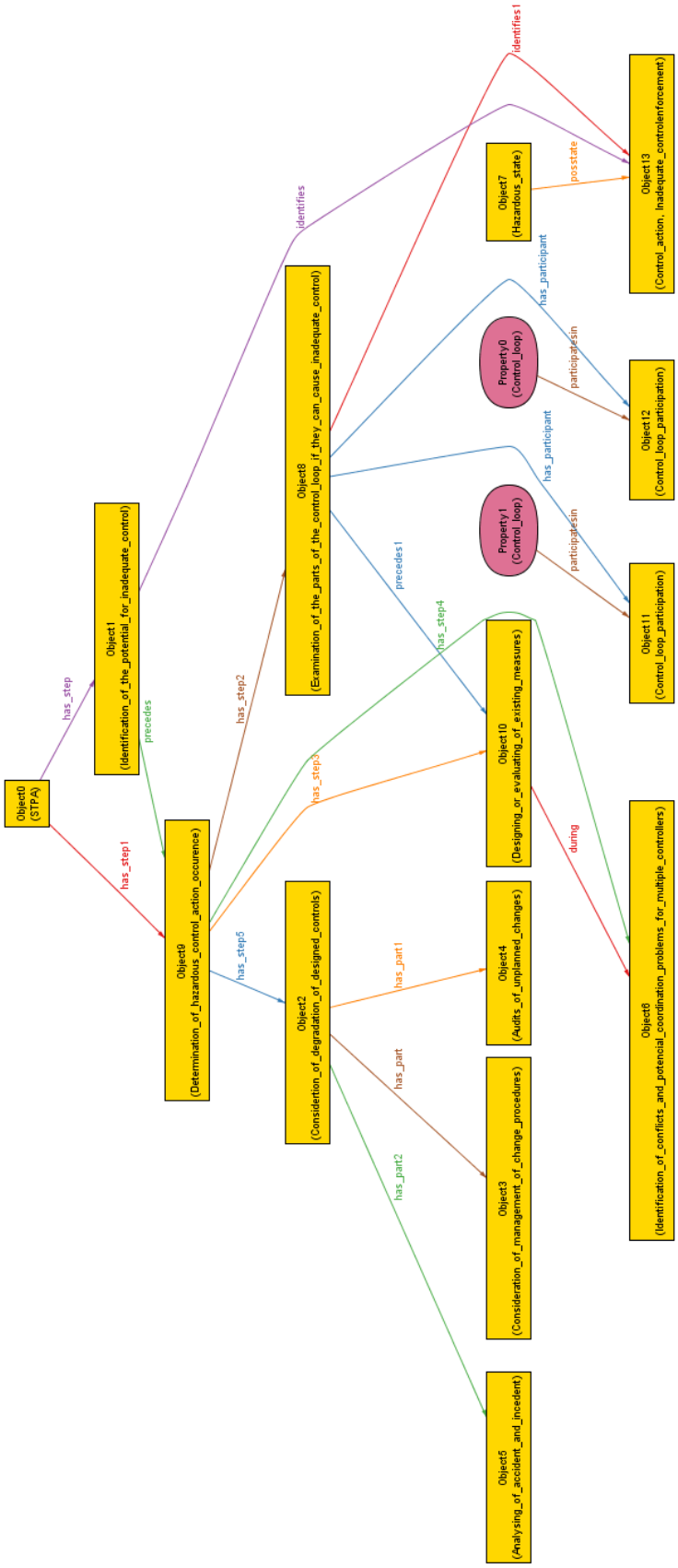
Příloha 2: Konceptuální model analýzy STPA



Příloha 3: Konceptuální model druhého hlavního kroku analýzy STPA



Příloha 4: Validace pomocí vizualizace Alloy konceptuálního modelu STPA



Zdroje

1. MASLOW A.H. Classics in the History of Psychology (1943) A Theory of Human Motivation. [cit. 2018-08-03]
   Dostupné z: https://psychclassics.yorku.ca/Maslow/motivation.htm
2. Letecká informační služba. 2018 Letecká informační služba, Řízení letového provozu ČR, s.p.. [cit. 2018-08-03] Dostupné z: http://lis.rlp.cz
3. LEVESON, Nancy G. a John P. THOMAS. STPA handbook [online], 188 [cit. 2018-08-05]
   Dostupné z:http://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf
4. LEVESON, Nancy. Engineering a safer world: systems thinking applied to safety. Cambridge, Mass.: MIT Press, 2011. Engineering systems. [cit. 2018-08-03] ISBN 978-0-262-01662-9.
5. Nancy Leveson: Professor of Aeronautics and Astronautics [online]. [cit. 2018-08-03] Dostupné z: http://sunnyday.mit.edu/bio-serious.html
6. LEVESON, Nancy G. A New Accident Model for Engineering Safer Systems. *Safety Science* [online]. [cit. 2018-08-08]
   Dostupné z: http://sunnyday.mit.edu/accidents/safetyscience-single.pdf
7. LEVESON, Nancy G. CAST Analysis of the Shell Moerdijk Accident [online]. [cit. 2018-08-11] Dostupné z: http://sunnyday.mit.edu/shell-moerdijk-cast.pdf
8. ARISTOTELÉS. Metafyzika. 2. vyd. Praha: Jan Laichter, 1946, 498 s. [cit. 2018-08-03]
9. Aristotle's Biology. Stanford Encyclopedia of Philosophy [online]. [cit. 2018-08-03] Dostupné z: https://plato.stanford.edu/entries/aristotle-biology/
10. Aristotle's Metaphysics. Stanford Encyclopedia of Philosophy [online]. [cit. 2018-08-03] Dostupné z: https://plato.stanford.edu/entries/aristotle-metaphysics/
11. Příruční slovník naučný. Praha: Nakladatelství Československé akademie věd, 1966. [cit. 2018-08-03]
12. GUIZZARDI, Giancarlo. Ontological foundations for structural conceptual models. Enschede, The Netherlands: Centre for Telematics and Information Technology, Telematica Institut, 2005. [cit. 2018-08-11] ISBN 9075176813.
13. Wiki | OntoUML Community Portal. OntoUML Community Portal | community site [online]. 2017 [cit. 2018-08-11] Dostupné z: https://ontouml.org/ufo/wiki/
14. KŘEMEN, Petr. Introduction [online]. [cit. 2018-08-14] Dostupné z :https://cw.fel.cvut.cz/old/courses/osw/program
15. BERNERS-LEE, Tim, James HENDLER a Ora LASSILA. Semantic Web [online]. Scientific American, 2001. [cit. 2018-08-14]

16. Ontology Engineering activities [online]. [cit. 2018-08-20]. Dostupné z: http://mayor2.dia.fi.upm.es/oeg-upm/files/pdf/NeOnGlossary.pdf
17. FERNÁNDEZ-LÓPEZ (CEU), Mariano, Asunción GÓMEZ- PÉREZ (UPM), Klaas DELLSCHAFT (UKO-LD), Holger LEWEN (UKARL) a Martin DZBOR (OU). Lifecycle Support for Networked Ontologies [online]. [cit. 2018-08-20]. Dostupné z: http://neon-project.org/web-content/images/Publications/neon_2009_d532.pdf
18. JEAN-MARY, Yves R., E. Patrick SHIRONOSHITA a Mansur R. KABUKA. *Ontology matching with semantic verification* [online]. [cit. 2018-08-20]. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1570826809000146?via%3Dihub
19. GRÜNINGER, Michael, Torsten HAHMANN, Ali HASHEMI a Darren ONG. *Ontology Verification with Repositories* [online]. [cit. 2018-08-20]. Dostupné z: http://www.cs.toronto.edu/~torsten/publications/MGruninger_FOIS10.pdf
20. *Ontology and Conceptual Modeling Research Group: Dr. Giancarlo Guizzardi* [online]. [cit. 2018-08-20]. Dostupné z: https://www.inf.ufes.br/~gguizzardi/
21. FOWLER, Martin. *Destilované UML*. Praha: Grada, 2009. Knihovna programátora (Grada). ISBN 978-80-247-2062-3.
22. TEVFIK, Bultan. *CS 267: Automated Verification: Alloy Analyzer* [online]. [cit. 2018-08-20]. Dostupné z: http://www.cs.ucsb.edu/~bultan/courses/267/lectures/l18-1.pdf
23. FERNANDES, Paulo C. Barbosa, Renata S.S. GUIZZARDI a Giancarlo GUIZZARDI. Using Goal Modeling to Capture Competency Questions in Ontology-based Systems [online]. Journal of Information and Data Management, 2010, , 16 [cit. 2018-08-20].
24. GUIZZARDI, Giancarlo, Gerd WAGNER. *Towards an ontological foundation of discrete event simulation.* [cit. 2018-08-20] ISBN 978-1-4244-9864-2 2010