

Autoservis Plus

1 Popis domény

Autoservis Plus je moderní servisní firma, která se specializuje na údržbu a opravy vozidel. Jejich zákazníci mohou využít široké nabídky služeb, které zahrnují diagnostiku, opravy, údržbu a další související služby. Cílem společnosti je poskytovat spolehlivé a kvalitní služby, aby vozidla zákazníků byla v co nejlepším technickém stavu. Společnost Autoservis Plus spolupracuje jak s právními osobami tak s osobami fyzickými. Doména popisuje proces poskytování servisních služeb, správu zákazníků, vozidel a jejich servisní historie, včetně objednávek. Zároveň pokrývá organizaci pracovníků a správu zásob dílů a celkovou strukturu firmy.

1.1 Popis firmy

Autoservis Plus funguje na základě několika klíčových oddělení. Oddělení zákaznické podpory zajišťuje komunikaci s klienty, jejich objednávky a následnou správu zakázek. Technické oddělení zahrnuje mechaniky a certifikované specialisty, kteří se starají o samotné provedení údržby a oprav vozidel. Důležitou roli hraje také skladové oddělení, které spravuje zásoby a objednávání náhradních dílů a dalších provozních materiálů potřebných pro opravy.

Nejdůležitější zaměstnanci servisu jsou pochopitelně mechanici, ti pracují při směně vždy ve dvojici, přičemž alespoň jeden z nich musí být mistr a druhý mistr nebo učeň, který k nám byl poslán z učiliště. Během směny mohou pracovat na více zakázkách, které si mistr vybírá ze systému, ovšem během směny má tým z provozních důvodů k dispozici pouze jeden hever.

1.2 Proces Poskytování služeb

Běžný pracovní den v Autoservisu začíná příjmem objednávek od zákazníků. Zákazník si může naplánovat návštěvu autoservisu buď telefonicky, online nebo osobně, tím se vždy spojí s jedním z našich operátorů. Při objednání služby je zaznamenána konkrétní poptávka/služba (např. výměna oleje, diagnostika motoru, oprava brzd) a zákazník dostane přibližný odhad času a ceny.

Jakmile zákazník dorazí, pracovník zákaznické podpory přijme vozidlo a zapíše všechny detaily zakázky, které jsou poté předány do technického oddělení. Tým mechaniků přidělený k vozidlu/objednávce provede příslušné opravy nebo údržbu. Během procesu oprav jsou využívány zásoby ze skladu, které jsou průběžně sledovány, aby nedošlo k nedostatku důležitých dílů.

Jakmile je oprava dokončena, zákazník je informován a může si vozidlo vyzvednout. Všechny provedené práce jsou zaznamenány do historie služeb vozidla pro budoucí reference. Nakonec je zákazníkovi ještě odeslán formulář hodnocení služby.

1.3 Proces objednávek a servisní historie

Když zákazník vytvoří objednávku, je zaznamenána konkrétní poptávka po službách, jako je oprava nebo údržba. Objednávky jsou přiřazeny mechanikům podle jejich dostupnosti a dovedností. Po provedení opravy je zákazníkovi vystavena faktura a všechny detaily o provedené práci jsou uloženy do historie vozidla. Historie zahrnuje všechny opravy a údržby, což umožňuje autoservisu i zákazníkovi přehled o předchozích zásazích.

1.4 Omezení

V doméně se vyskytují následující netriviální omezení:

1. Mechanici musí být certifikováni pro provádění specializovaných oprav.
2. Součet odhadovaných časů na opravu za všechny zakázky nesmí přesáhnout 12 hodin pro jeden pracovní tým během jedné směny
3. Zákazník který má více než 3 neukončené objednávky nemůže vytvořit další dokud se alespoň 1 z nich nedokončí
4. Vozidlo nemůže být rychlejší než ono samo
5. CostEstimate ani timeEstimate nesmí být záporné

2 OCL omezení

OCL omezení jsou také v příloze **constraints.oc1**.

2.1 OCL omezení 1

context CertifiedRepair

```
def: getRelatedCertificates(): Set(Certificate) = self.certificates->asSet()
```

```
def: getMechanicCertificates(): Set(Certificate) =  
self.order.shift.assignedTeam.members->collect(m | m.certificates)->flatten()->asSet()
```

```
inv MechanicsMustHaveRightCertificates: self.getRelatedCertificates()->forAll(c | self.getMechanicCertificates()->includes(c))
```

2.2 OCL omezení 2

context Shift

```
inv MechanicsCantWorkOvertime: self.orders->collect(o |  
o.contractSubject.orderCreation.timeEstimate)->sum() <= 12
```

2.3 OCL omezení 3

context Customer

```
inv CustomCantHaveMoreThen3ActiveOrders: self.orderCreation->collect(oc |  
oc.order)->select(o | o.phase <> 'Finished')->size() <= 3
```

2.4 OCL omezení 4

context Vehicle

```
inv VehicleCantBeFasterThenSelf: not self.fasterThan->includes(self)
```

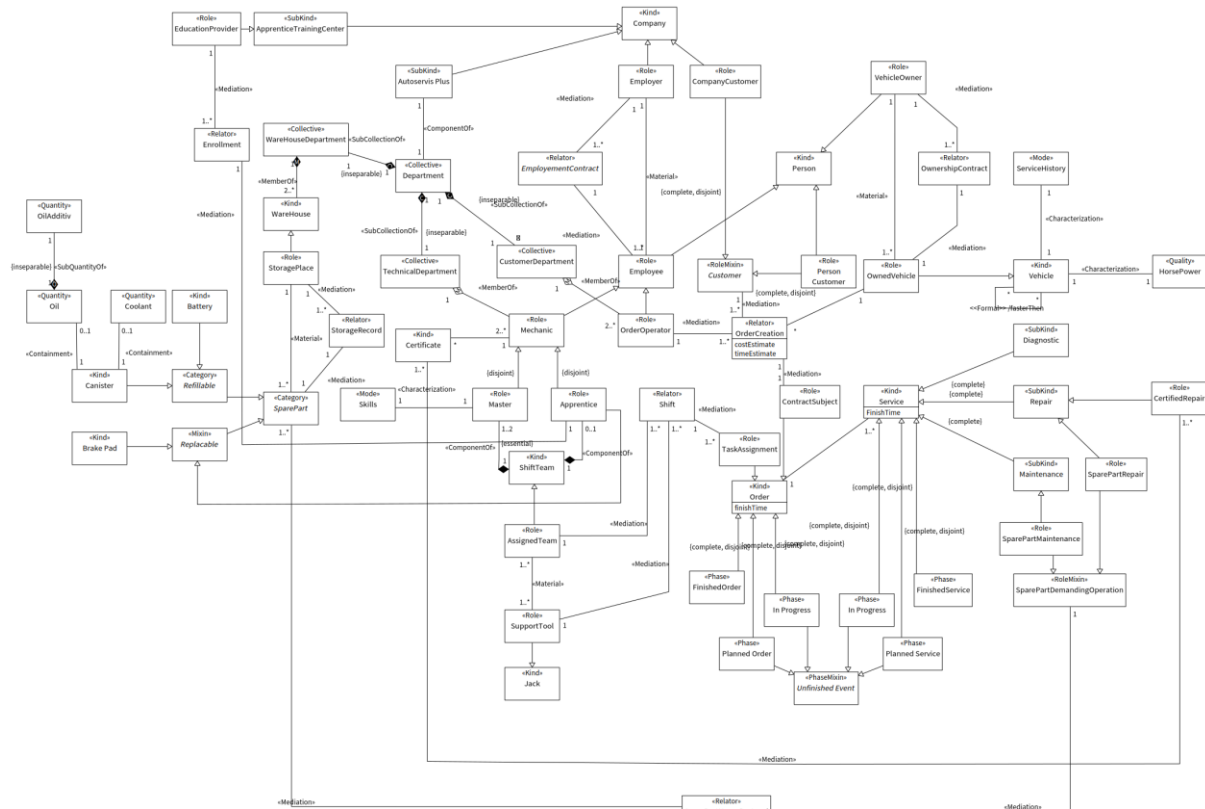
2.5 OCL omezení 5

context OrderCreation

```
inv NonNegativeValues: self.costEstimate >= 0 and self.timeEstimate >= 0
```

3 OntoUML model

Kompletní OntoUML model se nachází v příloze **model-ontouml.opp**.



Komentář k AntiPatterns:

BinOver – u Vehicle je řešen pomocí OCL pravidla

WholeOver - konzultováno - jedná se o chybu detekce – Master a Apprentice jsou totiž disjoint

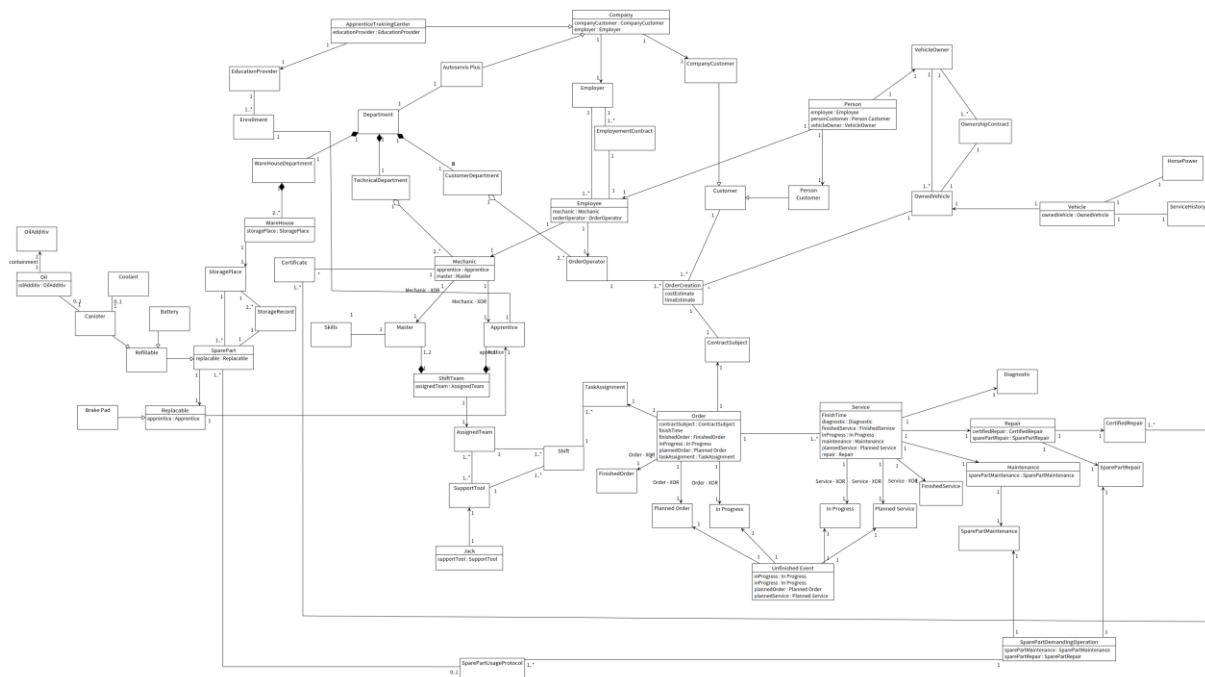
ImpAbs - nespadá mezi antipatery které jsme měli řešit, nicméně u všech vazeb zkontrolováno a tyto situace mohou nastat/nejsou problematické

3.1 Tabulka použitých konstruktů

Koncept	Popis	Počet
Kind	povinné alespoň 2 různé druhy Např: Company, Person, Vehicle	12
SubKind	alespoň 2 různé poddruhy jednoho druhu Např: Service nebo Company	6
Role	alespoň 2 různé role Např: Mechanic, Employee, Employer, Apprentice	12
Phase	alespoň 2 různé fáze Např: FinishedOrder, PlannedOrder	6
Category	alespoň 1 kategorie Např: SparePart	2
RoleMixin	alespoň 1 mixin rolí Customer – mixin (Company & Person) Customer	2
PhaseMixin	alespoň 1 mixin fází Unfinished Event – Mixin fází Order a Service	1
Mixin	alespoň 1 mixin Replacable – SparePart a Apprentice	1
Functional Complex	funkční celek složený alespoň ze dvou komponent s různou povinností části ShiftTeam	1
Part	alespoň 2 různé povinnosti celku vůči Částem Např: OilAdditiv, ShiftTeam	3
Quantity	alespoň 1 kvantita Např: Oil, Coolant	3
Collective	alespoň 1 kolektiv Např: Department	4
Quality	alespoň 1 kvalita HorsePower	1
Mode	alespoň 1 mód Např: Skills u Master Mechanic	2
Relator	alespoň 2 relátory pro materiální vazby (kompletní vzor) Např: OwnershipContract, EmploymentContract	6
Formal Relation	alespoň 1 formální relace Vehicle - fasterThen	1

3.2 UML model převedený z OntoUML

Kompletní UML model se nachází v příloze **model1-uml.opp**.



Komentář k převedení ONTOUML do UML:

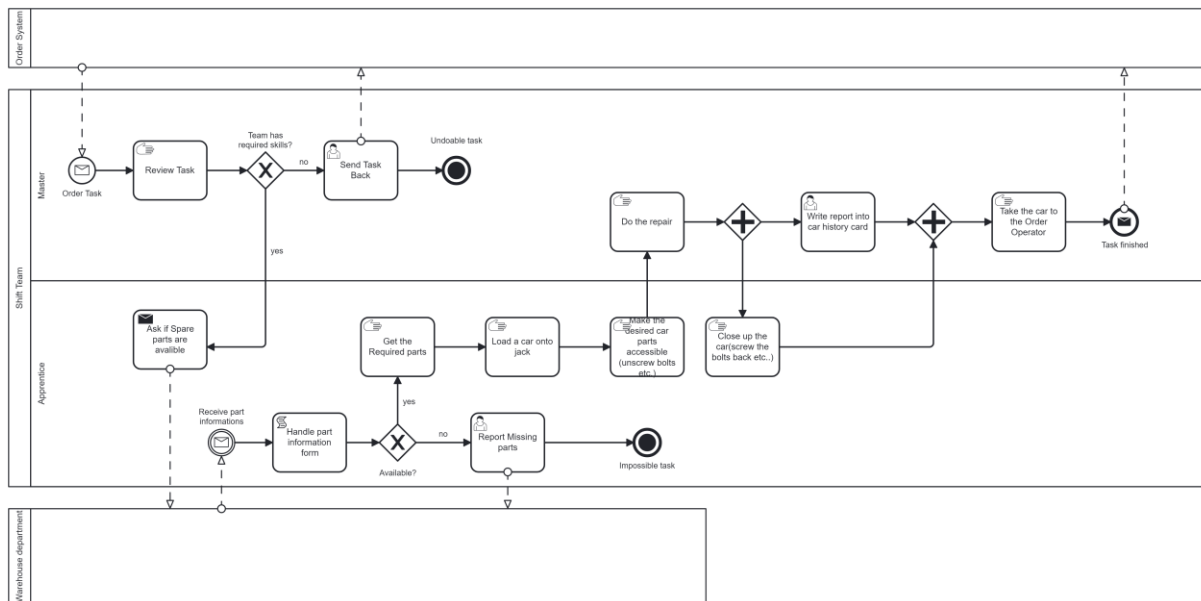
Např.: třída Employee - může být současně Mechanic(role) a OrderOperator(role) – což chceme připustit - zaměstnanec může jeden den pomoci mechanikům a druhý den volat se zákazníky. Převod do UML tuto situaci řeší skládáním kde Employee má atribut Mechanic a atribut OrderOperator

Třída Service - má podtypy diagnostika, oprava, údržba(které nejsou disjoint) - Při modelaci musíme zvolit - buď vícenásobnou dědičnost (Pokud to jazyk podporuje) nebo pomocí skládání, případně vytvořit všechny možnosti různých podtříd(např class DiagnosticRepair) které dovolí třídě chovat se jako kombinace 2 a více samostatných podtypů. Já jsem modeloval pomocí skládání

4 BPMN model

Textový popis diagramu je součástí popisu domény na začátku

BPMN model popisující průběh 1 opravy v rámci týmu a směny



5 DEMO model

Tabulky transakcí do větších podrobností popisují transakce zmíněné v popisu domény

5.1 Rozšířené TPT

ID (of transaction kind)		T01
Transaction kind		Objednávka služby
Product kind		Služba je objednána
Initiator (actor role)		Zákazník
Executor (actor role)		Operátor zákaznické podpory
Order	Request	Zákazník kontaktuje servis s poptávkou po službě (např oprava vozidla)
	Promise	Operátor potvrdí, že služba může být provedena a poskytne zákazníkovi přibližný odhad času a ceny
	Decline	Operátor se omluví zákazníkovi a objednávku zamítne z časových nebo jiných důvodů
Result	Declare	Operátor vytvoří záznam objednávky a zašle ho emailem zákazníkovi pro oficiální potvrzení
	Accept	Zákazník souhlasí a objednávku potvrdí
	Reject	Zákazník odmítne nabídku služby (např.: si myslí že cena je příliš vysoká nebo to bude příliš dlouho trvat)
Rev	Revoke-rq	Zákazník si rozmyslel co chce na autě opravit a změnil poptávku po službě

	Revoke-pm	Operátor zjistí že specializovaný mechanik má dovolenou a musí upravit časový odhad
	Revoke-da	<i>Pokud je to v možnostech servisu operátor upraví nabídku (sníží cenu opravy nebo vtěsná zákazníka dříve do fronty oprav)</i>
	Revoke-ac	<i>Zákazník změnil rozhodnutí a objednávku zruší po původním potvrzení(z finančních nebo časových důvodů)</i>

ID (of transaction kind)		T02
Transaction kind		Žádost o přidělení objednávky
Product kind		Objednávka je přidělená
Initiator (actor role)		Mistr
Executor (actor role)		Systém přidělování objednávek
Order	Request	Mistr požádá o přidělení objednávky
	Promise	Systém začne načítat dostupné objednávky
	Decline	Systém nemá žádnou dostupnou objednávku a žádost zruší
Result	Declare	Systém vybere první dostupnou objednávku a počká na potvrzení o vytvoření záznamu o objednávce
	Accept	Mistr souhlasí s přidělením objednávky a potvrdí vytvoření záznamu
	Reject	Mistr odmítne tuto objednávku (např: nemají dostatečnou certifikaci nebo zkušenosti nebo čas)
Revokes	Revoke-rq	Mistr zruší žádost o přidělení objednávky (zrovna začala pauza) a pošle tedy žádost až pauza skončí
	Revoke-pm	Systém zruší načítání objednávek (Zjistí že tým pracuje déle než 12 hodin což je nepřípustné)
	Revoke-da	<i>Systém nabídne jinou objednávku kterou může mistr přijmout</i>
	Revoke-ac	<i>Mistr odvolá souhlas s přidělením objednávky (objednávka se zařadí zpět do systému)</i>

ID (of transaction kind)		T03
Transaction kind		Příprava vozidla k opravě
Product kind		Vozidlo je připraveno k opravě
Initiator (actor role)		Mistr
Executor (actor role)		Učeň
Order	Request	Mistr požádá učně o připravení vozidla k opravě
	Promise	Učeň souhlasí a začne s přípravou
	Decline	Učeň odmítne začít s přípravou protože nemá dostatečné znalosti nebo nástroje
Result	Declare	Učeň ohlásí mistrovi že auto je připravené k opravě
	Accept	Mistr pochválí učně a potvrdí že vozidlo je připraveno k opravě
	Reject	Mistr učni vynadá že není vše dostatečně připraveno a odmítne začít s opravou
Revokes	Revoke-rq	Mistr zruší žádost na přípravu vozidla (už je připravené nebo chce připravit jiné vozidlo)
	Revoke-pm	Učeň zjistí že neumí připravit tento typ vozidla a odvolá svůj slib na přípravu
	Revoke-da	<i>Učeň se omluví mistrovi, opraví všechny nedostatky které mistr našel a opět ohlásí, že auto je připraveno</i>
	Revoke-ac	<i>Mistr odvolá pochvalu protože zjistil, že nejsou splněny všechny požadavky na přípravu</i>

ID (of transaction kind)		T04
Transaction kind		Objednání náhradního dílu
Product kind		Náhradní díl je objednán
Initiator (actor role)		Systém skladu
Executor (actor role)		Pracovník skladu
Order	Request	Systém skladu zjistí nedostatek konkrétního dílu a zašle požadavek na jeho objednávku pracovníkovi skladu
	Promise	Pracovník potvrdí požadavek a začne připravovat objednávku
	Decline	Pracovník odmítne vytvořit objednávku pokud je díl momentálně nedostupný nebo příliš drahý
Result	Declare	Pracovník odešle vytvořenou objednávku do systému
	Accept	Systém potvrdí přijetí objednávky a její zpracování
	Reject	Systém odmítne objednávku pokud zjistí nesrovnalosti (např: špatné číslo dílu)
Revokes	Revoke-rq	Po manuálním provedení inventury se najdou ještě naskladněné náhradní díly a systém zjistí, že je už není třeba objednávat
	Revoke-pm	Pracovník zruší přijmutí požadavku jelikož zjistil že se daný náhradní díl už nevyrábí
	Revoke-da	<i>Pracovník upraví objednávku ve které se nachází chyba a odešle ji</i>
	Revoke-ac	<i>Nelze (pokud systém již objednávku jednou přijal musela být v souladu s interními pravidly skladu)</i>

ID (of transaction kind)		T05
Transaction kind		Hodnocení služby servisu
Product kind		Služba je ohodnocena
Initiator (actor role)		Operátor zákaznické podpory
Executor (actor role)		Zákazník
Order	Request	Operátor zašle žádost o vyplnění dotazníku spokojenosti zákazníkovi
	Promise	Zákazník souhlasí a přislíbí že dotazník vyplní
	Decline	Zákazník nereaguje nebo nesouhlasí s vyplněním dotazníku
Result	Declare	Zákazník zašle vyplněný dotazník zpět operátorovi
	Accept	Operátor poděkuje a uloží výsledky dotazníku do systému
	Reject	Dotazník není vyplněn správně nebo kompletně a operátor nezanese výsledky do systému
Revokes	Revoke-rq	Nelze (Nedává smysl v rámci domény aby operátor rušil žádost o vyplnění dotazníku, firma by nevypadala před zákazníky sebevědomě)
	Revoke-pm	Zákazník si rozmyslel vyplnění dotazníku a již ho vyplnit nechce
	Revoke-da	<i>Zákazník požadá o vymazání jeho odpovědí v rámci dotazníku ze systému nebo že chce nějaké svoje odpovědi ještě změnit</i>
	Revoke-ac	<i>Operátor po uložení dotazníku zjistí chyby v informacích a smaže záznam ze systému</i>

5.2 Organisation Construction Diagram (OCD)

Textový popis diagramu je součástí popisu domény na začátku

