

Zadání 1. projektu ATA LS 2024/25

Cílem projektu je praktické cvičení testování založené na modelech i požadavcích a automatizace testů integrované do CI pipeline.

Na základě specifikace chování testované komponenty v přirozeném jazyce a specifikace požadavků navrhnete automatizovanou testovací sadu.

Popis testovaného produktu

Řízení vozíku v robotické továrně

V robotické továrně je autonomní motorizovaný vozík pro převoz materiálu potřebného k výrobě. Vozík má předem určené trasy a více zastávek. Řídicí systém vozíku má na starosti plánování další zastávky a monitorování zátěže a využití kapacity vozíku. Vozík zpracovává požadavky o přesun materiálu. Materiál, resp. požadavek na přesun, je určený svojí zdrojovou a cílovou stanicí, váhou a binární prioritou (přesun je nebo není prioritní). Každý materiál má při převozu alokován jeden slot vozíku. Pokud je materiál pro převoz větší než jeden slot nebo je příliš těžký, bude rozdělen do více slotů. O toto rozdělení se však kontrolní systém vozíku nestará. Uvažujte, že vozík je používán v ideálním prostředí (systémy s ním komunikující splňují veškeré předpoklady, data odpovídají reálným stavům). Vozík má pro převoz materiálu celkem 1 až 4 sloty. Vozíky jsou 3 druhů podle maximální nosnosti: 50 kg, 150 kg a 500 kg. Vozíky s nejmenší nosností mají nejméně 2 sloty, vozíky s největší nosností mají maximálně 2 sloty.

Specifikace chování

Řídicí systém automaticky generuje požadavky na přemístění materiálu mezi zastávkami v robotické továrně. Po přijetí požadavku vozík vyhodnotí trasu a zahájí nakládku materiálu. Pokud vozík do 1 minuty nenaloží materiál, řídicí systém nastaví materiálu prioritní vlastnost na hodnotu 1. Pokud takový materiál stále není naložen do další 1 minuty a vozík má volnou kapacitu, materiál se okamžitě naloží a vozík přepne do režimu pouze-vykládka. V tomto režimu vozík ignoruje jiné požadavky na převoz a zamíří přímo k cílové zastávce prioritního materiálu. Po vyložení veškerého prioritního materiálu přepne zpět do režimu normální. V tomto režimu optimalizuje trasu tak, aby minimalizoval celkovou přepravní vzdálenost a umožnil současnou nakládku a vykládku více materiálů v rámci své plánované cesty. Pořadí vyzvednutí materiálů nezávisí na pořadí vytvoření požadavků, ale na optimalizaci trasy a aktuálním zatížení vozíku. Pokud jsou všechny sloty vozíku obsazené nebo by naložením materiálu došlo k překročení maximální nosnosti, vozík materiál nenaloží a pokračuje k nejbližší cílové zastávce pro vyložení aktuálního nákladu. Po uvolnění kapacity se znovu vyhodnotí možnost naložení čekajícího materiálu.

Tabulka požadavků¹

(v EN, protože má blízko k technické dokumentaci, implementaci a testům)

Functional Requirements – Requirements defining the behaviour and operations of the system.

| Req. ID | Requirement Description | Priority | Notes |
|---------|---|----------|-------|
| F-01 | The system shall automatically generate requests for cargo relocation between stops. | High | |
| F-02 | The cart shall pick up the cargo within 1 minute of receiving the request. | High | |
| F-03 | If the cart does not load the cargo within 1 minute, the system shall set the cargo's priority. | Medium | |
| F-04 | The cart shall load priority cargo within 1 minute of priority setting. | High | |
| F-05 | The cart shall switch to unloading-only mode if the cart loads priority cargo. | High | |

¹29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering, <https://doi.org/10.1109/IEEESTD.2018.8559686>

| Req. ID | Requirement Description | Priority | Notes |
|---------|--|----------|--|
| F-06 | In unloading-only mode, the cart shall not perform any other loading or unloading except for priority cargo. | High | |
| F-07 | The cart shall switch back to normal mode after unloading all priority cargos. | High | |
| F-08 | In normal mode, the cart shall optimise its route to minimize the total transport distance. | Medium | The optimisation algorithm is not specified. |
| F-09 | The cart shall allow simultaneous loading or unloading of multiple cargos at a stop. | Medium | |
| F-10 | The cart shall reject the cargo if all cart slots are occupied or loading exceeds the maximum capacity. | High | |

Performance and Time Requirements – Requirements related to performance, capacity, and response time.

| Req. ID | Requirement Description | Priority |
|---------|--|----------|
| P-01 | The system shall process a cargo transfer request within 1 second of its creation. | High |
| P-02 | The cart shall respond to a priority request within 1 minute of its setting. | High |
| P-03 | The route optimisation algorithm shall run within a time less than 1 second. | Medium |
| P-04 | Switching the cart to unloading-only mode shall occur immediately after loading priority cargo. | High |
| P-05 | Switching back to normal mode shall occur within 1 second after unloading the last priority cargo. | Medium |

Constraints and Operational Requirements – Requirements on constraints, safety, and environment.

| Req. ID | Requirement Description | Type | Priority |
|---------|--|-------------|----------|
| C-01 | The cart shall have a capacity of 1 to 4 slots. | Constraint | High |
| C-02 | The cart's maximum capacity shall correspond to its type (50 kg, 150 kg, or 500 kg). | Constraint | High |
| C-03 | Carts with the lowest capacity shall have at least 2 slots. | Constraint | Medium |
| C-04 | Carts with the highest capacity shall not have more than 2 slots. | Constraint | Medium |
| C-05 | The control system shall not submit invalid requests (e.g., cargo transfer exceeding the cart's capacity). | Operational | High |
| C-06 | The system shall communicate correctly with the factory's external systems and respond to their requests. | Operational | High |

Prioritization:

- High – Critical for the system's proper function.
- Medium – Important, but does not prevent basic operation.

System Under Test

V informačním systému, v e-Learningu u zadání projektu je v archivu `cartctl.zip` k dispozici příklad implementace vozíku, resp. jeho řízení (control, ctl). SUT je třída `CartCtl` v souboru `cartctl.py`. V archivu je rovněž k dispozici příklad testovacího souboru:

```
python3 -m unittest tests/test_cartctl.py
```

- Chyby v implementaci řídicího systému vozíku mohou být dvojího druhu: zamýšlené a nezamýšlené. Zamýšlené chyby jsou dány volbou i implementací plánovacího algoritmu, který je velmi jednoduchý a nedokáže uspokojit všechny rozumné požadavky na převoz materiálu. Nezamýšlené jsou dány nedůkladnou kontrolou při implementaci. Pokud se vám podaří takovou chybu najít a opravit, prosím zařaďte do archivu také patch soubor a dodatečný popis opravy. Aktuální verze, kterou máte k dispozici, je dána výběrem některých patchů vašich předchůdců.

Úkoly

1. Vytvořte diagram popisující chování vozíku (např. sadu stavových diagramů s ohodnocenými hranami). Diagram(y) preferujte ve formátu pro nástroj `dot` (projekt Graphviz).
2. Identifikujte testovací cesty dle vámi vytvořených grafů.
3. Identifikujte vstupní parametry testu. Parametry mohou být přímé (jejich hodnota se přímo objeví v testu) nebo nepřímé (další data z nich budou odvozena), např. “čas vytvoření požadavku č. 3” nebo “maximální váha všech požadavků”.
4. Navrhněte testovací sadu pro plánovač vozíku (viz `cartctl.py` a `test_cartctl.py`).
5. Vytvořte CI/CD pipeline ve vlastním připraveném prostoru v systému Gitlab na serveru `pajda.fit.vutbr.cz` (přihlašovací údaje vám budou poslány).
6. Odevzdejte archiv s testy a jejich dokumentací. Proveďte commit všech vašich výsledků do repozitáře.

Dokumentace testů

V jeden z odevzdávaných souborů (`cartctl/README.md`) bude vaše dokumentace testů. Skládá se z několika částí:

1. Diagramy chování a jejich stručný popis.
2. Identifikace testovacích cest (kritérium pokrytí zvolte sami). Testovací cesty budou dokumentovány ve formě tabulky obsahující sloupce: identifikátor cesty, cesta/uzly, pokryté požadavky (mezerou oddělený seznam).
3. Identifikace vstupních parametrů, které budou tvořit test, ve formě tabulky dvou sloupců (název/identifikátor parametru, stručný popis).
4. Tabulku testů se sloupci: identifikátor testu, sloupce pro vstupní parametry, očekávaný výsledek, pokryté testovací cesty, zdrojový soubor nebo název testovací metody implementující test.

Tabulky mějte ve formátu dle <https://docs.github.com/en/github/writing-on-github/organizing-information-with-tables>.

Odevzdání

Odevzdejte archiv `proj1.zip` obsahující následující strukturu:

```
proj1.zip
├── cartctl/                - Git repozitář obsahující testovací scénáře vč.
│                           |                           kódu SUT a jeho okolí
├── README.md              - dokumentace testů (formát Github Markdown)
│                           |                           zahrnující diagramy
├── diagrams/              - adresář obsahující diagramy chování
│   ├── *.dot              - diagramy chování, každý diagram v
│   └── *.png              - jednom souboru
├── .gitlab-ci.yml          - předpis pro CI/CD pipeline
├── cartctl/                - původní zdrojové soubory
└── tests/                 - testovací soubory
    ├── test_cartctl.py     - hlavní testovací soubor
    └── test_*.py           - pomocné testovací soubory (volitelné)
```