

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

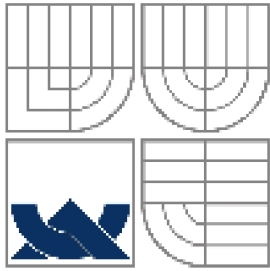
**UŽIVATELSKÉ ROZHRANÍ**  
**PRO DIVADELNÍ ŘÍDICÍ SYSTÉM**

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

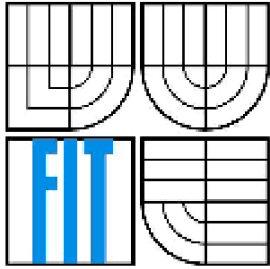
**AUTOR PRÁCE**  
AUTHOR

**LUKÁŠ KOBZA**

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

UŽIVATELSKÉ ROZHŘANÍ  
PRO DIVADELNÍ ŘÍDICÍ SYSTÉM  
USER INTERFACE FOR THEATRE CONTROL SYSTEM

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

LUKÁŠ KOBZA

VEDOUCÍ PRÁCE  
SUPERVISOR

Doc. Dr. Ing. PAVEL ZEMČÍK

BRNO 2010

## **Abstrakt**

Tématem této bakalářské práce je technické vybavení divadelní scény. Je zde nabídnut přehled o jevištní technice a jejím řízení a také přehled o uživatelských rozhraních. Konkrétním cílem pak bylo vytvořit nové uživatelské rozhraní pro řídicí systém horní i dolní mechanizace jevištního zařízení se zaměřením na možnosti aplikace v distribuovaných řídicích systémech s více ovládacími stanovišti.

## **Abstract**

The topic of this bachelor's thesis is the technical equipment of theatre stage. In this work is offered the overview of stage technology and its control and furthermore the overview about user interfaces. The main goal was to create a new user interface for control system of upper and bottom machinery with the intention of application possibilities in distributed control systems with several operator stations.

## **Klíčová slova**

Divadlo, divadelní technika, technologie, jeviště, horní mechanizace, dolní mechanizace, zařízení, pohon, pohyb, řízení, řídicí systém, uživatelsky orientované, uživatelské rozhraní, GUI, ovládání, komunikace

## **Keywords**

Theatre, theatre technique, technology, stage, upper machinery, bottom machinery, equipment, drive, motion, control, control system, user oriented, user interface, GUI, operating, communication

## **Citace**

Kobza Lukáš: Uživatelské rozhraní pro divadelní řídicí systém, bakalářská práce, Brno, FIT VUT v Brně, 2010

# Uživatelské rozhraní pro divadelní řídicí systém

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Doc. Dr. Ing. Pavla Zemčíka.

Další informace mi poskytli Ing. Jiří Ševcovic, Miroslav Souček a Ing. Pavel Župka.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Lukáš Kobza  
12. května 2010

## Poděkování

Rád bych zde poděkoval panu Doc. Dr. Ing. Pavlu Zemčíkovi za jeho vedení a konzultace. Za další cenné informace patří poděkování Ing. Jiřímu Ševcovicovi, panu Miroslavu Součkovi a Ing. Pavlu Župkovi.

© Lukáš Kobza, 2010

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

Obsah .....	1
1 Úvod.....	2
2 Divadlo a jeho technické vybavení .....	4
2.1 Horní mechanizace .....	4
2.2 Dolní mechanizace .....	5
2.3 Bezpečnostní požadavky na jevištní techniku .....	6
3 Řízení divadelní techniky.....	8
3.1 Prvky a principy řídicího systému divadla .....	9
3.2 Uživatelské rozhraní.....	11
4 Uživatelská rozhraní .....	14
4.1 Současné trendy.....	15
4.2 Prostředky pro implementaci.....	18
5 Uživatelské rozhraní pro řídicí systém.....	19
5.1 Návrh vizualizační aplikace .....	20
5.2 Komunikace a princip činnosti.....	21
6 Implementace uživatelského rozhraní.....	23
6.1 Uživatelské prostředí.....	23
6.2 Komunikační rozhraní.....	25
6.3 Režie zařízení, pohyb a vizualizace .....	26
6.4 Správa skupin zařízení .....	28
6.5 Zhodnocení výsledků .....	30
7 Závěr .....	31
Příloha A.....	34

# 1 Úvod

Lidé se již odpradávná snažili ulehčit si svůj život, snažili se ulehčit si činnost, kterou vykonávali. Po dlouhá staletí vynálezci neustále přicházeli s novými a novými objevy a vynálezy. Postupně se objevovali stále nové možnosti mechanizace a později i automatizace v oblasti nejen řemeslné či průmyslové, ale i v oblasti prostého každodenního života. Spousta vynálezů v průběhu historie lidstva způsobila jistou revoluci. Jedním z takovýchto revolučních vynálezů byl počítač, který potenciálně představoval bohaté možnosti řízení a automatizace.

Ke stavu, kdy mohl být počítač v oblasti automatizace efektivně využíván, vedla ještě dlouhá cesta vývoje a zdokonalování technologie. Ovšem dnes se již za velikým kusem této cesty můžeme ohlédnout. Počítače od doby svého vynálezu prošly dlouhým vývojem a v současné době již hrají neodmyslitelnou roli prakticky ve všech oblastech našeho života, na které si jen vzpomene. Počítače jsou nyní schopny namísto člověka vykonávat i poměrně komplikované činnosti a tímto mu velice usnadnit práci. Často je jim svěřována i činnost velmi odpovědná, kde nepatrná chyba či odchylka může způsobit i velké neštěstí. Může v těchto případech být řeč o automatizovaných přístrojích v oblasti zdravotnictví, mohou to být speciální systémy, které pomocí semaforů řídí dopravu ve městech nebo může jít třeba o řídicí počítač na palubě letadla. Programy, které na těchto počítačích běží a odpovídají za bezpečně vykonanou práci, proto musejí být zpracovány velmi důmyslně, tudíž jsou na jejich návrh kladeny velmi vysoké nároky.

Počítač pod vládou člověka je tedy schopný zajišťovat leccjakou činnost. A protože divadlo skýtá veliký prostor pro využití těchto technologií, pronikla automatizace i sem. Scéna divadelní hry je totiž obvykle doplněna o kulisy, se kterými je často potřeba pohybovat, pomalu je spouštět z vrchu dolů, či znenadání vysunout z podlahy jeviště, otáčet s nimi, apod. Často je též potřeba podobným způsobem pohybovat i se samotnými herci. Všechno toto je realizováno pomocí pohonů, které je potřeba nějak souhrnně ovládat. A to by bez počítačem řízené automatizace nebylo dost dobře možné, jelikož těchto pohonů se může nad jevištěm či pod jevištěm nacházet větší množství. Obvykle v řádu desítek kusů. A proto se v současnosti v oblasti jevištní techniky využívá sofistikovaných řídicích systémů, které ovládání a koordinaci těchto zařízení zajišťují. Protože se zde manipuluje mnohdy s těžkými předměty ba dokonce s lidmi, patří jevištní technika k oněm potenciálně nebezpečným oblastem automatizace a proto tyto systémy podléhají řadě bezpečnostních předpisů a norem, které je potřeba dodržovat.

Problematika jevištní technologie je z mého pohledu zajímavá především proto, že skýtá možnost studia a hlubšího nahlédnutí do zákulisí divadelní scény. K umění obecně, a drama nevyjímaje, mám velmi kladný vztah. Divadlo poměrně často navštěvuji a tak zejména z těchto důvodů pramenilo mé rozhodnutí zabývat se touto problematikou podrobněji a umožnit si tak rozšíření svých obzorů jak ve zminěné oblasti divadla, tak v oblasti uživatelských rozhraní a řídicích systémů.

Tato práce byla vypracována za podpory firmy Elseremo, a.s., jejíž oborem je právě výroba a montáž divadelní techniky. Tímto jsem měl možnost z firmou spolupracovat a inspirovat se tak jejich stávajícím divadelním řídicím systémem.

Hned jako druhou kapitolu zařazuji uvedení do divadelní techniky. Jsou zde popsány jednotlivé kategorie scénického zařízení, jeho principy a také požadavky na bezpečnost. Následující kapitola nabízí obecný teoretický přehled řídicích systémů a poté rozebírá prvky řídicích systémů divadelních. Dotýká se možností jejich celkového technologického řešení a je zde věnována pozornost také uživatelským rozhraním. To je podpořeno praktickými příklady. Jelikož předmětem této práce je uživatelské rozhraní, je čtvrtá kapitola zaměřena na tuto problematiku. Shrnuje současnou situaci v oblasti uživatelských rozhraní. Zmiňuje jednotlivé typy uživatelských rozhraní, přičemž se

zaměřuje na uživatelská rozhraní počítačů, zejména na rozhraní grafická. Poukazuje na současné trendy při návrhu i možnosti implementace. Pátá kapitola postihuje důvody a motivace pro tvorbu nového řídicího systému a představuje čtenáři návrh uživatelského rozhraní. Práci uzavírá kapitola popisující postupy implementace nového uživatelského rozhraní.

## 2 Divadlo a jeho technické vybavení

Následující rozdělení a popis scénické techniky je založený na současných obecně platných zvyklostech a standardech. Většina dnešních divadel vychází z tzv. kukátkového typu divadelní scény, který se vyvíjel již od 16. století zejména ve Francii a Itálii a jehož pojetí divadelního prostoru je v této práci popisováno. S následujícími prvky mechanizace se tedy běžně setkáme v divadlech po celém světě.

Technické zařízení divadelní scény sestává ze dvou hlavních částí. Jednak jsou to různé pohony, světelné zařízení, apod., tedy hardware. Druhou část představuje řídicí systém, který má na starosti bezpečný provoz hardwaru a poskytuje uživatelské rozhraní pro jeho ovládání.

Hardwarová zařízení je možné nejjednodušeji rozdělit podle toho, v kterém prostoru se nachází. Jevišťe představuje prostor kde se odehrává děj představení. Vespuďu pod podlahou jeviště jsou umístěna zařízení dolní mechanizace. Zpravidla se jedná o propadla a zvedací stoly, díky nimž je možné z pod podlahy vyvézt na jeviště nebo opět pod jeviště schovat herce či kulisy nebo například pomocí nich vytvořit na podlaze jeviště schody. Ve větších divadlech se objevují i točny zabudované v podlaze jeviště, které umožňují kulisy či herce podle potřeby natáčet a tím dotvářet scénu dle požadavků.

Dalším takovým důležitým prostorem pro divadelní techniku je provaziště. Jde o prostor přímo nad jevištěm, jež slouží jako úkryt pro horní mechanizaci. Dále pak jsou zde světla, dotvářející atmosféru scény či hereckého výstupu. I u nich je potřeba měnit jejich polohu, a proto se obvykle umísťují na celé kovové konstrukce a dohromady tak tvoří osvětlovací baterie, které je možné zavěsit na lana tahů horní mechanizace a umožnit tak jejich pohybem dle potřeby měnit polohu zdroje světla a úhel, pod kterým bude scéna nasvícena.

Také by bylo možné zmínit zvukovou aparaturu. Tato zařízení ale zpravidla bývají umístěna staticky po stranách jeviště a hlediště, nikoli na pohyblivých zařízeních. Posledním zařízením spadajícím pod hardware je ovládací pult. Na něm je nainstalovaná aplikace řídicího systému, jež zprostředkovává vizualizaci pohybu zařízení na scéně, jejich chybových stavů atd. [1]

### 2.1 Horní mechanizace

Je umístěna v provazišti nahoře nad jevištěm, jednu z jejích hlavních složek představuje zpravidla větší počet tahů, tedy pohonů s navijáky a lany, na kterých mohou být zavěšeny kulisy a jakákoliv potřebná dekorace. Základní členění tahů je na: [1]

- **Prospektové tahy** – jsou zakončeny tahovou tyčí, jejíž délka je rovna šířce hrací plochy jeviště. Na tahovou tyč jsou pomocí dvou či více lanových závěsů zavěšovány dekorace. Případně lze zátěž připevnit pevně do kleštin na tahové tyči, což je vhodnější například pro osvětlovací baterie.
- **Bodové tahy** – jedná se pouze o jeden lanový závěs, na jehož konci je upínací mechanismus pro bodové zavěšení dekorace.

Prospektové tahy jsou připevněny napevno k roštu provaziště s pravidelnou roztečí tak, aby pokrývaly pokud možno celou plochu jeviště od portálu až dozadu po horizont. Bodové tahy mohou být i mobilní. Rychlost pohybu tahů bývá obvykle od 0,5 m/s do 1 m/s, někdy dokonce až 2 m/s. Nosnost tahů bývá poměrně vysoká, často více než jednu tunu. Mechanismy tahů tedy musejí být



konstruovány s ohledem na značný výkon. A jelikož mnohdy bývá zapotřebí dosti jemných posunů, je zřejmé, že nároky na spolehlivost zde budou vysoké.

Pohon tahů je zajištěn zpravidla motorově. V současné době je hojně využíváno asynchronního elektromotoru pohánějícího buben pro navíjení závěsného lana. Rychlost motoru je regulovaná pomocí frekvenčního měniče. Toto řešení (je využíváno i firmou Elseremo) se v posledních letech ukazuje být výhodné i díky přijatelné ceně. Lze se též setkat s hydraulickým motorem (lineárním nebo rotačním), dříve i s motory stejnosměrnými. Součástí každého tahu jsou i bezpečnostní prvky jako brzda (bývá zdvojená) a senzory v mezních polohách. Stále může být ale užitečný i ruční pohon. A to ať při nějaké speciální potřebě přesného nebo rytmického spuštění předmětu třeba podle hudby, tak třeba v případě náhlého problému či nehody. Motorové tahy proto obsluze též nabízejí možnost ručního pohonu. Méně často, spíše na menších scénách, se i dnes můžeme setkat s ryze ručními pohony. [2, 3]



Obrázek 2.1 Horní mechanizace - vlevo prospektové tahy nad jevištěm [1],  
vpravo pohony v provazišti [4]

## 2.2 Dolní mechanizace

Má své místo v dolní části scény, v své podstatě spoluprotví podlahu jeviště. Prvky dolní mechanizace společně uzpůsobují plochu a prostor na jevišti. Opět lze toto zařízení rozdělit do několika skupin: [1]

- **Zvedací stoly** – realizují vertikální pohyb ve spodní oblasti jeviště, jsou schopny vyvýšit se nad úroveň podlahy, rozšiřují variabilitu uspořádání objektů na jevišti nebo mohou tvořit reliéf podlahy jeviště.
- **Propadla** – také realizují vertikální pohyb, ale zajišťují spíše spojení mezi prostorem pod jevištěm a úrovní hrací plochy, umožňují vyjetí herce či jiného objektu na hrací plochu a zpět pod jeviště.
- **Točny** – realizují otáčivý scénický pro nejrůznější scénografické potřeby.
- **Jevištní vozy** – realizují horizontální pohyb po jevišti. Bývají umístěny obvykle na bočních jevištích a pomocí lan mohou pohybovat s objekty na scéně.

K pohonu propadel bývá podobně jako u tahů horní mechanizace použito asynchronních elektromotorů a k jejich regulaci frekvenčního měniče, přičemž může podporován i neregulovaný režim pohybu. Aktuální poloha zařízení je snímána inkrementálním snímačem polohy. Součástí

zařízení musí nezbytně být bezpečnostní prvky, senzory hlídající, aby zařízení při pohybu nepřekročilo povolené meze, a jiné.

Zvedací stoly bývají poháněny hydraulickými motory, ruční pohon je zde prakticky nemyslitelný už kvůli tomu, že nosnost bývá zpravidla v řádu několika tun. Rychlost pohybu stolů bývá nízká, obvykle maximálně několik dm/s, což vyplývá z jejich účelu.

Točny mohou být buďto jednoduché, čili kruhové, anebo složené, tedy prstenek a kruh uvnitř něho. Rozměry bývají rozličné v závislosti na celkové velikosti jeviště. K pohonu slouží obvykle též asynchronní elektromotory. Jevištní vozy mohou být, v případě menších scén, poháněny ručně. [3]



Obrázek 2.2 Dolní mechanizace - vlevo zvedací stoly, vpravo mechanismus propadla [4]

## 2.3 Bezpečnostní požadavky na jevištní techniku

V oblasti automatizace, zejména v té oblasti automatizace, kde figurují zařízení větších rozměrů a tím pádem i vyšší hmotnosti, jsou na bezpečnost přirozeně kladeny vysoké nároky. Jako příklad poslouží automatizace v průmyslu. Ať už u těžkého průmyslu, třeba v lomu nebo i u montážní linky ve výrobní hale – všude hrozí vysoké riziko ujmy na zdraví. Je ale potřeba vzít v potaz, že v jakémkoli odvětví průmyslu je s automatizovaným zařízením spojena celá řada jak vizuálních, tak zvukových varovných mechanismů. Obsluhující pracovníci nejenže jsou patřičně proškoleni, navíc jsou před jakýmkoli nebezpečím vždy důsledně varováni.

Pokud se nyní přeneseme do prostředí divadelního jeviště, nemůže nám ujít že kulisy a nejrůznější další vybavení, se kterým se zde manipuluje, si se svými rozměry a hmotností nic nezadá se zařízením výrobní linky v továrně. Ovšem divadlo je stánkem umění. Herec stojící na jevišti uprostřed svého výstupu je soustředěn na svůj výkon a na fakt, že neustále kolem něho ba dokonce vysoko nad jeho hlavou zavěšené čeká potenciální nebezpečí, nemá v tu chvíli myšlenky. Hlavní odpovědnost za bezpečí lidí, ale i materiálu pohybujícího se na divadelní scéně, nesou osoby obsluhující jevištní techniku a k úspěchu by jim měl pomáhat důmyslný řídicí systém.

Chyby a potenciální nebezpečí havárie může souviset už se samotným hardwarem. Některým těmto nebezpečím lze předejít instalací bezpečnostních prvků jako snímačů mezních poloh, zdvojených brzd, záchranných stop tlačítek, nastavením frekvenčního měniče tak, aby v případě přetížení tahu motor okamžitě zastavil, atp. U velké části těchto chyb by tedy při jejich výskytu mělo postačit jako opatření okamžité zastavení činnosti zařízení. Používá se také tzv. bezpečnostní stopkový okruh, což představuje systém stop tlačítek příhodně rozmístěných po prostoru jeviště a okolí. [3]

Řídicí systém by měl být schopen nejen tyto, ale i další chyby detekovat a informovat o nich obsluhu, aby na ně bylo možné nějakým způsobem zareagovat. Tento úkol je mnohdy nelehký, některé chyby bývá velmi obtížné detekovat, a proto je třeba při návrhu řídicího systému této problematice věnovat pozornost. Typickým a dnes již tradičním opatřením proti výpadkům systému je zdvojení řídicího systému (Takové řešení je využito i v řídicím systému firmy Elseremo). První z možných režimů je potom takový, že pracuje jeden ze systémů a druhý kontroluje činnost toho prvního. Druhá možnost je, že pracuje jeden systém a druhý je v nečinnosti a slouží tak jako systém záložní. [2]

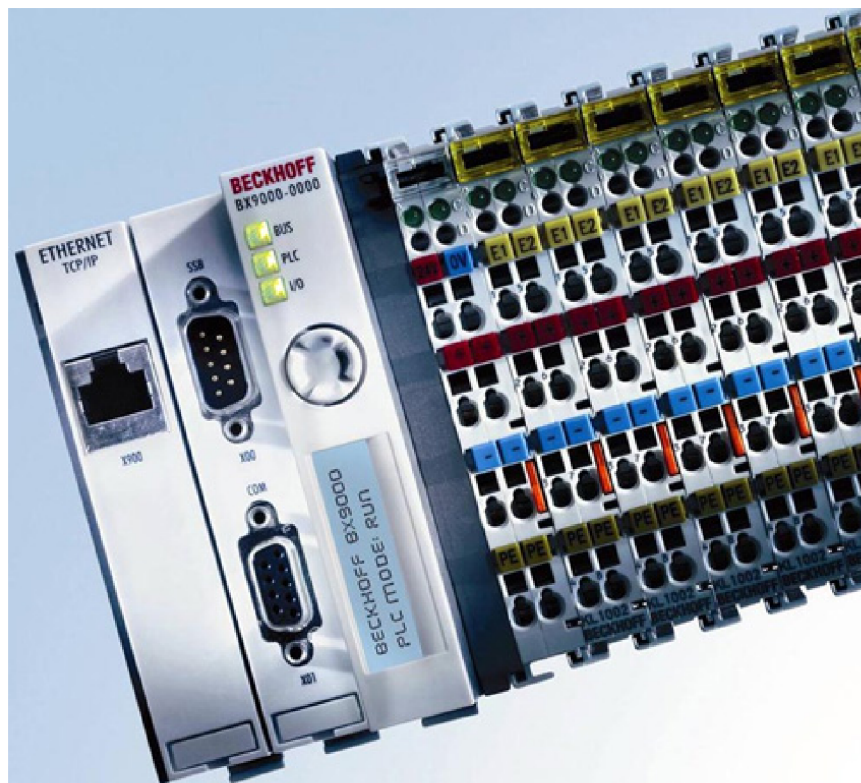
### 3 Řízení divadelní techniky

Tato kapitola popisuje obecné možnosti v oblasti řídicích systémů. Po úvodním přehledu existujících typů řídicích systémů následuje stručný popis principů hardwarového řešení a technologií používaných v rámci divadelních řídicích systémů. Jelikož předmětem této práce je uživatelské rozhraní, jsou dále pak probírána používaná uživatelská rozhraní a příklady z praxe.

V současné době je vesměs využíváno řídicích systémů zpětnovazebních, které nějakým způsobem reagují na podněty zpětné vazby z daného zařízení. Tyto systémy lze podle principu jejich činnosti členit na: [5]

- **Logické řídicí systémy** – získávají informace o aktuální situaci na zařízení pomocí snímačů polohy či teploty nebo jiných senzorů a reagují na ně prostřednictvím logických obvodů zpracovávajících binární informace přicházející ze senzorů.
- **Sekvenční řídicí systémy** – řízení a reagování na podněty probíhá podle předem vytvořené sekvence akcí.
- **Zpětnovazební** – pomocí zpětné vazby vytváří kontrolní signál, matematicky založený i na dalších proměnných s přihlédnutím k udržení řízeného procesu v přijatelných mezích.

Tyto principy lze vzájemně kombinovat, přičemž zřejmě větší praktický užitek přináší řešení logického řídicího systému. Čistě sekvenční přístup v současné praxi mnoho uplatnění nenalézá. Vedle tohoto členění lze ŘS dělit i z pohledu technologické realizace.



Obrázek 3.1 PLC značky Beckhoff - řídicí modul s připojenými dalšími moduly  
[<http://news.thomasnet.com/images/large/484/484222.jpg>]

Prvně Jsou to PLC (Obrázek 3.1), tedy programovatelné logické automaty. Jedná se o digitální elektronická zařízení s programovatelnou pamětí a digitálními i analogovými vstupy a výstupy, jež jsou schopná řídit činnost rozličných typů zařízení. Program je vykonáván cyklicky. PLC existují v kompaktním provedení (CPU, vstupy a výstupy jsou umístěny v jednom modulu) nebo v provedení modulárním. U druhé varianty jsou pro CPU, napájecí zdroj a vstupy i výstupy vyhrazeny samostatné moduly a lze připojit i další moduly v závislosti na nabídce výrobce daného PLC a na potřebách konkrétního systému. Jedná se pak o jakousi stavebnici, která přináší vysokou variabilitu. Uživatelské rozhraní pro PLC je označováno zkratkou HMI, tedy Human-Machine Interface. Běžná je realizace této interakce prostřednictvím ovládacích panelů, často s dotykovým displejem a dalšími ovládacími prvky. Tyto systémy se vyznačují spolehlivostí a relativně nízkou cenou. Ovšem naproti tomu programátorský komfort je poměrně nízký.

Další používanou technologií, kterou je třeba zmínit, je technologie poměrně moderní. Průmyslové počítače (IPC) uzpůsobené tak, aby splňovali nutné bezpečnostní normy a vysoké nároky na spolehlivost, slibují vysoký programátorský i uživatelský komfort, neboť jejich architektura je s běžnými počítači příbuzná. IPC, i přestože jsou několikanásobně dražší než běžné počítače, představují dnes levnější variantu k PLC a do budoucna se dá očekávat, že jim budou konkurovat stále silněji.

Poslední technologickou variantou, kterou zmíním, jsou SCADA systémy (Supervisory Control And Data Acquisition). Obvykle představuje prostředky pro většinou centralizované řízení a správu rozsáhlých komplexních systémů. S centrálním řídicím systémem spolupracuje uživatelské rozhraní (označované HMI, z angl. Human-Machine Interface) a dále tzv. RTU (Remote Terminal Unit), což jsou většinou nejrůznější senzory pro sběr dat. Toto vše je propojeno důmyslnou komunikační infrastrukturou. [2]

## 3.1 Prvky a principy řídicího systému divadla

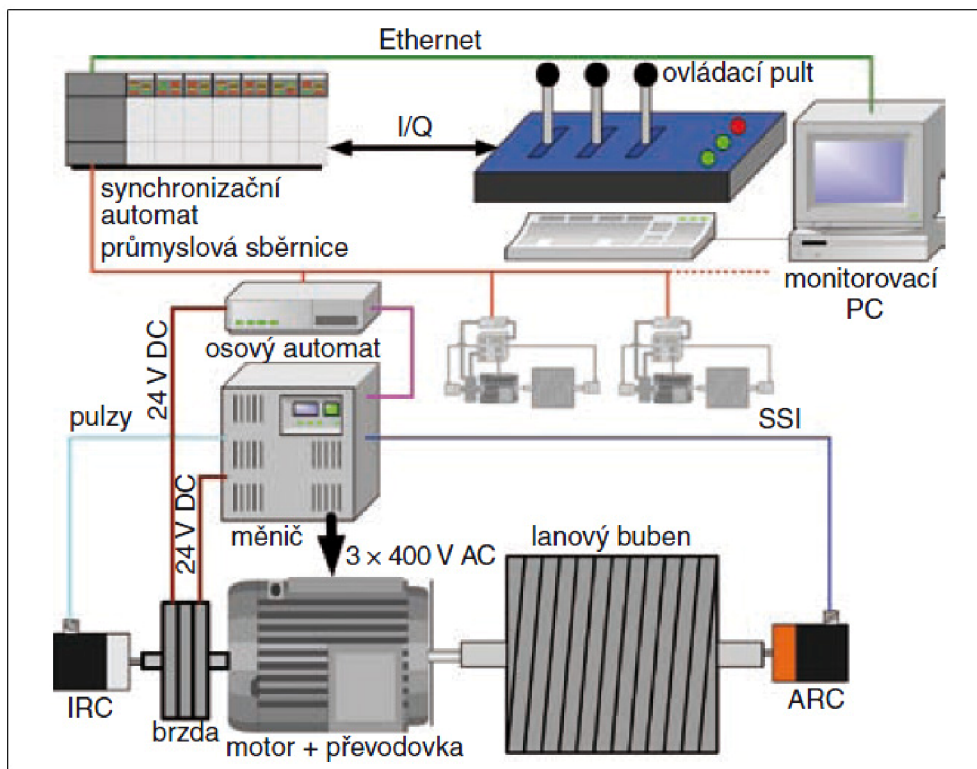
Technické zařízení divadelní scény popisuje kapitola 2. Toto zařízení je potřeba nějak řídit. Nabízejí se dva základní přístupy, a to manuálně nebo automatizovaně. K manuálnímu ovládní bývá používána soustava ovládacích prvků jako tlačítka, potenciometry a podobně. Avšak tento způsob ovládní v současnosti obvykle plní spíše úlohu záložního řešení. Nás z pohledu této práce zajímá spíše řízení automatizované.

Z hlediska struktury řízení rozlišujeme automatizované divadelní řídicí systémy centralizované a distribuované. V řízení scény lze rozlišit několik úrovní [3]:

1. Vlastní pohon
2. Řízení jednoho pohonu
3. Řízení skupiny pohonů
4. Řízení představení
5. Uživatelské rozhraní

Centralizace je myšlená v tom smyslu, že obvykle body 2 až 4 jsou sloučeny do jednoho centrálního řídicího systému. Dokonce v některých případech zde může být zakomponované i uživatelské rozhraní. Veškerý řídicí výkon je tak soustředěn na jednom místě. V poslední době jsou v praxi více prosazovány i distribuované systémy, kdy jednotlivé úrovně řízení jsou umístěny do různých hardwarových uzlů. Klíčovou rolí zde hraje komunikace mezi jednotlivými prvky, která musí

být maximálně spolehlivá. Možností jakým způsobem sloučit jednotlivé úrovně do hardwarových celků je skutečně mnoho. Jeden z možných návrhů je vidět na Obrázku 3.2.



**Obrázek 3.2** Možné schéma distribuovaného řízení tahů horní mechanizace [3]

Pokud jde o technologickou realizaci divadelního ŘS, je třeba odkázat na přehled v úvodu této kapitoly. V současnosti jsou divadelní řídicí systémy často postaveny na technologii PLC. Příkladem může být systém společnosti Elseremo, který využívá PLC značky Beckhoff. Pro připojení každého zařízení je zde vyhrazen samostatný modul připojený k modulu s řídicí jednotkou CPU. Toto jako celek tvoří veškerou logiku systému. K PLC jsou taktéž připojeny ovládací panely (může jich být i více), které představují uživatelské rozhraní pro ovládání systému. Využívány jsou ovládací panely buďto v provedení stacionárním, situovaném na pevném místě, které poskytuje dobrý přehled pro ovládání scény, nebo v provedení přenosném. Komunikace mezi těmito prvky probíhá pomocí sběrnice EtherCAT, což je klon Ethernetu používaný značkou Beckhoff a několika dalšími společnostmi. Obsluha má též možnost zařízení od řídicího systému odpojit a ovládat ho samostatně prostřednictvím ovládacího panelu přímo na zařízení. Zařízení, ať už tahy horní mechanizace nebo propadla mechanizace dolní, jsou opatřena inkrementálními snímači polohy pro zjišťování aktuální pozice závěsu lana resp. plošiny propadla a senzory hlídajícími mezní polohy zařízení. To je nezbytné pro zajištění zpětné vazby pro řídicí systém, který pak situaci vyhodnocuje a zprostředkovává informace obsluze. [2, 6]

Ovšem stále více jsou využívány i průmyslové počítače, alespoň pro realizaci centrálních řídicích systémů. Zde bych jako příklad zmínil Německého výrobce BBH Systems. Jejich systém již využívá IPC a užívají i decentralizovaného přístupu, kdy prostřednictvím Ethernetu a Switche lze k centrálnímu IPC s řídicím systémem připojit až 216 dalších boxů spravujících jednotlivá zařízení. [7]

Zajímavé řešení a prvky řízení jevištní techniky nabízí i britská firma Stage Technologies. Z jejich nápadů můžeme jmenovat třeba automatické nebo rádiem ovládané jevištní vozíky či software pro simulace a 3D vizualizace. [8]

## 3.2 Uživatelské rozhraní

Uživatelská rozhraní divadelních ŘS jsou zpravidla tvořena ovládacími pulty, na němž běží softwarová aplikace, umožňující ovládání pohonů a často nabízí přehlednou vizualizaci jejich pohybů a stavů. Firma Elseremo nabízí dva základní typy pultů. Mobilní variantu představuje ovládací pult AMBIT (viz. Obrázek 3.3). Je vybaven kolébkovými ovladači umístěnými vedle dotykového LCD displeje. Nahoře se nachází tlačítko TOTAL STOP pro zastavení všech zařízení v případě hrozícího nebezpečí. Na boční straně je port pro vstup USB klíče sloužícího k zabezpečení systému proti používání nepověřenou osobou. Stacionární variantou je pak pult ALPHA STAT. Ten navíc nabízí mimo jiné klávesnici a namísto kolébkových ovladačů jsou zde regulérní ovládací páky s bezpečnostním hardwarovým zámek. Ty jsou, stejně jako v případě mobilního pultu, jednoosé s nulovou polohou uprostřed. Ke stacionárnímu pultu je také možné připojit i druhý displej. [6]



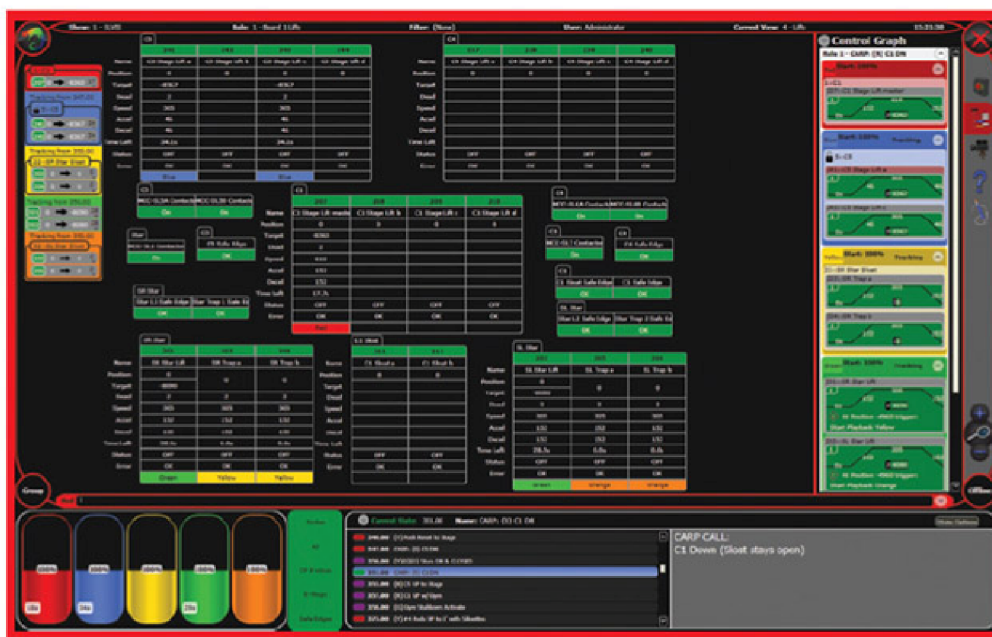
Obrázek 3.3 Mobilní ovládací pult AMBIT od firmy Elseremo

Britská Stage Technologies má v nabídce dokonce více různých pultů a ovládacích zařízení. Buď jsou také vybavena displeji, některá i více než jedním, anebo jsou zcela bez displeje. Na obrázku 3.4 si lze prohlédnout pult Akrobat G6 se dvěma 20" monitory a z hlediska bezpečnosti a spolehlivosti je užitečná i přítomnost dvou procesorů a záložního druhého zdroje napájení. [8]



**Obrázek 3.4 Ovládací pult Acrobat G6 od firmy Stage Technologies**

Vlastnosti a principy aplikací běžících na ovládacích pultech se taktéž u různých výrobců liší. Opět uvedu jako příklady značky Stage Technologies a Elseremo. Britský výrobce v současnosti instaluje na své ovládací panely software eChameleon. Ten umožňuje náhled na scénu ve 3D či v tabulkové podobě nebo vizualizaci v reálném čase, pomocí těchto prostředků může obsluha se zařízeními připojenými do systému pohodlně pracovat a stále má přehled o jejich činnosti. 3D model scény lze importovat z některého CAD systému nebo lze rychle vytvořit jednoduchý model pomocí univerzální vestavěné knihovny standardních objektů. Pro pokročilou obsluhu je zde k dispozici i rozhraní příkazového řádku pro rychlé programování pohybů. [8]



**Obrázek 3.5 Ukázka ovládací aplikace eChameleon firmy Stage Technologies v tabulkovém zobrazení**



Aplikace Elserema je také poměrně všestranná. Poskytuje přehlednou nabídku dostupných zařízení a taktéž umožňuje vizualizaci jejich pohybu v reálném čase. Poskytuje širokou paletu synchronních i asynchronních režimů pohybu zařízení ve skupině. Umožňuje vytváření scénářů, tedy předem připravených po sobě jdoucích akcí, typicky pohybů zařízení, postupně v čase. Systém je navíc možno ovládat za pomoci bezdrátového ovladače, který najde využití pro některé jednodušší úkony, umožňuje spíše základní možnosti ovládání. Jako zabezpečení zde slouží autentizace uživatele za pomoci USB klíče, který musí být v ovládacím pultu vložen po celou dobu manipulace. [6] Systémy Stage Technologies i Elserema běží na platformě operačního systému Windows Embedded.

Krom ovládání prostřednictvím stacionárních či mobilních panelů lze využívat i alternativní způsoby kontroly nad divadelním zařízením. Obvykle bývá možné pohony řídit i pomocí místního ovládacího panelu umístěného přímo na zařízení. Toto může být užitečné třeba při výpadku řídicího systému nebo i v případě potřeby rychlé manipulace s konkrétním zařízením ať už z jakéhokoli důvodu. Součástí uživatelského rozhraní by měli být samozřejmě i bezpečnostní tlačítka pro okamžité zastavení všech pohonů rozmístěná na strategických místech jeviště i zákulisí [3] a v praxi se mohou vyskytovat i další ovládací prvky.

## 4 Uživatelská rozhraní

Člověk s počítačem nebo obecně se strojem komunikuje a ovládá jej pomocí určitých prostředků. Souhrnně se tyto prostředky nazývají uživatelské rozhraní, též známé pod zkratkou UI, z anglického user interface. Pojem uživatelského rozhraní se tedy týká strojů obecně, nicméně předmětem této práce je uživatelské rozhraní počítačové aplikace (byť doplněné o hardwarové prvky jako např. ovládací páky), proto se tato kapitola zaměřuje zejména na uživatelská rozhraní počítačů a počítačových aplikací.

Uživatelská rozhraní lze dělit na UI orientovaná na stroj a na UI orientovaná na uživatele. [9] Princip strojově orientovaného UI je postavený na principech činnosti daného stroje a příliš se nebere v úvahu lidský přístup a způsob uvažování. Takto orientovaná rozhraní tudíž často vyžadují určitou kvalifikaci obsluhujících pracovníků. Naproti tomu uživatelsky orientovaná UI jsou postavená na logickém lidském úsudku a jsou vytvářena se snahou udělat je co nejpřehlednější a uživatelsky přívětivé. Strojově orientovaná rozhraní nalézají uplatnění například v průmyslu, kde je často vhodnější provádět specifická nastavení stroje způsobem stroji blízkým a kde se zároveň počítá s tím, že obsluhující pracovník je pro obsluhu stroje patřičně proškolený. Naproti tomu rozhraní uživatelsky orientovaná je vhodná pro zařízení, se kterými přichází do styku obyčejný člověk. Mohou to být třeba přístroje v domácnosti, telefony, apod.

Dáme-li ale za příklad počítače, tak zástupce strojově orientovaných rozhraní bychom mohli najít v dobách, kdy bylo pokolení počítačů ještě v plenkách. Interakce mezi člověkem a počítačem tenkrát probíhala zprvu pomocí děrných štítků či magnetických pásek. Způsoby komunikace to byly poněkud krkolomné, ryze strojově orientované. Během dalších let se postupně začaly objevovat klávesnice a dokonalejší monitory. Stále však se dlouhou dobu jednalo pouze o rozhraní příkazové řádky. Později, s příchodem textového uživatelského rozhraní (TUI, z anglického Text User Interface), už lze hovořit o prvcích uživatelsky orientovaného rozhraní. Textové uživatelské rozhraní funguje na principu pevného rastru, v jehož polích se zobrazují jednotlivé znaky (viz. Obrázek 4.1). Toto rozhraní též umožňuje pomocí speciálních znaků sestavovat ovládací a členící prvky jako rámečky, posuvníky nebo tlačítka. Vzniká tak jakési grafické uspořádání jednotlivých prvků na obrazovce a tím se toto rozhraní stává na rozdíl od rozhraní příkazové řádky uživatelsky mnohem přívětivější. Jako příklady TUI lze uvést řadu programů běžících pod operačním systémem MS DOS, např. souborový manažer Norton Commander nebo textový editor T602 ale také třeba unixový textový editor Vim. [10]



Obrázek 4.1 Ukázka textového uživatelského rozhraní – souborový manažer Norton Commander

[<http://upload.wikimedia.org>]

Následně se objevila uživatelská rozhraní grafická (též GUI, z anglického Graphical User Interface, viz. kapitola 4.1), která přinesla skutečný rozvoj uživatelsky orientovaných rozhraní. A v dnešní době jdou možnosti UI ještě dále. Lidské smysly jako zrak, sluch či hmat jsou využívány stále důmyslněji. Rozvíjí se rozpoznávání objektů v obraze, tedy počítačové vidění. S tím souvisí možnost komunikovat s počítačem a sdělovat mu informace pomocí gestikulace rukou a pohybů těla. Počítač lze ovládat hlasem, ba dokonce nevidomí lidé mohou ke čtení výstupu využívat i hmat.

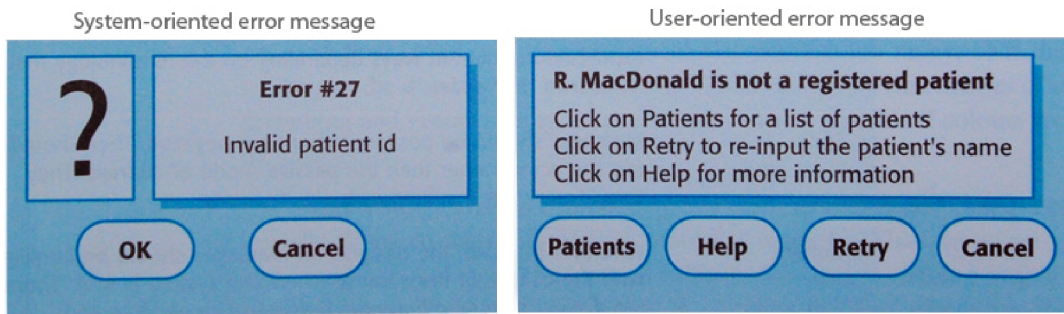
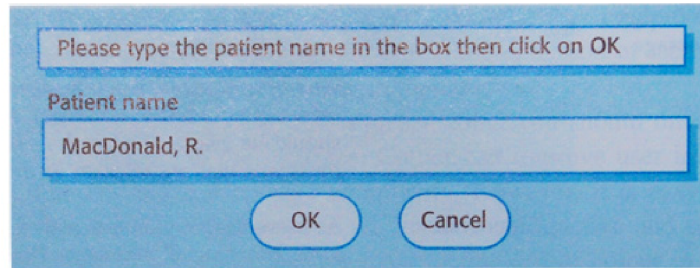
Zde pro ilustraci stručný chronologický přehled vývoje uživatelských rozhraní počítačů:

- **1. generace (40. léta 20. století)** – první počítače nepřinášeli příliš mnoho uživatelského komfortu. Používali se dřemé štítky či dřemé a později magnetické pásky. Komunikace se strojem prostřednictvím těchto způsobů k jeho obsluze vyžadovala poměrně vysoce kvalifikované pracovníky.
- **2. a 3. generace (50. až 70. léta 20. století)** – od dřemých štítků se upouštělo, začaly se objevovat monitory a klávesnice, nastává rozmach textového uživatelského rozhraní. Tím se obsluha počítače stala značně pohodlnější. Interakce mezi člověkem a počítačem se díky psanému vstupu z klávesnice a rychlé odezvě počítače pomocí monitoru stala bezprostřednější a přirozenější.
- **4. generace (70. až 80. léta 20. století)** – doba systém MS DOS. Éra textových UI vrcholí a brzy se již postupně začínají objevovat první vlašťovky grafických uživatelských rozhraní. S prvním GUI přichází roku 1984 firma Apple na svých počítačích Macintosh. Následujícího roku přichází se svou grafickou nástavbou pro systém MS DOS i firma Microsoft.
- **Současnost** – probíhá neustálé zdokonalování GUI. Konečně se ale též při vývoji UI, využívají další moderní způsoby komunikace člověka s počítačem, jako třeba jeho ovládání hlasem či gesty rukou.

Podrobný rozbor všech typů a možností uživatelských rozhraní zřejmě není předmětem této práce, a tak obsahem následujících podkapitol je tudíž přiblížení problematiky uživatelských rozhraní grafických, poukázání na zásady, které je třeba vzít v úvahu, na současné trendy, prvky moderních GUI a na směr, kterým se jejich vývoj ubírá.

## 4.1 Současné trendy

Na uživatelském rozhraní stojí možnosti interakce uživatele se softwarem a celkový komfort při jeho ovládání. Návrh UI proto představuje velmi podstatnou část celého procesu návrhu softwaru. Dnes dominují rozhraní grafická. Pomocí grafického výstupu lze uživateli předat v jednom okamžiku větší množství informací a to navíc, pokud jsou správně dodržovány určité zásady, tak i poměrně efektivně a přehlednou formou. I grafické uživatelské rozhraní lze navrhnout buďto více strojově orientované (zde možná vhodnější označení – systémově orientované) nebo spíše uživatelsky orientované. Může to být vidět například na chybových hlášeních systému. Dodržovat zásady uživatelsky orientovaného rozhraní znamená, že tyto informace budou nanejvýš jasné, srozumitelné a lidsky podané, pokud možno konkrétní a především konstruktivní (příklad viz. Obrázek 4.2).



**Obrázek 4.2** Systémově vs. uživatelsky orientovaný přístup. Nahoře dialog pro vstup, vlevo systémově orientované oznámení o chybném vstupu, vpravo uživatelsky orientovaná varianta [11].

Designér grafického uživatelského rozhraní, má-li být výsledek jeho práce kvalitní, si musí uvědomit několik faktů: [11]

- Lidé mají omezenou kapacitu krátkodobé paměti – v průměru je člověk schopný operovat asi se sedmi různými informacemi. Čili prezentovat uživateli příliš mnoho informací v jednom okamžiku nemá smysl a spíše to přispívá k nepřehlednosti.
- Každý máme jiný rozsah smyslového vnímání a každý reagujeme na podněty trochu jinak. Někdo vidí barvy jasněji, někdo je rozlišuje hůře. Prvky, které někdo přehlédne, mohou jinému odvádět pozornost. Designér UI nemůže při návrhu vycházet pouze ze svých dispozic a spoléhat, že ostatní se přizpůsobí.
- Každý preferujeme jiné způsoby interakce. Někdo raději pracuje s obrázky, jiný s textem. Pro někoho je přirozená přímá manipulace s objekty či interakce založená na zadávání příkazů systému.

S přihlédnutím k těmto faktům lze popsat některé základní styly interakce mezi uživatelem a systémem prostřednictvím grafického uživatelského rozhraní. Přehled těchto stylů je k dispozici v Tabulce 4.1.

Styl interakce	Hlavní výhody	Hlavní nevýhody	Příklady
Přímá manipulace	Rychlá a intuitivní interakce, snadná na naučení	Může být náročné na implementaci. Použitelné pouze pokud pro danou úlohu existuje nějaká vizuální metafora.	Počítačové hry, CAD systémy
Výběr z nabídky	Předchází uživatelským chybám díky malé nutnosti psaní	Zdlouhavé pro zkušené uživatele. Příliš komplikovaná nabídka se může stát nepřehlednou.	Nejběžnější aplikace různých typů
Vyplňování formuláře	Přirozený záznam dat	Často v okně zaujímá značný prostor. Může nastat problém s formátem vkládaných dat.	Databázové aplikace
Příkazový jazyk	Výkonný a flexibilní	Složitě na naučení, nedostatečná režie chybových stavů.	Operační systémy, řídicí systémy
Přirozený jazyk	Přístupné náhodným uživatelům, snadno rozšiřitelné	Vyžaduje více psaní. Systémy pro zpracování přirozeného jazyka nejsou úplně spolehlivé.	Systémy pro získávání informací

**Tabulka 4.1** Styly interakce mezi uživatelem a systémem, výhody a nevýhody jednotlivých stylů [11]

Člověk má ze svého života zaryté určité přirozené úkony. Třeba když chce přemístit mísu ze skříně na stůl, tak pro ni jednoduše natáhne ruku, uchopí ji, položí ji na stůl a pustí. V současnosti je trendem se při návrhu uživatelských rozhraní těmito zažitými zvyklostmi nechat inspirovat. Uživatel tak potom nemusí tápat a při obsluze aplikace může jednat intuitivně. Od onoho příkladu s mísou byla ve světě uživatelských rozhraní odvozena metoda „Drag and Drop“, v češtině tedy „Uchop a táhni“. Toto bývá použito typicky pro přesouvání nejrůznějších grafických objektů či elementů reprezentujících určitá data. S tímto principem inspirace reálným světem pak souvisí pojem „Look and Feel“, který zastupuje myšlenku, že uživatel by se měl „jen podívat“ a hned „cítit jak na to“. Uživatelská přívětivost nemá smysl pouze pro uživatele – laika. Uživatelská přívětivost usnadňuje práci i uživateli zkušenému a jeho práce s prostředím se tak stává rychlejší a efektivnější. [9]

S oblibou se též v moderních uživatelských rozhraních využívá multimédií. Ty totiž prostřednictvím více druhů médií umožňují, aby uživatel zapojil více smyslů najednou a tím prohloubil vjem přijímaných informací. A objevují se i další způsoby zpestření. Neustále rostoucí výkon počítačů v blízké době slibuje i běžnému uživateli např. využití prostorových grafických rozhraní, která přinášejí snadnou a přirozenou obsluhu a schopnost sdělovat naráz ještě více informací a jejich vjem tak prohloubit ještě dále.

## 4.2 Prostředky pro implementaci

K úspěšné implementaci uživatelského rozhraní vede několik více či méně efektivních cest: [9]

- **Použití standardního programovacího jazyka** – jedná se o jakýsi nejobecnější způsob implementace. K implementaci GUI je využit stejný jazyk jako pro implementaci zbytku systému, ať už je to jazyk Pascal, C nebo jiný. V některých případech je tento jazyk podpořen specializovanou knihovnou, která již obsahuje implementace prvků rozhraní. Tato cesta i přes možnost přesné realizace požadavků aplikace je časově náročná a nepříliš efektivní.
- **Použití specializovaného programovacího jazyka** – pro takový speciální účel, jako je tvorba GUI existují speciální programovací jazyky. Například jazyk Tcl s knihovnou Tk (známé jako Tcl/Tk), prostředky pro tvorbu GUI nabízí také JAVA v podobě knihovny SWING. Standardně tyto jazyky nepodporují grafickou editaci rozhraní. Na druhou stranu ale obvykle mají výhodné vlastnosti jako platformní nezávislost a snadnou modularizaci. Bývají dostupné i programové nadstavby, které zjednodušují tvorbu rozhraní v daném jazyku a některé nabízejí dokonce i grafický editor rozhraní. Tvorbu GUI pomocí těchto prostředků lze ve srovnání s použitím standardního programovacího jazyka považovat jistě za efektivnější a vhodnější variantu.
- **Integrované prostředí pro vývoj aplikací s GUI** – jde často o robustní systémy, které v sobě integrují jednak editor zdrojového kódu v jazyce, na který jsou orientované a jednak grafický editor GUI. Integrují tak v sobě prostředí, ve kterém je programátor schopen pohodlně navrhnout a editovat grafické uživatelské rozhraní a zároveň vytvořit zdrojový kód pro samotné jádro aplikace zajišťující její funkčnost. Navzdory veliké robustnosti se dá říci, že jejich užití představuje velmi efektivní cestu k úspěšné implementaci zdařilého GUI. Zpravidla jsou tyto prostředí postaveny na některém konkrétním programovacím jazyku. Pro jazyk Pascal poslouží za příklad prostředí Borland Delphi. Firma Microsoft pak nabízí své vývojové prostředí Visual Studio, které v sobě integruje prostředí postavené na C++, C#, Basic (zde Visual Basic) a dalších jazycích.
- **Další možné cesty implementace** – jmenovat lze ještě například programovací jazyky aplikačních programových systémů. Některé aplikační systémy nabízejí svůj integrovaný programovací jazyk. Databázový systém Access integruje své prostředí pro tvorbu grafických formulářů zprostředkovávající vstup a výstup dat do/z databáze. Balík kancelářských aplikací MS Office obsahuje jazyk Visual Basic for application (VBA), který bohatě rozšiřuje možnosti těchto aplikací. Další cestu představují automaticky generovaná rozhraní. Ovšem toto lze použít pouze v omezené míře specifických případů.

V oblasti implementace GUI se v současnosti staly trendem specializované programovací jazyky, popř. celá integrovaná vývojová prostředí. Tyto prostředky obvykle pro implementaci grafického rozhraní nabízejí nástrojové sady (tzv. toolkit). Při tvorbě je pak nutné postupovat v souladu se vzory, které do návrhu sady nástrojů vnesl její tvůrce. Každý toolkit definuje své aplikační programátorské rozhraní (API) a množinu návrhových pravidel, které je nutné znát. Zvládnutí nové nástrojové sady může být dokonce obtížnější, než zvládnutí nového programovacího jazyka. Z toho důvodu je snaha tvořit standardy na základě nástrojových sad, které jsou dostupné pro více programovacích jazyků (např. .NET). [12]

## 5 Uživatelské rozhraní pro řídicí systém

Tato kapitola ve své první části přináší zhodnocení současné situace v oblasti divadelní techniky a jejího řízení a postupně se dostává k motivaci pro vytvoření nového řídicího systému. Dále shrnuje zásady dobrého GUI a následně na začátku své druhé části kapitola analyzuje prostředky pro návrh a implementaci GUI, a tím se dostává k návrhu uživatelského rozhraní pro nový řídicí systém.

Již v předchozích kapitolách bylo řečeno, že řízení divadelní techniky je poměrně komplikované a vyžaduje proto adekvátní technologické řešení, zejména s ohledem na bezpečnost herců, rekvizit i zařízení samotného. Tyto bezpečnostní požadavky specifikuje několik norem (mezi ty základní patří zejména ČSN EN 954-1, ČSN EN 61511 a ČSN EN 61508). Další kritéria pak stanovuje například certifikace SIL3 (Safety Integrity Level 3), která je většinou již dnes v zemích Evropské unie standardem. Že činnost divadelního řídicího systému musí být bezpečná a musí se držet v mezích předpisů, je tedy jasné. Dalším kritériem je ale samozřejmě funkčnost, jdoucí ruku v ruce s efektivním a pohodlným uživatelským rozhraním. Funkční schopnosti systému by jistě měly jeho obsluze umožnit pokrýt požadavky scénografů, přičemž obsluze by k tomuto mělo zároveň co nejvíce napomáhat optimálně navržené uživatelské rozhraní.

Po celém světě působí v oblasti divadelní techniky celá řada firem, například německá firma BBH Systems GmbH nebo světově uznávaná Stage Technologies Ltd. z Velké Británie a také české Elseremo a.s. Pokud jde o funkční stránku, má systém každého výrobce svá specifika (i když vychází z podobných základních principů). Ovšem pokud jde o bezpečnostní požadavky, tak ty platí pro všechny stejně a pokud se chce výrobce prosadit například v EU, tak musí těmito kritériím bezvýhradně dostát. Pokud systém nespĺňuje patřičné bezpečnostní požadavky, firma ztrácí zakázky. To je například problém společnosti Elseremo, jejichž stávající systém nespĺňuje podmínky pro certifikaci SIL3, což je značnou překážkou v cestě na evropský trh.

Mezi výrobci jsou i rozdíly v použitém technologickém řešení. V poslední době se již hojně využívá architektury, kdy logika řídicího systému je zajištěna softwarem instalovaným na centrálním počítači (typicky nějaké IPC), případně s možností distribuce příslušných prvků logiky na počítače obsluhující přímo jednotlivá zařízení. Jak již bylo popsáno výše, jeví se tato technologie velmi výhodná jak z ekonomického hlediska, tak i z hlediska realizace a náročnosti instalace a údržby. Další cíl pro nový systém je tedy změnit technologický přístup – stávající řešení postavené na PLC nahradit technologií využívající IPC a na něm instalovaný software řídicího systému.

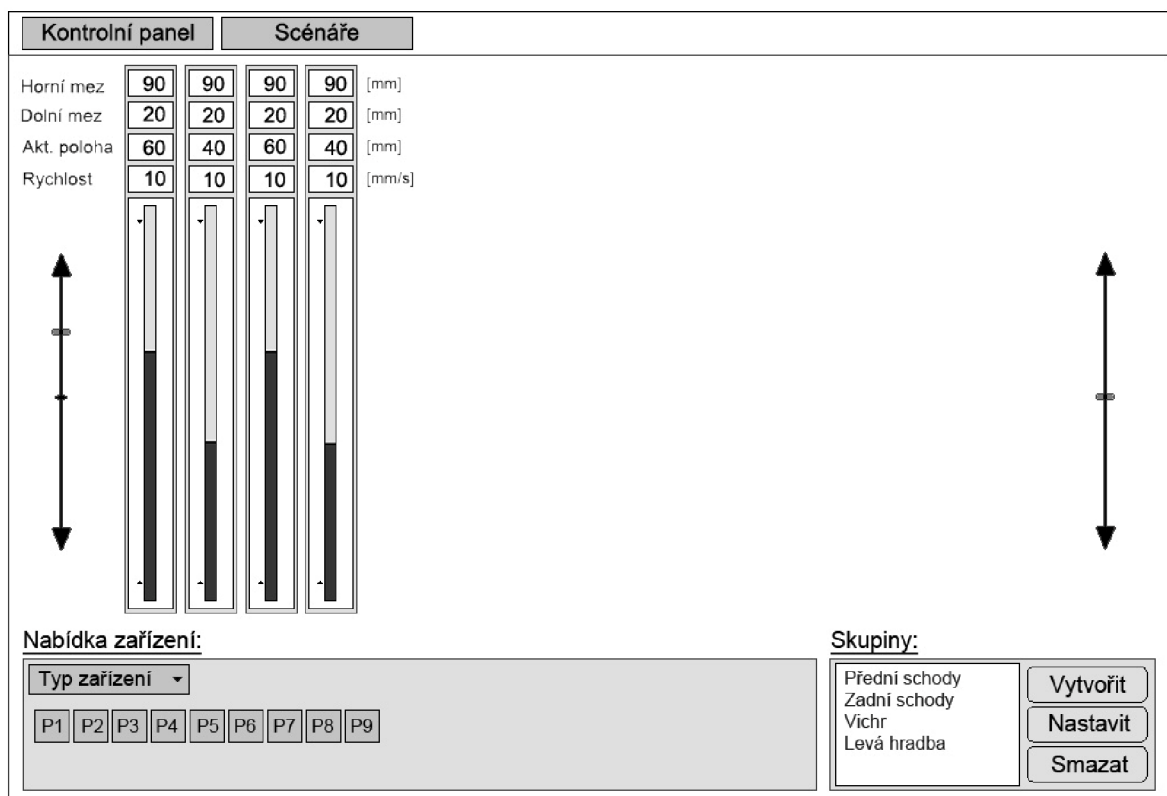
Smyslem této práce je implementovat jakousi první verzi uživatelského rozhraní, vizualizační aplikace. Tato první verze je určena především pro činnost spolu se simulátorem divadelní scény, který simuluje a graficky zobrazuje pohyb scénického zařízení podle požadavků UI. Činnost těchto dvou jednotek má za úkol koordinovat řídicí systém. Každá tato aplikace typicky běží na jiném počítači a komunikace mezi nimi probíhá prostřednictvím Ethernetu.

Nové GUI by mohlo být ve svém základním principu inspirováno stávající vizualizační aplikací společnosti Elseremo, neboť toto pojetí je funkční a v praxi osvědčené. Zároveň ale také by návrh nového prostředí měl korespondovat se zásadami a současnými trendy grafických uživatelských rozhraní, které popisují v předcházející kapitole. Aby aplikace splňovala požadavky Look and Feel, je třeba myslet mimo jiné na vhodné rozložení objektů v okně a volbu použitých barev. Objekty, které spolu logicky souvisí, by měly nějakým způsobem graficky ladit. Zároveň je užitečné myslet na pravidla kompozice a estetického dojmu. Mohl by zde být uplatněn i zlatý řez. Líbivé a dobře uspořádané prostředí totiž zaujme. Ovšem toto by mělo jít ruku v ruce s ergonomií a neměla by tím být potlačována funkčnost. Volba barev hraje také důležitou roli především proto, že člověk je ze života navyklý na určitou symboliku barev. Může to být symbolika odvozená z barev na

semaforu a dopravních značkách nebo může být využito faktu, že na vodovodní baterii červená představuje teplou a modrá studenou. Na základě nějaké z těchto a podobných symbolik lze barvám přiřknout určitý význam, což může výrazným způsobem pomoci prostředí uživatelsky příjemnit.

## 5.1 Návrh vizualizační aplikace

Jako vhodný jazyk pro implementaci řídicího systému bylo zvoleno C++ a stejný jazyk byl vybrán i pro implementaci vizualizační aplikace a uživatelského rozhraní. Existuje více možných prostředí a knihoven umožňujících implementaci GUI v C++. Snaha byla zvolit některé z integrovaných prostředí pro tvorbu GUI, která jsou dnes velmi oblíbená pro svůj výkon a efektivitu práce s nimi. Aplikace bude běžet pod operačním systémem Windows. Z toho důvodu nejsou kladeny zvláštní nároky na přenositelnost. Po zvážení těchto aspektů a po konzultaci s vedoucím práce jsem pro implementaci zvolil prostředí MS Visual Studio.



Obrázek 5.1 Návrh uspořádání hlavního okna vizualizační aplikace

Aplikace má zobrazovat nabídku dostupných zařízení a umožňovat jejich výběr a přiřazení k jedné ze dvou ovládacích pák. Pro páku by měly být implementovány dva stavy. V tom prvním by byla odemčena a přístupná přiřazování zařízení a nastavování jejich cílových poloh. Ve druhém stavu by pak byla uzamčena, bylo by zamezeno přidávání dalších pohonů a pohony pro páku vybrané by byly připraveny k činnosti a vychýlení páky v tomto režimu by znamenalo odeslání požadavku k pohybu. Zpětně aplikace musí zobrazovat aktuální stav a polohu zařízení prostřednictvím přehledné grafické vizualizace.

Těž zde má být podpora práce se skupinami zařízení. Obsluha by si měla moci vybrat libovolná dostupná zařízení, vytvořit skupinu a rozlišit ji názvem. Skupině má být možné nastavit



synchronizovaný pohyb a ten poté zahájit vychýlením ovládací páky. Aby možnosti nastavení tohoto pohybu poskytovaly dostatečnou volnost, měli by požadované polohy zařízení v rámci jejich synchronního pohybu být počítány matematickou funkcí, které bude možno nastavit potřebné parametry. Poslední parametr pak bude rychlost tohoto pohybu zadaná mírou vychýlení ovládací páky. Těchto základních matematických funkcí může být implementováno několik, výsledky výpočtů požadovaných poloh budou ovlivněny vstupními parametry funkce. Počítá se s dostupností vytvořených skupin pouze pro dané „sezení“. K vytváření trvalých skupin pohonů a sesazování jejich činností do posloupností například podle výstupů a scén divadelní hry by měli sloužit scénáře. Ty jsou ovšem v návrhu pouze výhledově pro implementaci v rámci dalšího pokračování této práce.

Na obrázku 5.1 je vidět návrh grafického uspořádání uživatelského prostředí. Ve spodní části okna by měla být situovaná nabídka dostupných zařízení filtrovaná dle jejich typu a také seznam skupin zařízení a tlačítka pro jejich vytváření a mazání. Na stranách jsou znázorněny ovládací páky a mezi nimi prostor pro vizualizaci pohybu jednotlivá zařízení s grafickým zobrazením aktuální pozice a také informacemi o cílových polohách či současné rychlosti pohybu.

## 5.2 Komunikace a princip činnosti

Komunikace ve směru OP -> ŘS	
INIT_REQUEST	Žádost Vizualizační aplikace o inicializaci seznamu zařízení. Odesílá se před zahájením činnosti aplikace nebo kdykoli je potřeba nová inicializace.
CHOOSE_ON_DESK	Zpráva o přiřazení zařízení ovládacímu pultu. Odesílá se při výběru zařízení na ovládací páku. Obsahuje identifikaci zařízení a ovládacího panelu a vykonanou operaci (vybráno / zrušen výběr).
MOTION	Příkaz k pohybu zařízení. Pokud se ovládací páka uzamkne, je opakovaně v pravidelných intervalech vysílán příkaz motion, který obsahuje informace o požadované poloze a rychlosti pohybu k této poloze.
LEAVING	Zpráva o ukončení činnosti ovládacího pultu posílaná řídicímu systému. Obsahuje ID pultu jako identifikaci.
Komunikace ve směru ŘS -> OP	
INIT_OK	Inicializace seznamu zařízení. Vysílá řídicí systém jako odpověď na žádost o inicializaci. Obsahuje seznam zařízení a informace o nich.
FEEDBACK	Zpětná vazba od řídicího systému. Vysílán neustále v pravidelných intervalech řídicím systémem. Obsahuje informace o aktuálním stavu všech zařízení. Vizualizační aplikace tak získává přehled o výběru zařízení na ostatních pultech, o chybách a v neposlední řadě o aktuálním stavu zařízení, jeho poloze a rychlosti.

Tabulka 5.1 Význam zpráv aplikačního protokolu pro komunikaci s řídicím systémem

Zásadní součástí vizualizační aplikace je její komunikační rozhraní. Jeho prostřednictvím by měla probíhat komunikace s řídicím systémem. Navrhnutý aplikační protokol je postaven nad transportním protokolem UDP, jež na rozdíl od protokolu TCP neztrácí čas tak vysokou režii a

potvrzováním přenášených dat. To je poměrně důležitý faktor, neboť zprávy zde budou odesílány v poměrně krátkých intervalech a proto je potřeba, aby komunikace probíhala co nejrychleji. Když nějaká zpráva nedorazí, není to příliš vážný problém, protože v příštím okamžiku je odeslána nová. Jelikož se počítá s tím, že ovládacích panelů je možné připojit více a teoreticky je možné zapojit i více řídicích systémů, jsou pakety vysílány jako UDP Broadcast. Z toho důvodu, pokud má komunikace úspěšně fungovat, je nutné, aby stanice s řídicím systémem i ovládací pult byli zapojeny ve stejné síti. Odeslanou zprávu tak může přijmout ta součást systému, která ji potřebuje. Tabulka 5.1 stručně popisuje význam jednotlivých zpráv v komunikaci mezi ovládacím pultem (OP) a řídicím systémem (ŘS). Podrobný popis aplikačního protokolu je pak k nalezení v Příloze A, kde je detailně popsán obsah všech možných zpráv i použité datové typy. Pro odchozí a příchozí komunikaci jsou využívány dva rozdílné implicitně přednastavené porty. Tyto porty by mělo však být možné změnit.

Dále popíší návrh samotné aplikace a princip její činnosti. Aplikace si před zahájením své činnosti vyžádá od ŘS inicializovaný seznam dostupných zařízení (odešle žádost INIT\_REQUEST). Jako odpověď dostane zprávu INIT\_OK, která obsahuje seznam dostupných zařízení a informace o nich. Poté tuto nabídku zpřístupní uživateli. Nabídka zařízení by měla být tříděná do třech základních skupin a sice na Horní mechanizaci, Dolní mechanizaci a Točny, zároveň se zobrazeným stavem zařízení (volné, vybrané zde nebo na jiném pultu, chyba). Uživatel může z této nabídky vybrat libovolné dostupné zařízení a přiřadit ho ovládací páce. V tu chvíli se řídicímu systému odešle zpráva CHOOSE\_ON\_DESK s informací o přiřazení zařízení danému stolu. Když ŘS tuto zprávu obdrží okamžitě tu informaci rozešle zpět všem pultům prostřednictvím zprávy FEEDBACK. Běžící aplikace na všech připojených pultech totiž neustále musejí dostávat aktuální informace o tom které zařízení je přiřazeno kterému pultu a obsluhujícího uživatele o tom informovat.

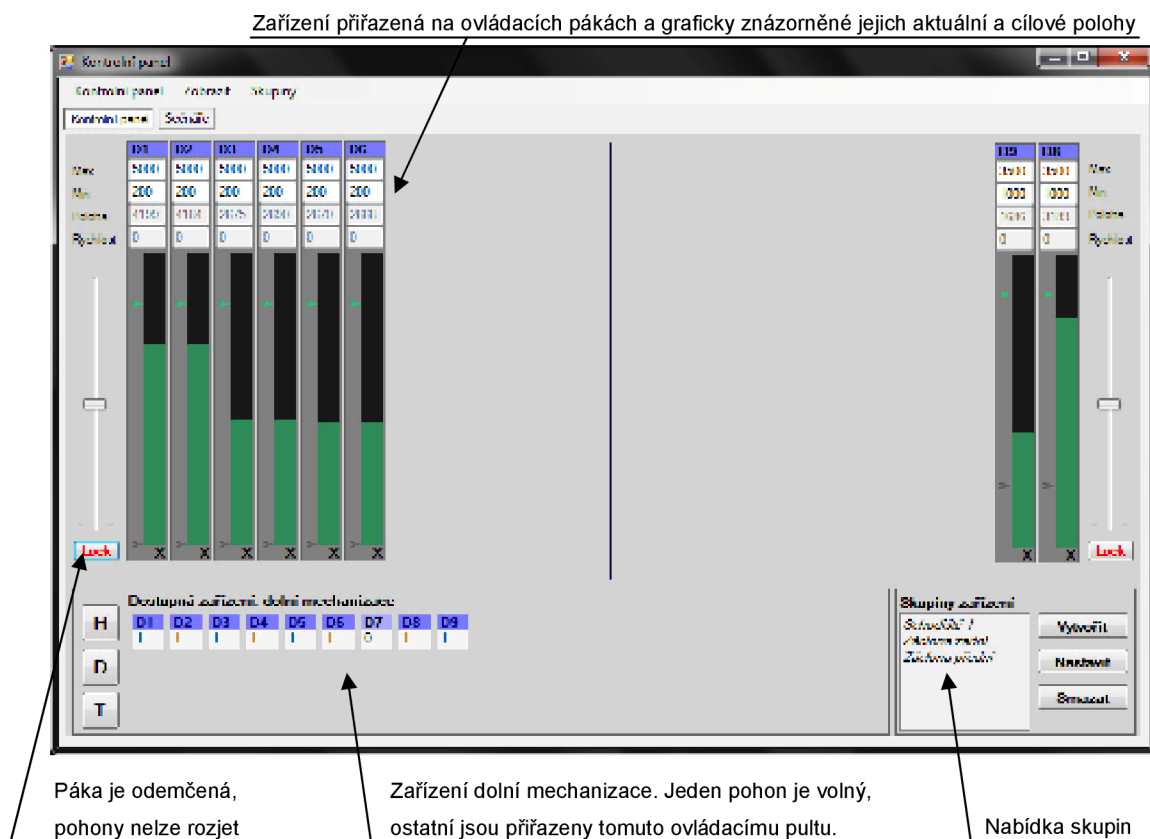
Po výběru zařízení na páku se zobrazí hodnota horní a dolní mezní polohy s možností tyto hodnoty změnit a zadat tak cílové polohy dle požadavků uživatele. Následně po tomto nastavení bude třeba páku uzamknout. Tím budou pohony uvedeny do aktivního stavu a v tu chvíli se začnou v pravidelných intervalech odesílat příkazy MOTION. Po uzamčení bude možné páku vychýlit a zadat tak požadovaný směr pohybu a mírou vychýlení také požadovanou rychlost. Do odesílaných příkazů MOTION se tím přidají informace o zvolené cílové poloze a požadované rychlosti, kterou mají pohony k této poloze dorazit. Aplikace zároveň neustále musí přijímat zprávy FEEDBACK, které krom informace o přiřazeném pultu obsahují i informace o aktuálním stavu zařízení, jeho aktuální poloze a rychlosti. Tyto informace budou obsluze taktéž zobrazovány a průběžně aktualizovány.

## 6 Implementace uživatelského rozhraní

V této kapitole se dostávám k implementaci nového uživatelského rozhraní. Nezabývám se zde všemi oblastmi implementace ale podrobněji rozebírám pouze několik těch podstatných. Nejdříve popisuji obecně strukturu celé aplikace a uživatelské prostředí. Poté se zaměřuji na komunikační rozhraní a na něho navazuje popis implementace ovládání a pohybu zařízení.

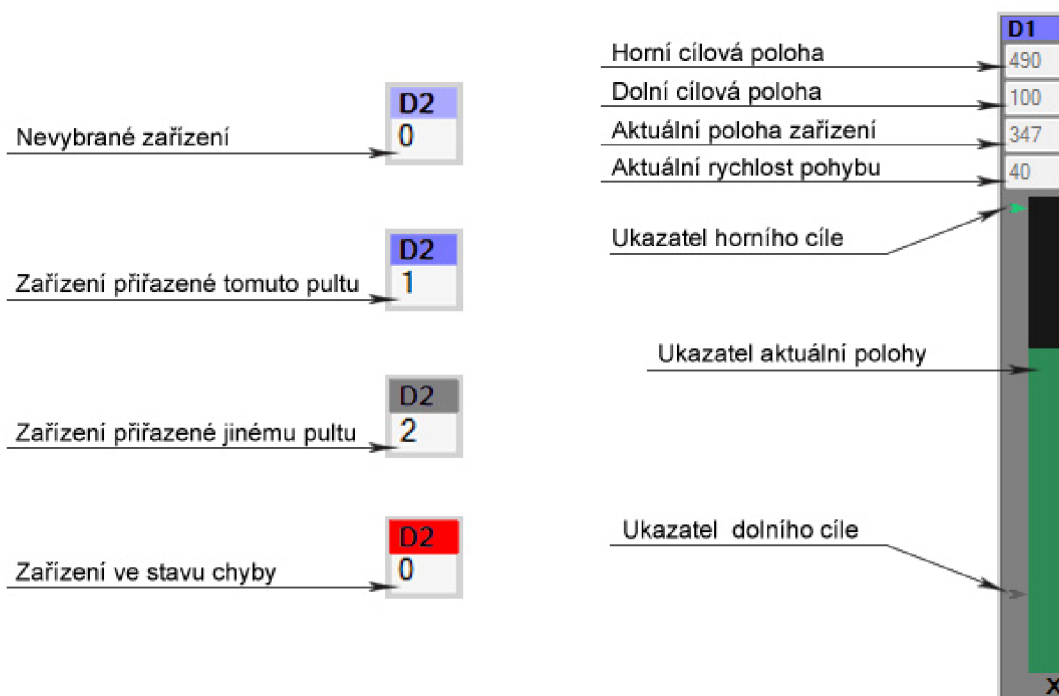
Vizualizační aplikace je implementovaná v prostředí MS Visual Studio jako aplikace s GUI pro windows (Windows Forms Application). Její hlavní okno tvoří základ uživatelského rozhraní. Třída `formControlPanel`, která toto okno implementuje, obsahuje režii událostí, pro jejichž obsluhu využívá instance dalších tříd, potřebných pro zajištění veškeré funkčnosti aplikace (viz. Obrázek 6.3). Jedná se zejména o třídu `classCommInterface` která implementuje komunikační rozhraní vizualizační aplikace, dále pak například `classDeviceVector` zapouzdřující seznam zařízení a metody pro operace s tímto seznamem nebo `classLeverAbstract`, která představuje funkčnost ovládací páky a vlastně zajišťuje ovládání a pohyb zařízení.

### 6.1 Uživatelské prostředí



Obrázek 6.1 Screenshot hlavního okna vizualizační aplikace

Nyní popíši realizaci uživatelského rozhraní, tedy pojetí a grafické uspořádání prostředí. Snímek okna je pro ilustraci uveden na obrázku 6.1. Většina interakce mezi uživatelem a aplikací se děje v hlavním okně, následující popis se tedy povětšinou týká tohoto okna. Seznam zařízení, který aplikace získala od řídicího systému prostřednictvím inicializace je uživateli zobrazen v panelu v levé spodní části okna. Nalevo jsou tlačítka „H“, „D“ a „T“ pro filtrování zobrazení zařízení podle typu (Horní, Dolní mechanizace a zvláště Točny). Konkrétní zařízení v nabídce je reprezentováno panelem, v jehož spodní části je zobrazena identifikace ovládacího pultu, ke kterému je přiřazen, přičemž hodnota „0“ značí, že zařízení není přiřazené k žádnému pultu. Horní část obsahuje podbarvený název zařízení. Podle barvy lze vyčíst stav zařízení (viz. Obrázek 6.2 vlevo).



**Obrázek 6.2 vlevo grafické rozlišení stavu zařízení v nabídce, vpravo panel reprezentující zařízení přiřazené páce**

Po stranách okna jsou dva TrackBary – posuvníky znázorňující levou a pravou ovládací páku. Pod každým z nich se nachází tlačítko „Lock“ pro uzamčení páky (viz. kapitola 6.3). Mezi pákami je prostor pro zařízení přiřazená některé z ovládacích pák. Protože by bylo nepřehledné, kdyby se mezi těmito zařízeními muselo rolovat pomocí ScrollBaru, omezil jsem počet pozic pro obě páky na 22. Prostor je pak opticky dělen na dvě části pomocí vertikální osy, která je znázorněna pomocí PictureBoxu s obrázkem čáry. Každé přiřazené zařízení je reprezentováno panelem (viz. Obrázek 6.2 vpravo) obsahujícím ve své horní části pod názvem zařízení dvě editovatelná textová pole zobrazující horní a dolní cílovou polohu zařízení, pod nimi další dvě pole, již needitovatelná, zobrazující aktuální polohu a rychlost pohybu zařízení. Ve spodní části panelu je graficky zobrazen zvolený horní a dolní cíl vzhledem ke krajním mezním polohám zařízení a aktuální poloha zařízení.

Vpravo dole našel své místo panel pro správu skupin. Je zde seznam zobrazující existující skupiny, tlačítka umožňující vstup do dialogu pro vytvoření skupiny či nastavení jejích parametrů a tlačítko pro smazání skupiny.

## 6.2 Komunikační rozhraní

Jak už bylo zmíněno v návrhu, komunikace v rámci řídicího systému je postavena nad protokolem UDP, na nějž padlo rozhodnutí nejen kvůli nižší režii a časové úspoře ale i proto, že ke komunikaci je využito UDP Broadcastu.

Komunikační rozhraní je implementováno třídou `classCommInterface`. Ta v sobě integruje jak rozhraní pro odchozí komunikaci, tak rozhraní pro posluchače zpráv příchozích. Hned po spuštění programu, v konstruktoru třídy `formControlPanel`, tedy formuláře hlavního okna, jsou vytvořeny dvě instance komunikačního rozhraní. Jedna slouží jako posluchač a je vytvořena pomocí konstruktoru s jediným parametrem – číslem portu, na kterém se bude naslouchat. Druhá instance představuje odesilatele zpráv a jako druhý parametr přetíženého konstruktoru dostává IP adresu pro broadcast, první parametr je číslo portu.

### BackgroundWorker a posluchač příchozí komunikace

Od posluchače se očekává, že bude neustále cyklicky naslouchat. Tato nepřetržitá činnost si vyžaduje běh v samotném vlákne. Visual C++ nabízí komponentu `BackgroundWorker`, která je určená pro vykonávání výpočetně náročných činností na pozadí aplikace. Tato komponenta se sama stará o vytvoření a veškerou režii vlákna, ve kterém je ona činnost vykonávána. Pro otevření nového vlákna a zahájení činnosti je určena metoda `RunWorkerAsync`. Ta je volána ihned po vytvoření komunikačního rozhraní posluchače a vytvoření a nastavení socketu metodou `setServerSocket` třídy `classCommInterface`. Běh činnosti je pak řízen událostmi:

- **DoWork** – nastává po volání metody `RunWorkerAsync`, její handler obsahuje kód té činnosti, která se má na pozadí vykonávat. Zde se tedy nachází kód posluchače. Metoda `recvResponse` třídy `classCommInterface` přijme zprávu, která je následně vyhodnocena. Je určeno o kterou zprávu jde a jsou provedeny příslušné operace (inicializace zařízení, aplikace feedbacku, ...).
- **RunWorkerCompleted** – nastává, když `BackgroundWorker` dokončí prováděnou činnost, což v tomto případě znamená, že posluchač obdržel a zpracoval příchozí zprávu. Zde, pokud se jednalo o inicializaci nebo o feedback je volána metoda `deviceListRepaint` třídy `formControlPanel` pro překreslení nabídky zařízení a metoda `choosedItemsRepaint` třídy `classLeverAbstract` pro překreslení zařízení přiřazených na pákách. Nakonec je voláním metody `RunWorkerAsync` vždy znovu spuštěn posluchač. Třidu `classLeverAbstract` blíže popíši v podkapitole 6.3.

### Odchozí komunikace

Odchozí komunikaci zajišťuje druhá instance komunikačního rozhraní. K odeslání zprávy jsou potřeba vždy dvě metody třídy `classCommInterface`. Jednak je to metoda pro vytvoření zprávy, přičemž pro vytvoření každého typu zprávy existuje jiná metoda. Jako parametry přijme potřebná data, zformátuje je a vytvoří řetězec zprávy. Nakonec je odtud zavolána ona druhá potřebná metoda, `sendRequest`. Ta je pro všechny typy zpráv stejná. Vždy vytvoří socket, pomocí funkce `setsockopt` ho nastaví pro vysílání broadcastu a nakonec jeho prostřednictvím požadovaný řetězec odešle.

## 6.3 Režie zařízení, pohyb a vizualizace

Předně zmíním něco o inicializaci a uchování dat o zařízeních. Data o každém jednotlivém zařízení zapouzdřuje třída `classDevice`. Celý seznam zařízení je reprezentován kolekcí typu `vector`, což je vlastně seznam objektů této třídy. Tento celý vektor je zapouzdřen třídou `classDeviceVector`, která obsahuje další prostředky a metody pro potřebné operace s vektorem zařízení, jako inicializace, aplikace feedbacku, nastavování a získávání dat o zařízeních apod. K nahlédnutí viz. diagram tříd na Obrázku 6.3.

Základním prvkem pro ovládání zařízení je ovládací páka. Krom její grafické reprezentace je tu zejména třída `classLeverAbstract` reprezentující veškerou funkčnost s pákou spojenou. v konstruktoru hlavního okna aplikace jsou tedy vytvořeny dvě instance této třídy, pro levou a pravou páku. K vizualizaci pohybu pak slouží panely – pozice pro zařízení, která uživatel vybere a přiřadí jedné z ovládacích pák. Každý tento panel si uchovává ID zařízení, kterým je právě obsazen, ve formě atributu `Tag` u `Labelu` zobrazujícího název zařízení, přičemž hodnota `-1` značí, že pozice je neobsazená. Díky tomu lze pozice procházet a snadno identifikovat příslušné zařízení a zajistit tak potřebnou režii. O tu se stará také třída `classLeverAbstract`. Krom obsazování a uvolňování pozic obsahuje tato třída další důležité metody. Jednak pro zaznamenávání požadavků na pohyb zařízení v závislosti na vychýlení páky (metoda `devicesMove`) a jednak pro vizualizaci pohybu zařízení, tedy překreslování a aktualizace dat zobrazených v panelech na jednotlivých pozicích (metoda `choosedItemsRepaint`). Během přidávání dalších zařízení je také nastavována pozice dělicí čáry `pictureMiddleLine`. Pokud počet zařízení na jedné páce překročí hranici páce vymezeného prostoru, který čára opticky znázorňuje, je tato hranice o potřebný kus posunuta.

### Požadavek na pohyb, zpráva MOTION

Mírou vychýlení ovládací páky je daná požadovaná rychlost tak, že maximální vychýlení nahoru znamená požadavek na pohyb směrem vzhůru rychlostí rovnající se maximální povolené rychlosti pro to které zařízení. Požadovaná rychlosti pohybu je tedy úměrná vychýlení vzhledem k rychlosti maximální. Vychýlení páky opačným směrem generuje požadavky na pohyb podobným způsobem, avšak hodnoty požadované rychlosti mají záporné znaménko, které symbolizuje pohyb směrem dolů. Proces generování požadavku na pohyb probíhá tak, že při změně míry vychýlení páky je volána metoda `devicesMove` třídy `classLeverAbstract`. Ta dostane jako parametr hodnotu vychýlení a podle hodnot maximální povolené rychlosti pro každé zařízení postupně počítá požadované rychlosti a ukládá je do vektoru zařízení.

Aby mohlo probíhat generování, musí být ale páka zamknuta. Pokud není, tak po vychýlení páky nedojde k žádné reakci. K uzamknutí dojde po stisku tlačítka „Lock“ a v tuto chvíli už zároveň nelze přidávat k páce další zařízení a nelze již editovat horní a dolní cílové polohy. Co nás ale nyní zajímá více je to, že po uzamčení se páka dostane do aktivního režimu. V tu chvíli se začínají generovat a odesílat příkazy `MOTION`, zatím s nulovým požadavkem na rychlost, dokud páka není vychýlena. Příkazy `MOTION` jsou generovány v pravidelném časovém intervalu. Aby řídicí systém měl dostatečně aktuální informace po požadavcích uživatele, je nutné aby intenzita generování byla dostatečně vysoká. Nastavil jsem tedy tento interval na 100ms.

Nastavený časový interval zajišťuje časovač `timerMotionSender`, jeho handler pro událost `Tick` implementuje generování a následné odeslání příkazu `MOTION`. Aby se zbytečně nezatěžovala přenosová linka a aby se neztrácel čas s režií kolem odesílání příkazu `MOTION` pro každé zařízení zvlášť. Jsou dílčí zprávy pro jednotlivá zařízení spojovány do jednoho řetězce a odesílány vždy společně jako jeden packet. Maximální počet zařízení na páce je 22, což

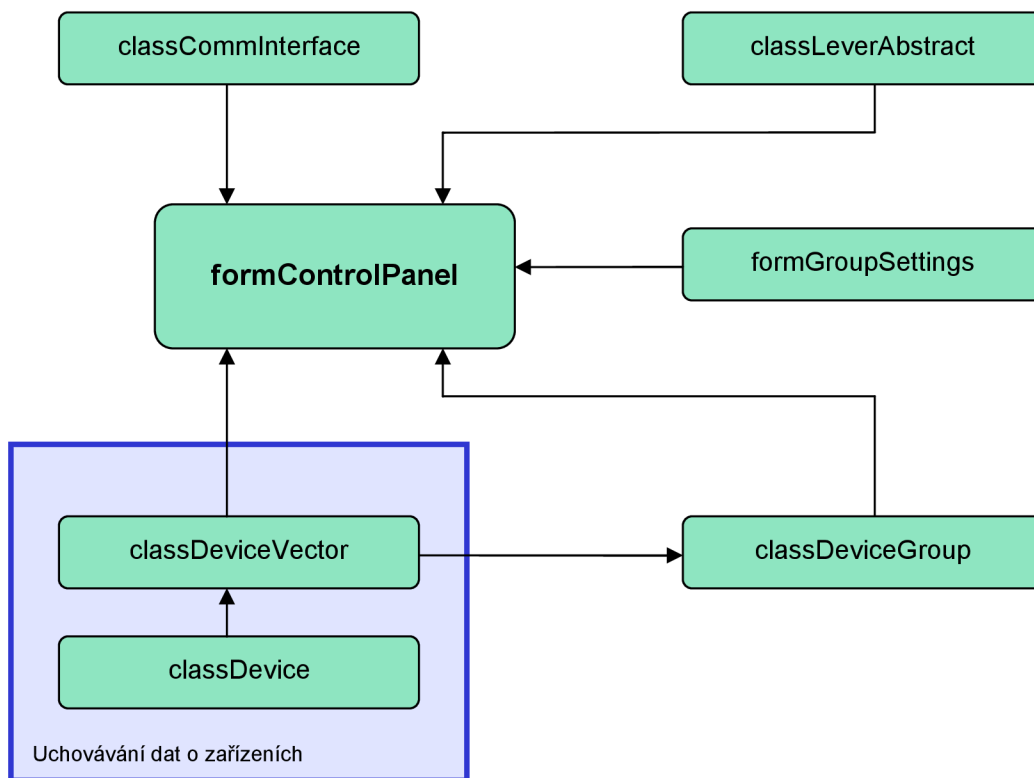
nepředstavuje nebezpečí, že by velikost celého packetu překročila snesitelnou mez. Je zde tedy opět nutné projít všechny pozice na páce. Jak už bylo výše zmíněno, ID zařízení je uloženo v atributu Tag u Labelu s názvem zařízení v každé pozici. Podle této identifikace získáme z vektoru zařízení informace o požadavku na pohyb, který je v tento okamžik aktuální (zároveň v těchto chvílích může být páka jakkoliv vychylována, požadavky na pohyb se mohou neustále měnit). Získáme tedy požadovanou rychlost, směr pohybu a cílovou polohu a tyto informace společně s identifikací zařízení a kontrolní informaci o aktuální poloze utvoří příkaz pro dané zařízení. Taky se sem vkládá informace o další následující cílové poloze a rychlosti, ale to má smysl spíše v případě pohybu skupiny. Z těchto dílčích příkazů je postupně skládána celá společná zpráva MOTION a nakonec je volána metoda `sendMotion` komunikačního rozhraní, která tuto zprávu odešle.

## FEEDBACK a vizualizace

Zprávy FEEDBACK, přichází od řídicího systému, obsahují aktuální data o zařízení, přičemž se jedná o informace více typů. Jednak jsou to základní údaje o stavu zařízení, zda je přiřazené nějakému ovládacímu pultu a kterému nebo zda bylo na některém pultu vybráno do skupiny. Tyto informace je potřeba mít stále aktuální, proto jsou zprávy FEEDBACK přijímány po celou dobu běhu aplikace. Ostatní údaje, které se týkají polohy a rychlosti, jsou ve vektoru zařízení rovněž aktualizovány, využívány jsou ovšem pouze pro vizualizaci pohybu, tedy až pokud je příslušné zařízení vybrané na páce.

Aktualizaci dat ve vektoru zařízení má na starosti metoda `receiveFeedback` třídy `classDeviceVector`, která vektor zapouzdřuje. Volaná je vždy po přijetí zprávy FEEDBACK přímo z `handlenu` události `DoWork` komponenty `backgroundWorkerCommlistener`. Aktualizovaná data je potřeba zobrazit uživateli, proto je poté zavolána metoda `deviceListRepaint`, která překreslí nabídku zařízení, kde se zohlední například případné chybové stavy nebo přiřazení na jiných pultech.

Zároveň je zde provedeno překreslení vybraných zařízení, čímž se dostáváme k vizualizaci pohybu. Obsah panelu představujícího pozici pro zařízení už byl popsán v kapitole 6.1, nyní vysvětlím, jak to funguje. Princip vizualizace je v aktualizaci dat, tedy postupné překreslování obsahu všech těchto panelů. To je úkol metody `choosedItemsRepaint` třídy `classLeverAbstract`. Data bere z vektoru zařízení, který je mezitím pravidelně aktualizován. Cyklicky projde každý panel, v příslušných textových polích vždy aktualizuje aktuální polohu a rychlost. Vlastní vizualizaci pak představuje grafické znázornění, nastaví se tedy poloha ukazatelů cílové polohy a výše vertikálního sloupce ukazujícího aktuální polohu. Překreslení vizualizace proběhne vždy po výběru zařízení na páku, tím se zobrazí jeho výchozí poloha. Dále proběhne vždy po změně cílových poloh, neboť je nutné aktualizovat i polohu grafických ukazatelů cílů. A jak bylo zmíněno výše, proběhne samozřejmě také vždy po přijetí zprávy FEEDBACK v případě, že je páka uzamčena, tedy v aktivním režimu.



Obrázek 6.3 Struktura důležitých tříd, které program využívá

## 6.4 Správa skupin zařízení

Funkcionalita skupin spočívá v možnosti zadat několika sdruženým zařízením určitý parametrizovaný synchronní pohyb, který bude zahájen vychýlením ovládací páky. Daný bod implementace je ovšem realizován spíše rámcově. Tato podkapitola obsahuje popis realizované implementace správy skupin i popis implementace realizované jen částečně, tedy určené dále pro budoucí vývoj.

Kliknutím na odpovídající tlačítko se otevře formulář dialogového okna `formGroupSettings` pro vytvoření a nastavení skupiny. Tomuto formuláři je parametrem předán ukazatel na instanci vektoru zařízení. V dialogovém okně je tedy možné zobrazit seznam dostupných zařízení, podobně jako v hlavním okně, filtrovaný podle jejich typu. Kliknutím na některé zařízení dojde k jeho výběru do seznamu vpravo. Dále je potřeba zvolit název skupiny a poslední položkou při vytváření skupiny je volba typu synchronního pohybu z nabídky rozbalovacího seznamu `comboBoxGroupType`. Poté se přejde k nastavení vytvořené skupiny. Parametry nastavení se liší podle zvoleného typu pohybu. Například vlnobití je postavené na funkci sinus. Jako parametry je možné nastavit výšku v milimetrech a počet vln. Tyto hodnoty jsou pak použity ve vzorci pro výpočet polohy postaveném na funkci sinus. Dále třeba u synchronního pohybu přímky se nastavuje úhel, který tato přímka svírá s vodorovnou rovinou.

Správu skupiny zařízení implementuje třída `classDeviceGroup`. Objekt této třídy je vytvořen ještě před otevřením dialogového okna pro vytvoření skupiny (`formGroupSettings`) a ukazatel na něho je tomuto oknu předán jako další parametr. Veškeré nastavení skupiny je tedy do tohoto objektu uloženo a po zavření dialogu, pokud se podařilo skupinu úspěšně vytvořit, je objekt přidán do pole `arrayGroups`, obsahujícího objekty všech vytvořených skupin. Toto pole je implementováno jako kolekce typu `ArrayList`.



Objekt reprezentující skupinu si musí nějak uchovávat seznam zařízení, které obsahuje. Třída `classDeviceGroup` Má tedy v sobě integrovanou svoji vlastní instanci třídy `classDeviceVector`, která tento seznam (`vector`) zapouzdřuje. Objekt skupiny si uchovává informaci o nastaveném režimu synchronizace pohybu a parametry potřebné k realizaci tohoto pohybu. Dále tato třída obsahuje metody, které v závislosti na těchto parametrech zajišťují samotný pohyb, jsou to metody dvou typů:

- **Elementární metody pohybu** – implementují přímo výpočet příští cílové pozice podle příslušné matematické funkce v závislosti na pořadí daného zařízení ve skupině. Je to například metoda `motionBaseLine` pro přímku nebo `motionBaseSinus` pro sinus.
- **Metody zajišťující nastavený pohyb** – transformují nastavené parametry do podoby vhodné pro jejich předání příslušné matematické funkci, tedy jedné z elementárních metod pohybu. Poté pro každé zařízení tuto metodu volají a tím získají pro každé zařízení příští cílovou polohu. Například je to metoda `synMotionLeaningLine` pro nakloněnou přímku nebo `synMotionWaves` pro vlnobití.

Když je tedy skupina přiřazená na páce a páka je uzamčena, je potřeba, podobně jako u pohybu samostatných zařízení, při každé změně vychýlení páky znovu přepočítat požadované cílové polohy pro zařízení ve skupině. Proto je při změně polohy páky volána metoda zajišťující příslušný nastavený pohyb, a ta přepočítá cílové polohy postupně pro každé zařízení ve skupině. Zároveň již od chvíle uzamčení páky jsou odesílány řídicímu systému požadavky na pohyb. Zde již, oproti samostatným zařízením, nabírají smysl i další položky ve struktuře zprávy MOTION. Je to sekundární cílová poloha a s ní související požadovaná rychlost pohybu k této poloze. Toto signalizuje řídicímu systému, například při pohybu symbolizujícím vlnobití, že po první cílové poloze v hodnotě amplitudy dané funkce sinus bude opět následovat pohyb směrem dolů.

Když je skupina vybrána a přiřazena ovládací páce, příslušný počet pozic bude obsazen všemi zařízeními z dané skupiny. Přičemž název těchto zařízení je pro zpřehlednění podbarven světlejším odstínem a pod panelem posledního zařízení se zobrazí název skupiny. Díky tomu, že jsou uživateli zobrazena všechna zařízení, je vidět graficky znázorněná aktuální poloha každého z nich a vzniká tak přehledná vizualizace výsledného synchronního skupinového pohybu.

V tuto chvíli jsou v nabídce synchronních režimy:

- **Souhlasný pohyb** – všechna zařízení ve skupině se pohybují stejnou rychlostí, ani nedochází k jejich „zarovnání“ do nějakého tvaru
- **Nakloněná přímka** – po zahájení pohybu se zařízení postupně dostanou do pozic, kdy vzájemně tvoří přímku nakloněnou pod nastaveným úhlem, který svírá přímka s vodorovnou rovinou. Tato přímka se pak pohybuje vertikálním směrem požadovanou rychlostí dále až do nastavené cílové polohy (ta platí pro zařízení, které ji dosáhne jako první).
- **Vlnobití** – dochází k pohybu připomínajícím vlnění. Zařízení v řadě vedle sebe kmitají nahoru a dolů a vzniká tak vlnění. Ze zadaného počtu vln a z celkového počtu zařízení ve skupině se spočítá perioda funkce sinus, osu x pomyslného grafu funkce tvoří „řada pohonů“ a požadované cílové polohy každého pohonu pak odpovídá hodnota funkce sinus v daném bodě. Jedná se o princip podobný postupnému příčnému vlnění. Vyšší míra vychýlení páky znamená vyšší frekvenci kmitání

## 6.5 Zhodnocení výsledků

Vytvořená aplikace běžící na ovládacím pultu je schopná prostřednictvím řídicího systému, ke kterému je pult připojen, ovládat zařízení jevištní techniky. Ovšem v rámci této práce nebylo prováděno testování na skutečném zařízení. Jak jsem již zmínil v páté kapitole, tato verze je určená zejména pro spolupráci s grafickým simulátorem divadelní scény. Tento simulátor je postaven na platformě CAD systému Lexocad. Díky tomuto přístupu je možné poměrně jednoduše namodelovat divadelní scénu podle momentálních požadavků a poté tento model pomocí ovládací aplikace a řídicího systému „rozhýbat“ a simulovat tak, jak by se chovali skutečné pohony na reálné scéně. Testování bylo tedy prováděno tímto způsobem. Bylo otestováno ovládání jednotlivých zařízení horní i dolní mechanizace a byl přitom kontrolován jak výstup na simulátoru, tak samozřejmě i grafická vizualizace jejich pohybu zprostředkovaná vytvořenou ovládací a vizualizační aplikací.

Rozšiřující funkcionalitu představuje možnost vytváření skupin a uvedení pohonů dané skupiny ve společný synchronizovaný pohyb typicky založený na nějaké matematické funkci. Nutno zde však znovu připomenout, že implementace této funkcionality je realizovaná jen částečně. Problematika synchronních režimů pohybu je pojata jako námět a prostor pro další zdokonalování v rámci pokračování této práce. Je potřeba dořešit všechna úskalí, doimplementovat příslušné metody a celou funkcionalitu synchronního pohybu poté řádně otestovat.

Po dokončení aplikace a před jejím nasazením do reálného provozu bude nezbytně nutné provést rozsáhlé a důmyslné testy, neboť se jedná o ovládání divadelní techniky. A jak již bylo výše několikrát zmíněno, bezpečnostními nároky na řídicí systémy tohoto typu jsou velmi vysoké.

## 7 Závěr

Jako cíl této práce bylo vytyčeno vytvoření uživatelského rozhraní pro nový divadelní řídicí systém. Dále pak přinést náhled do problematiky jevištní techniky a jejího řízení a shrnout současný stav v oblasti uživatelských rozhraní., tento cíl byl splněn.

Uvedené informace a fakta pramení z dostupné literatury a na těchto znalostech stavím při plnění dalších bodů zadání. Druhým bodem je navržení koncepce uživatelského rozhraní. Při tomto byl kladen důraz na jednoduchost, přehlednost a funkčnost. Bohatý zdroj informací také přinesly konzultace ve firmě Elseremo, a.s. Jejich doporučení a zkušenosti z praxe byly při návrhu vzaty v potaz.

Celá vizualizační a ovládací aplikace byla implementována jako aplikace s grafickým uživatelským rozhraním pro operační systém Windows v prostředí MS Visual C++. Vytvořené rozhraní instalované na ovládacím pultu představuje prostředek pro ovládání řídicího systému, se kterým komunikuje prostřednictvím Ethernetu. Komunikace probíhá podle pravidel speciálního aplikačního protokolu. Vhodná struktura tohoto protokolu byla jednou z hlavních met při návrhu celé aplikace.

Vznikla funkční aplikace umožňující ovládání divadelní techniky, horní i dolní mechanizace. Zpětně pak přehledně vizualizuje skutečnou polohu při pohybu zařízení. Její funkčnost byla testována za pomoci grafického simulátoru divadelní scény. Přínos práce lze najít například v použité technologii. Realizace řídicího systému a jeho uživatelského rozhraní formou softwaru instalovaném na IPC slibuje jak přijatelnou finanční náročnost, tak i poměrně snadnou instalaci a uvedení do provozu.

Myslím, že díky této práci jsem se dověděl mnoho nového z prostředí divadla, jeho zákulisí a technologických řešení scény. Dále jsem si utvořil přehled mezi současnými trendy uživatelských rozhraní a naučil jsem se některé tyto poznatky aplikovat v praxi.

Výstup této bakalářské práce představuje určitý směr, kterým by se měl další vývoj ubírat. Ovšem k definitivní verzi vizualizační aplikace povede ještě dlouhá cesta. Jako výhled pro pokračování v budoucnu se nabízí další práce na implementaci synchronizace skupin zařízení či dokonce zamyšlení se nad návrhem dokonalejšího principu této funkcionality. Počítá se též s implementací zabezpečení užívání aplikace pomocí USB klíče anebo s tvorbou scénářů ve formě uložených posloupností dopředu nastavených uspořádání pohonů na scéně. A v neposlední řadě stále zbývá prostor pro další zdokonalování uživatelského prostředí aplikace.

# Literatura

- [1] Bezděk, P.: Jevištní technologické zařízení – divadla. Typizační směrnice všeobecná. Ministerstvo kultury ČSR, Praha, 1987.
- [2] Ševcovic, J.: Řídicí systém divadelní scény s grafickým rozhraním, diplomová práce. Fakulta Informačních technologií VUT, Brno, 2009.
- [3] Jančík, J.: Moderní řídicí systémy jevištní techniky. Magazín AUTOMA 3/2009. FCC Public s.r.o., Praha, ČR, 2009. Dostupné na URL: <[http://www.odbornecasopisy.cz/index.php?id\\_document=38732](http://www.odbornecasopisy.cz/index.php?id_document=38732)>
- [4] Elseremo a.s. [online]. *Webová prezentace firmy*. 2008 [cit. 2010-04-08]. Dostupné na URL: <<http://www.elseremo.com>>
- [5] *Control system* [online]. 2010 [cit. 2010-04-18]. Dostupné na URL: <[http://en.wikipedia.org/wiki/Control\\_system](http://en.wikipedia.org/wiki/Control_system)>
- [6] *Příručka k divadelnímu řídicímu systému*. Elseremo a.s. [cit. 2010-04-21].
- [7] BBH Systems GmbH. [online]. *Webová prezentace firmy*. [cit. 2010-05-04]. Dostupné na URL: <<http://www.bbh-systems.de/>>
- [8] Stage Technologies Ltd. [online]. *Webová prezentace firmy*. 2010 [cit. 2010-05-04]. Dostupné na URL: <<http://www.elseremo.com>>
- [9] Zemčík, P.: Tvorba uživatelských rozhraní, studijní opora. Fakulta Informačních technologií VUT, Brno, 2006.
- [10] *Textové uživatelské rozhraní* [online]. 2009 [cit. 2010-04-14]. Dostupné na URL: <[http://cs.wikipedia.org/wiki/Textov%C3%A9\\_u%C5%BEivatelsk%C3%A9\\_rozhran%C3%AD](http://cs.wikipedia.org/wiki/Textov%C3%A9_u%C5%BEivatelsk%C3%A9_rozhran%C3%AD)>
- [11] Sommerville, I.: Software Engineering. Pearson Education Limited, Harlow, UK, 2004. Seventh Edition. ISBN 0-321-21026-3.
- [12] Faule, A.: Learn to Program Using Python. Překlad: Příkryl, P. Addison-Wesley, Canada, 2000. ISBN 0-201-70938-4. Dostupné na URL: <<http://www.freenetpages.co.uk/hp/alan.gauld/czech/cztutcont.html>>
- [13] Kruglinski, D.J., Shepherd, G., Wingo, S.: Programujeme v Microsoft Visual C++. Computer Press, Praha, ČR, 2000. ISBN 80-7226-362-5

# Seznam příloh

Příloha A: popis aplikačního protokolu pro síťovou komunikaci.

Příloha B: vytvořená vizualizační aplikace se zdrojovými kódy na přiloženém DVD.

# Příloha A

## Popis komunikačního protokolu

INIT_REQUEST		
<i>Směr komunikace:</i>	OP -> ŘS	
<i>Popis:</i>	žádost o inicializaci dat o zařízeních připojených k řídicímu systému	
<i>Formát:</i>	"INIT_REQUEST"	
<i>Název</i>	<i>Datový typ</i>	<i>Komentář</i>
INIT_OK		
<i>Směr komunikace:</i>	OP <- ŘS	
<i>Popis:</i>	odpověď na žádost o inicializaci dat o zařízeních připojených k řídicímu systému	
<i>Formát:</i>	"INIT_OK,givenDeskID;deviceID,status,deviceType,maxPosition,minPosition,maxSpeed,groupID,deskID;"	
<i>Název</i>	<i>Datový typ</i>	<i>Komentář</i>
givenDeskID	int [1 az n]	ID ovládacího pultu přidělené řídicím systémem
deviceID	int [1 az n]	jednoznačná identifikace zařízení
status	int ["1"-stojí, "2"-jede, "0"-chyba]	stav daného zařízení (stojí, v pohybu, chyba)
deviceType	int ["1"-horní, "2"-dolní, "3"-točna]	typ zařízení (horní, dolní mechanizace, točna)
maxPosition	int [mm]	horní mezní poloha zařízení
minPosition	int [mm]	dolní mezní poloha zařízení
maxSpeed	int [mm/s]	maximální rychlost pohybu zařízení
groupID	int [1 az n; 0 - neprirazeno]	identifikace případného přiřazení do skupiny
deskID	int [1 az n; 0 - neprirazeno]	identifikace přiřazení na ovládacím pultu
CHOOSE_ON_DESK		
<i>Směr komunikace:</i>	OP -> ŘS	
<i>Popis:</i>	zpráva řídicímu systému obsahující informaci o přiřazení zařízení ovládacímu pultu nebo o zrušení jeho přiřazení	
<i>Formát:</i>	"CHOOSE_ON_DESK;deviceID,deskID,operation;"	
<i>Název</i>	<i>Datový typ</i>	<i>Komentář</i>
deviceID	int [1 az n]	jednoznačná identifikace zařízení
deskID	int [1 az n]	jednoznačná identifikace ovládacím pultu
operation	int ("0"-zrušeno, "1"-vybráno)	prováděná operace - vybráno/zrušen výběr
MOTION		
<i>Směr komunikace:</i>	OP -> ŘS	
<i>Popis:</i>	požadavek na pohyb zařízení, obsahuje informace o požadované cílové pozici a rychlosti pohybu a další potřebná data	
<i>Formát:</i>	"MOTION,devicesCount;deviceID,actualPosition,position1,speed1,position2,speed2,time,mode;"	
<i>Název</i>	<i>Datový typ</i>	<i>Komentář</i>
devicesCount	int [1 az n]	počet zařízení v jedné zprávě MOTION
deviceID	int [1 az n]	jednoznačná identifikace zařízení
position0	int [mm]	aktuální poloha zařízení (předpokládaná)
position1	int [mm]	1. požadovaná poloha zařízení
speed1	int [mm/s]	požadovaná rychlost pohybu zařízení k 1. poloze
position2	int [mm]	2. požadovaná poloha zařízení
speed2	int [mm/s]	požadovaná rychlost pohybu zařízení k 2. poloze
time	Time	časová synchronizace
mode	int	režim chování zařízení

## FEEDBACK

**Směr komunikace:** OP <- ŘS

**Popis:** zpětná vazba od řídicího systému, obsahuje aktuální informace o stavu zařízení, o přiřazení k ovládacímu pultu, o aktuální poloze a rychlosti zařízení

**Formát:** "FEEDBACK;devicesCount;deviceID,status,actualSpeed,actualPosition,groupID,deskID,time,errorNum;"

<i>Název</i>	<i>Datový typ</i>	<i>Komentář</i>
devicesCount	int [1 az n]	počet zařízení v jedné zprávě FEEDBACK
deviceID	int [1 az n]	jednoznačná identifikace zařízení
status	int ["1"-stojí, "2"-jede, "0"-chyba]	stav daného zařízení (stojí, v pohybu, chyba)
actualSpeed	int [mm/s]	aktuální rychlost pohybu
actualPosition	int [mm]	aktuální poloha zařízení
groupID	int [1 az n]	identifikace případného přiřazení do skupiny
deskID	int [1 az n]	identifikace přiřazení na ovládacím pultu
time	Time	časová synchronizace
errorNum	int [0 az n]	číslo chyby, "0" znamená bez chyby

## LEAVING

**Směr komunikace:** OP -> ŘS

**Popis:** zpráva řídicímu systému o ukončení činnosti ovládacího pultu

**Formát:** "LEAVING;deskID"

<i>Název</i>	<i>Datový typ</i>	<i>Komentář</i>
deskID	int [1 az n]	jednoznačná identifikace ovládacím pultu