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

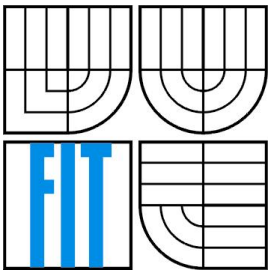
# ANALÝZA SENZORICKÝCH DAT PRO POKROČILÉ UŽIVATELSKÉ ROZHRANÍ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. FILIP CHMIEL
AUTHOR

BRNO 2013

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

# ANALÝZA SENZORICKÝCH DAT PRO POKROČILÉ UŽIVATELSKÉ ROZHRANÍ

SENSOR DATA ANALYSIS FOR ADVANCED USER INTERFACES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. FILIP CHMIEL

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. VÍTĚZSLAV BERAN Ph.D.

BRNO 2013

# Abstrakt

Práce se zabývá tvorbou uživatelského rozhraní založeného na více vstupních signálech, tedy multimodálním rozhraním. Za tímto účelem nejprve rozebírá výhody daného přístupu ke komunikaci s přístroji. Dále práce obsahuje přehled úrovní, na kterých lze fúzi dat provádět, a různé přístupy k rozvržení architektury systému pro zpracování multimodálních dat. Důležitou částí je samotný návrh systému, kde pro výsledné rozhraní byla zvolena distribuovaná architektura používající softwarové agenty pro zpracování vstupů. Ze studovaných metod pro integraci dat byla vybrána hybridní fúze. Cílem má být rozhraní umožňující ovládání multimediálního centra a interakci s dalšími zařízeními v okolí uživatele.

# Abstract

The paper deals with the creation of interface based on multiple input signals, i.e. multimodal interface. For this purpose analyzes the benefits of the approach to communicate with the device that way. The work also includes an overview of the level at which you can perform data fusion, and different approaches to the layout of the system architecture for multimodal data processing. The important part is the actual design of the system, where for the interface was chosen distributed architecture using software agents for processing inputs. As a method for data integration was picked hybrid fusion based on dialog driven and unification strategy. The result should be an interface for media center control and interaction with other devices around the user.



# Klíčová slova

Multimodální rozhraní, HCI, fúze dat, melting pot, unifikace, dialogem řízená fúze, hybridní fúze

# Keywords

Multimodal interface, HCI, data fusion, melting pot, unification, dialog driven fusion, hybrid fusion


# Citace

Chmiel Filip: Analýza senzorických dat pro pokročilé uživatelské rozhraní, diplomová práce, Brno, FIT VUT v Brně, 2013

# Analýza senzorických dat pro pokročilé uživatelské rozhraní

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Vítězslava Berana, Ph.D.
Další informace mi poskytl Ing. Igor Szőke, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Filip Chmiel
7. 1. 2013

## Poděkování

Přednostně děkuji vedoucímu práce Ing. Vítězslavu Beranovi Ph.D. za odborné rady a motivaci nejen v podobě nápadů, ale i dalších publikací. Také bych chtěl poděkovat rodině a všem přátelům za podporu a vstřícnost během testování navrhovaného systému.

# Obsah

# 1 Úvod

Ovládání počítače a dalších zařízení je pro dnešního člověka neodmyslitelnou součástí denních úkonů. V posledních letech došlo v oblasti tvorby rozhraní mezi člověkem a přístroji ke značnému pokroku. Nastoupily dotykové displeje, které se nadále rozvíjí a přicházejí s dalšími gesty pro usnadnění ovládání přístrojů. Na trh byly uvedeny i další periferie umožňující sledování pohybu uživatele bez nutnosti použití ovladače. Většina běžných rozhraní je dnes založena na klávesnici, myši či variantě dotykového displeje. Komunikačních kanálů je ale dostupných mnoho a výkon počítačů pokročil natolik, aby umožnoval bez větších potíží získávat informace z více senzorů náročných na zpracování (například webkamera), aniž by celý systém byl vytížen natolik, že by nebyl schopen v reálném čase vykonávat i další užitečnou práci. Avšak současná řešení rozhraní se často zaměřují na zpracování jednoho kanálu. To má za následek poněkud menší pohodlí ovládání a vytěsnění těchto přístupů do specifických úloh, jako například pomoc handicapovaným nebo ovládání specializovaných zařízení.

Cílem této práce je navrhnout a realizovat uživatelské rozhraní založené na souběžném zpracování více signálů různého charakteru. Určit z audio či video signálu vstupní akce pro systém však pro vytvoření rozhraní nemusí postačovat, zvláště pokud chceme vytvořit uživatelsky přívětivé rozhraní. V tomto případě je potřeba vstupní data vhodným způsobem zkombinovat – použít některé z technik fúze dat. Proto se práce zaměřuje zejména na zpracování dat na vyšší úrovni a samotné získávání základních informací ze vstupů postihuje jen okrajově. Celkově je fúze dat poměrně rozsáhla oblast. Techniky lze rozlišit nejen dle způsobu získávání vstupních dat, ale i podle úrovně zpracování, na které dochází ke sloučení informací. Z tohoto se odvozují různé mechanismy vhodné pro integraci dat dané úrovně. Proto je v technické zprávě věnována značná pozornost přehledu dnes používaných metod pro fúzi dat. Výsledkem práce je systém umožňující ovládání přístrojů hlasem a gesty rukou a technická zpráva vytvořená za účelem, aby čtenáře zajímavou formou seznámila s mechanizmy fúze a samotnou tvorbou systémů.

V dokumentu je po stručném úvodu nejprve rozvedeno, proč má význam věnovat se problematice multimodálních rozhraní – důvody pro tvorbu ovládání takového typu i způsob komunikace a druhy modalit. Následně je uveden přehled architektur pro návrh rozhraní a metod pro fúzi dat. Na teoretickou část navazuje návrh systému, kde popisuji svůj koncept systému pro fúzi multimodálních dat. Kvůli širokému spektru možných řešení při návrhu zmiňuji různé způsoby, jakými lze navrhovaný systém realizovat. Dále je uveden popis implementace, tedy vlastní realizace mnou navrženého systému. Kromě popisu řešení jednotlivých bloků obsahuje tato část i popis a výsledky několika testů, kterými jsem ověřoval funkčnost řešení. Tato práce popisuje způsob analýzy senzorických dat pro tvorbu multimodálního rozhraní nejen z teoretického, ale i z praktického hlediska a vše je doplněno názornými ilustracemi.

# 2 Ovládání počítače

Přístupů k tvorbě rozhraní či na čem má být založeno je mnoho. Pro účely této práce se zaměříme na problematiku tvorby multimodálního rozhraní – jaké senzory jsou k dispozici, jaké jsou výhody daného přístupu, úrovně, na kterých dochází k fúzi, a samotné metody fúze dat.

Nejprve však uveďme, o jaký systém by se mělo jednat. Navrhovaný systém by měl využívat informace o poloze uživatele a jeho pohybech (získaných nejlépe z hloubkové mapy snímaného prostoru), dále údaje o vyslovených příkazech a informaci o směru pohledu uživatele. Z těchto dat by měl systém vhodným způsobem určit, jaký pokyn uživatel systému zadal a vykonat ho. Pokyny by se měly týkat ovládání multimédií a přístrojů, které se nacházejí v okolí uživatele. Bližší specifikaci systému naleznete v kapitole 3.

## 2.1 Senzory

Pro komunikaci s počítačem či přístroji již nejsme odkázáni pouze na klávesnici a myš, existuje totiž celá řada různých vstupních zařízení (ukázka Obrázek 1). Již delší dobu například oblast výzkumu věnující se počítačovému vidění používá video signál z webkamery nejen pro usnadnění ovládání přístrojů handicapovanými osobami. Dalším běžně se vyskytujícím přístupem k tvorbě rozhraní je využití audio signálu z mikrofonu. Díky rozvoji mobilních zařízení se zdá, že ovládání hlasem získává na popularitě.

Několik posledních let se začal prosazovat trend pohybových senzorů (hlavně v oblasti herních konzolí), s možným uplatněním i u počítače. Oproti experimentálně vyvíjeným ovladačům (různé rukavice snímající pohyb či přístupy založené na počítačovém vidění vyžadující speciální značky) jsou přizpůsobeny pro větší svobodu pohybu. Mohou poskytovat údaje přímo o pohybu uživatele, nebo lze přistupovat k datům hloubkové mapy či naměřeným hodnotám z gyroskopů. Zajímavým vstupním zařízením je i elektronické pero a dotykový displej umožňující přesnější snímání pohybu ruky.

Existují i řešení založená na senzorech snímajících zvuk v pásmu ultrazvuku, vytvářeného pohybem ruky před soustavou mikrofonů, a díky tomu ovládat počítač gesty podobně jako v případě dotykového displeje.

Pro návrh rozhraní používající data z méně běžných senzorů, jako například kamer pro snímání hloubky prostoru, není již potřeba specializovaný hardware vyvíjet, ale lze využít komerčně dostupné produkty, často přímo poskytující SDK (Software Development Kit) pro tvorbu aplikací[30].

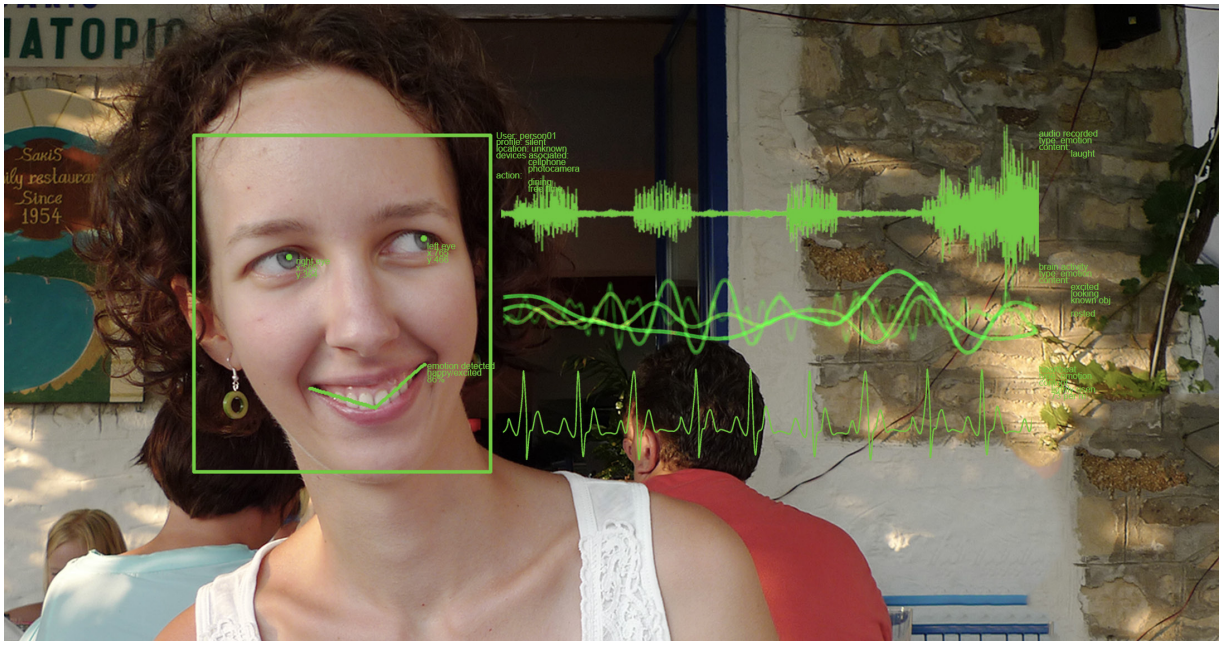

**Obrázek 1:** Ukázka běžně dostupných senzorů poskytujících různé informace o prostředí, ve kterém se nacházejí.

**Senzorická fúze**

Jedná se o kombinaci senzorických dat či dat odvozených ze senzorových signálů za účelem dosažení v jistém smyslu lepšího výstupu než kdyby vstupy byly vyhodnoceny zvlášť [9]. V literatuře se objevuje řada termínů, jako „senzorická fúze“, „fúze dat“, „fúze informací“, „multisenzorová fúze dat“ nebo „multisenzorová integrace“, pro označení technik či systémů a aplikací, které využívají data získaná z více než jednoho zdroje. Fúze nachází své uplatnění v mnoha oblastech, od analýzy signálů v reálném čase, pro navigaci robotů, po zpětné zpracování zpravodajských informací.

## 2.2 Důvody pro multimodální rozhraní

Člověk, aniž by si to uvědomoval, komunikuje se svým okolím multimodálně. Pro komunikaci využíváme řeč, gesta, výrazy tváře. Bez využití těchto dalších komunikačních kanálů působí řeč nepřirozeně a uměle. I v roli posluchače nasloucháme tónu hlasu řečníka a současně si všímáme výrazu jeho tváře.




**Obrázek 2:** Ukázka možných vstupů pro rozhraní využívající informace z různých zdrojů.

V případě rozhraní mezi člověkem a strojem je situace rozdílná. Běžně používáme pouze jeden komunikační kanál – psaní, pohyb myší, řečové příkazy apod. Tato situace není pro obyčejného uživatele příliš ergonomická. Může způsobovat problémy při ovládání počítače, kdy použití přirozeného způsobu komunikace nahrazuje uměle vytvořená metoda dorozumívání. Názorným příkladem je situace, kdy nechtěně stiskneme špatnou klávesu nebo musíme projít sérií nabídek pro pouhou změnu filtru či barvy objektu.

Proto při zvažování, jaký způsob ovládání v aplikaci použít, mělo by se vzít v úvahu několik výhod hovořících pro multimodální rozhraní (ukázka možných vstupů pro multimodální rozhraní viz Obrázek 2).

**Praktické hledisko**

Praktický důvod pro multimodální HCI (human-computer interface) vychází z jistých nevýhod současných rozhraní, které nekladou důraz na efektivitu práce. Jsou založena na zařízeních navržených v době počátku počítačové éry - myši, klávesnici či joysticku, která omezují způsob, jakým člověk může ovládat současné výpočetní prostředí.

Existuje již několik studií, které potvrzují, že lidé preferují vícekanálové rozhraní pro interakci s virtuálními objekty [2], [3]. Tyto experimenty jsou založeny na „Wizard of Oz“ experimentech, v nichž jsou reakce rozhraní simulované jiným člověkem. Ve zmíněných studiích Hauptman a McAvinney došli k výsledku, že 71% z testovaných subjektů preferovalo použití jak řeči, tak i gestikulace rukou pro ovládání raději než použití pouze jednoho. Další ukazuje, že 95% subjektů v úloze založené na interakci s mapou má obvykle sklon používat gesta spolu s hlasovými povely.

Modality se často doplňují – gesta jsou vhodná k manipulaci s objekty, zatímco řeč na popisovací úlohy či jednotlivé příkazy. Tohoto faktu lze využít pro zlepšení současných pokročilých jednomodálních HCI. Například při automatickém rozpoznávání gest rukou, které v poslední době získává na popularitě, jsme často vázání na několik předdefinovaných pohybů rukou. Současně se při použití dodatečného hardwaru jako stylus nebo speciální rukavice musíme vypořádat s dalšími kabely. Oproti tomu souběžné využití dvou interakčních kanálů může tyto nedostatky odstranit: vyslovením slova můžeme potvrzovat příkazy gestem a naopak pohyb rukou může doplnit vstup v případě šumu v signálu z mikrofonu pro analýzu řeči.

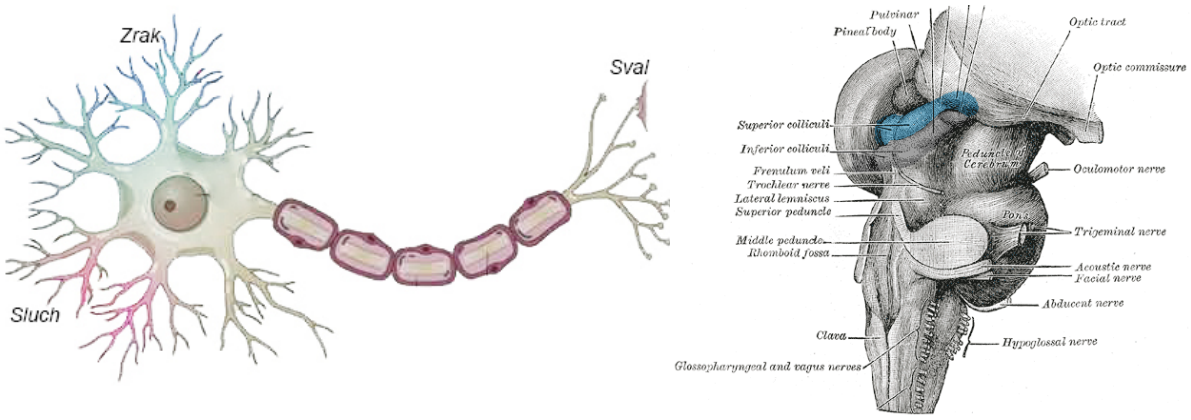
Použití multimodální komunikace dokáže snížit náročnost a zlepšit přirozenost rozhraní. Názorným příkladem je sledování gest rukou založené na počítačovém vidění jako vstup pro prohlížení snímků. Bez doplnění vstupů by se vlastní ovládání muselo skládat ze dvou gest (objekt vybraný prvním gestem by byl přiblížen či jinak manipulován druhým gestem). Doplnění pohybu rukou slovním příkazem „ten“, může být použito za stejným účelem, ale ve výsledku s větší intuitivností a přirozeností.

Dalším praktickým důvodem pro použití multimodálního HCI, zejména s redundantními vstupy, je pomoc s přístupem k počítači tělesně nebo kognitivně postiženým. Například detekce směru pohledu s rozpoznáváním řeči nebo použití ovládání založeného na snímání elektrických signálů mozku může ve výsledku vést k vytvoření rozhraní výrazně usnadňujícího ovládání počítače[28][29].

**Biologické důvody**

Důvod realizovat rozhraní s využitím více senzorů lze nalézt v přírodě, přesněji v biologické stavbě člověka. Lidé, jako i zvířata, vnímají svět za pomocí více smyslů.

Výzkumu v této oblasti se věnuje R. Murphy [4]. Studie mozku (konkrétně tecta čili střední části mozku, viz Obrázek 3) ukazují, že podněty pro odlišné smysly jsou z počátku na nervové úrovni odděleny. Neurony asociované jedním smyslem neintereagují s neurony souvisejícími se smysly jinými. Jakmile podněty dosáhnou mozku, sbíhají se do stejné oblasti - colliculum superior. Je to část tecta, která, jak se zdá, je zodpovědná za pozornostní a orientační funkce. Neboli část mozku, kde si kočka všimne pohybu a otočí se, aby prozkoumala, kde se nachází myš. Colliculum superior také dostává signály z kůry mozkové. Ta moduluje nebo ovlivňuje výsledné chování. Zatímco většina signálů vcházejících do colliculum superior je specifická v rámci jednoho smyslu, odhadem 75% vycházejících neuronů jsou vícesmyslové (ilustrace Obrázek 3) - to znamená, že reagují na stimuly od více než jednoho smyslu. Navíc výstup těchto neuronů může být silnější, když více vstupních neuronů obdrží slabý stimul, než když jeden vstupní neuron obdrží silný signál. Ikdyž výstup tecta směřuje do různých části mozku, zdá se, že tyto vícesmyslové neurony formují cesty přímo do svalů a center pro kontrolu chování. Z toho lze vyvodit, že použití multimodálních rozhraní by odpovídalo struktuře mozku, což je žádoucí, pokud chceme dosáhnout co největší přirozenosti v komunikaci mezi člověkem a strojem.



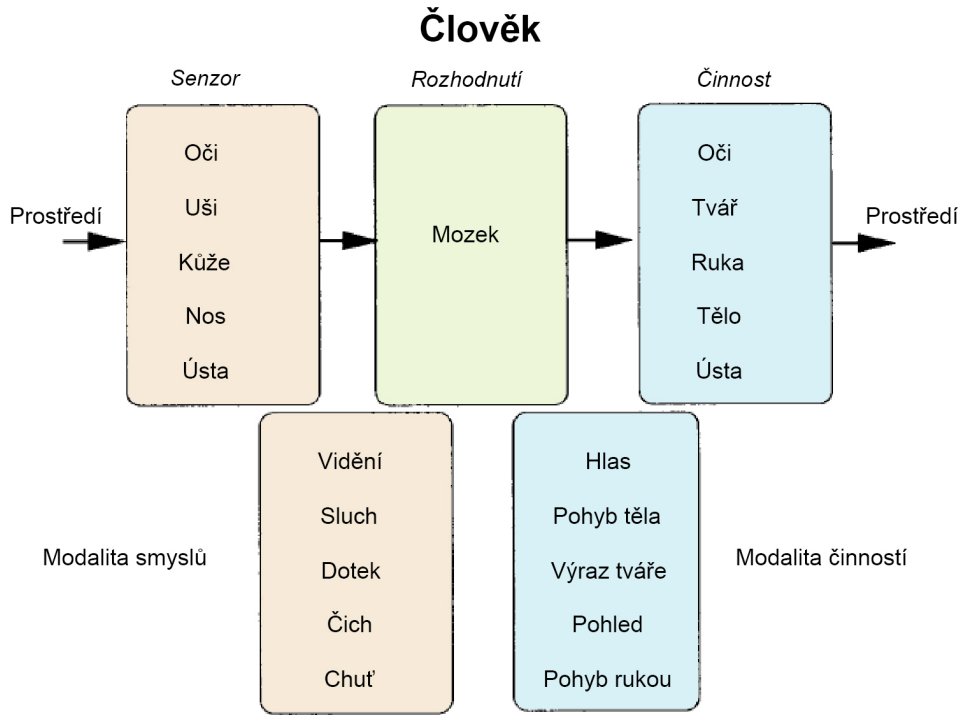

**Obrázek 3:** Vlevo: ilustrace vícesmyslového neuronu,
vpravo: střední část mozku s modře zvýrazněným colliculum superior [10].

**Matematické důvody**

Další pohled na to proč, jak a kdy použít více signálů, pochází z oblasti fúze senzorických dat. Fúze dat jako oblast výzkumu existuje již několik desítek let, ale jejím hlavním zaměřením byla detekce cíle. Detekce cíle se zabývá nalezením optimálního způsobu integrace dat z rozdílných senzorů (radar, infračervené kamery atd.), za účelem dosažení „nejlepší“ míry detekce. Důvod pro kombinaci různých senzorů pochází ze statistické analýzy dat. Nevýhoda použití pouze jednoho senzoru je taková, že tento nemusí být schopný dostatečně redukovat nejistotou správnosti v rozhodování. Nejistota přichází v momentě, kdy některé příznaky chybí – když senzor nedokáže určit veškeré potřebné vlastnosti, nebo když jsou vyhodnocení dvouznačná. Na druhou stranu je dobře známo, že je statisticky výhodné, pro získání lepšího odhadu díky redundantnosti pozorování, kombinovat více měření ze stejného zdroje. Je také známo, že použití různých typů senzorů může zvýšit přesnost výsledků [5].

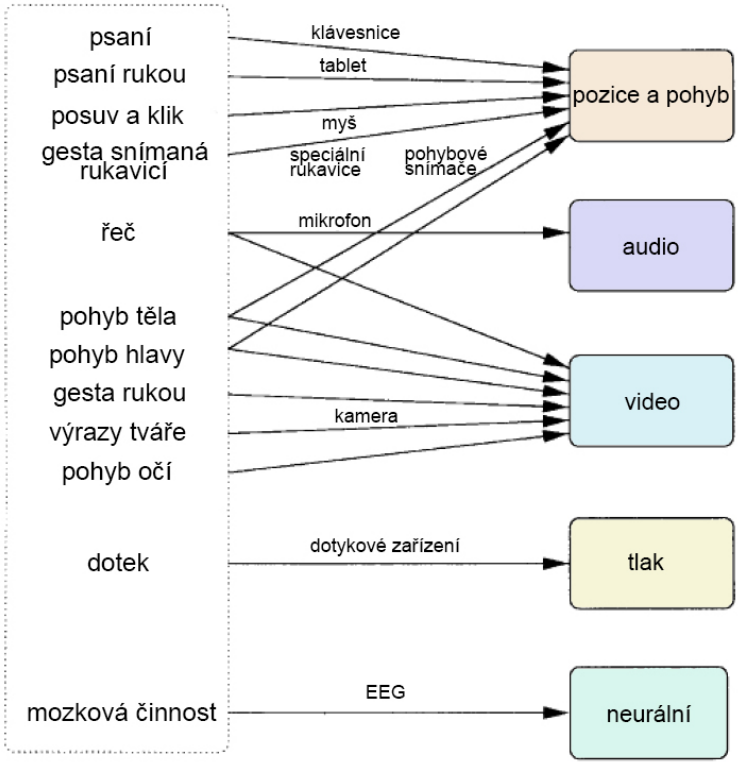
## 2.3 Komunikace člověka a počítače

Člověk vnímá prostředí, ve kterém se pohybuje, za pomoci smyslů a interaguje s ním prostřednictvím aktuátorů, jako jsou ruce, tělo, tvář a hlas (více Obrázek 4). Mezilidská komunikace v sobě zahrnuje vysílání a přijímání verbálních a neverbálních signálů, často navíc v kontextu prostředí, v jakém se nachází. V případě komunikace člověka se strojem vnímá přístroj akce člověka. Aby interakce mezi člověkem a strojem byla co nejpřirozenější, bylo by vhodné, aby počítač byl schopný interpretovat všechny akce, jaké člověk během sdělování informací vykonává. Proto je sledován pohyb hlavy, výrazy tváře či řeč, ale také projevy, které člověk vnímat nedokáže, například aktivita mozkových vln či přesná poloha ruky.




**Obrázek 4:** Modality v lidských smyslech a činnostech [5].

Komunikaci lze rozdělit do dvou kategorií, a to lidských akcí a počítačové detekce. Obrázek 5 ukazuje, jak tyto dvě kategorie spolu souvisí. Konkrétní lidská akce, může být počítačem detekována různými prostředky (například řeč lze snímat pomoci mikrofonu, kamerou sledovat pohyb rtů). Toho lze využít pro fúzi dat za účelem zvýšení robustnosti detekce lidských činností [1].




**Obrázek 5:** Zobrazení lidských činností na rozhraní počítače [5].

Velké množství lidských akcí může být začleněno do výsledného rozhraní díky vhodnému návrhu snímacích metod. Původně nejsledovanější akcí byl pohyb ruky. Nejspíše kvůli přesnosti s jakou jí dokážeme manipulovat a možnosti snadného polohování a výběru objektů. Klávesnice poskytuje přímý způsob zadávání textu, kdežto analýza řeči jej umožňuje generovat jen do určité, technologiemi určené, úrovně. Podobně je omezen i pohyb ruky, který je běžně limitován na pohyb po dvourozměrné ploše. Jako další vstup se nabízí gesta rukou, a to od jednoduchých manipulačních a komunikačních gest po komplexní příkazy, zadávané pomocí snímání kamerou nebo speciální rukavice.

Snímání audia díky mikrofonu a jeho zpracování pomocí technik automatického rozpoznávaní řeči (Automatic Speech Recognition - ASR) dovoluje interpretovat lidskou řeč. Řeč je nejspíše nejpřirozenější způsob komunikace, proto se již delší dobu věnuje pozornost vývoji ASR technik pro využití v rámci HCI. Dnešní ASR techniky jsou sice stále náchylné na šum v signálu, ale vývoj nadále pokračuje a díky nárůstu výpočetního výkonu dnešních přístrojů lze očekávat zlepšení této situace [8].

Videokamera, spolu s technikami zpracování obrazu a počítačového vidění, poskytuje další zdroj informací o činnosti člověka. Vedle již zmíněných gest rukou, pohybu hlavy, očí či rtů, existují kamery určené pro sledování polohy rohovky nebo pohybová periferie Kinect, poskytující komplexnější představu o pohybu celého těla či vybraných partií. Jedná se ovšem o velký tok informací vyžadující značný výpočetní čas a také odezva je závislá na rychlosti snímaní kamery.

Dále existují zařízení určená na snímání síly či tlaku, poskytující zpětnou vazbu, umožňující vnímat dotykem tvar virtuálních objektů. Tyto přístroje jsou využívány například pro výstavbu chirurgických simulátorů.

Snímání mozkové aktivity lze v dnešní době sledovat i neinvazivně – pomocí speciální helmy, v míře umožňující základní ovládání přístrojů. Tento způsob se nabízí jako pomoc handicapovaným při ovládání počítače. Nevýhodou je však náročnost osvojení si daného způsobu ovládání, často vyžadující dlouhodobý tréning [5][26].

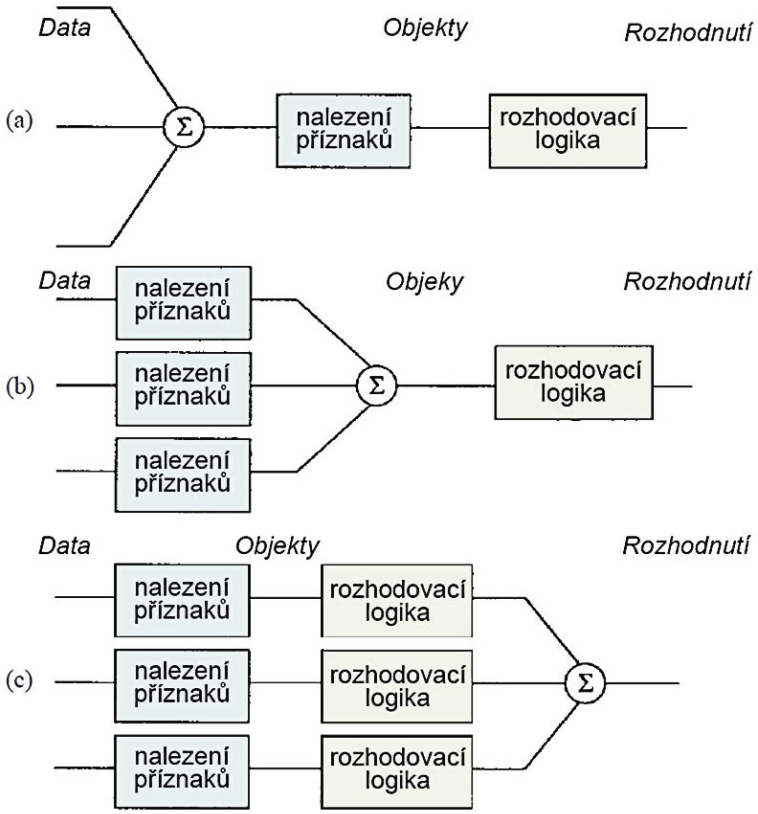
## 2.4 Architektura multimodálního HCI

Zatím jsme se seznámili s různými druhy modalit, které můžeme použit pro výstavbu rozhraní. Ta však mají různé formy signálu a frekvence aktualizace. To z tvorby HCI dělá náročnou úlohu. Musíme určit, kdy sloučit jednotlivé signály, a tím také stanovit úroveň abstrakce fúze. Před samotnou fúzi je dobré si odpovědět na několik otázek.

Za prvé – jak spolu souvisejí signály, pokud se jedná o přirozenou mezilidskou komunikaci. Odpověď zpravidla získáme z oblasti psychobehavioralního výzkumu [6]. Například bylo zjištěno, že gesta rukou a řeč jsou blízce svázány, a pochází z jednoho společného mentálního konceptu. Gesta se vyskytují souběžně s jejich řečovými ekvivalenty, a také pohyb rtů úzce souvisí s mluveným slovem.

Druhou otázkou je, zda spojitost mezi komunikačními kanály je zachována pokud se jedná o komunikaci mezi člověkem a strojem. Experimenty potvrzují, že opravdu, ikdyž se jedná o HCI, jsou souvislosti v mozku zachovány. Například při pokusech s dotykovými gesty a hlasovým ovládáním se došlo k závěru, že ke sjednocení dochází na sémantické úrovni, kdy jsou gesta použita pro sdělení informace o pozici, zatímco řeč předává informaci o činnosti v dané části komunikace [3].

Dalším problémem je určení úrovně fúze dostupných dat. Většinou se uvádí tři úrovně integrace dat: senzor neboli data fusion, object nebo feature fusion a decision fusion [5] (viz Obrázek 6).




**Obrázek 6:** Tři úrovně fúze senzorických dat [5].

a) Data fusion – nejnižší úroveň integrace, zahrnuje spojení nezpracovaných vstupů ze senzorů. Tohoto se historicky používalo v případě, kdy signály byly stejného typu (audio ze dvou blízce umístěných mikrofonů či video ze stereokamery).

   Pro multimodální rozhraní se nejedná o příliš vhodný typ, neboť chceme vytvořit rozhraní založené na zcela odlišných vstupech. Přesto ze zmíněných druhů integrace datová fúze poskytuje nejlepší úroveň detailů získaných informací. Na druhou stranu je datová fúze nejnáchylnější na chyby způsobené šumem přítomným ve vstupním signále či na výpadky jednotlivých senzorů.

b) Feature fusion – s tímto typem fúze se v rámci multimodálních HCI setkáváme častěji. Předpokladem je, že z každého zdrojového signálu jsou nejprve extrahovaný objekty a teprve ty potom následovně sloučeny.

   Oproti fúzi datové nemusí vstupní signály pocházet z totožných senzorů, ale stále musejí být úzce svázány (například audio vstup řeči a snímání pohybů rtů). Tento typ integrace dat je méně náchylný na chyby způsobené šumem, ale takto získaná informace obsahuje méně detailů (datová fúze by ze dvou vstupů byla schopná kromě přepisu řeči na text určit i polohu řečníka; feature fusion poskytne přesnější výstup i při hluku na pozadí, ale postrádá dodatečné informace).

c) Decision fusion – fúze rozhodování je nejrozšířenějším typem. Je založená na integraci jednotlivých pozorování. Například jakmile je pohyb rukou vyhodnocen jako ukazovací a řečový příkaz rozpoznán jako „přibliž“, mohou být tyto dva vstupy interpretovány spolu pro přiblížení zvýrazněné oblasti.

   Tento druh integrace je nejodolnější na chyby ve vstupech či výpadky jednotlivých senzorů (při výpadku může poskytnout alespoň částečnou funkčnost, kdežto ostatní druhy mohou být zcela ochromené). Nevýhodou však může být, že při ztrátě informace na nižší úrovni

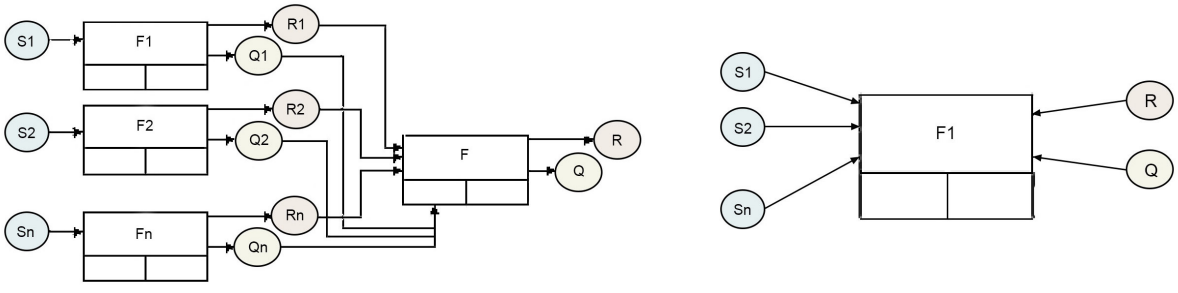
se nemusí podařit nalézt souvislost mezi jednotlivými vstupy (když je člověk mimo rozsah kamery a zadá příkaz „zapni“, nelze dohledat, o jaký objekt se má jednat) [5][25].

Kromě zmíněného způsobu rozdělení metod určených pro fúzi senzorických dat, existuje i obecnější klasifikace - podle způsobu zpracování, jak popsal L. Wald [12].

a) Centralizovaná – výpočet je soustředěn na jednom místě či v jednom uzlu. Tam je zpracovávána sada dat ze všech senzorů. Obrázek 7 ukazuje, jak *n* vstupů ze *Si* (vstupy mohou být data ze senzorů nebo už získaná rozhodnutí) je předáváno jednomu uzlu. Výsledky *R* a parametry kvality *Q* jsou získány ze zpracování všech zdrojů dostupných v daném okamžiku. Výhodou centralizovaného zpracování teoreticky je, že poskytuje optimální výsledky díky dostupnosti všech informací. Ztráta informace je minimalizována, neboť data jsou zpracována přímo, bez aproximace atributů vektory stavů, případně jiným způsobem. Na druhou stranu, pokud má jeden ze zdrojů vysokou chybovost nebo nízký odstup signálu od šumu a výsledek fúze na něm z větší míry záleží, může znehodnotit celou sadu dat.

b) Decentralizovaná – nabízí vysokou míru flexibility i modularity. Tato architektura je také nazývána autonomní, protože zahrnuje samostatné zpracování každého ze zdrojů informací (případně skupiny zdrojů), dokud nedojde k fúzi, sémanticky na vyšší úrovni, dat. Zpravidla se decentralizovaný způsob používá v situaci, kdy je problém s komunikací (malá šířka pásma, chybovost kanálů). Navíc se tímto způsobem lze vyhnout problému s rozdílnou rychlostí pořizování dat (kdy by mohlo docházet k opětovnému zpracování dat z pomalejších senzorů, při změně signálu ze senzorů s rychlejší frekvencí snímání). Zpracování dat v uzlu a následný přenos získaného rozhodnutí zpravidla nevyžaduje vysoký tok dat. Další uplatnění se nachází v nebezpečných prostředích (například na bojišti), kdy výpadek jednoho uzlu neochromí celek.

c) Hybridní – kombinací fúze centralizované a decentralizované, lze ještě lépe přizpůsobit systém dané úloze (stejné nebo podobné signály zpracovávat, pokud je to vhodné, v jednom uzlu umístěném do centralizovaného či decentralizovaného celku).

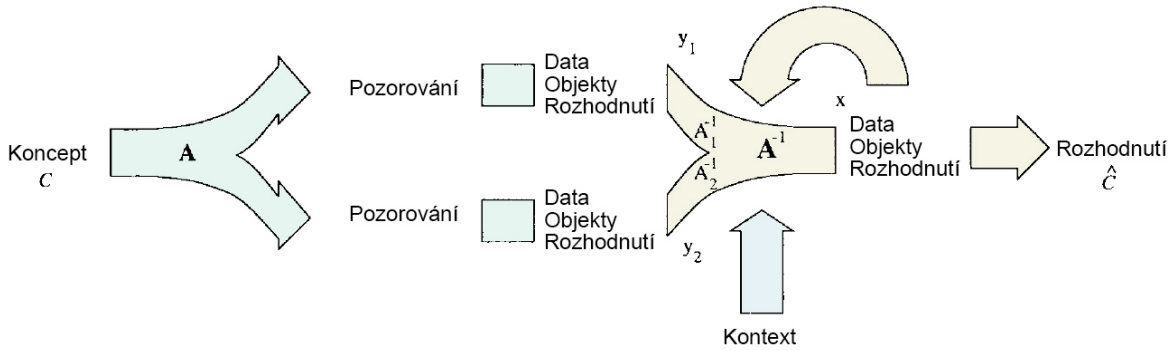



**Obrázek 7**: Vlevo: centralizovaná architektura, vpravo: decentralizovaná [12].

# 2.5 Integrace vstupů

Pro každý druh fúze, zmíněného v minulé sekci, je nutno použít jiný způsob, jakým se získanými informacemi pracovat. Přehled principů obsažených v této kapitole popsali R. Sharma, V. Pavlovic a T. Huang [5].

Obecný model fúze (Obrázek 8) je postaven na předpokladu, že koncept, skrývající se za každou lidskou akcí, je vyjádřen pomocí více komunikačních kanálů a také je přijímán pomocí více smyslů. Cílem fúze je tedy sloučit rozdílné abstrakce pozorované činnosti pro co nejlepší rozpoznání komunikovaného konceptu. Pro zredukování možných výsledků lze také využít kontext, v jakém se systém právě nachází.




**Obrázek 8:** Obecný model fúze [5].

Fúzi na úrovni objektů lze rozdělit do dvou kategorií:

a) FIFO (feature in, feature out) – vyžaduje další klasifikátor pro dosažení rozhodnutí. Dobře známou metodou používanou pro tento typ fúze je Kalmanův filtr. V tomto případě se neprovádí fúze časové řady vektoru příznaků, ale slučuje se příznaky patřící různým zdrojům.

b) FIDO (feature in, decision out) – nepotřebuje další klasifikátor, protože je založena na pravděpodobnostních sítích, jakými jsou neuronové sítě nebo skryté Markovovy modely.

Fúze rozhodování, jak již bylo naznačeno na obrázku Obrázek 6, spočívá v integraci rozhodnutí získaných z jednotlivých komunikačních kanálů v jednu multimodální zprávu. Tento typ fúze se ukazuje vhodný v případě, kdy příznaky pocházející z jednotlivých signálů spolu nesouvisí natolik, aby mohly být asociovány na úrovni objektů. Namísto toho jsou vyhodnoceny zvlášť a jejich výsledky sloučeny v jedno rozhodnutí. Tento přístup však vyžaduje odlišný způsob vyhodnocení. Často se používají dva mechanizmy:

a) Rámce – pojem rámec původně pochází z oblasti umělé inteligence. Tam se rámcová metoda používá pro reprezentaci znalostí. Rámec je speciální datová struktura, například ve formě tabulky, s jasným označením a rubrikami obsahující hodnoty atributů i odkazy na další tabulky. V případě rozhraní lze informace získané z jednotlivých zdrojů přiřadit atributům rámce. Mohou se doplňovat (řeč může popisovat barvu nebo tvar objektu, gesta jeho polohu) nebo mohou určovat stejnou vlastnost.

b) Softwaroví agenti – jedná se o softwarovou entitu pracující samostatně a průběžně, nacházející se v určitém prostředí, často spolu s dalšími agenty. V principu by měli pracovat samostatně, bez jakéhokoli zásahu člověka. Možným řešením by bylo použití jednoho agenta pro analýzu řeči, druhého pro sledování pohybu rukou a třetího pro sledování umístění člověka v prostoru. Agenti zpracovávající signály komunikují s centrálním agentem. Ten zprostředkovává zasílání zpráv mezi agenty (například i agentovi pro interpretaci výsledků).

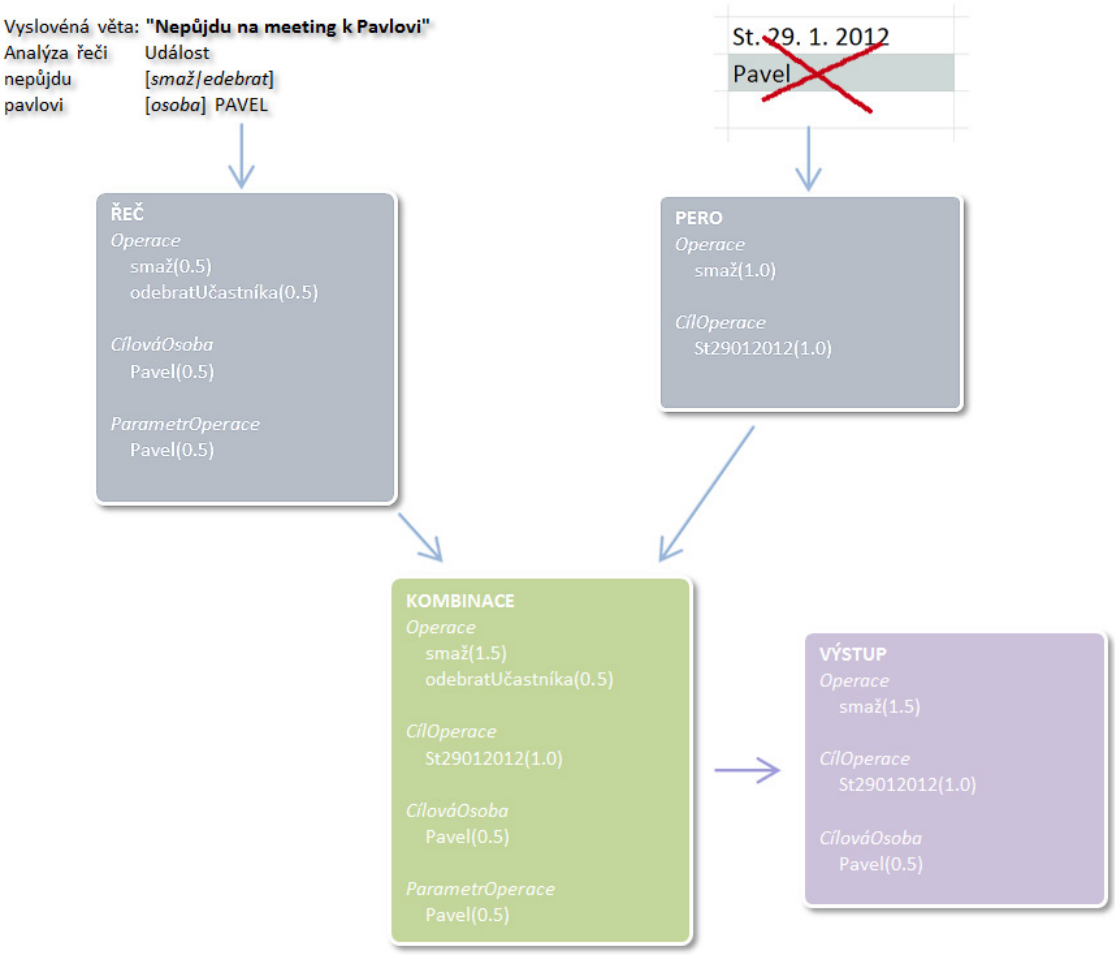
Pokud jsou k dispozici jednotlivé modality, je potřeba určit způsob, jakým bude získaná výsledná informace. Je nutno se vypořádat jednak s časovými rozdíly (intervaly mezi výskyty jednotlivých částí příkazu nebo i s rozdílnou rychlostí, jakou lze pomocí daných kanálů komunikovat; slovní příkazy se mohou vyskytovat rychleji než gesta či naopak), tak i pracovat se sémantickou rozdílností jednotlivých částí komunikované zprávy. Ty mohou nést významově stejnou informaci (gestem i slovním příkazem lze najednou předávat příkaz, například pro vypnutí – redundance) nebo samostatně nepostačují pro rekonstrukci celého příkazu.

**Frame a Slot filling**

Integrace vstupů je často založená na strukturách nazývaných frame – dále rámec (nejen u rozhraní člověk počítač, ale i články věnující se komunikaci mezi člověkem a strojem zpravidla zmiňují tuto metodu). Ty lze definovat jako abstraktní reprezentaci entit reálného světa a jejich vztahů. Každá entita je složena ze sady atributů nazývaných sloty. Ty nabývají hodnot získaných skrze senzory. Hodnota slotu může být určena přímo, nebo zděděna pomocí vztahů mezi rámci [22].

Sloty mohou být založené na široké škále druhů dat. Například může být v podobě slova získaného z rozpoznání řeči stanovena pozice z polohy myši či elektronického pera. Sloty mohou také obsahovat časové značky určené podle toho, kdy byly pořízeny atributy či vytvořen rámec.

Rámce lze rozdělit do tří kategorií: vstupní, integrační a výstupní. Vstupní slouží k uchování informace či události, generované vstupním zařízením nebo senzory umístěnými v prostředí. Každý rámec může obsahovat i prázdné sloty (neobsahující požadovaná data), pokud jsou získané informace pouze částečné. Integrační rámce jsou vytvářeny dialogovým systémem v průběhu interakce s uživatelem. Hodnota slotů v těchto rámcích je určena pomocí kombinace slotů ze vstupních rámců. Výstupní rámce se používají pro generování reakce systému (například pro změnu stavu zařízení – rozsvícení světla, pro přenos informace k uživateli – syntéza řeči, zobrazení grafické informace na displeji) [15].



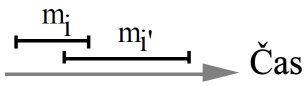

**Obrázek 9:** Ukázka interpretace založené na rámcích. Kalendářní aplikace používající více vstupů. Modře vstupní rámce, zeleně integrační rámce, fialově výstupní rámce [14].

Samotný mechanizmus slučování vstupních rámců, jak ho demonstruje Obrázek 9, lze pojmout jako sjednocení jednotlivých sad hodnot a sečtení korespondujících ohodnocení. Pokud jsou rámce slučovány rekurzivně, je výsledkem rámec zahrnující i alternativní interpretace vstupních dat. Při tvorbě výstupního rámce se zvolí kombinace s nejlepším ohodnocením [14], [15].

**Melting pot**

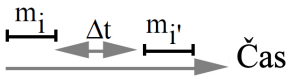
Další přístup, jak reprezentovat informace, navrhli Nigay a Coutaz [16]. V článku pojednávají o takzvaných melting pot. Melting pot (dále MP) lze považovat za rozšíření principu rámců, kdy jsou přijaté informace kombinované s časovým razítkem. To umožnuje, aby fúze byla založena na čase, kontextu a kombinaci vhodných MP. U této metody kombinace vstupů se vyskytují tři druhy fúze: mikrotemporální, makrotemporální a kontextová.

1) Mikrotemporální fúze se používá ke kombinaci souvisejících informací obdržených paralelním či pseudoparalelním způsobem. Provádí se, pokud se části vstupních melting potů doplňují nebo když jsou si blízké v čase (časy trvání se překrývají). Obrázek 10 demonstruje možnou konfiguraci dvou MP s označením $m_i$ a $m_{i'}$ která určuje aplikaci mikrotemporální fúze.




**Obrázek 10:** Dva MP vybraní pro mikrotemporální fúzi kvůli překrývajícímu se času trvání [17].

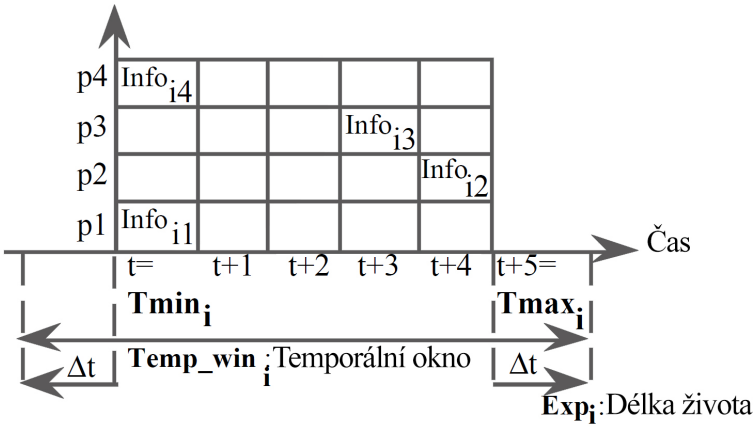
2) Makrotemporální fúze je použita v případě, kdy se snažíme skombinovat související informace pořízené sekvenčně nebo případně paralelně, ale zpracované sekvenčně. Také i v případě opoždění způsobeném časovou náročnosti analýzy dat (například kdy čas potřebný pro analýzu řeči přesahuje dobu zpracování vstupu zadaného prostřednictvím klávesnice či myši). Makrotemporální fúze se provádí v případě, kdy se jednotlivé části MP doplňují a jejich časové intervaly spolu nesouvisí, ale zároveň náleží do stejného temporálního okna. Obrázek 11 ukazuje závislost mezi dvěma MP, kdy lze uvažovat o makrotemporální fúzi.




**Obrázek 11:** Příklad makrotemporální fúze dvou MP [17].

3) Kontextová fúze slouží ke kombinaci korespondujících informací, bez ohledu na časovou informaci (například, pokud uživatel přeruší činnost zadávání informace a vrátí se k ní za několik minut). Kontextová fúze vychází z aktuálního kontextu. V systémech pro fúzi dat kontext koresponduje s požadavkem, který systém v daném okamžiku vyřizuje (třeba když uživatel zadává adresu pro vyhledání trasy, jedná se o jiný kontext a jiná vstupní data, než když uživatel prohlíží fotografie). Tento přístup sloučí nový vstupní MP $m$ s dalšími $M$ obsaženými v aktuálním kontextu, pokud informace obsažené v $m$ doplňují informace jednoho z MP v $M$ [17].

Máme-li definovány principy mechanizmu fúze, je potřeba ještě specifikovat metriky samotných MP. Jak demonstruje Obrázek 12, lze melting pot $m_i$ určit jako:
$m_i = (p1, p2 \dots pn)$: $m_i$ je tvořeno n strukturami $p1, p2 \dots pn$.
$info_{ij}$: informace uložena v $p_j$, která je součástí $m_i$
T$info_{ij}$: časové razítko $info_{ij}$
T$max_i$: časové razítko nejaktuálnější informace obsažené v $m_i$
T$min_i$: časové razítko nejstarší informace obsažené v $m_i$
T$emp_win_i$: délka temporálního okna pro $m_i$
$\Delta t$: zbylá doba života pro $m_i$




**Obrázek 12:** Ilustrace metrik použitých pro definování MP $m_i$.

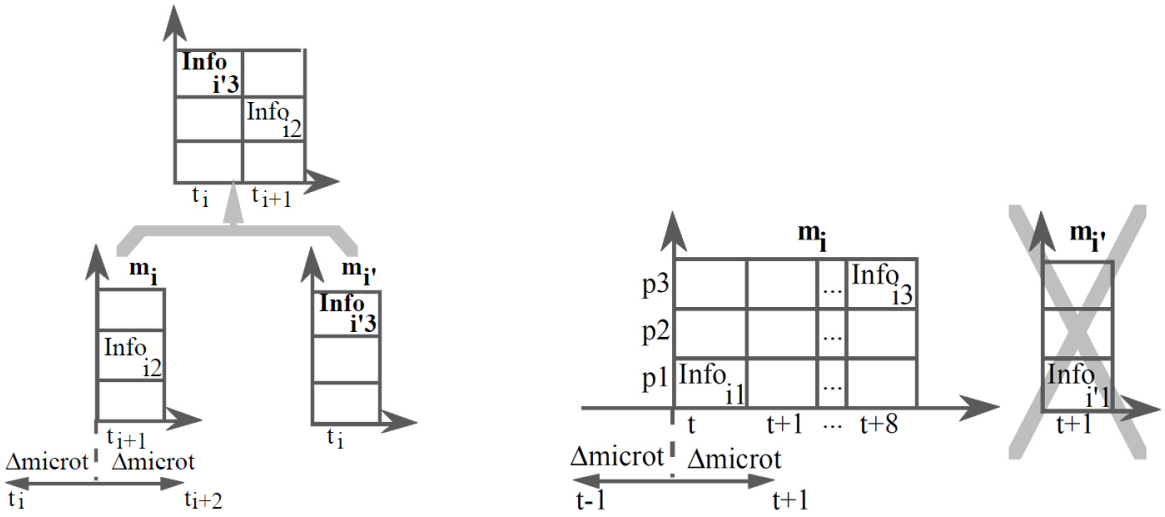
Melting pot zapouzdřuje sadu informací $pn$. Každá z nich má přiřazenou hodnotu o času vzniku. Pro usnadnění práce lze dopočítat dobu výskytu ($Tmax$ a $Tmin$, čili časové okno, ve kterém byla pořízena jednotlivá data).

Temporální okno udává časovou vzdálenost mezi dvěma sousedními melting poty (pro $m_i = (p1, p2 \dots pn)$ je temporální okno $Temp\_win_i = Tmin_i - \Delta t, Tmax_i + \Delta t$ ), tato informace se používá pro aplikaci makrotemporální fúze.

Posledním pojmem týkajícím se MP je délka života ($Exp_i = Tmax_i + \Delta t$), sloužící pro stanovení, zda je potřeba melting pot odebrat ze sady kandidátů pro fúzi[17].

Samotná fúze se řídí několika pravidly. Jelikož je kladen důraz na paralelizmus na úrovni akcí uživatele, systém se při příchodu nové události pokouší jako první uplatnit mikrotemporální fúzi. Jak ukazuje Obrázek 13, ke sloučení dvou MP dochází, když obsah příchozí události doplňuje obsah již existujícího melting potu, a jejich časové razítko je dostatečně blízké (uvnitř časového intervalu $\Delta microt$). Tuto situaci lze ilustrovat příkladem, kdy uživatel vysloví operaci a současně ukáže na předmět. Obě události nastávají téměř současně a obdržené informace jsou komplementární.

Je však potřeba se vyrovnat i s možnou redundancí. Jelikož uživatel má možnost komunikovat paralelně, mohou obdržené informace popisovat stejnou vlastnost. V tomto případě je nutno jednu z těchto událostí ignorovat, nebo již existující informaci přepsat novější [20][24].




**Obrázek 13:** Příklad mikrotemporální fúze a redundance příchozí informace [17].

Makrotemporální fúze se provádí stejně jako mikrotemporální, pouze místo časového intervalu $\Delta microt$ se používá delší časové období – temporální okno.

Oproti mikro a makro temporální fúzi se kontextová neřídí časem. Ta, jako poslední krok v mechanizmu fúze, operuje s pojmem kontext. Jak již bylo zmíněno, kontext nám může udávat, jaký požadavek systém vyřizuje a naopak. V tomto případě pak z kontextu vyplývá, jaké informace je potřeba získat (jaké strukturální části musí mít melting pot vyplněné) proto, aby mohl být výsledný MP považován za úplný a provést pak požadovanou akci [17].

**Grammar-driven strategy**

O dalších přístupech k fúzi dat pojednávají Portillo, García a Carredano [18], když shrnují stávající přístupy (grammar-driven strategy - fúze unifikací a dialogue-driven strategy - dialogem řízená fúze) a navrhují další (hybridní). Grammar-driven strategy (dále jen unifikace) je oproti předchozím metodám integrace informací postavena na pravidlech definujících vymezení modalit a času pro jednotlivé operace. Tento přístup umožnuje výstavbu širokého spektra systémů využívajících různé vstupy (od systémů založených pouze na řeči po aplikace využívající události obsahující různé modality, například řeč a zároveň klik).

Při použití této strategie pro integraci vstupů je potřeba určit, jakým způsobem jsou multimodální události vnímány. Jestli jako jedna jednotka informace, či několik komplementárních akcí.

Pro první případ by se zdálo vhodné použít jednu gramatiku schopnou se vyrovnat s kombinovanými modalitami. Obsahovala by proto pravidla s různými zakomponovanými modalitami (například pravidlo jako $Příkaz \rightarrow Akce(\text{Řeč})Parametr(\text{Klik})$ ).

Problém by však mohl nastat v případě, jestliže by systém umožnoval několik souběžných činností. Vyrovnat se se vzniklou dvojznačností by nemuselo být možné bez dodatečných informací. Proto se v tomto případě ke vstupu přiřazuje i druh modality, čas počátku a čas konce události. Spolu s logickými operátory lze takto definovat omezení času a modalit pro jednotlivá pravidla gramatiky. Obrázek 14 demonstruje, jak by mohla výsledná pravidla vypadat a jak by vypadala odezva systému na vstup uživatele v podobě vyslovení „zapni to“ a ukázání na lampu.

```
(Pravidlo1: Příkaz-> ComdOn DevSpec)
   { @up = @self-1;}
(Pravidlo2 : Příkaz-> DevSpec ComdOn)
   @up.DevSpec =a @self-1;
   @if((@self-1.MOD == CLICK) &&
   (@self-2.MOD == VOICE))
      @then {
         @if ((@self-1.INIT-@self-2.INIT <= 5) &&
         (@self-1.INIT-@self-2.INIT <=-5))
            @then { @break();}
            @else { @up.MOD =a [VOICE,CLICK];
         @if((@self-1.INIT <= @self-2.INIT))
            @then { @up.INIT =a @self- 1.INIT;}
            @else { @up.INIT =a @self-2.INIT;}
         @if((@self-1.END >= @self-2.END))
            @then { @up.END =a @self-1.END;}
            @else { @up.END =a @self-2.END;}
      }
   }
   @else {break();}
}
```

```
DMOVE: specifyCommand
TYPE: CommandOn
ARGS: specifyParameter
MOD: řeč
INIT: 00:01
END: 00:03
specifyParameter:
      DMOVE: specifyParameter
      TYPE: Zařízení
      CONT: Lampa_1
      MOD: click
      INIT: 00:02
      END: 00:02
```

**Obrázek 14:** Příklad unifikačních pravidel a výstupu unifikačního procesu [18].

**Dialogue-driven strategy**

Dialogem řízená fúze, ikdyž využívá údaje o různých modalitách, nezahrnuje záznamy kombinující informace různých modalit, jak tomu bylo v minulém případě. Pracuje pouze s jednomodálními

záznamy (v případě „zapni to“ spolu s ukázáním na lampu se vstup zpracuje jako dvě nezávislé, spolu nesouvisející události, ale se stejným časem počátku a konce).

Všechny uživatelské vstupy se uchovávají ve vstupní frontě. Možná souvislost mezi samostatnými událostmi se určuje z časového razítka přiřazeného každé události. Pokud jsou dostatečně blízké v čase, považují se za souběžné nebo pseudo-souběžné a následně se rozhoduje, zda jsou komplementární.

Časové okno, ve kterém se dá různé vstupy považovat za potenciálně související, se určuje empiricky (přístup podobný k mikro a makro temporální fúzi u melting pot). Jakmile jsou tímto způsobem vybrány události, které mohou spolu souviset, rozhodne se podle následujících pravidel, zda se mají opravdu za komplementární považovat:

- Jedna z událostí spouští Dialogové Pravidlo (podobně jako pravidla z minulého přístupu), další patří do jeho předpokladů.
- Obě jsou součástí již aktivovaného Dialogového Pravidla.
- Neexistuje Dialogové pravidlo, se kterým by nově vzniklé bylo v konfliktu.

Jestli události nespadají do stejného časového okna (byly vykonány příliš dlouho po sobě), uvažuje se jiné možnosti závislosti:

- Jedna z událostí ukončuje minulou úlohu, druhá začíná další.
- Každá z události dokončuje odlišnou úlohu.
- Každá z události může dokončit více z úloh, což implikuje zjevnou dvouznačnost.
- Dvě na sobě nezávislé úlohy jsou započaty souběžně.

Pokud jsou události vyhodnoceny jako doplňující se (například jako by mohlo nastat u Obrázek 15), jsou spojeny do jedné informace s výsledkem podobným jako v předchozím přístupu. Ve výsledku pak dialog manager (jednotka zajišťující aktualizaci kontextu dialogu v rámci interpretace komunikace a rozhodující jakou akci provést a kdy) určí, jestli vstupy lze skombinovat ve validní konstrukci a vyvolat požadovanou akci[23].

DMOVE: specifyCommand
TYPE: CommandOn
ARGS: specifyParameter
MOD: Řeč
INIT: 00:01
END: 00:03

DMOVE: specifyParameter
TYPE: Zařízení
MOD: Gesto
INIT: 00:02
END: 00:02

**Obrázek 15:** Dvě události, které pro daný vstup byly vyhodnoceny jako související a ze kterých pak systém kombinací získá požadovanou činnost [18].

**Hybridní**

Hybridní systémy se snaží kombinovat unifikační a dialogový přístup se zachováním co nejmenší složitosti. První je založen na gramatice, která pracuje s multimodálními informacemi pomocí definovaných časových a modálních podmínek. Takto nevyžaduje další rozhodovací logiku, pouze definování pravidel. Oproti tomu dialogem řízená fúze pracuje s jednomodálními událostmi a podle pravidel určuje, jestli spolu souvisí a následně je zpracovává. Dá se předpokládat, že díky pokročilejší logice, bude takový systém poskytovat lepší výsledky.

Hybridní přístup se snaží kombinovat multimodální záznamy s přihlédnutím k dalším informacím používaným v Dialogovém rozhodovacím procesu.

## 2.6 Existující aplikace

Pokud nebereme v úvahu všechny druhy rozhraní, jaké se mohou vyskytovat, a zaměříme se pouze na rozhraní založená na fúzi multimodálních událostí, dostaneme stručnější seznam existujících řešení (navíc s datem vzniku v posledních letech).

| Název aplikace | Mechanizmus | Vstupy | Typ aplikace |
|---|---|---|---|
| Pac-Amodeus | Melting pot | řeč, klávesnice, myš | Vyhledávání letu |
| Quickset | Unifikace | elektronické pero, řeč | Tréninkové simulace |
| Tycoon | Dialogem řízená | řeč, klávesnice, myš | Editace GUI |
| iMap | Frame | řeč, gesta | Krizové řízení |
| HephaisTK | Frame | řeč, myš, hardwarový ovladač | Organizování meetingů |
| WCI | Unifikace | řeč, myš, klávesnice | Multimodální DB |
| Pate | Unifikace | elektronické pero, řeč | Design koupelen |
| PUMPP | Unifikace | řeč, gesta | Řízení dopravy |
| ICARE | Melting pot | Vizor, dotyková plocha, GPS, magnetometr, myš, klávesnice | Simulace kokpitu |
| MIMUS | Hybridní | řeč, myš | |
| FAME | Hybridní | řeč, myš, klávesnice | Digitální mluvicí kniha |

**Tabulka 1:** Stručná tabulka shrnující již existující aplikace založené na fúzi multimodálních dat [19].

Tabulka 1 ukazuje zkrácený seznam aplikací využívajících multimodální vstup. Jejich určení se od sebe dost liší, ale obecně se jedná o úlohy vyžadující komplexnější vstupní data.

# 3 Návrh systému

Tato kapitola popisuje návrh systému, jeho rozdělení na bloky a způsob, jak by měly fungovat. Návrh vychází z informací získaných studiem metod fúze dat, existujících aplikací a počáteční specifikace v úvodu. Možností jak systém realizovat je více a volba, která se pro danou aplikaci hodí lépe, by vyžadovala výzkum v rozsahu této práce, proto je návrh bloků ve formě přehledu možných realizací.

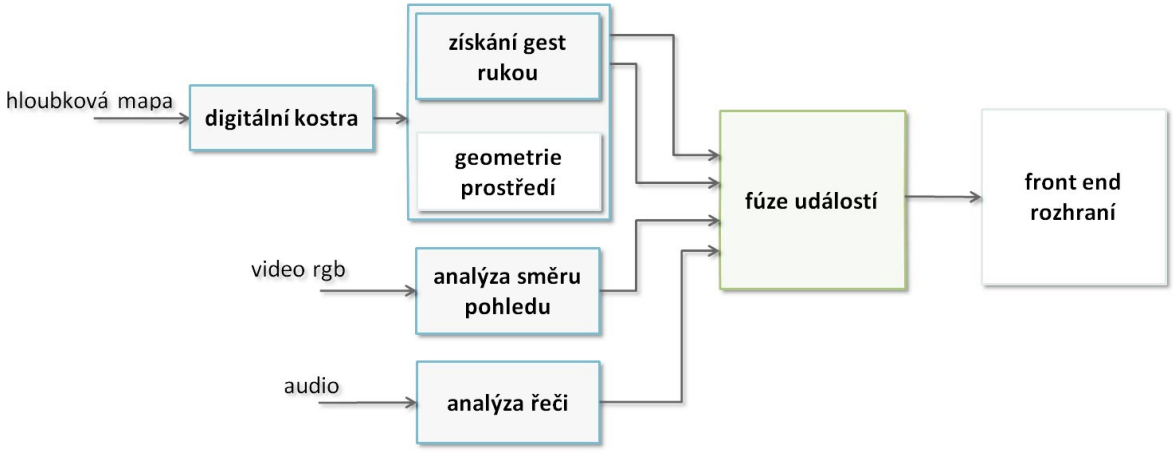
## 3.1 Specifikace systému

Před přistoupením k samotnému návrhu systému, by bylo vhodné specifikovat, co bude výsledný systém vykovávat.

Jak již bylo zmíněno, mělo by se jednat o systém, který z dat získaných od různých senzorů určí, jakou akci chce uživatel vykonat. Měl by tak poskytovat uživatelsky přívětivější rozhraní, obecně mezi člověkem a prostředím. U takového systému je potřeba stanovit jednak jakou bude mít architekturu, funkci jednotlivých bloků či formát a obsah zasílaných zpráv, ale dále je nutné nadefinovat alespoň základní příkazy, pomocí kterých může uživatel se systémem interagovat. Určení příkazů zde navrhovaného systému je komplikováno i faktem, že se má jednat o sekvenci či kombinaci samostatných akcí s odlišným způsobem provádění (vyslovení příkazu, gesto rukou, pohyb hlavy). Pro opravdu uživatelsky přívětivou komunikaci bude potřeba pořídit dostatek vstupů v podobě nahrávek běžných lidí.

Zde navrhovaný systém by měl poskytovat možnost ovládání multimediálního centra spolu se základními možnostmi interakce s prostředím. Co se týče postavení uživatele vůči systému, měl by mít možnost pohybovat se volně v prostoru (zadávat příkazy z různých míst, ne z fixně určené pozice). Poloha je ale mírně limitována zorným polem senzorů poskytujících data pro analýzu pohybu. Pro navrhované rozhraní lze ale uvažovat, že se uživatel bude nacházet ve vzdálenosti 1,5 až 4 metry od senzoru, měl by mít uživatel dostatečnou svobodu pohybu, a současně budou i vstupní data dostatečně přesná. Ovládání multimediálního centra by mělo poskytovat možnosti běžných přístupů (volba hlasitosti, pohyb v nabídce, vyvolání menu a podobně). Interakce s prostředím by spočívala v určení, se kterým objektem chce uživatel pracovat a specifikace požadované akce (například ukázaní na lampu a gesto pro ztlumení).

Na základě informací získaných z dostupných publikací jsem se rozhodl pro hybridní architekturu založenou na samostatném zpracování signálů a jejich následném centralizovaném vyhodnocení (pro bližší představu viz Obrázek 16). Přístup je možno také nazvat decision fusion se samočinnými agenty pro zpracování signálu ze senzorů. Jednotlivé vstupy jsou tedy zpracovány specifickými agenty odděleně. Získané informace o akcích uživatele jsou následně předány v podobě události centrálnímu agentu obstarávajícímu fúzi dat. Ten z příchozích událostí sestavuje komplexnější činnost, kterou má systém vykonat. Výsledek je postoupen grafickému rozhraní pro vykonání či vizualizaci.




**Obrázek 16:** Ilustrace návrhu systému pro fúzi dat z více různých zdrojů.

Jedná se o systém zaměřený na komplexní činnost. Už jen samotná analýza jednotlivých signálů je značně komplikovaný proces, výsledky je ještě potřeba následně zpracovat a výstup provést. Proto jádro této práce představuje fúze událostí získaných od jednotlivých senzorů. Pro implementaci jednotlivých bloků je co nejvíce využito dostupných knihoven či frameworků.

## 3.2 Vstupy systému

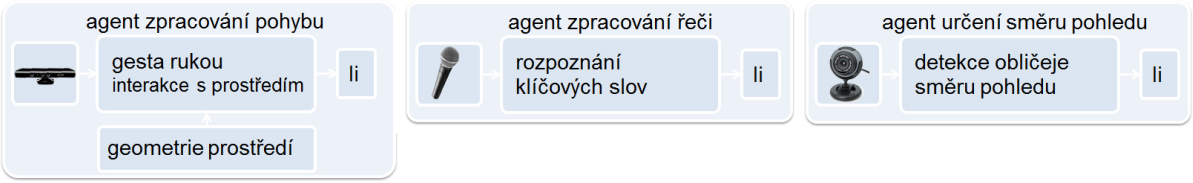
Před určením jak bude systém vypadat a co vykonávat je vhodné říci, jaké vstupy má k dispozici. Tedy kromě seznamu senzorů i jaké data poskytuji:

1) Microsoft Kinect – pro snadnější, ale hlavně přesnější sledování gest vykonaných uživatelem je vhodné využít senzor poskytující komplexnější informaci o sledovaném prostoru. Hlavní výhodou však jsou sady pro vývoj softwaru na něm postavené, poskytující přímý přístup k datům digitální kostry uživatele. Jejich formát se liší podle použité knihovny:
    a. Kinect for Windows SDK – kostra uživatele obsahuje 19nact kloubů, každý s údaji o X, Y a Z souřadnici. Souřadnice jsou v metrech - X a Y hodnota není tedy závislá na vzdálenosti od senzoru (pokud se ruka uživatele nachází 0.5 od ramene, zůstane hodnota na 0.5, ikdyž se posune dopředu nebo dozadu).
    b. OpenNi – opensource framework pro práci s kinectem. Poskytuje údaje o X, Y a Z souřadnici 15 kloubů. Souřadnice jsou v rozmezí 0 až 640 u X, 0 až 480 u Y. Ikdyž rozsah odpovídá rozlišení hloubkové mapy, jedná se o hodnoty typu float, takže přesnost není menší než u Kinect for Windows SDK.
2) Webkamera – základní zdroj dat pro počítačové vidění. Z RGB obrazu lze získat informace o poloze objektu či sledovat jeho pohyb. Nemusí se jednat o špičkovou kameru, pro účely detekce a sledování objektů postačuje obraz v rozlišení 640 na 480 bodů (pokud tedy uvažujeme nasazení systému v domácnosti a ne sledování na velké vzdálenosti).
3) Mikrofon – poskytuje audio vstup umožňující převod zvuku na řeč. Při použití mikrofonního pole lze s jistou pravděpodobností určit, odkud zvuk přichází. Formát, v jakém mikrofon poskytuje vstupní data, není příliš důležitý, zvlášť když je odstíněn zvukovou kartou či knihovnou pro zpracování řeči. Pro dosažení dobrých výsledku by bylo vhodné, kdyby umožnoval filtraci šumu prostředí (což dnes je často zahrnuto v ovladačích zvukové karty).

## 3.3 Softwaroví agenti

Po obecném rozvržení architektury je potřeba přesněji určit, co budou jednotlivé bloky vykonávat a jakým způsobem budou fungovat.

První stupeň analýzy senzorických dat probíhá v oddělených blocích. Přesněji by se mělo jednat o agenty snímající pohyb člověka, směr pohledu a rozpoznání mluveného slova. Každý z nich analyzuje data ze svého senzoru, zpracovává je a odesílá řídící jednotce pomocí komunikačního kanálu (jak ukazuje Obrázek 17).




**Obrázek 17:** Ukázka agentů pro jednotlivé vstupy.

**Agent zpracování pohybu**

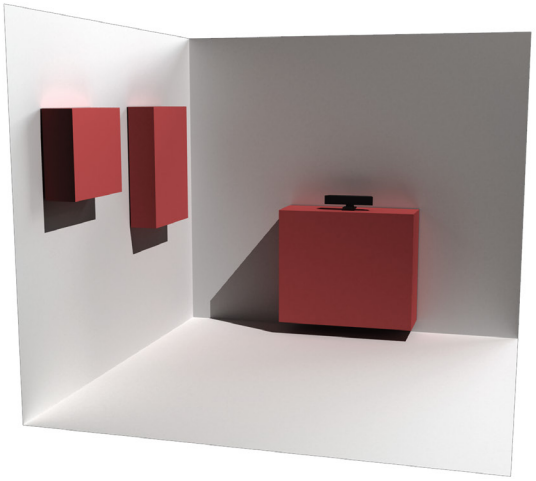
Díky senzorům schopným získat hloubkovou mapu snímaného prostoru, navíc přímo podporujících tvorbu digitální kostry uživatelů, je možno přímo sledovat pohyb jednotlivých lidí nacházejících se před senzorem. Zařízení poskytujících možnost jednoduššího sledování akcí uživatele je již komerčně, za poměrně přijatelnou cenou, dostupných více.

Z digitální kostry pak lze přímo, bez potřeby složitějších knihoven, určit pohybová gesta (za základní řešení lze považovat sledování splnění sekvence podmínek pro jednotlivá gesta – například dlaň se nachází napravo od ramena, dlaň se nachází mezi rameny, dlaň je vlevo od druhého ramena; ve výsledku určí swipe doleva). Pro rozšíření možností interakce by bylo vhodné implementovat detekci gest založenou i na jiném principu, umožňující širší škálu gest či jemnější motoriku. Pokročilé určování gest z hloubkové mapy přesahuje rozsah této práce, proto v případě nedostupnosti knihovny pro přesnější či pokročilejší detekci bude postačovat základní detekce gest. Zajímavým faktem je, že při využívání údajů o poloze jednotlivých kloubů digitální kostry jsou, dle mé zkušenosti, získané informace zatížené menší chybovostí či nejistotou než při použití technik počítačového vidění na dvourozměrný obraz.

Dalším konceptem založeným na poloze kloubu je interakce uživatele s prostředím. Postačuje jednoduché nadefinování geometrie prostředí (znázorněno na Obrázek 18 představujícím, jak by rozvržený prostor vypadal a jak by bylo možné pozice přístrojů zapisovat) k tomu, aby se zjistilo, zda se přímka daná body kloubu ruky, protíná s některým z definovaných objektů, a tím se určilo, jestli uživatel ukazuje na objekt nebo se přímo dotýká určitého přístroje.

Důležitou informací, která by neměla být při zpracování vynechána, je určení ruky vykonávající daný pohyb. Proto jsou i v obecném návrhu systému (Obrázek 16) nastíněny dvě datové cesty pro každou ruku zvlášť. Oddělení gest rukou nalezne uplatnění, pokud bude vytvořený systém podporovat i komplexní příkazy, zadávané čistě pohyby rukou (v případě konkurence jednotlivých vstupů - možnost zadávání stejného příkazu různými způsoby, například rukou ukázat na přístroj a vyslovit „zapnout“ nebo druhou rukou provést swipe nahoru).

Aby nedocházelo k mylným konstrukcím v případě, že by příkazy zadávalo současně více lidí, lze pro další zpřesnění informace k získaným gestům přiřadit uživatele, který je vykonal.



```
<?xml version="1.0" ?>
<objects>
  <object name="television">
    <x0>-1</x0>
    <y0>-1</y0>
    <z0>-1</z0>
    <x1>1</x1>
    <y1>1</y1>
    <z1>1</z1>
  </object>
  <object name="radio">
    <x0>-3</x0>
    <y0>-1</y0>
    <z0>-3</z0>
    <x1>-2</x1>
    <y1>1</y1>
    <z1>-2</z1>
  </object>
</objects>
```

**Obrázek 18:** Ilustrace rozvržení virtuálního prostoru pomocí krychlí reprezentujících zařízení a Kinectem ve středu souřadného systému, vpravo demonstrace zápisu geometrie prostředí pomocí XML.

**Agent zpracování řeči**

Zpracování dalšího významného vstupu v podobě audia lze také pojmout různými způsoby. První možností je použít detekci klíčových slov a tato posílat spolu s jistotou detekce centrálnímu agentu. Tento přístup umožnuje zadávat jednoslovné či kratší slovní příkazy, což by obecně pro ovládání postačovalo.

Další variantou je převod řeči na text, kdy ze vstupu je získána posloupnost znaků či v lepším případě přímo slov. Výhodou by byla možnost větší variability příkazu, či použití příkazu mimo předem definovanou gramatiku. Problém nastává, při zpracování farfield signálu, pro který nejsou knihovny na přímý přepis řeči na text zcela běžné, a chybovost výstupu je vyšší, než když se systém snaží detekovat předem definovaná slova.

Zajímavou informací při zpracování řeči s využitím mikrofonního pole je pozice zdroje zvuku. Takto lze alespoň základně určit mluvčího a případně v dalším zpracování přiřadit pohybová gesta a slovní příkazy stejné osobě.

**Agent určení směru pohledu**

Pro odstranění nutnosti inicializace komunikace jsem zvolil přístup z běžného mezilidského dialogu, kdy určujeme, s kým se řečník dorozumívá pomocí očního kontaktu (směr pohledu určuje, koho se sdělovaná informace týká). Proto navrhovaný systém obsahuje agenta pro určení směru pohledu. Znovu se nabízí více variant. Vývojářská sada pro zpracování hloubkových dat umožnuje dokonce namapování trojrozměrného modelu obličeje na nalezeného uživatele (toto řešení však vyžaduje více výpočetních prostředků a dostatečně kvalitní vstup).

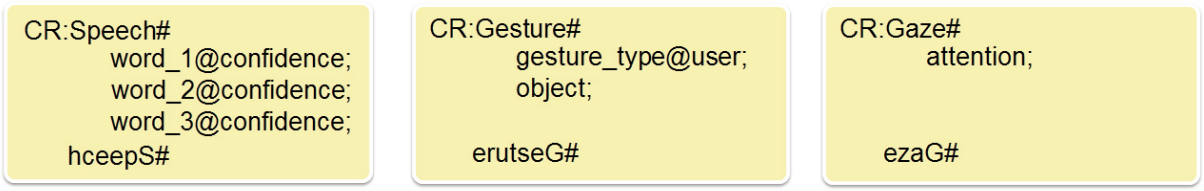
Pro určení, že uživatel chce komunikovat se systémem, by měla stačit detekce očí v RGB obraze. Navíc detekce nemusí běžet neustále, ale s postačující periodou.

**Komunikace mezi agenty**

Komunikaci mezi softwarovými agenty a centrálním agentem (či přímo mezi agenty) lze realizovat několika způsoby. Mezi procesy na jednom počítači lze zasílat zprávy. Takto jsme ale vázaní na to, že agenti se budou fyzicky nacházet na jednom stroji. Pokud chceme snímat větší prostor nebo

výpočet distribuovat mezi více strojů (nabízí se i myšlenka realizovat systém pomocí energeticky úsporných embedded zařízeni), lze využít síťové komunikace, například pomoci protokolu UDP (pro navrhovaný systém není nutné bezpečné doručení zpráv, senzory se zpravidla vyskytují v jedné budově čili v jedné síti, UDP také méně vytěžuje síť).

Díky správné definici komunikačního kanálu lze také docílit jednodušší modifikace výsledného systému. Implementaci agentů lze zcela změnit, ale pokud zprávy zůstanou stejné, tak zbytek systému není potřeba modifikovat.



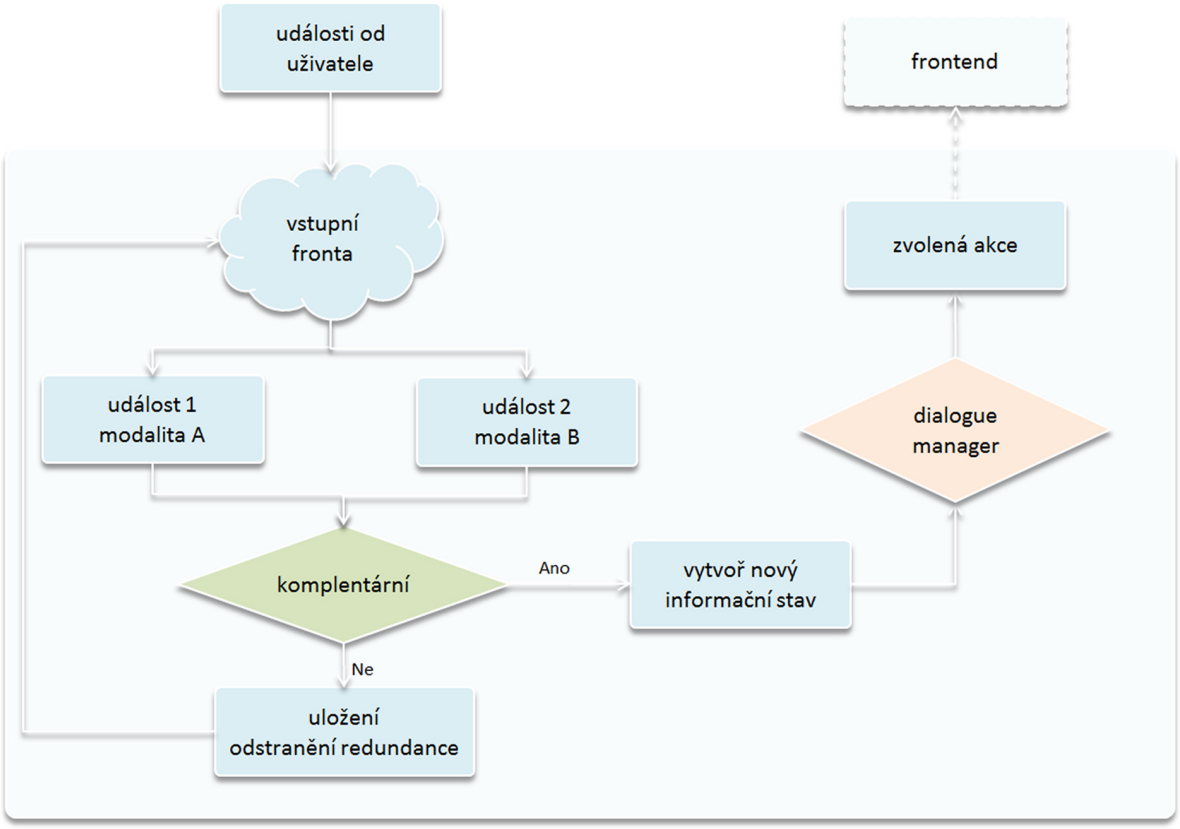

**Obrázek 19:** Formát zasílaní zpráv jednotlivých agentů. Návěští CR určující, že se jedná o zprávu navrhovaného systému, následované identifikaci senzoru, seznamem objektů a ukončením paketu.

Jak ukazuje Obrázek 19, přenášená informace se podle zdrojového agenta liší. Pro odlišení původu zprávy slouží označení (Speech,Gesture,Gaze). Detekovaná slova jsou zasílaná spolu s jistotou, s jakou byla zjištěna, i případnými dalšími interpretacemi vstupu. Gesta (kromě informace, který uživatel gesto vykonal) obsahují identifikaci objektu, kterého se týkají, tedy jestli se vztahovala k některému z objektů uložených v geometrii prostředí.

## 3.4 Centrální agent

Jakmile jsou události popisující akce uživatele dostupné, je potřeba je vhodným způsobem vyhodnotit. Proto se v systému nachází centrální agent určený k fúzi dat (Obrázek 16). Mechanizmů pro integraci událostí existuje celá řada (významné jsou již popsané v předchozí kapitole). Rozhodnout, který se pro účely zde navrhovaného rozhraní hodí nejlépe, není jednoduché. Některé přístupy jsou navrženy pro co nejlepší reflektovaní časové informace týkající se příchozích zpráv, jiné zase mají blíže k přirozenému uvažování člověka a tak mohou teoreticky poskytovat základ pro přirozenější komunikaci mezi člověkem a strojem.

Pro své řešení jsem zvolil hybridní fúzi, která pomocí sady pravidel vybírá z příchozích událostí ty, které spolu souvisejí a následně pomocí dialog managera stanoví koncové rozhodnutí o akci, kterou uživatel chce provést.




**Obrázek 20:** Algoritmus navrhované dialogem řízené fúze.

Jak demonstruje Obrázek 20, návrh se příliš neodlišuje od principů popsaných u hybridní fúze v kapitole 2.7. Vstupní události jsou bez ohledu na to, z jakého zdroje pocházejí, vloženy do fronty (v tomto případě nezáleží na tom, zda jsou události seřazeny podle času příchodu či jinak). Z této se berou události a pomocí pravidel se určuje, zdali obsahují komplementární informace, nebo je potřeba je vrátit do fronty či případně z ní zcela odstranit. Pokud se data doplňují, je vytvořen nový informační stav sdružující dané informace. Ten může být doplňován o další data nebo už může být kompletní a být přijat dialogue managerem. Pokud je v procesu fúze informační stav dialogue managerem přijat, vygeneruje se požadovaná akce a stav managera je případně aktualizován.

Dialog manager obecně je komponenta spravující: interpretaci vstupu, správu kontextu dialogu, určení, jestli informace dostatečně identifikují úlohu; určení, co je očekáváno nebo přiměřené s ohledem na daný kontext, správu průběhu dialogu. Jedná se o širokou škálu úloh a volba, jak je provést, záleží na zaměření celého systému a provedení jednotlivých částí. V případě mnou navrhovaného systému využívajícího hybridní fúze, kde dialogue manager nemá stěžejní roli v procesu fúze, nemusí být dialogue manager realizován jako komplexní jednotka. Měl by postačovat konečný automat aktualizující stav dialogu a reflektující kontext, v jakém se dialog nachází.

Pro rozhodování o komplementárnosti je potřeba ještě definovat dialogová pravidla (pravidla definující, jaké akce uživatel může provádět, jaká data jsou k nim potřeba). Formát, v jakém daná pravidla specifikovat, se často razantně liší mezi různými systémy. Mohou být ve formě XML, matice, datové struktury či jiné vhodné reprezentace.

```
<rule name="volume">
  <init>
    <event type="KEYWORD">VOLUME</event>
  </init>
  <expectations>
    <event type="GESTURE">SWIPEUP</event>
  </expectations>
  <action>VOLUMEUP</action>
  <enabled>TRUE</enabled>
</rule >
```

```
<rule name="light">
  <init>
    <event type="KEYWORD">LIGHT</event>
  </init>
  <expectations>
    <event type="KEYWORD">DISABLE</event>
  </expectations>
  <action>DISABLELIGHT</action>
  <enabled>FALSE</enabled>
</rule >
```

```
<rule name="gestohor">
  <init>
    <event type="GESTURE">SWIPEDOWN</event>
  </init>
  <expectations>
  </expectations>
  <action>MOVEDOWN</action>
  <enabled>TRUE</enabled>
</rule >
```

```
<rule name="volume">
  <init>
    <event type="KEYWORD">VOLUME</event>
  </init>
  <expectations>
    <event type="GESTURE">SWIPEDOWN</event>
  </expectations>
  <action>VOLUMEDOWN</action>
  <enabled>TRUE</enabled>
</rule >
```

```
<rule name="light">
  <init>
    <event type="POINTINGAT">LAMP</event>
  </init>
  <expectations>
    <event type="GESTURE">SWIPEUP</event>
  </expectations>
  <action>ENABLELIGHT</action>
  <enabled>TRUE</enabled>
</rule >
```

```
<rule name="gestohor">
  <init>
    <event type="GESTURE">SWIPEUP</event>
  </init>
  <expectations>
  </expectations>
  <action>MOVEUP</action>
  <enabled>TRUE</enabled>
</rule >
```

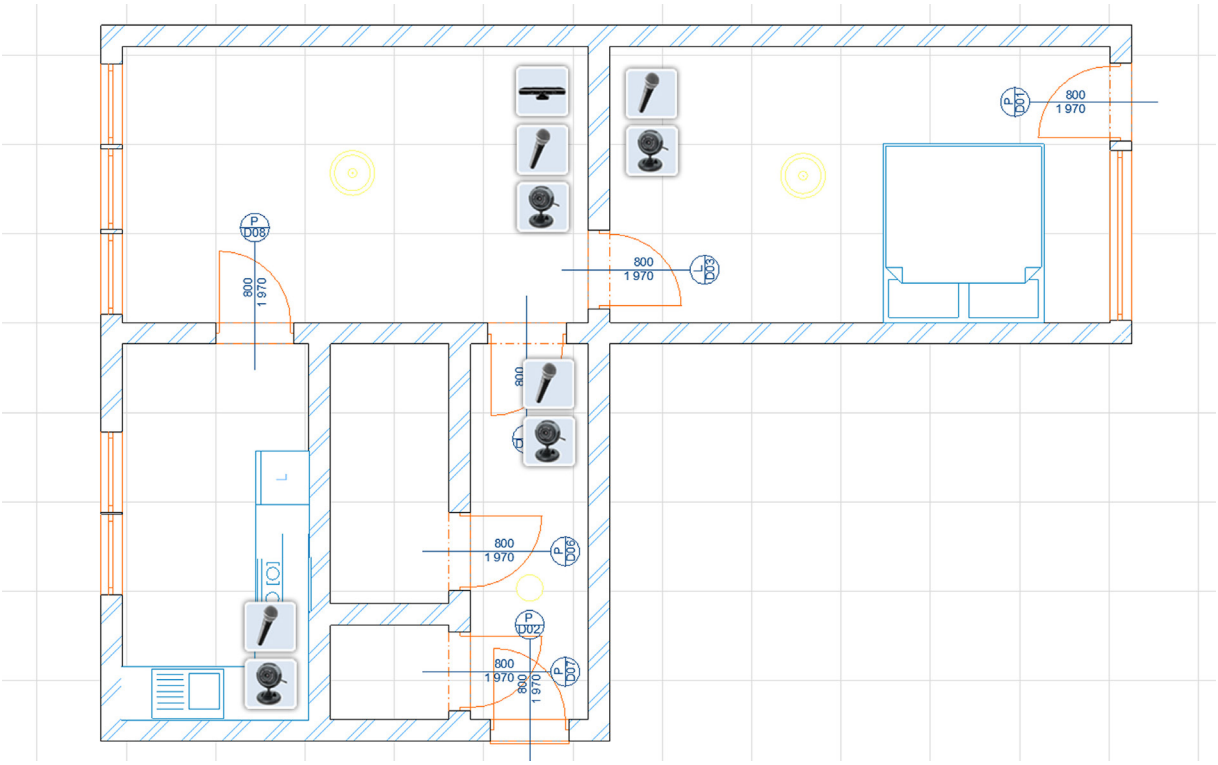
**Obrázek 21:** Základní pravidla určující akce, které by měl systém umožnovat.

Navrhl jsem základní sadu pravidel (Obrázek 21) umožňujících běžné operace s multimediálním centrem a prostředím. Byla však vytvořena jen po krátkých experimentech. Pro získání lepší sady příkazů by bylo vhodné věnovat delší čas zkoumání interakce uživatelů s tímto systémem.

Pravidla jsou ve formátu XML, ve kterém se po identifikaci pravidla nachází spouštěcí podmínka (obsah elementu init), další akce požadované pro provedení (element expectations) a výsledná operace (element action), která se má provést. Uživatelské akce definuje jejich typ (gesto, slovo, pohled) a mohou obsahovat upřesňující informace (jakými jsou směr pohybu či vyslovené klíčové slovo). Pravidla obsahují i doplňující informaci o tom, jestli jsou povoleny (to umožnuje dynamicky měnit sadu akcí, které může uživatel provést).

Kromě integrace jednotlivých typů vstupů centrální agent umožnuje i zpracování více vstupů stejného druhu. Systém teoreticky nemusí být vázán na snímání jedné místnosti, stejných agentů může být spuštěno více (například agent pro zpracování řeči může běžet kromě obývacího pokoje i v přilehlé kuchyni a dovolit tak vzdálené ovládání hlasitosti či přepínání kanálů). Kdyby byly senzory rozmístěny ve více místnostech, realizovala by se myšlenka inteligentního domu ovládaného hlasem a gesty. Realizace domu s funkcemi (jako ovládání teploty, světel a podobně) ovládanými centrálním systémem jsou běžně dostupné a často používají dálkové ovládání pomocí webové stránky, takže napojení na takový systém by nemusel představovat větší komplikace. Obrázek 22 ilustruje, jak by mohly být senzory v domě umístěny, šlo by tak ovládat funkce domu či multimediální centrum z více míst bez nutnosti dálkového ovladače.

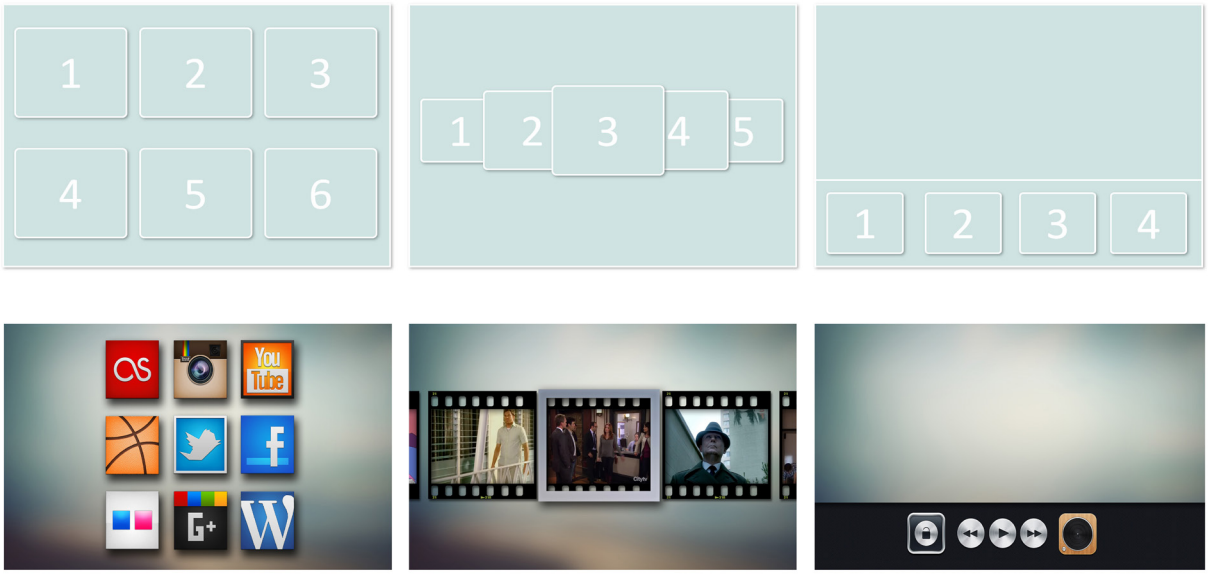
V tomto případě lze uvažovat o tom, které úlohy je vhodné tímto systémem provádět. Například ovládání světla v pokoji je pro většinu uživatelů pravděpodobně jednodušší pomocí klasického vypínače. Dostaneme-li se ale na úroveň celého bytu, kdy jediným příkazem by šlo vypnout světlo v jiné místnosti nebo i veškerá světla najednou, dostáváme se k zajímavým možnostem, které by už uživatele možná ocenili. Umožnit vzdáleně ovládat specifické spotřebiče či zařízení je jedna možnost, další zajímavou vlastností je oproštění se od fyzických ovladačů – tedy možnost zadávat příkazy, ikdyž uživatel nemá ruce k dispozici (například při práci v kuchyni, čtení knížky a podobně). Další situací, kde teoreticky může ovládání gesty a slovními příkazy přinést zlepšení oproti klasickému dálkovému ovladači, jsou krátké úkony, například: pozastavení filmu, ztlumení hlasitosti, vypnutí televizoru, zapnutí nahrávání. V těchto a dalších případech čas potřebný pro zadání příkazu gestem či slovem je přibližně stejný jako v případě dálkového ovládání, a když ovladač nemá uživatel přímo k dispozici, může být navrhovaný systém i rychlejší a pohodlnější. Naopak při výběru specifické položky z obsáhlého seznamu se může jevit klasický ovladač jako pohodlnější řešení.




**Obrázek 22:** Ukázka možného rozmístění senzorů v inteligentním domě.

## 3.5 Výstup systému

Jakmile jsou k dispozici informace o tom, jakou akci chce uživatel provést, je vhodné výsledek vizualizovat. Takto navržený systém by měl sloužit pro ovládání multimediálního centra. Za tímto účelem lze vytvořit grafickou aplikaci simulující přehrávač s několika animovanými obrazovkami (znázorňujícími nabídku filmů, aktuálně přehrávaný film, ovládací menu nebo prohlížení obrázků). Pro tuto úlohu lze také použít dostupný software pro přehrávání multimédií, pokud je umožněno jeho ovládání jiným programem (ideálně pomocí síťového protokolu, aby byla dodržena myšlenka distribuovaného systému, s možností umístění jednotlivých části na fyzicky rozdílné stroje).




**Obrázek 23:** Návrh obrazovek aplikace demonstrující multimodální rozhraní. Z leva – výběr akce gestem ukázání, prohlížení položek, menu pro ovládání média. Dole grafický koncept navrhovaných obrazovek.

Obrázek 23 popisuje možné rozvržení obrazovek multimediálního centra vhodného pro multimodální ovládání. Jedná se o menu s velkými položkami pro snadnější volbu při použití gesta ukázání (mělo by ovšem být umožněno provádět výběr i dalším způsobem, například vyslovením čísla položky). Další obrazovkou je nabídka položek, která se posouvá horizontálně, například gestem swipe či gestem chycení a dále pohybem ruky vpravo či vlevo. Poslední je nabídka zobrazující se při přehrávání v případě, kdy uživatel chce provést běžné akce (vyvolání menu by mohlo být spouštěno gestem swipe směrem nahoru a výběr opět ukázáním či vyslovením příkazu).

Otázkou je, jak obecně nabídky vyvolávat. Pro zobrazení nabídky lze vyčlenit jedno z gest, například mávání nebo swipe směrem nahoru, takto ale uživatel nebude mít k dispozici informaci o tom, jakou nabídku gestem vyvolá (během přehrávání filmu si uživatel bude muset pamatovat, jakou nabídku jakým gestem zobrazí, například swipe nahoru pro ovládání a swipe dolů pro hlavní nabídku, pokud se ale zmýlí, bude se muset vracet a proces opakovat což pro komfort ovládání nebude příliš dobré). Jako lepší řešení se mi zdá reagovat na pohyb uživatele a naznačit kde se jaká nabídka vyvolává. Přesněji začne-li uživatel pohybovat rukou před sebou, objeví po stranách obrazovky malé ikony s popisky, o jakou nabídku se jedná, takto by mělo být jasné, jakou nabídku jakým gestem lze vyvolat (to by mělo usnadnit ovládání prostředí složeného z více obrazovek, kde každá obsahuje různé nabídky, což je obecně případ multimediálního centra).

# 4 Realizace

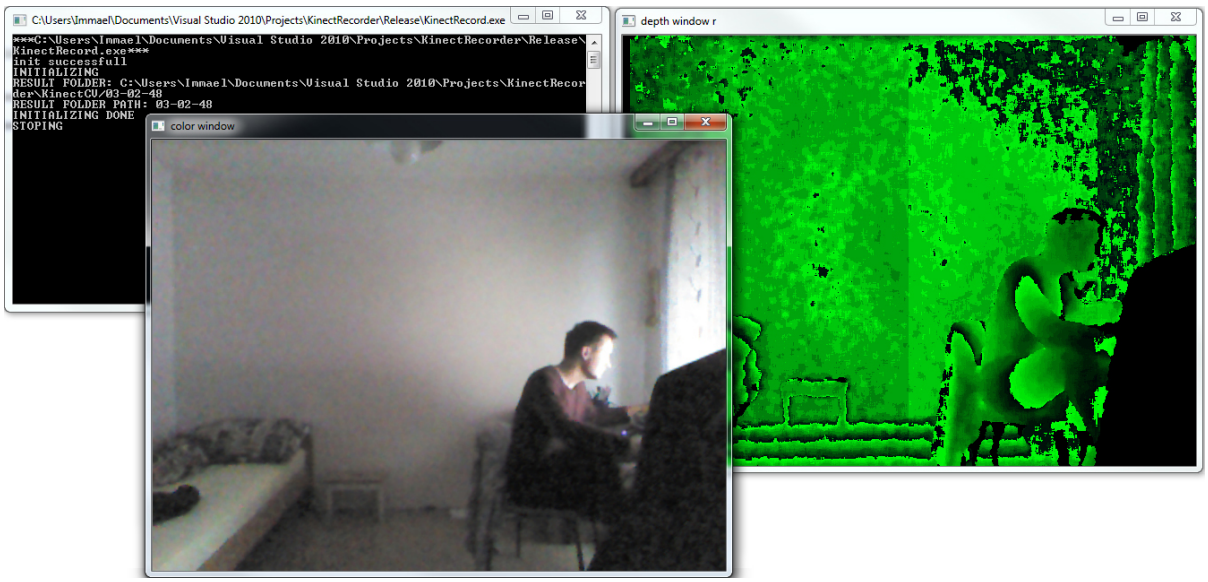
Systém navržený v minulé kapitole jsem realizoval nejprve pomocí knihoven třetích stran, ale po nedostačujících výsledcích jsem funkce většiny bloků realizoval sám. Proto zde budou zmíněny použité knihovny, ale hlavně realizace použitá při koncovém testování. Kromě samotných bloků systému bylo potřeba implementovat i pomocné aplikace pro usnadnění vývoje. Pozornost je zde věnována i provedení testů na uživatelích.

## 4.1 Pomocné aplikace

Pro účely tvorby systému bylo potřeba vytvořit aplikaci pro nahrávání dat a jejich anotaci (sloužících pro další experimenty a určování gest). Jedná se o jednoduchou aplikaci ovládanou pomocí klávesnice ukládající data spolu s XML souborem popisujícím nahraná data (pro lepší představu viz Obrázek 24).

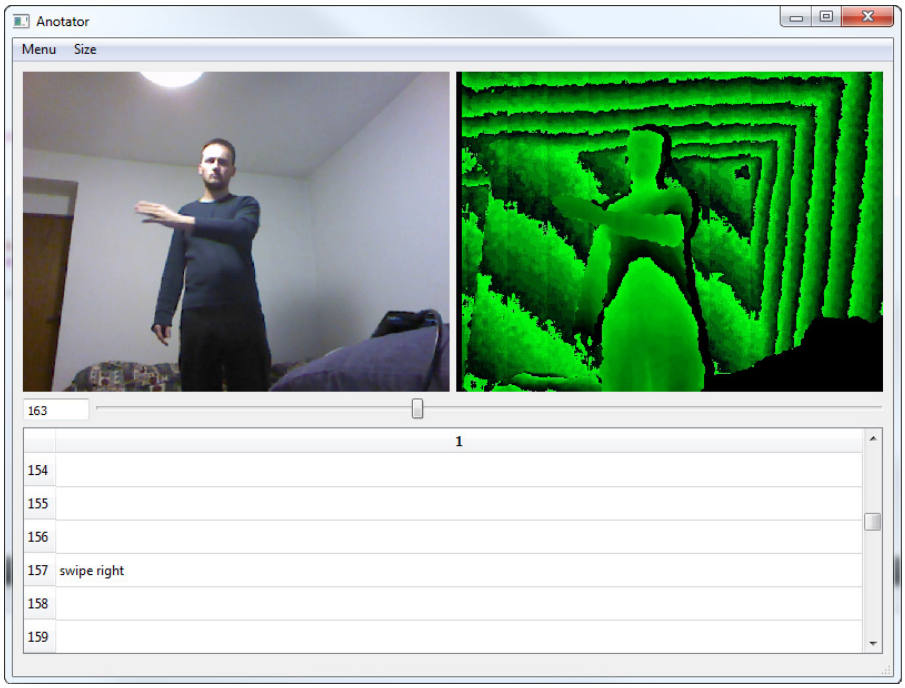
Ukázka XML popisujícího pořízená data:

```
<config version="1">
  <group name="color">
    <element>color.avi</element>
  </group>
  <group name="depth">
    <element>depth.avi</element>
  </group>
  <group name="skeleton">
    <element>skeleton.txt</element>
  </group>
  <group name="audio">
    <element>audio.wav</element>
  </group>
  <group name="actions">
    <element name="frame number">action performed</element>
  </group>
</config>
```




**Obrázek 24:** Aplikace pro nahrávání RGB, hloubkových, skeleton a audio dat

Samotné nahrávání se provádí do nekomprimovaných AVI souborů (pro případné další použití kolegy v rámci trénování detektorů gest) s tím, že data hloubkové mapy jsou kvůli 16 bitovému rozsahu rozložena do dvou kanálů videa (osm nejvýznamnějších bitů do červené složky videa, osm less significant bitů do zelené složky videa, modrá zůstává nulová). Soubory vytvořené v rámci jednoho nahrávání jsou pro zpřehlednění uloženy do oddělené složky, pojmenované podle času spuštění pořízení záznamu. Pro usnadnění vývoje jsem anotační aplikaci vyvíjel zvlášť za pomocí frameworku Qt. Aplikace (demonstrována na Obrázek 25) umožňuje snadný pohyb načtenou sadou dat, označování událostí pro každý snímek a pro zlepšení komfortu anotace i volbu velikosti zobrazovaných náhledů.




**Obrázek 25:** Jednoduchý anotátor nahraných dat.

## 4.2 Softwaroví agenti

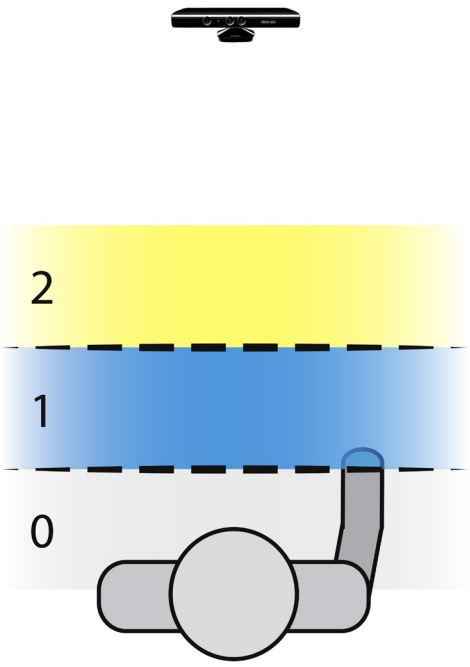
Každý z bloků navrhnutého systému jsem realizoval jako samostatnou aplikaci – samočinného agenta. Pro urychlení vývoje, respektive usnadnění tvorby a úprav prototypů, jsou agenti pro zpracování vstupů a zpětnou vazbu implementováni v jazyce C# s využitím Microsoft Kinect SDK. Oproti tomu hlavně rozebíranou část této práce, tedy fúzi, jsem pro dosažení lepší univerzálnosti implementoval v jazyce C++.

**Agent zpracování pohybu**

Gesta rukou jsem v počáteční fázi vývoje realizoval pomocí knihovny FizBin Kinect Gestures. Ta umožňuje snadnou definici a sledování základních gest, například swipe, zoom či mávání. Gesta jsou definována jako posloupnost konfigurací kloubů digitální kostry. V tom spočívalo i základní omezení řešení, takto definována gesta lze provést pouze jedním způsobem (například gesto swipe pravou rukou bylo nutné provést tak, aby pozice ruky zjištěna senzorem Kinect byla nejprve napravo od pravého ramene, následně mezi rameny a nakonec nalevo od levého ramene), provádění tak bylo dost těžkopádné a často gesto nebylo ani detekováno.

Z těchto důvodu jsem přistoupil k vlastní implementaci detekce gest. Ta vychází z konceptu rozdělení interakce pomocí gest do několika zón, tedy přesněji tří (v jiné realizaci by počet mohl byt jiný). Každá zóna (pro lepší představu viz Obrázek 26) je určená vzdálenosti ruky od těla a reaguje na pohyb rukou odlišně. V zóně 0 nedochází k žádnému vyhodnocení pohybů, pracovně je nazvána jako klidová zóna, tedy prostor pro přirozenou gestikulaci člověka. Pokud je ruka dostatečně daleko, nachází se v zóně 1. Zde systém ještě neprovádí žádné akce, pouze poskytuje uživateli zpětnou vazbu, že sleduje jeho pohyb (toto chování lze přizpůsobit pro jiné aplikace například tak, že v této zóně by se provádělo zvýraznění ovládacích prvků pod rukou uživatele, ale také by se pořád neprováděla žádná akce). Poslední zóna (nacházející se dostatečně daleko od těla, aby nedocházelo příliš často k mylnému provedení gest, ale zároveň dostatečně blízko pro lepší komfort ovládání) slouží k zadávání jednotlivých gest a tedy k ovládání systému. Přechody mezi zónami jsou ošetřeny hysterezí, aby vlivem šumu nedocházelo k náhlému opakovanému přepínání.

Samotné určení, v jakých vzdálenostech od uživatele se jednotlivé zóny nacházejí, vychází z délky ruky uživatele (ta je vypočtena ze vzdálenosti bodu ruky k lokti a vzdálenosti lokte k rameni). Každá zóna má definovanou vzdálenost v procentech délky ruky. Díky tomu je řešení použitelné pro více osob, tedy není nutné nastavovat vzdálenost jednotlivých zón při každé změně uživatele.



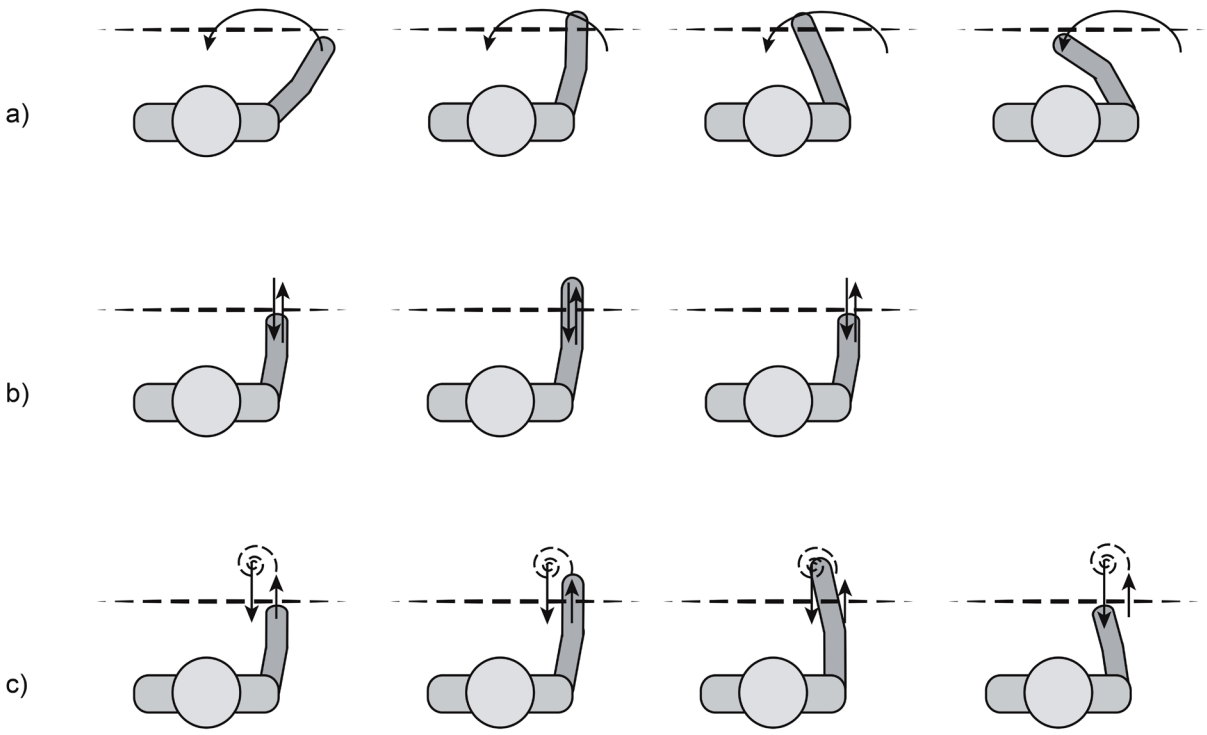

**Obrázek 26:** Ilustrace rozdělení prostoru do zón. Hranice mezi zónami jsou plochy kolmé na senzor.

Gesta jsou rozdělena do tří typů. Každé s odlišným způsobem zadávání, ale se společným základním konceptem, aby jejich zadávání uživateli způsobovalo menší problémy. Gesta jsou (názorná ukázka Obrázek 27):

a) Swipe – swipe vertikálně a horizontálně (doleva, doprava, nahoru dolů) uživatel provede tak, že rukou vstoupí do druhé zóny a bude dále rukou posouvat požadovaným směrem. Swipe je detekován jako pseudo-online gesto, tedy je-li pohyb dostatečně dlouhý, je postupně označen jako několik gest. Speciálním případem je swipe šikmo pod úhlem 45º, ten je označen jako „zpět“.

b) Klik – je detekován, jestli uživatel rukou vstoupí do druhé zóny a následně s ní vystoupí, aniž by rukou značně pohnul horizontálně či vertikálně.

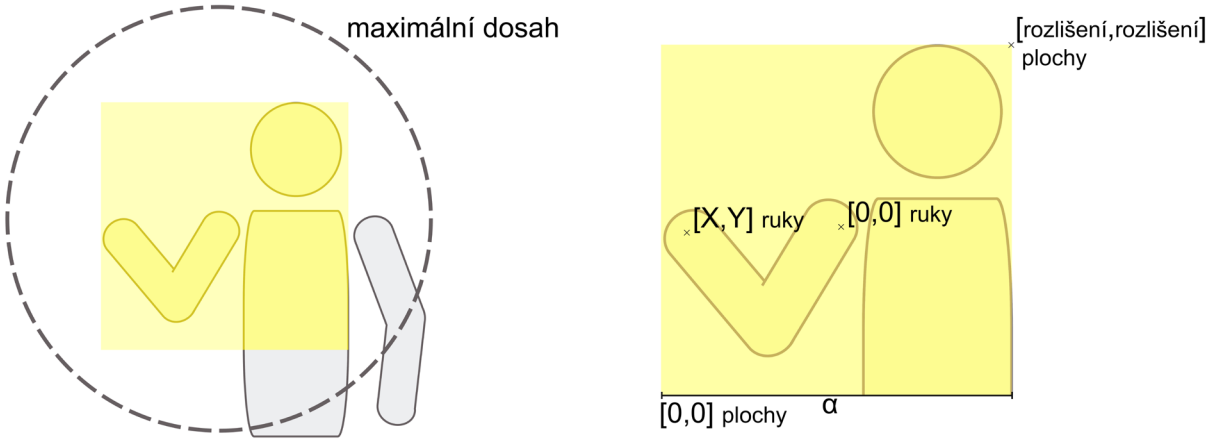
c) Specifický výběr – spíše než o gesto jedná se o absolutní pozicování kurzoru. Vstoupí-li uživatel do druhé zóny a ponechá ruku chvíli bez pohybu, vyhodnotí se jako gesto specifického výběru sloužící pro ovládání kurzoru. Ovládání je ukončeno vystoupením z druhé zóny.




**Obrázek 27:** Snímána gesta a) swipe vpravo, ale i vlevo, nahoru, dolu b) klik c) specifický výběr

Gesto specifického výběru se od ostatních liší nejen samotným zadáváním, ale i způsobem vyhodnocení. Gesta swipe a klik jsou ve formě jednorázové události, která je předána centrálnímu agentovi k vyhodnocení. Ovládání kurzoru oproti tomu je složeno z událostí zahájení či ukončení ovládání a aktualizace pozice mezi tím. Kromě rozdělení ovládání kurzoru do několika událostí jsem musel vhodným způsobem informaci o poloze ruky transformovat na pozici kurzoru na obrazovce. K tomu jsem opět použil informaci o délce ruky, pomocí ní stanovil maximální dosah uživatele a v něm vyčlenil prostor odpovídající prostoru obrazovky, přesněji čtverec se středem v rameni a velikostí menší než poloměr kruhu daného délkou ruky. Když je tedy následovně převedena souřadnice ruky do souřadného systému se středem v rameni, lze jednoduše převést polohu ruky, přesněji X a Y, na body v rámci obrazovky (pro lepší představu o prostoru pro ovládání kurzoru viz Obrázek 28). Postup lze popsat jako:

1. Souřadnice jsou v rozmezí $-maximální\ dosah$ až $+maximální\ dosah$
2. Omezit hodnoty $X$ a $Y$ polohy ruky, aby nepřesahovaly hranice vymezené plochy (od $-\alpha/2$ do $+\alpha/2$ kde $\alpha$ je délka strany plochy).
3. Posunout hodnoty polohy, aby byly v rozmezí 0 až $\alpha$ přičtením $\alpha/2$.
4. Transformace na body obrazovky ($\frac{X}{\alpha} * rozlišení$ a $\frac{Y}{\alpha} * rozlišení$).




**Obrázek 28:** Ovládání kurzoru pohybem ruky. Přerušovaná čára - teoreticky dosah ruky. Žlutě - namapovaný prostor obrazovky.

Další informací odvozenou z prostorových dat je informace, na který z definovaných objektů uživatel ukazuje. Objekty jsou pro snadnější přizpůsobení definovány v souboru XML jako trojrozměrná pozice a údaje o rozměrech. Pro zjištění, na který z objektů uživatel ukazuje, se provádí výpočet průsečíku přímky (definované body dlaně a prstů digitální kostry) a jednotlivých objektů.

Ukázka XML popisujícího rozložení objektů prostoru (u objektu se nejprve definuje pozice, následovně rozměry, první záznam obsahuje informaci pro úvodní nastavení úhlu Kinectu):

```
<?xml version="1.0" ?>
<objects>
	<kinect angle="0"></kinect>
	<object name="television">
		<x0>0</x0>
		<y0>30</y0>
		<z0>60</z0>
		<x1>31</x1>
		<y1>31</y1>
		<z1>31</z1>
	</object>
	<object name="computer">
		<x0>60</x0>
		<y0>0</y0>
		<z0>0</z0>
		<x1>31</x1>
		<y1>31</y1>
		<z1>31</z1>
	</object>
	<object name="lamp">
		<x0>-60</x0>
		<y0>30</y0>
		<z0>60</z0>
		<x1>31</x1>
		<y1>31</y1>
		<z1>31</z1>
	</object>
	<object name="lamp ceiling">
		<x0>0</x0>
		<y0>60</y0>
		<z0>-5</z0>
		<x1>31</x1>
		<y1>31</y1>
		<z1>31</z1>
	</object>
</objects>
```

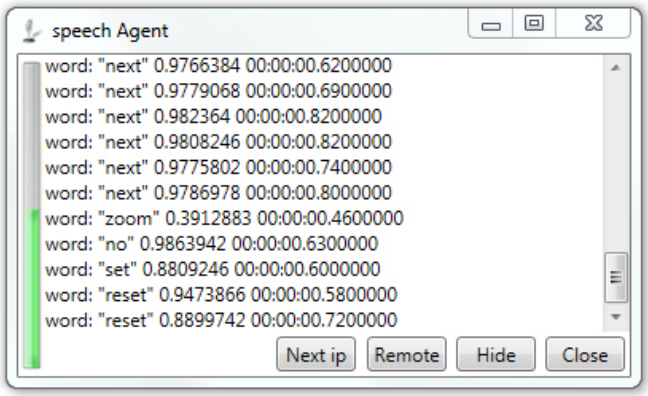
**Agent zpracování řeči**

Pro analýzu řeči jsem po konzultaci zvolil Microsoft Speech Api (zkráceně SAPI). Ta umožňuje relativně snadnou detekci klíčových slov z farfield signálu a v testech prokázala dobré výsledky. Klíčová slova pro detekci lze nadefinovat, v tomto případě pomocí externího souboru (formát a ukázka detekovatelných slov na Obrázek 29), a při spuštění načíst. Pro detekovaná slova jsou dostupné i další informace, například alternativní interpretace vstupu, pravděpodobnost, s jakou bylo slovo určeno či délka slova v sekundách. Nevýhodou je, že nepodporuje český jazyk, ovládání musí být založeno na anglických klíčových slovech.

```
-- [klicove slovo] | [text pro zobrazeni v agentovi] | [popis]
cristi|word: "cristi"|klicove slovo pro ziskani pozornosti kdyz se uzivatel nediva na "rozhrani"
cancel|word: "cancel"|slovo pro zruseni
forget it|word: "forget it"|slovo pro zruseni
volume|word: "volume"|hlasitost
temperature|word: "temperature"|pro pripadne nastavovani topeni
what is it|word: "what is it"|pro pripadnou identifikaci objektu
what is|word: "what is"|pro pripadnou identifikaci objektu
that is|word: "that is"
identify|word: "identify"|pro pripadnou identifikaci objektu
turn on|word: "turn on"|prikaz pro zapnuti
turn off|word: "turn off"
enable|word: "enable"|prikaz pro zapnuti
disable|word: "disable"
reset|word: "reset"
light|word: "light"
brightness|word: "brightness"
yes|word: "yes"
no|word: "no"
shut down|word: "shut down"
next|word: "next"
zoom|word: "zoom"
set|word: "set"
up|word: "up"
down|word: "down"
left|word: "left"
right|word: "right"
one|word: "one"
two|word: "two"
three|word: "three"
five|word: "five"
six|word: "six"
seven|word: "seven"
```

**Obrázek 29:** Formát zápisu klíčových slov pro detekci.

Grafické rozhraní agenta slouží hlavně pro účely vývoje, tedy ověření správnosti detekce a funkčnosti senzoru, nastavení adresy centrálního agenta pro zasílání detekovaných slov a případné skrytí agenta. Při nasazení v praxi by bylo potřeba aplikaci nastavit pro skrývání při spuštění či vytvořit čistě konzolového agenta.

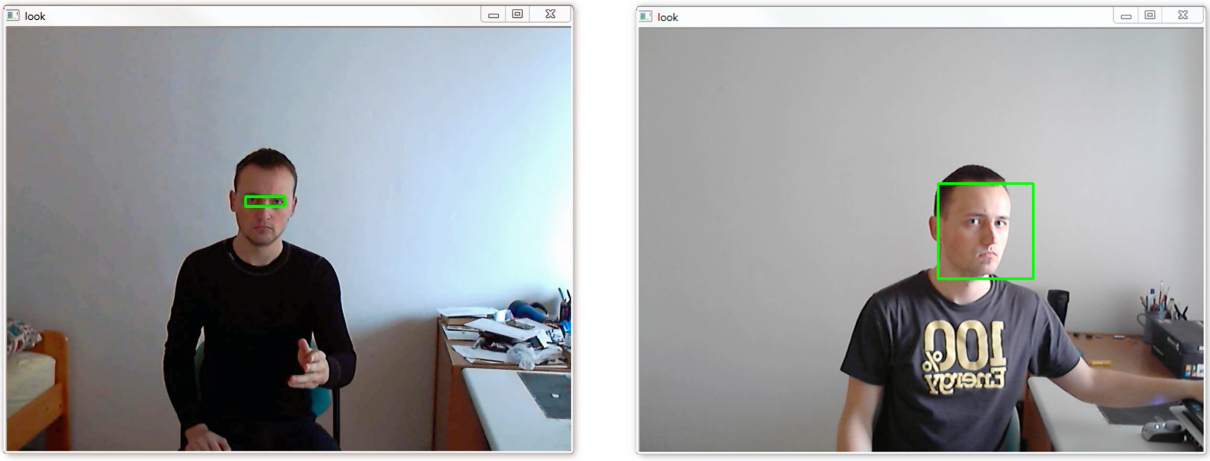



**Obrázek 30:** Agent zpracování řeči. Grafické rozhraní indikuje úroveň vstupu, detekovaná slova.

### Agent určení směru pohledu

Detekci směru pohledu jsem realizoval pomocí knihovny OpenCV. Nedetekuji přímo směr pohledu uživatele (to by umožnil například Kinect SDK, ale výpočetní nároky pro to potřebné jsou poměrně vysoké a požadavkem pro detekci je i menší vzdálenost uživatele od senzoru). Namísto toho pomocí detekce očí určuji, zda je uživatel v očním kontaktu se systémem (jestli se jej vyslovené příkazy a provedená gesta týkají). Zmenšila se tak i výpočetní náročnost běhu celého systému. Ten vytížil systém s Core i7 2630qm 2ghz z přibližně 15 až 20 procent. Náhled na výstup poskytuje Obrázek 31, grafická podoba však není důležitá, agent je určen pro běh na pozadí.

Oproti určení přesného směru pohledu toto řešení má jednu znatelnou nevýhodu a to absenci možnosti nastavení tolerance určení, jestli se uživatel dívá či nedívá na systém (nelze nastavit, jaké natočení hlavy ještě znamená, že se uživatel dívá na systém – detekce objektu buďto nalezne nebo nenalezne oči, při větším natočení hlavy mimo obraz detekce selhává, tento fakt nelze jednoduše ovlivnit). Toto omezení lze nepřímo obejít. Pro dosažení větší tolerance na natočení hlavy, lze namísto očí detekovat v obraze celý obličej. Při detekci obličeje je uživatel nalezen i při větším natočení hlavy než při detekcí očí, takže se uživatel nemusí dívat přímo na systém (otázkou je, jaký přístup uživatel požaduje, jedná se o poměrně subjektivní záležitost).




**Obrázek 31:** Detekce očí pro určení jestli je uživatel v očním kontaktu se systémem. Vlevo - detekce očí s menší tolerancí na natočení hlavy, Vpravo – detekce obličeje umožňující větší natočení bez ztráty kontaktu.

### Centrální agent

Centrálního agenta jsem implementoval podle návrhu z předchozí kapitoly. Základem jsou pravidla načtená z XML souboru. Podle nich se rozhoduje, jakou operaci s příchozí událostí provést (tedy jestli spolu s jinou události ve frontě tvoří celek a je potřeba pro ně vytvořit novou strukturu obsahující jejich údaje a jakou akci tvoří) nebo jestli ji zařadit do fronty příchozích událostí. Události ve frontě mají stanovenou dobu platnosti, po jejím uplynutí jsou z ní odstraněny.

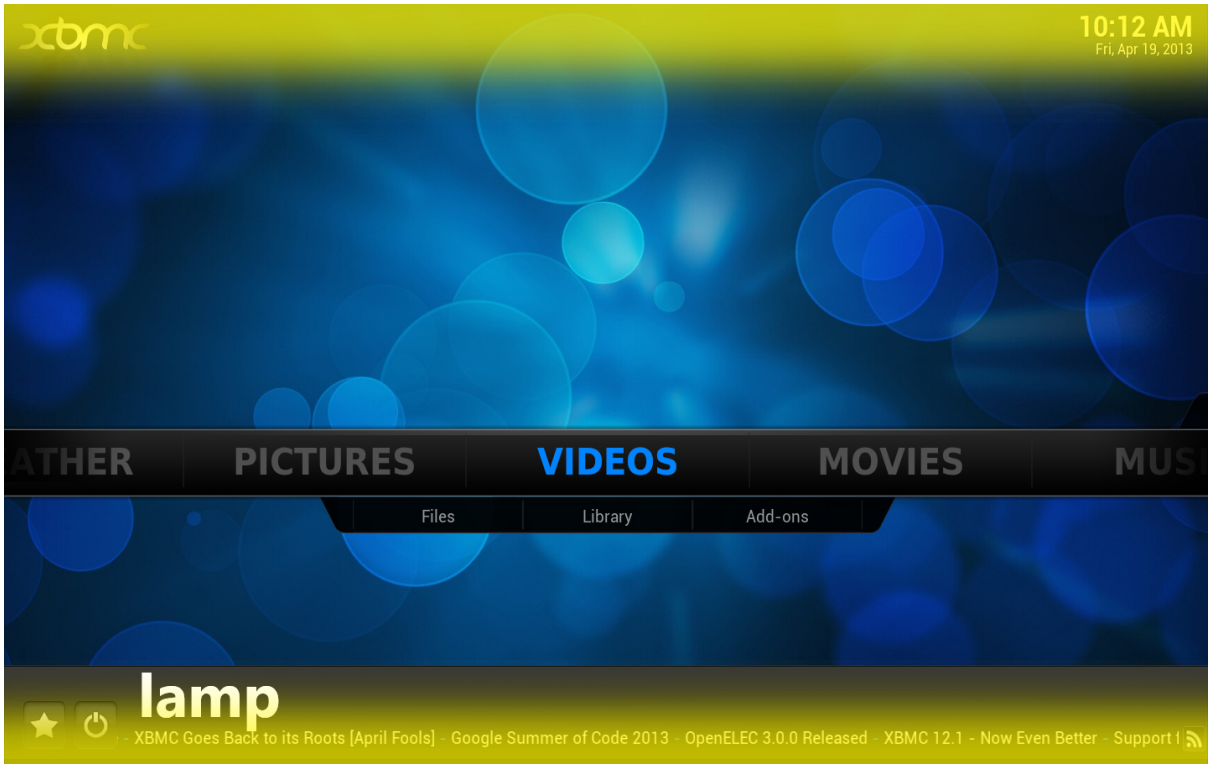
Samotné vstupní události jsou reprezentovány datovou strukturou obsahující informaci o tom, jakého je typu, parametru, který uživatel ji vykonal a v jakém čase. Pokud dvě události spolu tvoří celek, je pro ně vytvořená datová struktura kopírující strukturu unifikačního pravidla, podle kterého byly shledány za komplementární (oproti unifikačnímu pravidlu načtenému z XML nově vytvořená struktura má některé položky nevyplněné, obsahuje jen ty, které obsahovaly události, pomocí kterých byla vytvořena). Takto vytvořený nový informační stav je uložen, dokud není úplný (tedy dokud neobsahuje všechna potřebná data), nebo se stane neaktuální (po uplynutí časového intervalu od vytvoření).

Po zkompletování datové struktury akce je vyhodnoceno, jestli se má provést, jestli se má provést jinak než bylo definováno pravidlem, nebo zcela zahodit (je tedy posouzena dialog managerem – jednotka odpovědná za stav a průběh komunikace). Dialog manager jsem realizoval jako stavový automat, který podle provedené akce přechází mezi stavy. Oproti klasickému stavovému automatu se aktualizuje nejen podle prováděné akce, ale i podle času (nevyskytne-li se po jistu dobu žádná akce, vrátí se do původního stavu). Ze stavové informace odvozuje, jak akci provést, je-li například ve stavu „hlasitost“, tedy po provedení změny hlasitosti jsou gesta swipe nahoru a dolů interpretovány jako další úprava hlasitosti a ne jako navigace v nabídce. Oproti tomu může při vykonání posuvu určitým směrem zahazovat pohyb směrem opačným, aby zamezil chybám při navigaci využívajícím jiný princip vyhodnocení gest (gesta vycházející z konfigurace kostry byly často zatížené tím, že při opakovaném vykonání swipe jistím směrem, se při návratu ruky detekoval i swipe směrem opačným).

Pokud je akce přijata (datová struktura akce obsahuje všechny potřebné informace a dialog manager ji přijal), centrální agent ji provede. Funkce odpovídající jednotlivým akcím jsou definované přímo v kódu, jedná se o zasílání zpráv multimediálnímu centru, agentovi pro zpětnou vazbu a aplikaci reprezentující další přístroje v okolí uživatele.

**Zpětná vazba**

Kvůli použití frontendové aplikace, která neumožnuje reflektovat informace o vzdálenosti ruky od těla, což je princip fungování implementované detekce gest, chybí důležitá zpětná vazba. Proto jsem vytvořil jednoduchou aplikaci, která se zobrazuje jako průhledné okno přes vybraný monitor a pomocí změny průhlednosti barevných pruhů nahoře a dole obrazovky ukazuje uživateli, jak daleko v dané zóně se nachází (pro lepší představu viz Obrázek 32). V dolní části také zobrazuje, na který předmět v prostoru uživatel případně ukazuje (levý dolní roh) a jaké klíčové slovo vyslovil (pravý dolní roh).

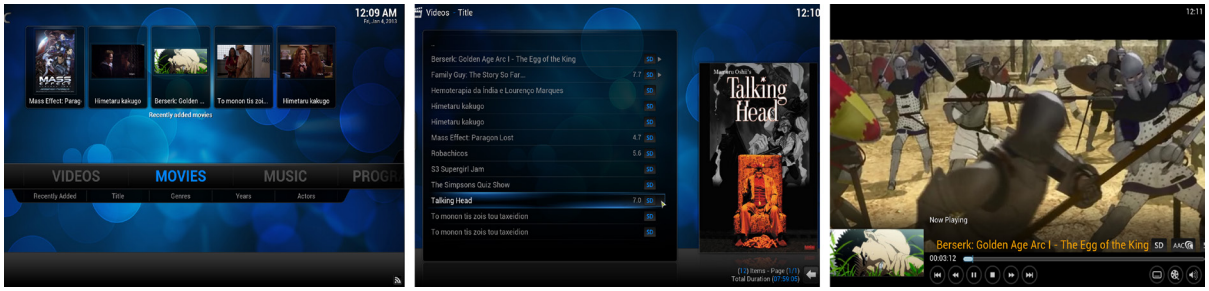



**Obrázek 32:** Snímek zpětné vazby s multimediálním centrem XBMC na pozadí. Žluté pruhy změnou průhlednosti poskytují informaci o vzdálenosti ruky.

**Multimediální centrum**

Aplikaci XBMC jsem zvolil jako vizuální výstup (tedy pro obstarání prohlížení souborů, přehrávání videa a podobně). Jedná se o opensource přehrávač s distribucemi dostupnými pro všechny platformy a širokou škálou rozšíření typu internetová rádia, přehrávání videí youtube. XBMC sice umožnuje běžné operace multimediálního centra, ale nemusí být zcela vhodnou aplikací pro multimodální rozhraní (ovládání je podobné dálkovému ovladači). Pokud je však tato aplikace dobře nastavená a provede se import medií, lze ji pro demonstrační účely dobře použít (pouze komfort ovládání ve srovnání s klasickým dálkovým ovládačem nebude příliš větší, ne-li dokonce menší). Jak je vidět na Obrázek 33, jedná se víceméně o klasické multimediální centrum s širokými možnostmi rozšíření a personalizace.

Při defaultním nastavení jsou nabídky ve formě seznamu, tedy pro výběr pomocí pozicování kurzoru, ne příliš vhodné. Nabídky pro výběr souborů lze ale nastavit na dlaždicový vzhled, který je pro ovládání gesty vhodnější. Horší je to v případě ovládacích prvků přehrávání, ty nejsou příliš velké a v základní verzi nenabízí žádné alternativní zobrazení (v tomto případě, pokud bychom uvažovali o nasazení v této podobě, nezbývá než přehrávání videa ovládat pouze gesty swipe).




**Obrázek 33:** Snímky z aplikace XBMC

**Komunikace mezi agenty**

Jak bylo navrženo, agenti spolu komunikují pomocí protokolu UDP a definovaných zpráv. Programově je komunikace realizovaná pomocí soketů. Pomocí UDP je také ovládána frontendová aplikace XBMC zajišťující funkčnost multimediálního centra.

Centrální agent naslouchá na portu 27015 (lze změnit, pokud by došlo ke konfliktu s jinou aplikací) a po interpretování příchozích zpráv získané rozhodnutí ve formě akcí předává aplikaci XBMC, ta naslouchá na portu číslo 9777.

Komunikace pomocí protokolu UDP se ověřila i v praxi kdy byl systém rozdělen mezi více zařízení propojených i pomocí bezdrátové sítě.

Zajímavou vlastností agentů, kterou sdílí více agentů (přesněji agenta pro detekci směru pohledu a agenta detekce gest) je regulace aktivity. Tedy podstata samočinných agentů – přizpůsobování se prostředí. Aby nevyužívali tolik výpočetních zdrojů, je proces detekce pozastaven, není-li v obraze nalezena žádná osoba. U detekce směru pohledu lze interval detekce očí prodloužit, pokud nebyl po určitou dobu detekován žádný uživatel. Oproti tomu u agenta pro detekci gest lze vypnout tvorbu digitální kostry, nenachází-li se nikdo v zorném poli senzoru, a pak periodicky kontrolovat, jestli je už uživatel před kinectem.

## 4.3 Testování na uživatelích

S výstupem systému souvisí i testování, tedy jak zhodnotit kvalitu získaných informací. Ze studia dostupných materiálů vyplynulo, že pravděpodobně nebyl specifikován jednotný způsob vyhodnocení multimodálního rozhraní. Proto lze uvažovat několik různých způsobů testování:

1) Cooperate evaluation – podobně jako usabillity testing je metoda určená pro získání zpětné vazby po počáteční implementaci. Výsledky mohou posloužit pro úpravu návrhu a urychlení jednotlivých cyklů vývoje. Toto testování spočívá ve vykonávání specifikovaných úloh uživatelem. Výběr správných úloh je pro tento typ vyhodnoceni základem. Je potřeba, aby byly proveditelné, reprezentovaly reálné úlohy a dostatečně prozkoumávaly možnosti prototypu. Z naměřených časů potřebných pro vykonání jednotlivých úkonů lze vyvodit, jestli rozhraní představuje zlepšení oproti existujícím aplikacím, nebo při opakování testů jak jsou uživatel schopni si dané rozhraní osvojit.

2) Testování modulů – jelikož se systém skládá z několika oddělených součástí, lze se zaměřit na chybovost jednotlivých modulů. Buď tedy testovat každý zvlášť, nebo všechny kromě jednoho najednou (pro navrhovaný systém pravděpodobně vhodnější kvůli přímé závislosti centrálního agenta na výstupech agentů). Z výsledků testů například úspěšnosti

detekce požadované akce lze vyvodit, jak jednotlivé moduly ovlivňují správnost fungování celého systému.

3) Wizard of Oz – často používána metoda při vytváření prototypů rozhraní. Jsou-li reakce systému simulovány jiným člověkem, odstíní se chyby zpracování vstupů. Lze tak vytvořit například sadu pravidel definujících možné akce výsledného rozhraní, které budou pravděpodobně lépe reflektovat požadavky uživatele.

## 4.4 Wizard of Oz

Testování navrhovaného systému jsem provedl v několika fázích. Jako první, před dokončením implementace, tedy bez funkčního prototypu, byly provedeny Wizard of Oz experimenty pro zjištění, jak uživatelé reagují na daný typ rozhraní a jaká gesta a příkazy by měl systém obsahovat. Experimentu se zúčastnilo 5 lidí (malý počet, ale pro získání základních informací postačující) a poznatky lze shrnout jako:

- Uživatelé se přirozeně snažili gesta doprovázet slovními příkazy (nevěděli, jak vyjádřit abstraktní příkazy, jako například „play“ gestem, proto inklinovali k slovním příkazům; gesta používali jako doprovodný prostředek pro vyjádření směru či míry).

- Raději používali gesta jedné ruky než dvou (i v případě ukázání na přístroj a následného gesta, sekvenci příkazu prováděli raději sériově než paralelně).

- Slovní příkazy používali ve formě klíčových slov, příliš se nesnažili o plynulou řeč. To je možná zapříčiněno tím, že jsou zvyklí na omezení dnešních rozhraní na bázi analýzy řeči.

- I bez předchozí demonstrace uživatele používají gesta swipe doprava, doleva, nahoru a dolů a ukázání, ale měli tendenci používat i online gesta, například pro postupné posouvání a nastavování hladiny hlasitosti.

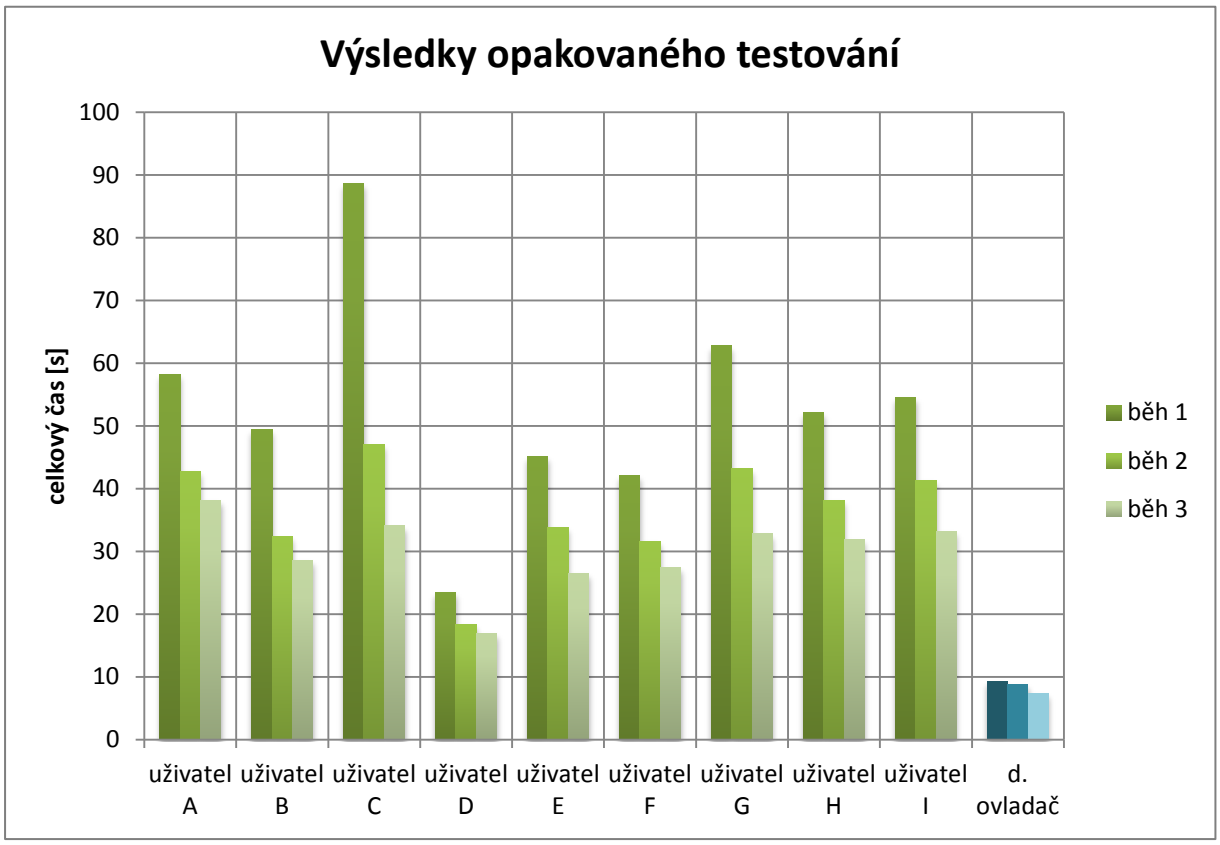
- Určení toho, že uživatel chce komunikovat se systémem a ne s někým jiným v místnosti pomocí určení očního kontaktu, se jeví jako dobré řešení. Testované subjekty se při zadávání příkazů zpravidla dívaly do kamery.

## 4.5 Osvojení rozhraní

Další testování proběhlo po dokončení funkčního prototypu a bylo zaměřeno na schopnost uživatelů si rozhraní osvojit. Testování bylo ve formě několika menších úloh zaměřených na běžné ovládání multimediálního centra. Pro ovládání byla použita aplikace XBMC přijímající příkazy prototypu, pomocí něj měl uživatel vykonat tyto úlohy:

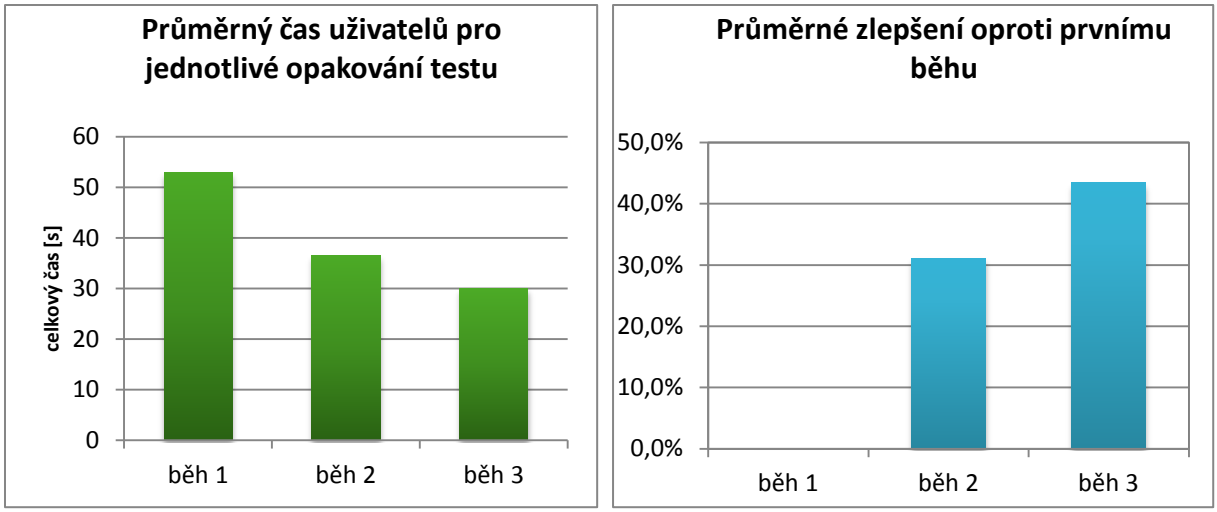
1. Navigace na položku menu „Movies“ a její výběr.
2. Spuštění prvního filmu z druhého řádku nabídky.
3. Návrat do hlavní nabídky.
4. Opětovná navigace na položku „Movies“.
5. Spuštění čtvrtého filmu z druhého řádku.

Aby se redukovaly problémy s ovládáním vzniklé tím, že i prostředí aplikace XBMC je pro uživatelé nové, byli před testováním s ním seznámeni. Také jim bylo předem předvedeno, jaké gesta systém detekuje (rozsah, v jakém byla gesta demonstrována, lze přirovnat k demonstračnímu videu či obsahu návodu k použití).



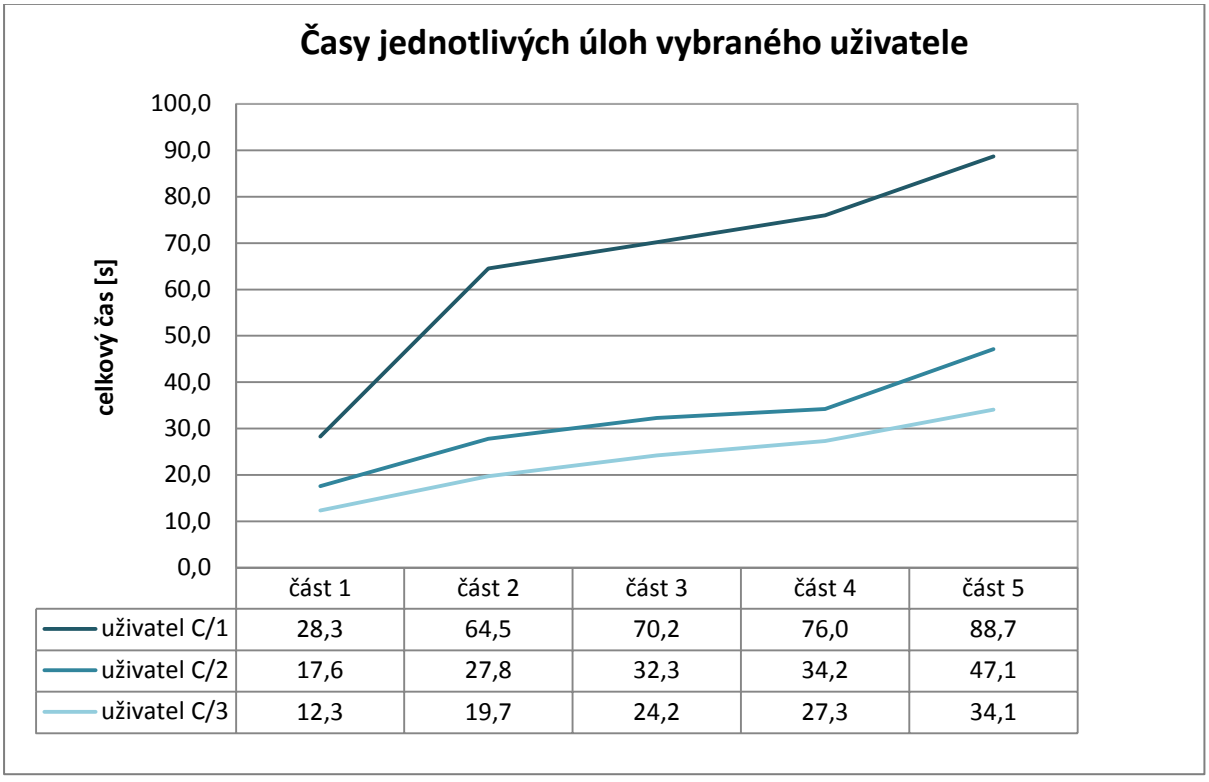

**Tabulka 2:** výsledky uživatelů při opakovaném vykonávání definované úlohy.

Když se podíváme na výsledky opakovaného vykonávání úlohy jednotlivými uživateli (tedy Tabulka 2), můžeme z grafu vyčíst, že ovládání šlo uživatelům většinou podobně dobře. U každého lze pozorovat výrazné zlepšení při opakování testu (v průměru jsou časy třetího experimentu lepší o 42% než u prvního). Dalo by se říci, že uživatele jsou relativně rychle schopní si dané ovládání osvojit (tedy rozhraní je pravděpodobně dostatečně intuitivní). Časy však nedosahovaly rychlosti klasického ovladače. Lze ale předpokládat, že kdyby měli uživatelé více času pro osvojení, přiblížili by se jejich výsledky ještě více pomyslné metě času konvenčního ovladače (potvrdil to i déle trvající experiment, kdy uživatel nebyl omezen počtem pokusů).




**Tabulka 3:** Celkové průměrné zlepšení, vlevo – průměrná čas při opakování testů, vpravo – zlepšení oproti prvnímu provedení testu v procentech.

Tabulka 3 ukazuje průměrný čas potřebný pro vykonání úlohy. Při zopakování testu se čas potřebný pro jeho vykonání zlepšil o 16 sekund a při dalším opakováni o dalších 6 sekund. Výsledky byly zprůměrovány bez předcházejícího rozdělení do skupin podle typu uživatele, protože žádné z testovaných skupin se nedařilo výrazně lépe (ať už uvažujeme rozdělení podle pohlaví, věku, dosaženého vzdělání či typu profesního zaměření). Tento fakt může být zapříčiněn tím, že s rozhraním takového typu se nikdo z testovaných uživatelů nesetkal a princip ovládání jim pouze vzdáleně připomínal dotykový displej.




| | část 1 | část 2 | část 3 | část 4 | část 5 |
|---|---|---|---|---|---|
| uživatel C/1 | 28,3 | 64,5 | 70,2 | 76,0 | 88,7 |
| uživatel C/2 | 17,6 | 27,8 | 32,3 | 34,2 | 47,1 |
| uživatel C/3 | 12,3 | 19,7 | 24,2 | 27,3 | 34,1 |

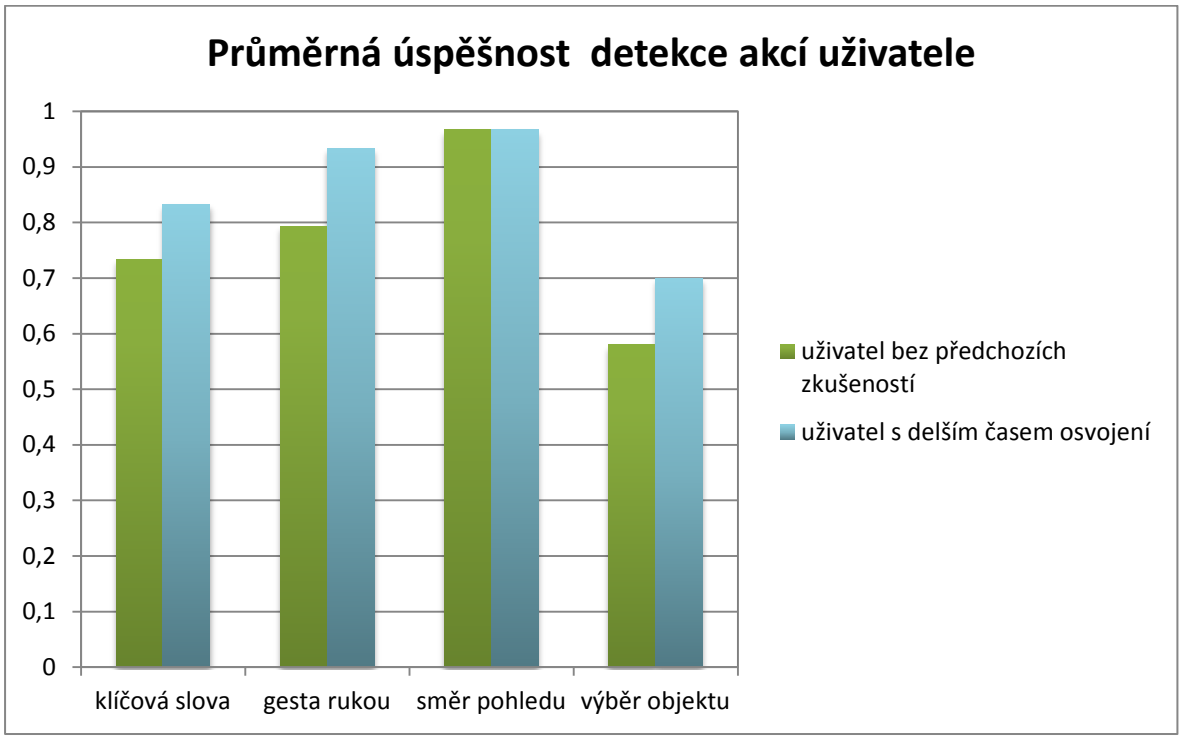
**Tabulka 4:** Podrobné výsledky vybraného uživatele reprezentující typický průběh.

Když se zaměříme na výsledky vybraného uživatele (Tabulka 4), lze u průběhu prvního běhu testu u jeho počátečních částí pozorovat horší výsledky než u ostatních částí. Bylo poměrně časté, že uživatelé měli z počátku obtíže s ovládáním, ať už z důvodu neúplného pochopení jak systém funguje, jak správně zadávat příkazy či chybě při prvních pokusech o ovládání (stávalo se, že uživatel omylem vstoupil do jiné nabídky a potom se musel vracet, což mělo za následek zhoršení výsledného času). V dalších částech si už uživatelé vedli zpravidla lépe, tedy osvojili si ovládání poměrně rychle (k tomu pravděpodobně přispělo i úvodní poučení o fungování systému, to by při reálném nasazení mohlo být nahrazeno piktogramy v návodu použití či instruktážním videem).

## 4.6 Spolehlivost detekce

Předchozí testy byly zaměřeny na schopnost uživatelů osvojit si vytvořený systém. Výsledky ukázaly, že je ovládání poměrně dobře osvojitelné a tedy snad i dostatečně intuitivní. Bylo ale ještě potřeba zjistit, jak dobře jednotlivé bloky fungují. Za tímto účelem jsem provedl sadu testů zaměřených na zpracování jednotlivých vstupů. Probíhaly opakovaným vyslovením klíčového slova, provedení gesta či výběru objektu. Pro lepší ověření vlastností opakoval uživatel testování pro několik variant vstupu:

1. Klíčová slova – volume, disable, next
2. Gesta rukou – swipe, klik, pozicování
3. Směr pohledu – otočení na systém
4. Výběr objektu – různé objekty




**Tabulka 5:** Výsledky analýzy vstupů. Jednotlivé akce (každé klíčové slovo, gesto) prováděl uživatel desetkrát.

Tabulka 5 ukazuje, jak je systém schopný sledovat jednotlivé akce uživatele. Nejhůře dopadl výběr objektu ukázáním, to bylo způsobeno pravděpodobně šumem ve vstupních datech (souřadnice dlaně a prstů nejsou příliš stabilní), možná i samotný algoritmus výpočtu průsečíku by mohl být implementován lépe. Lépe dopadla detekce klíčových slov, zde výsledek nejvíce ovlivnila výslovnost testovaného uživatele a snaha zřetelně artikulovat. Podobně se osvědčila detekce gest (swipe byl zpravidla detekován lépe než klik či pozicování). Detekce směru pohledu fungovala téměř bezchybně, selhávala v případě, měl-li uživatel více nakloněnou hlavu (při větším naklonění hlavy do strany detekce očí selhává i pře přímém pohledu do kamery).

U uživatele s většími zkušenostmi s testovaným rozhraním jsou výsledky lepší než u ostatních, to je způsobeno tím, že se účastnil i experimentů měřících časy při ovládání aplikace, takže měl lépe osvojený způsob komunikace. Takže lze říci, že čím déle má uživatel možnost rozhraní používat, tak roste nejen schopnost ovládat, ale i schopnost správného zadávání jednotlivých částí příkazu (zadávají příkazy rychleji a ve formě systémem lépe detekovanou).

Po provedení testů schopnosti ovládat multimediální centrum a chybovosti analýzy jednotlivých vstupů teoreticky zbývá zjistit, jaké chybovosti se dopouští centrální agent při provádění fúze. Centrální agent je však implementován tak, že závisí pouze na vstupních datech, tedy úspěšné provedení příkazu závisí pouze na správnosti vstupních dat. Proto jsem chybovost systému jako celku netestoval, z výsledků předchozího testu lze ale odvodit, že příkazy skládající se z více modalit budou zatíženy vetší chybovostí než příkazy využívající jeden či dva vstupy.

Kromě zjištění, jak jsou uživatelé schopní dané rozhraní osvojit a jak spolehlivě detekuje jejich akce, testování poskytlo i obecné poznatky o vytvořeném rozhraní:

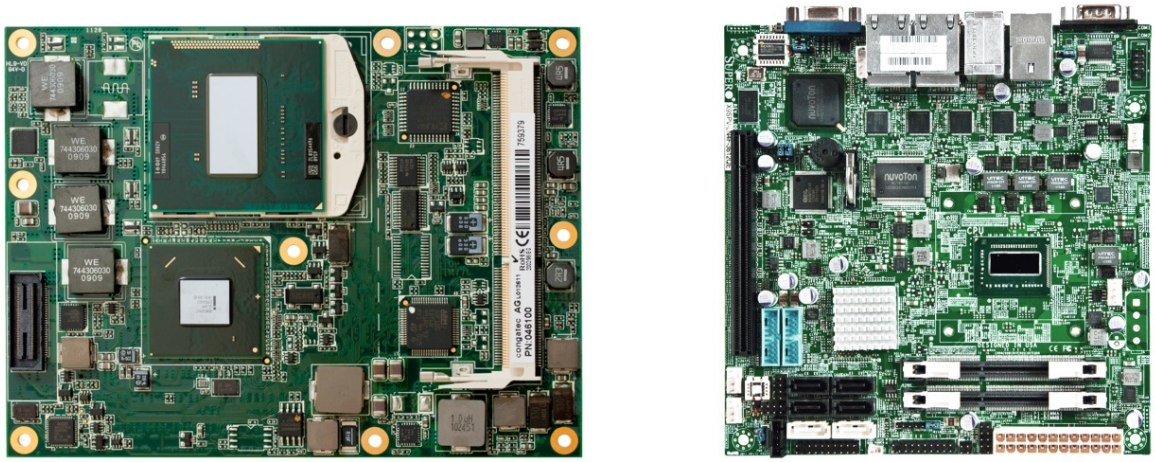
1) Princip rozdělení prostoru před uživatelem do zón pomáhá zabránit náhodnému zadávání gest, ale pro správné zadávání je potřeba uživatele poučit jak má gesta zadávat (při prvním kontaktu uživatel nemá představu o tom, jaká gesta systém sleduje, je potřeba ho instruovat ať už pomocí demonstrace či pomocných piktogramů).
2) Vzdálenost jednotlivých zón, ikdyž je přizpůsobena každému uživateli podle délky rukou, se liší i podle zvyklostí uživatele (někteří uživatele by uvítali, kdyby mohli gesta zadávat blíže těla, jiní jsou zase zvyklí více gestikulovat). Proto by bylo vhodné vytvořit profily s nastavením pro jednotlivé uživatelé.
3) Grafická zpětná vazba na pohyb uživatele má důležitou roli při osvojování ovládání. Díky ní měli uživatele alespoň tendenci pokoušet zadávat gesta požadovaným způsobem i bez toho aby věděli, jak bude systém reagovat.
4) Rozhraní XBMC není vhodné pro ovládání pomoci gest. Obrazovky obsahují mnoho prvků a tak vzrůstá riziko, že uživatel vybere jinou položku, než chtěl a bude nucen se vracet a proces opakovat. Multimediální centrum vytvořené pro navrhované ovládání by proto poskytlo lepší komfort.
5) Ovládání uživatelům zpravidla připadalo zábavné a zajímavé (ovládání některým z testovaných připomínalo pohybové hry konzoly Xbox). To by mohlo přispět k uplatnění v praxi.

## 4.7 Náměty k další práci

Téma multimodálního rozhraní nabízí mnoho možností další práce, jednak testovat různé přístupy k detekci akcí uživatele, kombinace různých modalit, design nového rozhraní a mnohé jiné.

**Vytvoření fyzického řešení**

V této práci byl popsán a realizován systém pro ovládání multimediálního centra, realizace však zůstala na úrovni prototypu, tedy bez kompletního řešení, které by šlo označit jako vhodné pro nasazení v domácnosti. Přizpůsobit centrálního agenta či agenta pro detekci směru pohledu pro běh na miniaturních počítačích by neměl představovat velký problém, ale zpracování vstupu ze senzoru Kinect vyžaduje značné výpočetní prostředky, které většina malých zařízení neposkytuje. Pro tyto účely lze využít malé desky s integrovanými procesory Intel i5 či i7 (jako například desky na Obrázek 34), pro ty nepředstavuje náročné zpracování dat problém, dále na nich může běžet i software multimediálního centra i další agenti.



**Obrázek 34:** Desky (vlevo Supermicro X9SPV-LN4F , vpravo Conga-TM67), které by mohly posloužit jako základ multimediálního centra a agenta detekce gest

**Jiné modality**

Pro realizaci byla použitá detekce klíčových slov a detekce očí. Lze využít ale i jiné přístupy a získat tak i prakticky odlišné modality. Pokud by byla vyvinuta vhodná knihovna pro převod farfield audio signálu na řeč a ne pouze na klíčová slova, získal by se tak zajímavý vstup poskytující teoreticky přirozenější způsob komunikace. V dalších verzích senzoru Kinect či jemu podobných lze očekávat zvýšení rozlišení, byla by pak i pro větší vzdálenosti dostupná detekce směru pohledu pomocí mapování trojrozměrného modelu. Umožní tak lepší určení, kam se uživatel dívá (nabízí se i možnost vybírat zařízení pro interakci pohledem). Zapojit do systému by samozřejmě šlo zcela odlišné vstupy, než jsou použité teď. Pero či dotykové displeje dovolují jiný způsob zadávání, ale s tím, že jsou potřeba další ovladače – systém by tedy nebyl handsfree. Oproti tomu rozpoznání mluvčího by nevyžadovalo další periferie a poskytlo by zajímavé informace například pro tvorbu profilů, či s kombinací s rozpoznáním uživatele podle obličeje by pomohlo určit, který uživatelů zadává slovní příkazy a gesta.

**Multimodální multimediální centrum**

Navrhovaný systém sice poskytuje nový způsob ovládání, ale ten je aplikován na klasické rozhraní aplikace XBMC. Aby vytvořené ovládání znamenalo skutečný pokrok oproti klasickým přístupům, je potřeba přizpůsobit i frontendovou aplikaci. Jak by přehrávač ovládaný gesty mohl vypadat, jsem již v technické zprávě zmínil, návrh by ale bylo potřeba implementovat a také přizpůsobit požadavkům uživatelů. Kromě multimodálního centra lze systém napojit na inteligentní dům, což by znamenalo další práci a teoreticky i spolupráci s firmou vyvíjející software pro zmiňované inteligentní domy. S tím je spojena i potřeba dalších spotřebičů, které by bylo možné dálkově ovládat.

**Pokročilá gesta**

Agent pro detekci gest poskytuje informace o jednoduchých gestech, které uživatel provedl, tento přístup by šlo rozšířit o detekci specifičtějších gest. Bylo by možné sledovat gesta založená na konfiguraci ruky, jako například chycení, ukázání prstem či rotace dlaně (podobně jako u regulace hlasitosti u hi-fi sestav), ty by dále rozšířili možnosti navrhovaného systému. Detekce takovýchto gest vyžaduje zpracování dalších dat, ne pouze analýzu pohybu ruky, je ale jistá šance že detekce právě těchto gest přibude v dalších verzích sady pro vývoj softwaru pro Kinect (Kinect for Windows SDK ve verzi 1.7 přinesl omezené možnosti detekce gesta chycení).

**Učení akcí a profily**

Pro zlepšení použitelnosti by pravděpodobně bylo vhodné systém rozšířit o možnost učení akcí za běhu nebo i pomocí trénovaní sady. Uživatelé mají různé požadavky a zvyky, proto by sada pravidel definujících akce, které systém podporuje, nemusela být statická, naopak mohla by se přizpůsobovat zvyklostem uživatele. S tím souvisí i tvorba profilů pro jednotlivé uživatelé, ty by mohly obsahovat sadu akcí, které uživatel vykonává, informace o něm (například pomocná data k detekci gest jako délka ruky či vzdálenosti jednotlivých zón) či osobní informace, jako například profil pro přehrávání audia a videa, používané aplikace nebo jejich profily.

# 5 Závěr

V této práci jsou diskutovány především principy potřebné pro tvorbu multimodálního rozhraní. Jsou zde uvedeny důvody, proč se věnovat návrhu systému pro analýzu signálů z více senzorů různého typu, dále úrovně, na kterých může docházet k integraci získaných informací a architektury, jak lze takový systém vytvořit.

Významnou části je kapitola věnující se metodám fúze dat na úrovni rozhodování. Tedy přístupům, jak ze získaných událostí popisujících akce uživatele v rámci jednotlivých vstupů získat jednu komplexní informaci. Pro tyto účely existuje více vhodných metod. Byly nastudovány přístupy jako melting pot, frame, unifikace, dialogem řízena fúze či hybridní fúze. Všechny poskytují zajímavé způsoby interpretace multimodálních vstupů. Navíc již byly použity v praxi, což demonstruje tabulka existujících aplikací založených na ovládání pomocí více vstupů.

Další kapitola se zabývala návrhem systému a rozdělením činnosti jednotlivých bloků. Systém jsem navrhl jako decentralizovaný, kdy jednotlivé vstupy jsou zpracovány samostatně, a až získané údaje o tom, jaké gesto uživatel vykonal či jaký příkaz vyslovil, slouží k rozhodnutí, jakou akci chce realizovat. Za účelem rozhodování o výsledné akci byl navržen centrální agent fungující na principu hybridní fúze, tedy kombinace dialogem řízené fúze a fúze unifikací. Kromě návrhu způsobu fúze jsem se zabýval i návrhem ostatních části systému, tedy agentů pro zpracování vstupních dat. Způsobů jak vstupní data analyzovat je více proto ke každému bloku jsem zmínil různé varianty řešení.

Co se týče samotné realizace, implementoval jsem nejen bloky systému, ale také pomocné aplikace pro nahrávání a anotaci získaných dat. Zaznamenávají se hloubková, barevná, audio data a polohy kloubů detekované kostry pro každý snímek. Data mohou být použita nejen pro určení gest vhodných pro ovládání, ale mohou pomoci i dalším kolegům pracujícím s hloubkovými daty, například pro případné natrénování detektoru gest. Samotný systém jsem realizoval jako několik samostatných aplikací, přesněji samostatné agenty zpracovávající každý z možných vstupů, dále centrální agent rozhodující, jakou informaci oddělené vstupy tvoří jako celek. Takto vytvořené rozhraní jsem použil pro ovládání běžného multimediálního centra XBMC. Pomocí něj jsem provedl několik testů určených pro zjištění, jak jsou uživatelé schopní daným způsobem ovládat přehrávač a také jaké chybovosti se jednotlivé bloky dopouštějí. Výsledky ukázaly, že uživatelé jsou již po menším počtu opakování testu schopni danou aplikaci ovládat. Při třetím opakování testovací úlohy se čas potřebná pro vykonání série úkonů v průměru zlepšil téměř dvojnásobně. Kromě schopnosti si dané rozhraní osvojit byly provedeny testy zaměřené na chybovost jednotlivých agentů zpracovávajících vstupy. Pro nasazení v praxi, ať už pro ovládání multimediálního centra, interakci s přístroji či reklamní účely, by bylo potřeba upravit řešení pro běh na menších zařízeních s menší spotřebou.

# Literatura


[1] WEI, Lai a Huosheng HU. *Towards Multimodal Human-Machine Interface for Hands-free Control: A survey*. 2011, ISSN 1744-8050.

[2] A. G. Hauptmann and P. McAvinney, "Gesture with speech for graphics manipulation," Int. J. Man-Machine Studies, vol. 38, pp. 231–249, Feb. 1993.

[3] S. Oviatt, A. DeAngeli, and K. Kuhn, "Integration and synchronization of input modes during multimodal human–computer interaction," in Proc. Conf. Human Factors in Computing Systems (CHI'97), Atlanta, GA, pp. 415–422.

[4] R. R. Murphy, "Biological and cognitive foundations of intelligent sensor fusion," IEEE Trans. Syst., Man, Cybern., vol. 26, pp. 42–51, Jan. 1996.

[5] SHARMA, Rajeev, Vladimir I. PAVLOVIC a Thomas S. HUANG. Toward multimodal human-computer interface. *Proceedings of the IEEE*. 1998, č. 5. ISSN 0018-9219.

[6] ALIBALI, Martha W., Sotaro KITA a Amanda J. YOUNG. Gesture and the process of speech production: We think, therefore we gesture. *Language and cognitive processes*. 2000, č. 5, 593–613. ISSN 0169-0965.

[7] HALL, David L. a James LLINAS. An introduction to multi-sensor data fusion. In: SOCIETY], [sponsored by the IEEE Circuits and Systems. *ISCAS '98: proceedings of the 1998 IEEE International Symposium on Circuits and Systems : May 31-June 3, 1998, Monterey Conference Center, Monterey, CA*. Piscataway, NJ: IEEE, 1998, 537 - 540. ISBN 0-7803-4455-3.

[8] RADHA, V. a C. VIMALA. A Review on Speech Recognition Challenges and Approaches. In: *World of Computer Science and Information Technology Journal*. New York: ACM, 2012, s. 1-7. ISBN 978-1-4503-1310-0ISSN 2221-0741.

[9] W. Elmenreich. An introduction to sensor fusion. Technical Report 47/2001, TechnischeUniversit¨at Wien, Institut f¨ur Technische Infor-matik, Vienna, Austria, 2001.

[10] Superior colliculus. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-10-08]. Dostupné z: http://en.wikipedia.org/wiki/Superior_colliculus.

[11] MOSALLAEI, Mohsen, Karim SALAHSHOOR a Mohammadreza BAYAT. Centralized and Decentralized Process and Sensor Fault Monitoring Using Data Fusion Based on Adaptive Extended Kalman Filter Algorithm. *Measurement Journal (ScienceDirect)*. 2008, roč. 41, č. 10, s. 1059-1076.

[12] WALD, Lucien. *Data fusion: definitions and architectures ; fusion of images of different spatial resolutions*. Paris: Les Presses de l'École des Mines, 2002. ISBN 29-117-6238-X.

[13] MICROSOFT. *Human Interface Guidelines: v1.7* [online]. 2013 [cit. 2013-03-28]. Dostupné z: http://www.microsoft.com/en-us/kinectforwindows/develop/learn.aspx

[14] TUE VO, Minh a C. WOOD. Building an application framework for speech and pen input integration in multimodal learning interfaces. In: ... *The 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings: May 7-10, 1996, Marriott Marquis Hotel, Atlanta, Georgia, USA*. [Piscataway, NJ: IEEE Service Center], 1996, 3545–3548. ISBN 0-7803-3192-3. DOI: 10.1109.

[15] LÓPEZ-CÓZAR, Ramón a Zoraida CALLEJAS. Multimodal Dialogue for Ambient Intelligence and Smart Environments. *Handbook of ambient intelligence and smart environments*. New York: Springer, 2010, č. 2. ISSN 978-0-387-93807-3.

[16] NIGAY, Laurence a Joëlle COUTAZ. Generic Platform for Addressing the Multimodal Challenge. In: EDITORS, Irvin R. *CHI '95 conference proceedings: ACM Conference on Human Factors in Computing Systems : mosaic of creativity, May 7-11, 1992, Monterey, California*. New York, N.Y: Association for Computing Machinery, 1995, s. 98-105. ISBN 0-201-84705-1.

[17] BLUMENDORF, Marco. *Multimodal interaction in smart environments: a model-based runtime system for ubiquitous user interfaces*. Berlin, 2009. Disertační práce. Berlin Institute of Technology.

[18] MANCHÓN PORTILLO, Pilar, Guillermo PÉREZ GARCÍA a Gabriel AMORES CARREDANO. Multimodal fusion: a new hybrid strategy for dialogue systems: In Proceedings of the 8th international Conference on Multimodal interfaces. In: *ICMI '06: Eighth International Conference on Multimodal Interfaces : conference proceedings : November 2-4, 2006, the Fairmont Banff Springs Hotel, Banff, Alberta, Canada*. New York, N.Y.: Association for Computing Machinery, c2006, s. 357-363. ISBN 1-59593-541-X.

[19] LALANNE, Denis, Laurence NIGAY, Philippe PALANQUE, Peter ROBINSON, Jean VANDERDONCKT a Jean-François LADRY. Fusion engines for multimodal input: a survey. In: GENERAL CHAIRS, James Crowley, Daniel Gatica-Perez PROGRAM CHAIRS, Sponsored by ACM SIGCHI a In cooperation with IEEE. --. *ICMI-MLMI '09: Proceedings of the International Conference on Multimodal Interfaces*. New York, N.Y: ACM Press, 2009, s. 153-160. ISBN 978-1-60558-772-1.

[20] RAISAMO, Roope. *Multimodal human-computer interaction a constructive and empirical study*. Tampere, [Finland]: University of Tampere, 1999. ISBN 95-144-4702-6. Disertační práce. University of Tempere.

[21] DUMAS, Bruno, Beat SIGNER a Denis LALANNE. Fusion in Multimodal Interactive Systems: An HMM-Based Algorithm for User-Induced Adaptation. In: *EICS '12 Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*. New York: ACM, 2012, s. 15-24.

[22] MEDL, A., I. MARSIC, M. ANDRE, C. KULIKOWSKI a J. FLANAGAN. Multimodal User Interface for Mission Planning. In: MICHAEL COEN, Chair. *Intelligent environments: papers from the 1998 AAAI Symposium, March 24 - 26, Stanford, California*. Menlo Park, Calif: AAAI Press, 1998, s. 103-109. ISBN 978-1-57735-047-7.

[23] TOPTSIS, Ioannis, Shuyin LI, Britta WREDE a Gernot A. FINK. A Multi-modal Dialog System for a Mobile Robot. In: *International Conference on Spoken Language Processing*. Korea: Jeju, 2004, s. 273-276.

[24] NIGAY, Laurence a Joëlle COUTAZ. A Generic Platform for Addressing the Multimodal Challenge. In: *Proceedings CHI'95*. Denver: ACM, 1995, s. 98-105.

[25] KRAHNSTOEVER, N., S. KETTEBEKOV, M. YEASIN a R. SHARMA. A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays. In: *Fourth IEEE International Conference on Multimodal Interfaces, 14-16 October, 2002, Pittsburgh, Pennsylvania*. Los Alamitos, Calif.: IEEE Computer Society, c2002, s. 349-355. ISBN 0-7695-1834-6.

[26] ALEJANDRO, Jaimes a Sebe NICU. Multimodal human-computer interaction: A survey. In: *Computer Vision and Image Understanding*. 108. vyd. New York: Elsevier Science Inc., 2007, s. 116-134.

[27] L. HALL, David a James LLINAS. An Introduction to Multisensor Data Fusion. In: *Proceedings of the IEEE*. New York: ACM, 1997, 6 - 23. ISSN 0018-9219.

[28] MAURI, César, Toni GRANOLLERS, Jesús LORÉS a Mabel GARCÍA. Computer Vision Interaction for People with Severe Movement Restrictions. In: *Human technology an interdisciplinary journal on humans in ICT environments*. Finland: University of Jyväskylä, Agora Center, 2006, 38–54.

[29] ACHARYA, Chitra, Harold THIMBLEBY a Patrick OLADIMEJI. Human computer interaction and medical devices. In: *BCS '10 Proceedings of the 24th BCS Interaction Specialist Group Conference*. UK: British Computer Society Swinton, 2010, s. 168-176.

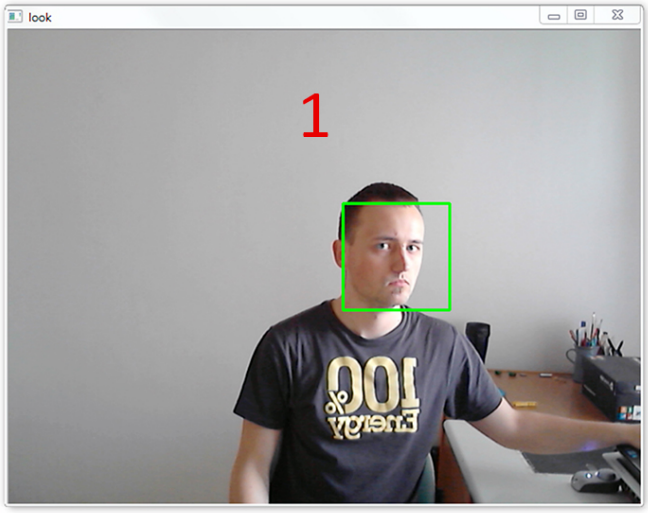
[30] DIX, Alan, Janet FINLAY, Gregory D. ABOWD a Russell BEALE. *Human-computer interaction*. 3rd ed. Harlow, England: Pearson/Prentice-Hall, 2004, xxv, 834 s. ISBN 01-304-6109-1.

[31] MÜLLER, Jörg, Robert WALTER, Gilles BAILLY, Michael NISCHT a Florian ALT. Looking Glass: A Field Study on Noticing Interactivity of a Shop Window. In: *CHI '12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM, 2012, s. 297-306. ISBN 978-1-4503-1015-4.

# Seznam příloh

Příloha 1. Manuál – ovládání aplikací
Příloha 2. CD – zdrojové kódy aplikací, prezentační video, plakát

## 5.1 Manuál – ovládání aplikací




1. ovládání klávesovými zkratkami
   a) x|q – ukončení
   b) n – další webkamera
   c) r – restart
   d) i – další Ip adresa


2. ukončení aplikace
3. skrytí okna
4. nastavení Ip na posílání na výchozí bránu
5. připojení na další dostupné rozhraní
6. nastavení sklonu senzoru Kinect