Zápočtová úloha z předmětu KIV/ZSWI

OBJEKTOVÝ NÁVRH APLIKACE

Android aplikace pro nastavení parametrů a ovládání stimulátoru pro neuroinformatické experimenty

31.5.2016

Tým: „Tým snů“

Členové:

Petr Štechmüller	pstechmu@students.zcu.cz
Zuzana Soukupová	soukupz@students.zcu.cz
Antonín Vrba	avrba@students.zcu.cz

Obsah

1. Úvod

Tento dokument popisuje softwarový návrh aplikace pro Android, která bezdrátově nastavuje mozkový stimulátor. Nejprve bude systém zařazen do kontextu pomocí diagramů a následně budou popsány programovací technologie. Nakonec bude znázorněna struktura packetu pro výměnu dat. Dokument obsahuje diagram užití (usecase), ITIL a diagram komponent, které tvoří aplikaci.

1.1 Účel systému

Účel systému se nezměnil, viz DSP.

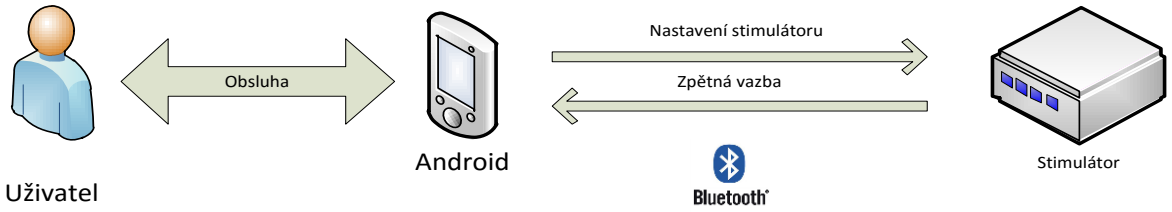
1.2 Slovníček definic, pojmů a zkratek

1.3 Odkazy na další dokumenty

2. Kontext a architektura systému

2.1 Kontext systému

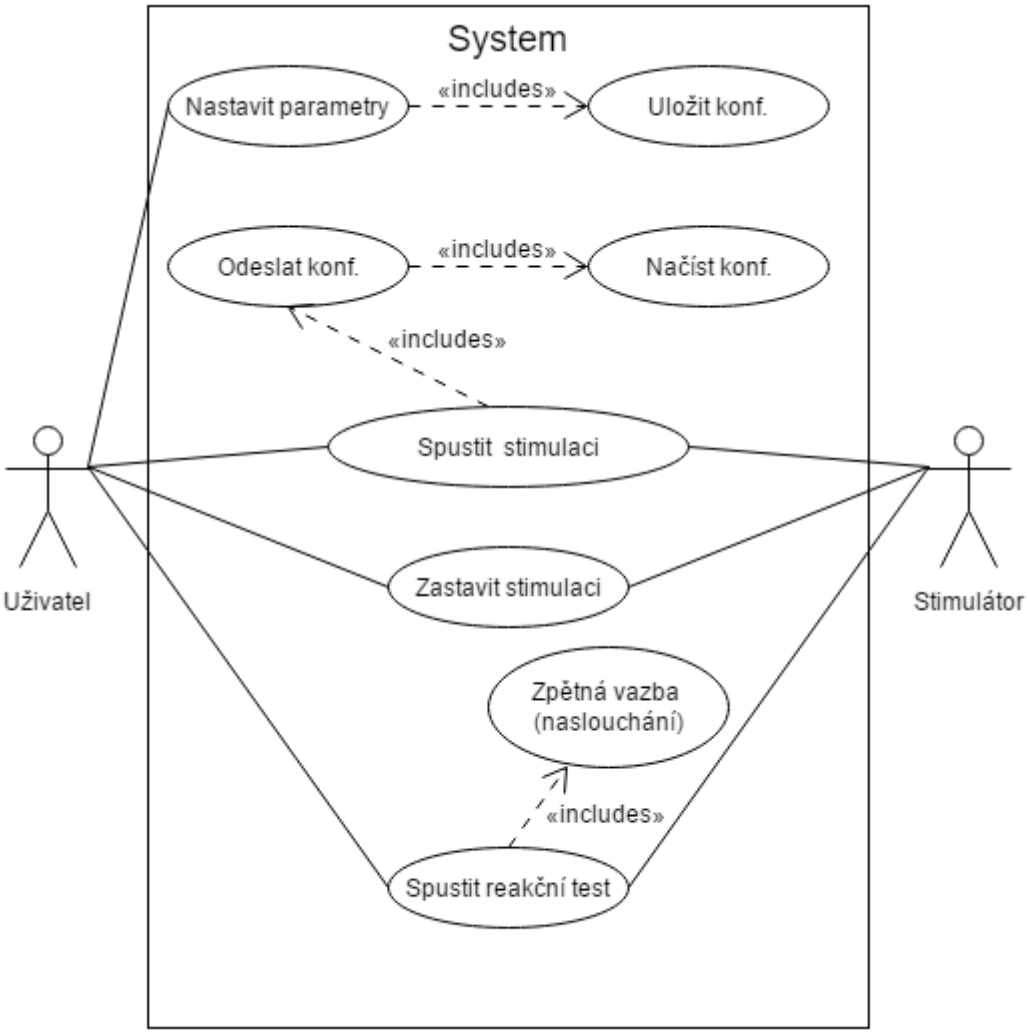
Nejobecnější popis přístupu nebo obsluhy systému je znázorněný pomocí ITIL diagramu:



ITIL Diagram

Uživatel obsluhuje stimulátor skrze aplikaci běžící na Android zařízení. Z diagramu je patrné, že zde kromě výstupních dat z aplikace existuje také zpětná vazba, kterou produkuje stimulátor. Je tedy nutné příchozí informace zobrazit uživateli.

Konkrétní uživatelské aktivity znázorňuje diagram užití:



Usecase diagram

Z Usecase diagramu je zřejmé, že uživatel skrze aplikaci provádí nastavování stimulačních parametrů a poté spouští a zastavuje požadované stimulace. Při spuštění reakčních testů je navíc informován o výsledcích experimentu a je mu prezentována zpětná vazba stimulátoru. Konkrétním typem zpětné vazby je v současném návrhu aplikace např. Reakční experiment, kdy subjekt reaguje na různé podněty, což je vyhodnoceno a posláno zpět do naší aplikace, kde se tyto údaje zobrazí.

2.2 Architektura systému, přehled podsystémů

Diagram komponent znázorňuje logické celky aplikace:

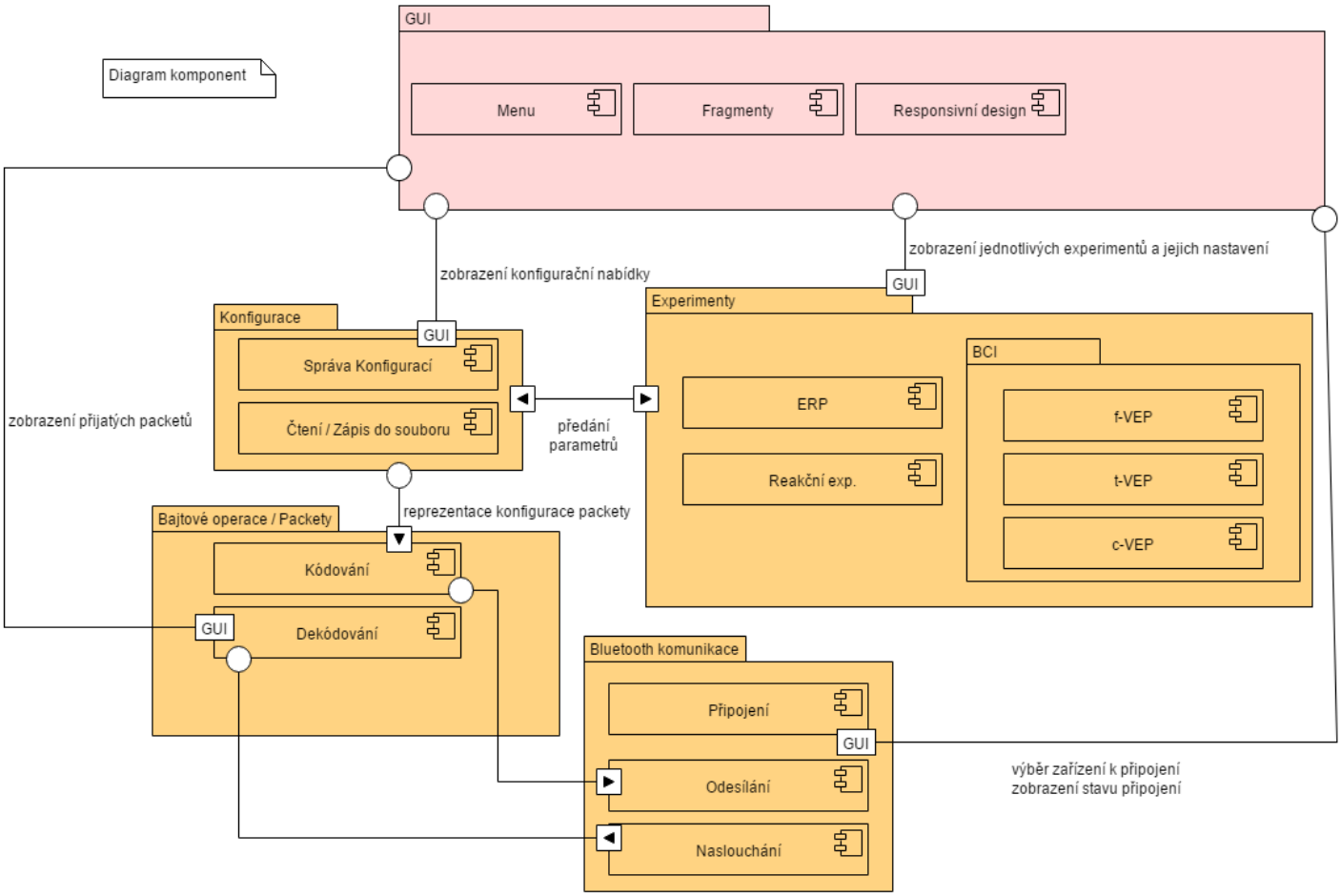


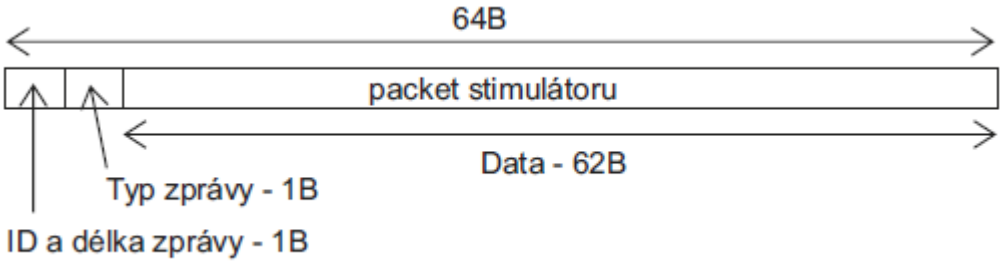
Diagram komponent

2.3 Zvolená technologie, programovací jazyk ad., důvody

Výchozím programovacím jazykem pro Android je **Java**, tedy kód běží pod virtuálním strojem. Do Androidu 4.4 to byl „Dalvik“, nyní se používá prostředí **ART**. GUI pro android se většinou píše pomocí **XML**, nebo je možné jej „naklikat“ přímo ve vývojovém prostředí, avšak tento způsob není všemocný. Projekt je sestavován pomocí nástroje **Gradle**, což lze považovat za nástupce dobře známého nástroje Ant. Dříve se pro vývoj na Androidu používala kombinace Eclipse + Android SDK, dnes se však používá přímo **Android Studio**, které pro Google vyvíjí společnost JetBrains. Pro ukládání konfiguračních souborů byl z důvodu komplikovaného parsování XML, byl zvolen dobře podporovaný formát **JSON**.

3. Typy informací zpracovávané systémem

Pro komunikaci se stimulátorem byl vytvořen speciální protokol, který má následující strukturu packetů:

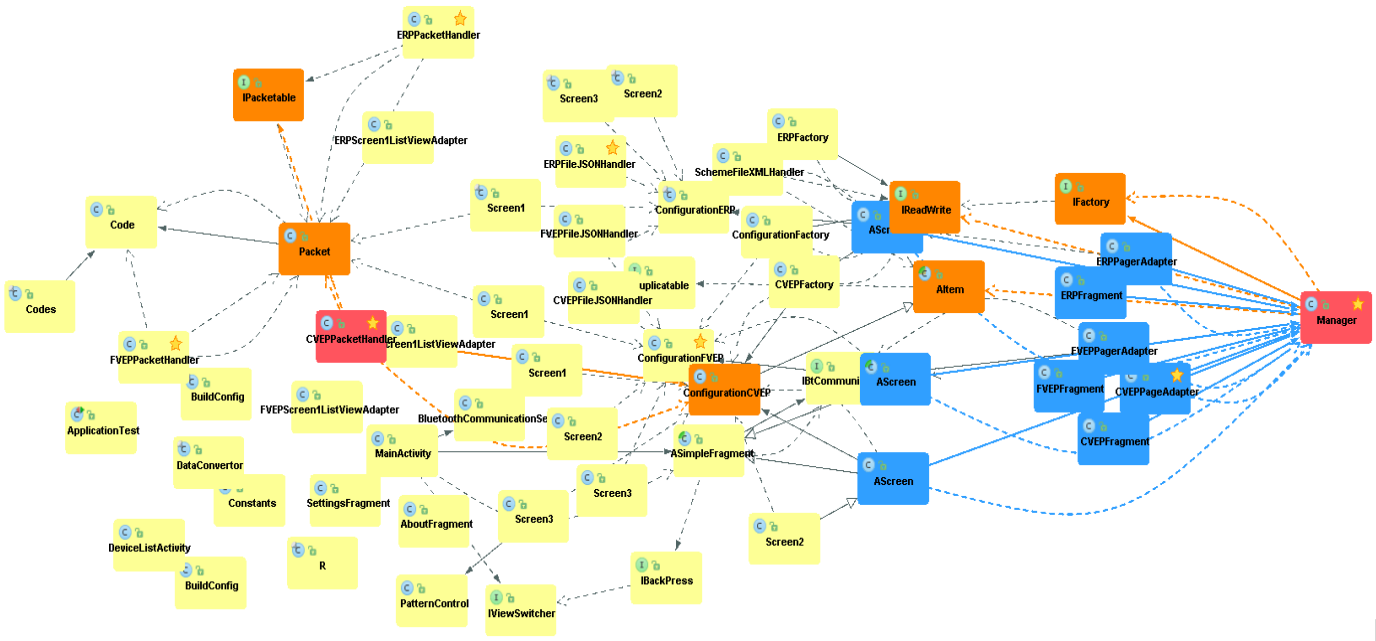


Struktura packetu

1. Byte – hlavička
    - ID – 2 bity (zatím nevyužíváno)
    - LEN – 6 bitů (počet datových bajtů)
2. Byte – typ zprávy
    - Do tohoto bajtu se vkládá hodnota odpovídající parametru např. pokud chceme nastavit jas u výstupu č.3 potom: 0x1F
3. Další Byty - datové

4. Podsystémy

Již v rané části vývoje aplikace (v0.3.2) obsahuje projekt desítky tříd a je velice těžké určit a přesně nadefinovat diagram tříd pro jednotlivé moduly (velmi se prolínají).



Náhled všech tříd, verze aplikace 0.3.2

4.1 Bluetooth komunikace

Aplikace potřebuje komunikační rozhraní a proto existuje tato komponenta. Každá lepší aplikace by tuto komunikační vrstvu měla mít nezávislou na zbytku kódu a stejně je to i v tomto případě. Dokáže vzít balík packetů a odeslat je do stimulátoru, stejně tak předat přijaté bajty ke zpracování.

4.2 Bajtové operace / Packety

Slouží pro převedení uložené JSON konfigurace do packetové podoby (list packetů). Zároveň dokáže dekódovat přijaté zprávy.

4.3 Konfigurace

Každý implementovaný experiment má svou vlastní konfiguraci. Proto musel vzniknout univerzální Manager konfigurací, který umí pracovat s generickým typem konfigurace.

4.4 Experimenty

Největší část kódu zabírají právě implementované experimenty, jelikož se často principiálně liší, nelze v nich nalézt společné prvky a využít prvků OOP k zjednodušení kódu.

4.5 GUI

Reagování na uživatelské pokyny, zobrazování načtených konfigurací, nastavování parametrů stimulací a mnoho jiných interakcí. To vše zabezpečuje GUI aplikace.

5. Přiřazení tříd/modulů programátorům

Petr Štechmüller - Architektura aplikace, GUI, Experimenty, Konfigurace, Komunikace
Antonín Vrba – Packetové / Bajtové operace