

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## DIPLOMOVÁ PRÁCE



Petr Kavánek

### **System pro správu a optimalizaci využití prodejních ploch**

Katedra softwarového inženýrství

Vedoucí diplomové práce: Prof. RNDr. Jaroslav Král, DrSc.

Studijní program: Informatika, softwarové systémy

Rád bych na tomto místě poděkoval vedoucímu mé diplomové práce, prof. RNDr. Jaroslavu Královi, DrSc., za cenné rady a náměty, které jsem ve své práci použil.

Podobně patří moje poděkování společnosti ExTech s.r.o., která podpořila vznik této práce.

Hlavní díky patří mojí ženě, za to, že vybojovala boj s těžkou nemocí a dala mi tak sílu a motivaci tuto práci dokončit.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 23.03.2009

*Petr Kavánek*

# Obsah

<b>1 Úvod.....</b>	<b>5</b>
1.1 Merchandising.....	5
1.2 Category management.....	5
1.3 Planogramy.....	5
<b>2 Popis problému a cílů práce.....</b>	<b>7</b>
2.1 Kdo a proč vytváří planogramy.....	7
2.1.1 Řetězce.....	7
2.1.2 Kooperace prodejen.....	7
2.1.3 Distributoři.....	8
2.1.4 Výrobci.....	8
2.1.5 Firmy specializované na merchandising.....	8
2.2 Faktory ovlivňující výběr produktů.....	8
2.2.1 Objektivní faktory.....	8
2.2.2 Subjektivní faktory.....	9
2.3 Nástroje usnadňující tvorbu planogramů.....	9
2.4 Nedostatky existujících nástrojů.....	11
2.4.1 Standardní planogramy a jejich nevýhody.....	12
2.5 Cíle práce.....	12
<b>3 Struktura práce.....</b>	<b>14</b>
<b>4 Analýza problému.....</b>	<b>15</b>
4.1 Požadované vlastnosti merchandisingového systému.....	15
4.1.1 Údaje o produktech důležité pro tvorbu planogramů.....	15
4.1.2 Údaje o skříních.....	16
4.2 Obecně platná pravidla rozmístování produktů.....	17
4.2.1 Produktbloky.....	17
4.2.2 Jedinečná poloha produktu ve skříní.....	17
4.2.3 Uchopitelnost produktu.....	17
4.2.4 Nosnost produktu.....	18
4.3 Pokročilé vizuální efekty využívané v planogramech.....	18
4.3.1 Vertikální produktbloky.....	18
4.3.2 Magnetické čáry.....	20
4.3.3 Zarovnání produktů.....	20
4.4 Výhody a nedostatky standardních planogramů.....	21
4.4.1 Příliš široký produktblok.....	22
4.4.2 Příliš vysoký produktblok.....	22
4.4.3 Nevyužití místo ve skříní.....	22
4.4.4 Těžko realizovatelné vizuální efekty.....	22
<b>5 Návrh řešení.....</b>	<b>24</b>
5.1 Univerzální bloky.....	24
5.1.1 Poloha a délka univerzálního bloku v rozmístovací mřížce.....	25
5.1.2 Mapování polohy z rozmístovací mřížky na polohu v planogramu.....	25
5.1.3 Priorita.....	26

5.1.4	Minimální počet viditelných kusů.....	26
5.1.5	Maximální počet viditelných kusů.....	26
5.1.6	Maximální počet kusů nad sebou a vzdálenost od stropu.....	27
5.1.7	Požadovaný počet kusů nad sebou.....	27
5.1.8	Výplň mezi jednotlivými kusy.....	27
5.1.9	Příslušnost k magnetické čáře.....	27
5.2	Magnetické čáry.....	28
5.3	Priority polic.....	29
5.4	Zarovnání bloků.....	29
5.5	Rozmíst'ovací algoritmus.....	29
5.5.1	Vstup.....	29
5.5.2	Odstranění bloků s příliš hlubokými produkty.....	30
5.5.3	Příprava skříně.....	30
5.5.4	Odstranění přebytečných řádků.....	32
5.5.5	Odstranění bloků s příliš vysokými produkty.....	32
5.5.6	Inicializace datových struktur pro expanzi bloků.....	32
5.5.7	Expanze bloků.....	34
5.5.8	Důkaz regulérnosti výsledku.....	40
5.5.9	Důkaz optimálnosti výsledku.....	43
5.5.10	Důvody pro neumístění produktů.....	44
<b>6</b>	<b>Popis prototypové implementace.....</b>	<b>45</b>
6.1	Uživatelská dokumentace.....	45
6.1.1	Instalace QuantPE.....	45
6.1.2	QuantPE s využitím demonstrační databáze.....	46
6.2	Programátorská dokumentace.....	49
6.2.1	Model skříně.....	49
6.2.2	Třídy reprezentující universální planogram.....	49
6.2.6	Implementace rozmíst'ovacího algoritmu.....	50
6.2.7	Vektorová reprezentace planogramu.....	50
<b>7</b>	<b>Uplatnění navrženého řešení.....</b>	<b>52</b>
7.1	Systémy využívající univerzální planogramy.....	52
7.1	QuantEE / i-merch.....	52
7.1.2	QuantSE.....	53
7.1.3	PPP.....	53
7.1.4	Prototyp inteligentního rozhodování pro merchandising.....	53
7.2	Případové studie.....	54
7.2.1	GEHE A-max.....	54
7.2.2	Sanacorp „Meine Apotheke“.....	54
<b>8</b>	<b>Závěr.....</b>	<b>56</b>
	<b>Příloha A.....</b>	<b>57</b>
	<b>Příloha B.....</b>	<b>58</b>
	<b>Seznam použité literatury.....</b>	<b>59</b>
	<b>Seznam ilustrací.....</b>	<b>60</b>

**Název práce:** Systém pro správu a optimalizaci využití prodejních ploch

**Autor:** Petr Kavánek

**Katedra:** Katedra softwarového inženýrství

**Vedoucí diplomové práce:** Prof. RNDr. Jaroslav Král, DrSc.

**E-mail vedoucího:** kral@ksi.ms.mff.cuni.cz

**Abstrakt:**

Planogramy jsou velmi důležitým a často užívaným prostředkem komunikace mezi výrobcí, distributory a prodejny. Planogramy velmi dobře a srozumitelně vyjadřují, jak mají být produkty v prodejně vystaveny. Kvalita rozmístění produktů výrazně ovlivňuje tržby prodejny.

Vytvořit přesné planogramy pro stovky či tisíce prodejen, je s existujícími merchandisingovými systémy často velmi náročné a proto se merchandisingová oddělení často uchylují pouze k tvorbě několika desítek standardních planogramů.

V práci je provedena analýza obecných vlastností planogramů a nedostatků a výhod plynoucích z distribuce standardních planogramů místo planogramů reálných. Navržené řešení odstraňuje nedostatky standardních planogramů. Umožňuje merchandisingovému oddělení namísto standardních planogramů vytvářet planogramy univerzální s vynaložením řádově stejného úsilí. Popsaný rozmisťovací algoritmus zajišťuje automatické převedení univerzálních planogramů na planogramy reálné.

Popsané řešení našlo již během psaní práce významné uplatnění zejména v oblasti tvorby planogramů pro lékárny, proto jsou v práci uvedeny i příklady tohoto uplatnění.

**Klíčová slova:** planogram, merchandising, category management

---

**Title:** System for management and optimisation of retail space

**Author:** Petr Kavánek

**Department:** Department of Software Engineering

**Supervisor:** Prof. RNDr. Jaroslav Král, DrSc.

**Supervisors e-mail address:** kral@ksi.ms.mff.cuni.cz

**Abstract:**

Planograms are very important and often used way of communication between producers distributors and stores. They are very good in expressing how presentation of products in store should look like. Quality of product presentation influences turnover of the store.

It is very difficult to produce hundreds or thousands precise planograms with existing merchandising tools. That is why many merchandising departments only create limited number of standard planograms.

This thesis contains general analysis of planograms and discussion of main advantages and disadvantages of standard planograms production. The proposed solution solves disadvantages of standard planograms. Merchandising departments can use the solution to produce precise planograms for all of their stores with need of nearly same effort as in case of standard planograms. Described algorithm is able to generate precise planograms automatically from universal planograms that are created instead of standard planograms.

The solution was already during writing of this work successfully used in pharmaceutical area which is illustrated in some contained case studies.

Keywords: planogram, merchandising, category management

# 1 Úvod

Rozmístění produktů v prodejně je jedním z nejdůležitějších faktorů ovlivňujících její úspěšnost. Je třeba, aby zákazník snadno našel to co hledá, nebo aby narazil na něco, co si koupí, i když to neměl původně v úmyslu.

Není tedy žádným překvapením, že většina prodejců, distributorů a výrobců věnuje rozmisťování produktů velkou pozornost a že existují celá komplexní odvětví věnovaná této problematice jako jsou merchandising a category management.

## 1.1 Merchandising

Pojem merchandising viz. [1], [2] v sobě zahrnuje velké množství činností jejichž cílem je zajistit co možná nejlepšího rozmístění produktů v prodejnách. Mezi tyto činnosti patří například průzkumy zabývající se chováním zákazníků, plánování rozmístění produktů na základě faktorů ovlivňujících jeho kvalitu, implementace připraveného plánu v samotné prodejně a doplňování produktů, aby nevznikala prázdná místa.

## 1.2 Category management

Category management viz. [3] úzce souvisí s merchandisingem. Jeho hlavním cílem je vytvoření jakési hierarchie produktů v níž na nejvyšší úrovni jsou kategorie, které se skládají ze subkategorií. Subkategorie se skládají z brandů a brandy z jednotlivých produktů. Řádově se jedná obvykle o desítky kategorií.

Příkladem kategorie je například pečivo, dámská kosmetika, vitamíny nebo mléčné výrobky. Příkladem subkategorie mléčných výrobků jsou jogurty, sýry nebo mléčné nápoje. Příkladem brandu jsou například jogurty Jogobella.

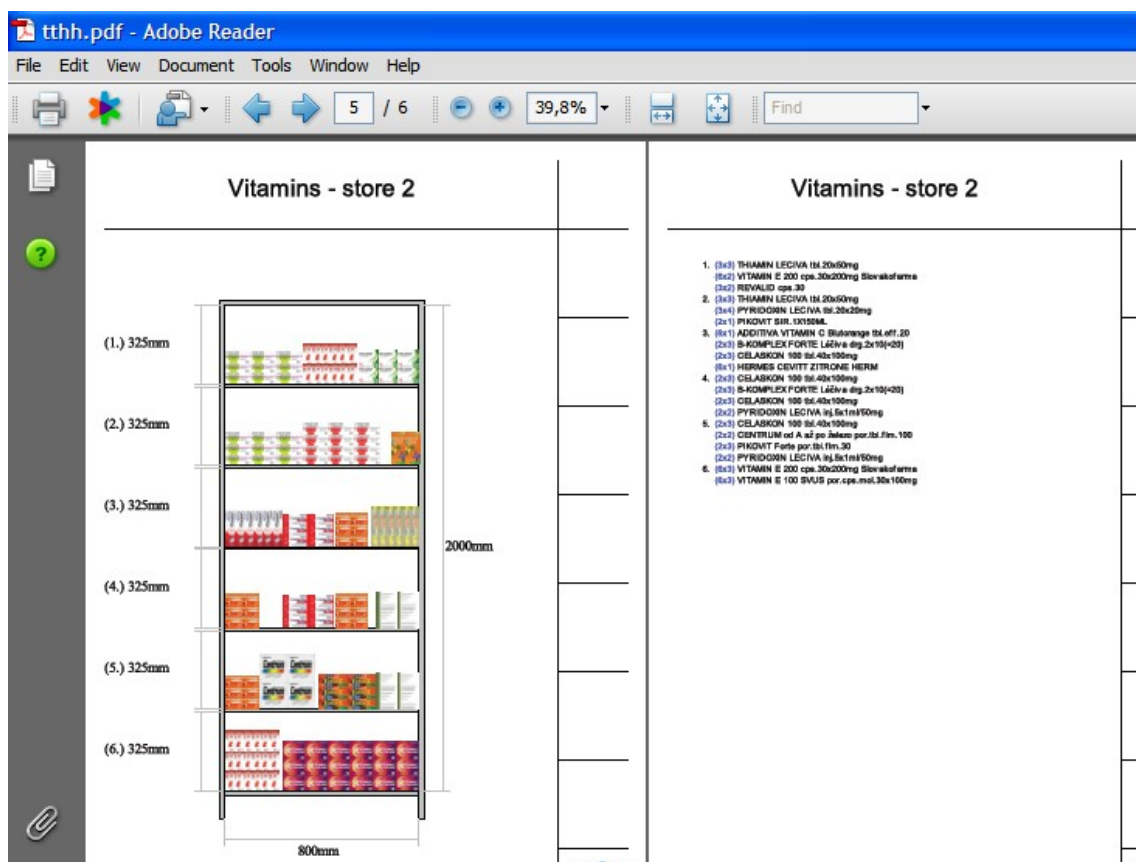
V rámci této hierarchie se pak analyzují a zkoumají data o jednotlivých úrovních, vytvářejí se pořadí důležitosti a pod.

Na základě analýz prováděných v rámci category managementu se merchandiseři rozhodují, jaké produkty umístit a jakým způsobem to provést. Produkty, které k sobě mají blíže v hierarchii jsou umístěny blíže k sobě i v prodejně a produkty, které jsou důležitější dostávají přednost před produkty méně důležitými.

## 1.3 Planogramy

Asi nejoblíbenější a nejčastěji využívanou technikou při plánování rozmístění je

tvorba planogramů viz.[1]. Planogram je nákres skříně obsahující produkty často doplněný ještě písemným seznamem viz. Ilustrace 1.



Ilustrace 1: Příklad planogramu vytvořeného v systému QuantEE

Hlavní výhodou planogramu je v jeho srozumitelnosti a snadné distribuci. Planogram je možno jednoduše zaslat prodejně vytištěný poštou, e-mailem nebo vystavit na webových stránkách. Pokud jsou planogramy připraveny přímo na míru dané prodejny, zvládne jejich realizaci běžný zaměstnanec. Problematikou tvorby planogramů na míru se zabývá tato diplomová práce.

## 2 Popis problému a cílů práce

### 2.1 Kdo a proč vytváří planogramy

Není obvyklé, aby planogramy vytvářely samostatné nezávisle prodejny, jelikož si nemohou dovolit investovat do vlastních analýz dat o prodejkách, nemají data z jiných podobných prodejen, nemají prostředky na provádění průzkumů chování zákazníků ani na nákup specializovaného softwaru.

Planogramy obvykle vznikají v kancelářích řetězců, větších kooperací, distributorů, výrobců, nebo firem specializovaných přímo pouze na merchandising. Tvorba planogramů, je jedním z nejpoužívanějších způsobů plánování a zajištění správné prezentace produktů v prodejkách. Správná prezentace produktů v prodejkách je dle studie [4] vnímána jako jedna z největších výhod v konkurenčním boji. Tedy tvorba planogramů je pro mnoho společností, jednou z klíčových činností.

#### 2.1.1 Řetězce

Řetězce obvykle disponují velmi podrobnými daty o prodejkách ve svých prodejkách. Prodejny navíc obvykle bývají velmi podobné co se týče prostoru a vybavení, což tvorbu planogramů na míru značně usnadňuje. Není totiž třeba řešit příliš mnoho specifik jednotlivých prodejen.

Hlavními faktory hrajícími roli při rozhodování o tom, které produkty vystavit, bývají vlastní data o prodejkách a obchodní strategie řetězce daná například specifickými dohodami s výrobcí, nebo snahou utvářet určitý obraz o sobě u potenciálních zákazníků.

#### 2.1.2 Kooperace prodejen

Kooperace prodejen představuje značně volnější koncept spolupráce skupiny prodejen, než řetězce. Prodejny na rozdíl od řetězců mívají značnou míru nezávislosti na vedení kooperace a vyžadují od vedení kooperace mnohem větší neutralitu. Řetězec může například realizovat strategie, které zvýší jeho výdělek jako celku, za cenu zhoršení výdělku některých svých poboček. Něco takového si vedení kooperace nemůže dovolit. Pokud tedy kooperace vytváří centrálně planogramy pro své členy, vychází obvykle z nezávislých dat o prodejkách a snaží se řešit v co možná největší míře specifika konkrétních prodejen.



### **2.1.3 Distributoři**

Distributoři často nabízejí merchandisingové služby jako jakousi přidanou hodnotu pro své zákazníky, která je má motivovat k odebrání produktů právě od nich. Velkou výhodou distributorů je, že disponují poměrně kvalitními daty o prodejkách a tento fakt využívají, jako jeden z argumentů, proč jsou jejich planogramy kvalitní a výhodné pro prodejny. Někdy distributoři vytvářejí planogramy tak, aby jejich nasazení v prodejních maximalizovalo výtěžek jejich samotných, což ne vždy koresponduje s maximalizací výtěžku prodejen. Často se jedná o vzájemně výhodnou dohodu, kdy prodejna vystavuje produkty preferované distributorem a ten tyto produkty na oplátku dodává se slevou.

### **2.1.4 Výrobci**

Výrobci obvykle nevytváří planogramy pro celé prodejny, avšak často ovlivňují planogramy vytvářené řetězci, kooperacemi nebo distributory. Častým modelem bývá „pronajmutí“ určitého prostoru, jako je několik polic nebo třeba celá skříň, v prodejně danému výrobcí. Výrobce pak vytváří vlastní planogramy pouze pro tuto část a umísťuje pouze své vlastní produkty dle svých obchodních zájmů.

### **2.1.5 Firmy specializované na merchandising**

Existují i nezávislé firmy, které nabízejí kompletní merchandisingové služby samostatným prodejnám, ale i kooperacím, či řetězcům. Firmy nabízejí zejména své zkušenosti v oboru, znalost práce se specializovaným softwarem a podobně. Pokud je jejich zákazníkem nezávislá prodejna, vyžaduje individuální, nezávislý přístup, který zlepšuje její tržby. Specializovaná firma pak buď kupuje data o prodejkách z jiných zdrojů, nebo pracuje přímo s daty konkrétní prodejny, případně sama sbírá a analyzuje data z podobných prodejen.

## **2.2 Faktory ovlivňující výběr produktů**

Abychom mohli začít tvořit planogramy, musíme mít nejprve představu, jaké produkty chceme umístit. Při rozhodování, které produkty mají pro nás při umísťování prioritu hrají roli různé faktory viz. [3], [5] a [6]. Tyto faktory je možno rozdělit do dvou skupin.

### **2.2.1 Objektivní faktory**

Objektivní faktory jsou obvykle výsledkem nějakých nezávislých statistických či sociologických studií. Patří sem zejména:

- data o prodejkách posbíraná z co možná největšího množství prodejen

- předpovědi vývoje dat o prodeji na základě dlouhodobých trendů
- probíhající, nebo plánované reklamní kampaně v masových médiích
- předpověditelné přírodní vlivy ovlivňující prodejnost specifických produktů, jako počasí (prodejnost zmrzliny stoupá v tropických vedrech) nebo zprávy o blížící se chřipkové epidemii (stoupá prodejnost vitaminů a léků proti chřipce)
- politický a ekonomický vývoj (zprávy o globální finanční krizi snižují prodejnost dražších produktů)

### 2.2.2 Subjektivní faktory

Subjektivní faktory nemají obecnou platnost, ale mohou hrát významnou roli pro konkrétní společnost. Častými subjektivními faktory jsou:

- specializace na určitý segment trhu například na luxusní značky, nebo naopak na nejlevnější, méně kvalitní produkty
- specifické dohody s určitými výrobci o prosazování některých jejich produktů
- specifické dohody s dodavateli o preferování jistých produktů díky akčním cenám
- preferování určitých produktů na základě subjektivní důvěry nebo ze zvyku
- netypická poloha prodejny, nebo jiné vlivy způsobující atypičnost klientely

## 2.3 Nástroje usnadňující tvorbu planogramů

Nakreslit planogram skříně obsahující desítky produktů s pomocí běžných kancelářských nebo grafických balíků není snadnou záležitostí. Takový postup by vyžadoval nakreslení skříně ve správných proporcích a ruční vkládání jednotlivých obrázků produktů ve správném měřítku. Uživatel by si musel sám ohlídat, aby se obrázky nepřekrývaly a nepřesahovaly hranice skříně ručním zadáváním jejich souřadnic. Vytvářet takto planogramy pro desítky, či stovky prodejen by bylo neúnosně náročné.

Merchandisingová oddělení proto téměř výhradně využívají specializovaný software, který tvorbu planogramů usnadňuje. Na trhu je k dispozici poměrně velké množství programů, které usnadňují tvorbu planogramů. Výběr sahá od poněkud inteligentnějších grafických editorů v ceně desítek euro až po komplexní softwarové balíky v ceně desítek i stovek tisíc euro řešící kromě samotné tvorby planogramů i mnoho jiných součástí merchandisingu.

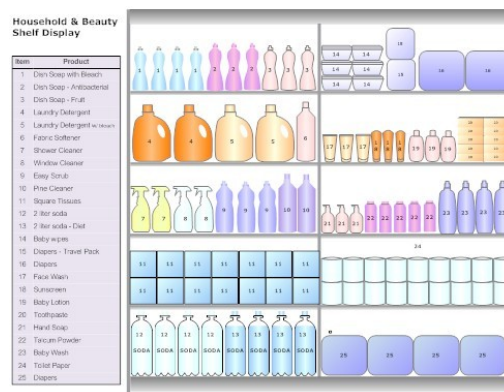
Následující přehled obsahuje nejznámější z těchto programů a stručný popis způsobu, jakým usnadňují tvorbu planogramů.

- SmartDraw je inteligentní grafická aplikace, která jako jednu ze svých předností udává podporu tvorby planogramů na základě využití velkého množství předpřipravených šablon skříní a symbolů produktů. Jak je vidět na příkladu planogramu vytvořeném v této aplikaci viz. Ilustrace 2, jedná se spíše o symbolický planogram, který neobsahuje přesné obrázky produktů a nepracuje ani s přesnými rozměry.

Takovéto planogramy mohou mít své opodstatnění pro zobrazení určitého prvotního záměru, při plánování rozmístění nábytku v prodejně nebo při konzultacích merchandisingového specialisty přímo na místě v prodejně. Pro práci velkého merchandisingového oddělení, které potřebuje pracovat s přesnými rozměry a obrázky reálných produktů je však tento software absolutně nedostatečný.

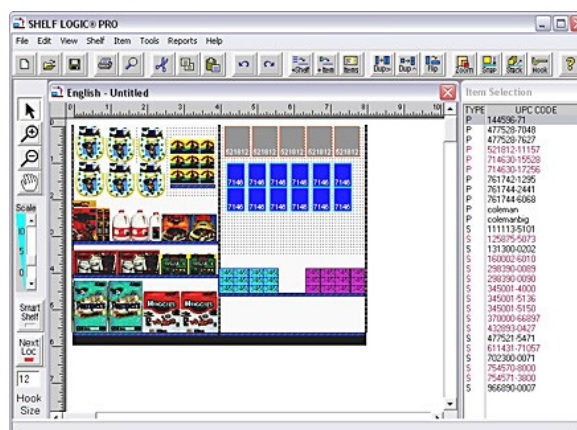
Cena: 297\$ za licenci pro jednoho uživatele

<http://www.smartdraw.com>



Ilustrace 2: Planogram ve SmartDraw

- ShelfLogic je oproti SmartDraw podstatně pokročilejším nástrojem umožňujícím tvorbu planogramů se skutečnými rozměry a obrázky produktů. Program umožňuje vytvoření databáze produktů, se kterými je následně možno pracovat, což pochopitelně značně šetří čas při vyhledávání a vkládání obrázků do planogramu. Uživatel nejprve zadá rozměry skříně a polic. Samotné vložení obrázku funguje na principu drag and drop - produkt je přetažen ze seznamu, do příslušné police. viz. Ilustrace 3. S vloženým obrázkem je



Ilustrace 3: Prostředí programu ShelfLogic

posléze možno provádět některé užitečné operace jako nastavení počtu kusů vedle sebe a nad sebou. Velmi užitečnou vlastností je, že program dokáže automaticky ohlídat, aby se obrázky nepřekrývaly. K zakreslenému planogramu, je možné automaticky vygenerovat seznam obsažených produktů s doplňkovými údaji.

Cena: 1000\$ za licenci pro jednoho uživatele

<http://www.shelflogic.com>

- my.spaceman je velký balík nástrojů, který se snaží řešit víceméně všechny aspekty merchandisingu. Jedná se o velmi rozšířený systém, používaný mnoha merchandisingovými oddělení velkých řetězců, či distributorů. Jako ostatní podobné balíky poměrně úzkostlivě tají podrobnější specifikace svých funkcí i cenu, která bývá obvykle stanovena individuální dohodou. Součástí balíku je dle marketingových materiálů i podpora tvorby planogramů specifických pro danou prodejnu a automatizace tvorby planogramů.

Dle zkušeností několika velkých merchandisingových oddělení, se kterými jsem v rámci své práce ve firmách ExTech s.r.o. a QuantPharma GmbH spolupracoval na vývoji nových merchandisingových systémů, je my.spaceman vhodný spíše pro řetězce velmi podobných prodejen např. supermarkety. Systém například dokáže při tvorbě planogramů napovídat, které produkty je nejvhodnější umístit a v jakém množství na základě posbíraných dat o prodejkách, nicméně není příliš dobře schopný brát v úvahu zejména subjektivní faktory popsané v odstavci 2.2.2. Tvorba

planogramů na míru pro víc jak 50 prodejen s různými rozměry skříní v tomto systému je příliš náročná na čas a lidské zdroje a proto i tento systém obvykle nakonec slouží spíše k tvorbě standardních planogramů.

Cena: Individuální. Dle zkušeností 20 000 EUR základní zavedení systému v merchandisingovém oddělení, 7000 EUR za každého uživatele, 6000 EUR za rok technické podpory.

<http://www.nielsen.com>

- Apolo designer workstation, Intactic, Galleria a další patří do rodiny balíků s velmi podobnými charakteristikami, jako my.spaceman. Konkrétní vlastnosti, podmínky a cena u těchto produktů vznikají na základě konkrétní dohody a způsobu nasazení v merchandisingovém oddělení. Jejich kvalita automatické produkce planogramů na míru prodejnám zřejmě závisí od schopnosti merchandisera nastavit komplexní pravidla pomocí podporovaných nastavení, či skriptovacích jazyků.

Tyto systémy společně s my.spaceman dle [3] patří v současnosti k nejpoužívanějším řešením v oblasti merchandisingu a category managementu.

<http://www.retailsmart.com>, <http://www.jda.com>, <http://www.galleria-rts.com>

## 2.4 Nedostatky existujících nástrojů

Obvyklá praxe velkých merchandisingových oddělení, se kterými jsem měl možnost spolupracovat (zejména v oblasti tvorby planogramů pro české a německé lékárny), ukázala, že není možné pomocí existujících nástrojů docílit plně automatické tvorby planogramů na míru jednotlivých prodejen tak, aby výsledné planogramy byly vyhovující.

Důvodem je zejména to, že tyto nástroje obvykle při automatickém generování nejsou schopny postihnout některá specifika a zvyky daného odvětví. Generování probíhá obvykle na základě dat o prodejnách a v pokročilejších případech může být ovlivněno ještě jakýmsi uživatelsky definovanými skripty. Problém však bývá zejména se subjektivními faktory ovlivňujícími výběr produktů, případně se zvyky některých prodejen, které například vyžadují umístění několika důležitých produktů na určité konkrétní místo.

Vygenerované planogramy také obvykle neřeší estetické faktory, o kterých bude řeč v odstavci 4.3. Těmto faktorům mnozí merchandiseři přikládají velký význam.

Úprava automaticky vygenerovaného planogramu pak i v pokročilých merchandisingových systémech znamená nutnost ručních úprav v editoru s logikou programu ShelfLogic popsanou v odstavci 2.3. Pokud například chceme ve všech automaticky vygenerovaných planogramech nahradit produkt A produktem B, musíme tuto úpravu provést postupně ve všech planogramech ručně a sami se musíme vypořádat s problémy vzniklými z různých velikostí produktů A a B. Je-li například produkt B větší, než produkt A, nemusí být možné jej ve všech skříních umístit ve stejném počtu kusů, jako byl původně umístěný produkt A.

Tvorba planogramů na míru prodejnám, tak často vyžaduje značné lidské zdroje, a proto mnohá merchandisingová oddělení na jejich tvorbu rezignovala a spokojila se s vytvářením tzv. standardních planogramů.

## 2.4.1 Standardní planogramy a jejich nevýhody

Standardními planogramy rozumíme sadu planogramů pro skříně určitých standardních rozměrů. Pomocí standardních planogramů, řeší mnohá merchandisingová oddělení přílišnou náročnost vytvoření velkého množství (stovek či tisíců) přesných planogramů během dostatečně krátké doby. Na vytvoření a distribuci nové sady planogramů má merchandisingové oddělení mnohdy pouze několik dní. Vytvoří proto například pro každou kategorii produktů pouze čtyři planogramy odpovídající zvoleným čtyřem standardním rozměrům skříní a prodejna obdrží buď celou sadu planogramů, nebo v lepším případě pouze ty, které jsou rozměry nejbližší jejím reálným skříním.

Velkou nevýhodou standardních planogramů je, že **přenášejí část rozhodování o rozmístění produktů ve skříní na potenciálně nekvalifikovaný personál prodejny**. Stává se pak například, že prodejna nemá ve své skříní dostatek místa na všechny produkty obsažené ve standardním planogramu a rozhodne se neumístit některý z důležitých produktů. Produkt může být důležitý například proto, že za jeho vystavení v prodejnách merchandisingové oddělení vyjednálo výhodnější cenové podmínky u některého z výrobců, o čemž nemá zaměstnanec prodejny ani ponětí.

Standardní planogramy jsou často také vnímány negativně prodejny, jelikož **nevytvářejí pocit individuálního přístupu**. Prodejna, která obdržela standardní planogram si uvědomuje, že stejný planogram obdržela spousta jiných prodejen a nemůže tedy odrážet lokální specifika.

## 2.5 Cíle práce

Přestože tvorba planogramů má své uplatnění pro všechny možné typy prodejen a zboží, je tato práce zaměřena pouze na tvorbu planogramů pro klasické skříně s policemi, které jsou nejčastějším typem nábytku v prodejnách s rychlo obrátkovým zbožím.

Práce si klade následující cíle:

- Provést obecnou analýzu možností, jak vylepšit koncept tvorby standardních planogramů tak, aby byly pokud možno odstraněny nevýhody uvedené v odstavci 2.4.1.
- Vytvořit návrh řešení v podobě jakéhosi univerzálního planogramu, který by náročností svého vytvoření víceméně odpovídal náročnosti vytvoření standardního planogramu a návrh algoritmu, který by dokázal z univerzálního planogramu automaticky vytvořit pro reálné rozměry skříně planogram na míru této skříní.
- Implementovat funkční prototyp navrhovaného řešení.

Při analýze požadavků na zlepšení zavedených způsobů tvorby planogramů a návrhu vhodného řešení budu vycházet zejména ze zkušeností získaných během vývoje několika systémů zaměřených na tvorbu planogramů a podporu merchandisingových aktivit ve společnostech ExTech s.r.o. a QuantPharma GmbH. Jedná se zejména o systémy QuantEE, QuantPE a PPP, které se v době psaní této práce úspěšně prosazují zejména na českém a německém trhu především v oblasti tvorby planogramů pro lékárny.

Jako vedoucí vývoje těchto systémů, jsem měl během několika posledních let možnost

konzultovat mnoho aspektů tvorby planogramů a merchandisingu obecně s mnoha merchandisingovými odděleními a odborníky zabývajícími se tvorbou planogramů. Takto nabyté znalosti byly hlavním zdrojem a ideovým motorem při vzniku této práce.

Vzhledem k tomu, že finální text této práce vzniká v době, kdy její výsledky již našly praktické uplatnění, pokusím se ho obohatit i o některé příklady z praxe a ilustrovat na nich vhodnost či nevhodnost některých rozhodnutí.

# 3 Struktura práce

Pro snadnější orientaci v textu této diplomové práce a pro představu čtenáře, co může v následujících kapitolách očekávat, uvedu nejprve stručný přehled jejich obsahu.

- Ve čtvrté kapitole bude provedena obecná analýza vlastností planogramů relevantních pro naši snahu o vytvoření univerzálního planogramu, pomocí něž bude merchandiser schopen vyjádřit svůj záměr zejména co do vzhledu planogramu a důležitosti umístěných produktů tak, aby mohl existovat algoritmus převádějící univerzální planogram na reálný planogram. Zároveň budou diskutovány vlastnosti a funkce, které by měl poskytovat komplexní merchandisingový systém, aby v jeho rámci byla tvorba takového univerzálního planogramu možná.
- V kapitole páté bude na základě výsledků provedené analýzy vytvořen návrh řešení splňujícího vytčené cíle. Bude popsána struktura navrženého univerzálního planogramu společně s definicí korektnosti odpovídajícího reálného planogramu. Dále bude popsán také algoritmus převodu univerzálního planogramu na planogram reálný včetně důkazu, že výsledný reálný planogram je korektní. Dostupnost takového algoritmu je totiž klíčová pro praktické uplatnění univerzálních planogramů.
- Šestá kapitola popisuje prototypovou implementaci tvorby univerzálních planogramů na níž je možné odzkoušet funkčnost provedeného návrhu.
- V sedmé kapitole je také uveden přehled a stručný popis existujících systémů v nichž se popsáný návrh univerzálních planogramů uplatnil a několik případových studií nasazení univerzálních planogramů v praxi.
- Závěrečná kapitola shrnuje přínosy celé práce.

# 4 Analýza problému

Cílem této kapitoly je provést obecnou analýzu vlastností planogramů relevantních pro naši snahu o vytvoření univerzálního planogramu, pomocí něž bude merchandiser schopen vyjádřit svůj záměr zejména co do vzhledu planogramu a důležitosti umístěných produktů tak, aby mohl existovat algoritmus převádějící univerzální planogram na reálný planogram.

Nejprve však popíšeme nezbytné podpůrné vlastnosti a funkce, které by měl poskytovat komplexní merchandisingový systém, aby jej bylo možné rozšířit o koncept tvorby univerzálních planogramů.

## 4.1 Požadované vlastnosti merchandisingového systému

Abychom mohli na základě univerzálních planogramů vytvářet reálné planogramy pro prodejny, musí nám merchandisingový systém poskytovat zejména nezbytné informace o produktech a skříních v prodejen.

### 4.1.1 Údaje o produktech důležité pro tvorbu planogramů

Pro tvorbu přesného planogramu je třeba o každém produktu znát následující údaje:

- **Výšku a šířku.**
- **Hloubku** – některá merchandisingová oddělení se hloubkou produktů nezabývají, jelikož předpokládají, že skříně jsou dostatečně hluboké, aby se do nich vešly všechny produkty a počet kusů produktu za sebou pro ně není důležitý. My však budeme brát hloubku produktů důsledně v úvahu, abychom podporovali i situace, kde je důležitý počet umístitelných kusů produktů za sebou například kvůli plánování skladu.
- **Obrázek resp. fotografii.**
- **Rotovatelnost produktu** - příznak udávající, zda je produkt možné umístit pouze v základní poloze, kdy je šířka vodorovný rozměr, nebo, zda je možné produkt umístit i v poloze otočené o 90 stupňů. Přítomnost tohoto údaje nám umožní automaticky umisťovat produkt do otočené polohy, pokud ho není možné umístit v poloze základní.
- Doplnkové údaje uváděné v automaticky generovaném seznamu produktu doprovázejícím planogram jako **id, jméno, typ, velikost balení** apod.

Běžné merchandisingové systémy obsahují databázi produktů s většinou vyjmenovaných údajů i nástroje pro její vytvoření, jako jsou různé editory produktů a importní utility.

Samozřejmostí nebývá pouze informace o rotovatelnosti produktů. Její absence však neznamená vážný problém, pouze ztrátu možnosti aplikace poněkud komfortnějšího řešení.



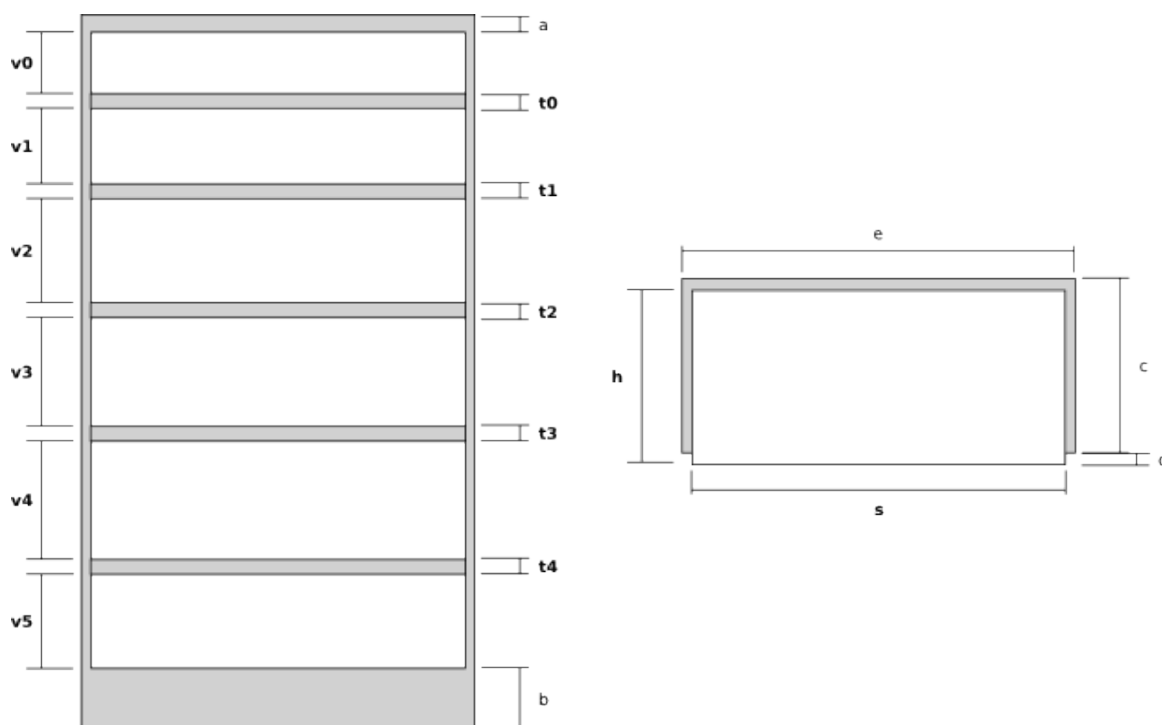
Kromě zmíněných údajů o produktech, které jsou pro tvorbu planogramů víceméně nezbytné, poskytují merchandisingové systémy obvykle i další podpůrné informace o produktech, které merchandiserovi pomáhají při rozhodování, které produkty je vhodné umístit a kam. Mezi tyto údaje patří zejména rozčlenění produktů do kategorií, subkategorií a brandů. Data o prodejích a pod.

#### 4.1.2 Údaje o skříních

Pokud chceme vytvořit přesný planogram pro určitou skříň, musíme znát zejména následující rozměry skříně:

- Vnitřní šířku, která je společná pro všechny police.
- Vnitřní hloubku, která je společná pro všechny police.
- Výšky jednotlivých polic v případě, že tyto jsou ve skříní fixní.
- Seznam pozic, do nichž je možno desky umístit společně s údajem o počtu dostupných desek pro danou skříň v případě, že pozice desek ve skříních nejsou fixní. Existují totiž speciální typy skříní s lehce nastavitelnými polohami desek. V takovém případě se předpokládá, že si merchandiser při tvorbě planogramů zvolí výšky polic, které mu budou nejlépe vyhovovat.
- Tloušťky desek oddělujících police.

Následující schéma viz. Ilustrace 4 zobrazuje poněkud bohatší sadu rozměrů skříní, která je schopna lépe modelovat většinu reálně používaných typů skříní s policemi. Nezbytné rozměry jsou ve schématu zvýrazněny tučně:



Ilustrace 4: Schéma modelující podporovaný typ skříní

## 4.2 Obecně platná pravidla rozmístování produktů

Pro umístování produktů do skříní platí určité obecné zákonitosti, které je třeba brát v úvahu při tvorbě planogramů. Jedná se o základní pravidla, jejichž dodržování lze předpokládat v drtivé většině situací. Jejich znalost by nám měla poskytnout základní rámec vlastností, které by měl podporovat hledaný univerzální planogram. Následuje výčet nejdůležitějších z nich.

### 4.2.1 Produktbloky

Produkty se ve skříních umísťují v jakýchsi konzistentních blocích tvořených několika kusy stejného produktu.

*V dalším textu bude pro tyto bloky použit pro jednoduchost výraz produktblok.*

Produktblok má následující základní charakteristiky:

- **Polohu ve skříní**, kterou je možno vyjádřit jednoduše jako číslo police a vzdálenost od levého okraje, jelikož je jasné, že dno produktbloku odpovídá dnu police a první řada produktů v bloku je maximálně ve předu tak, aby nepřesahovala ven ze skříně.
- **Šířkou** odpovídající počtu kusů produktu vedle sebe.
- **Výškou** odpovídající počtu kusů produktu nad sebou.
- **Hloubkou** odpovídající počtu kusů produktu za sebou.
- **Produkt**, jehož kusy tvoří obsah.

### 4.2.2 Jedinečná poloha produktu ve skříní

Produkty mají v rámci jedné skříně jedinečnou polohu. Nestává se, aby se mezi dvěma kusy stejného produktu vyskytoval nějaký jiný produkt. Produkty se umísťují v blocích, aby je zákazníci jednodušeji objevili. Umístění stejného produktu na různá místa ve skříní by vedlo k dezorientaci zákazníků a plytvání místem i doplňkovými informačními materiály, jako jsou cenovky.

Ve velkých prodejnách se občas stejný produkt objeví na více různých místech, nikoliv však v rámci jedné skříně. Obvykle se jedná o nějaký velmi významný produkt, který by zákazníci neměli přehlédnout, nebo o produkt v akci, který je krom svého obvyklého umístění vystaven ještě ve speciální skříní určené výhradně pro produkty v akci.

### 4.2.3 Uchopitelnost produktu

Produkty se z pochopitelných důvodů umísťují tak, aby je bylo možné bez větších

problémů vyndat z police. Obvykle k tomu stačí zachovat dostatečnou mezeru mezi horním okrajem produktbloku a stropem police tak, aby bylo možné vyndat jeden z kusů umístěný v nejvyšší řadě produktbloku. Ve výjimečných případech je třeba mezi jednotlivými kusy zachovat i mezery ve vodorovném směru, častěji však jsou jednotlivé kusy umísťovány těsně vedle sebe, aby se ušetřilo vzácné místo a zároveň zvýšila stabilita produktbloku, zejména jsou-li jednotlivé kusy vyskládány i nad sebou.

#### 4.2.4 Nosnost produktu

Ne všechny produkty jsou baleny tak, aby jejich kusy bylo možno stavět v policích jeden na druhý, nebo je nosností balení omezena výška stavěného sloupce. Při tvorbě planogramů je třeba s tímto faktem počítat. Nosností produktu a požadavkem na jeho uchopitelnost bývá obvykle dána výška produktbloku v dané polici.

### 4.3 Pokročilé vizuální efekty využívané v planogramech

Některá merchandisingová oddělení se snaží kvalitu planogramů zvyšovat i určitými pokročilými vizuálními efekty, které mají obvykle za cíl zvětšit přehlednost, estetickou kvalitu nebo viditelnost určitých důležitých produktů. V následujících odstavcích budou popsány nejdůležitější z nich, jelikož náš univerzální planogram by v ideálním případě měl využití těchto efektů podporovat.

#### 4.3.1 Vertikální produktbloky

V odstavci 4.2.1 jsme popsali produktbloky a předpokládali jsme, že jeden konkrétní produktblok je umístěn pouze v jedné polici. Při zkoumání postupů využívaných některými zkušenými merchandisery se ukazuje, že tomu tak nemusí být vždy. Existuje totiž teorie, že umístění více užších produktbloků ve více policích pod sebou je lepší než umístění jednoho širokého produktbloku v jediné polici. O využití vertikálních bloků a důvodech jejich užitečnosti se lze dočíst například v [5] a [7].

*V dalším textu budeme takovému umístění říkat vertikální produktblok.*

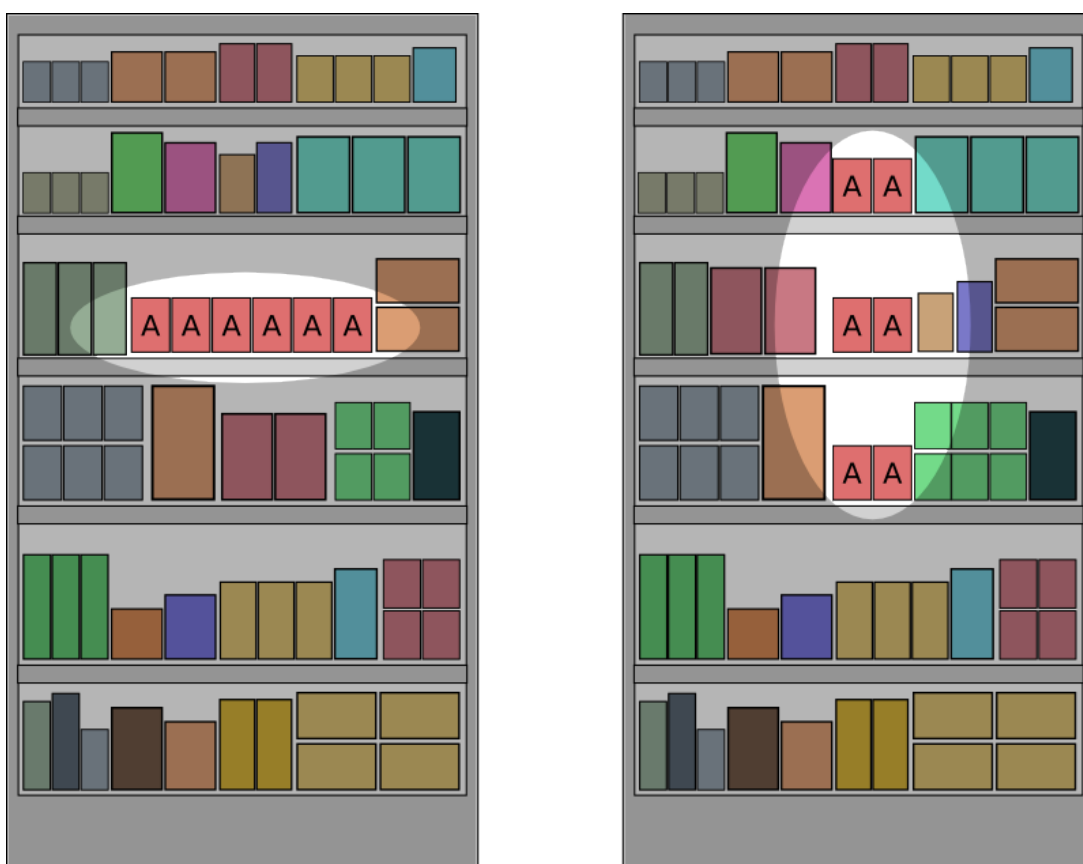
Vertikální produktblok se skládá z několika stejně širokých produktbloků umístěných v několika policích pod sebou ve stejné vzdálenosti od levého okraje skříně. Jeho základními charakteristikami jsou:

- **Poloha ve skříní**, kterou je možno vyjádřit nejvyšší polici v níž je první z obsažených produktbloků a vzdáleností od levého okraje skříně.
- **Délka** udávající, přes kolik polic se vertikální produktblok rozprostírá.
- **Šířka** společná pro všechny obsažené produktbloky.
- **Výšky obsažených produktbloků**, které se liší obvykle pouze v případě různých výšek polic. Obvykle jsou dány zejména nosností a uchopitelností produktu.

- **Hloubka** společná pro všechny obsažené produktbloky.
- **Produkt**, jehož kusy tvoří obsah všech produktbloků tvořících tento vertikální produktblok.

Je důležité uvědomit si, že umístění více stejně širokých produktbloků pod sebou neporušuje podmínku jedinečné polohy produktu ve skříni viz. 4.2.2. Také si můžeme všimnout, že produktblok, jak jsme ho popsali v odstavci 4.2.1 je vlastně speciálním případem vertikálního produktbloku, jehož délka je rovna jedné.

Levá skříň v Ilustraci 5 ukazuje umístění produktu A klasickým způsobem v jediné polici ve vertikálním bloku délky 1. Skříň napravo ukazuje příklad využití vertikálního produktbloku délky 3 pro umístění produktu A ve stejném počtu kusů.



*Ilustrace 5: Srovnání horizontálního a vertikálního produktbloku*

Rozmístění, jaké je vidět v pravé skříni využívají někteří merchandiseři k zviditelnění produktu A. Pro zákazníka je pak jednodušší produkt nalézt, případně si jej všimne, i když jej nehledal a zvýší se tak pravděpodobnost tzv. impulzivního nákupu tj. nákupu produktu, který zákazník původně vůbec koupit neplánoval.

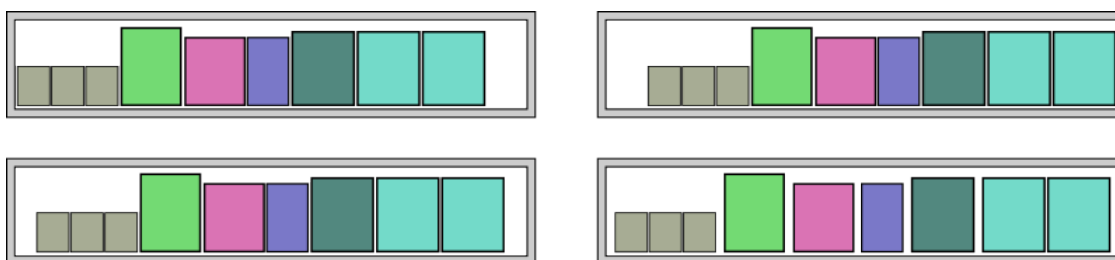
Využívání vertikálních produktbloků délky větší než jedna má skalní zastánce i skalní odpůrce, kteří tvrdí, že jde o zbytečné plýtvání místem, jelikož využití většího množství takových produktbloků zvětšuje podíl nevyužitého místa. Pro naše účely není příliš důležité, kdo z nich má pravdu. Podstatné je, že existuje nezanedbatelné množství merchandisingových oddělení viz. [5] a [7], které při tvorbě planogramů vertikální produktbloky využívají a náš návrh univerzálního planogramu by tento fakt měl brát v úvahu a vertikální produktbloky podporovat.



- zarovnání produktbloků k levému okraji police
- zarovnání produktbloků k pravému okraji police
- zarovnání produktbloků na střed, tak aby mezi produktbloky a okraji police byla na obou stranách stejná mezera
- rovnoměrné rozprostření produktbloků v polici

Obvykle se tedy při zarovnávání pracuje s produktbloky jako s celky. Vytváření mezer mezi jednotlivými kusy produktů nesmí totiž být výsledkem nějakého automatického zarovnávacího procesu, jelikož by mohlo vést k narušení stability jednotlivých kusů vyrovnaných nad sebou.

Ilustrace 7 ukazuje zmíněné typy zarovnání:



*Ilustrace 7: Typy zarovnání produktbloků*

Je poměrně překvapivé, že velké množství standardních planogramů, se kterými jsem měl možnost se setkat, používalo pro všechny police zarovnání k levé straně, přesto, že v reálných skříních prodejen se s takovýmto zarovnáním produktů téměř nesetkáme. Mohu pouze odhadovat, že tvůrci takovýchto planogramů prostě neměli k dispozici nástroj, který by by produktbloky v planogramu rozprostřel rovnoměrně automaticky. V reálných skříních je totiž zdaleka nejčastější rovnoměrné rozprostření produktbloků. Setkáme se i se zarovnáním na střed.

## 4.4 Výhody a nedostatky standardních planogramů

Myšlenka tvorby standardních planogramů vznikla z důvodu snahy omezit úsilí, které je nutné vynaložit na vytvoření planogramů tak, aby nerostlo s rostoucím počtem prodejen. Postup je založený na definování konečného množství standardních skříní se standardními rozměry.

Koncept standardních planogramů vychází ze správného předpokladu, že planogramy pro skříně podobných rozměrů jsou podobné. Podobnost však není identita a standardní planogram v sobě neobsahuje informaci, jak se vyrovnat s problémy plynoucími z rozdílů v rozměrech standardní skříně a reálné skříně. V důsledku tohoto rozdílu může docházet zejména k níže popsaným nepříjemnostem.

#### 4.4.1 Příliš široký produktblok

Je-li reálná skříň užší, než standardní skříň může se stát, že není možné umístit všechny produktbloky ve stejné šířce, jako je tomu ve standardním planogramu. V lepším případě je tuto situaci možné řešit pouhým zúžením některých bloků, v horším případě není možné některý z produktbloků vůbec umístit.

Při řešení této situace by se ideálně měly vzít v úvahu zejména následující pravidla:

- Je-li třeba zúžit některý z produktbloků v polici, měl by se zúžit ten produktblok, který má nejmenší důležitost.
- Produkt blok nemusí být možné zúžit, jelikož by přestal být dostatečně viditelný.
- Není-li možné některý z produktbloků umístit vůbec, je třeba, aby se zvolil produktblok s co možná nejmenší důležitostí.
- V některých případech lze situaci vyřešit otočením produktbloku o 90 stupňů.

#### 4.4.2 Příliš vysoký produktblok

Police reálné skříně může být nižší než police skříně standardní a může se tedy stát, že do reálné skříně není možné umístit produktblok tak vysoký jako je tomu ve standardním planogramu, případně nemusí být možné daný produkt umístit vůbec.

Při řešení této situace by se ideálně měly vzít v úvahu zejména následující pravidla:

- Je třeba dbát, aby byla zachována uchopitelnost produktbloku. Pokud lze dosáhnout uchopitelnosti snížením výšky produktbloku, mělo by se toto opatření učinit.
- V situaci, kdy není možné uchopitelnosti dosáhnout snížením výšky produktbloku, lze zvážit otočení o 90 stupňů. To však nesmí mít za následek vytlačení nějakého důležitějšího produktbloku.

#### 4.4.3 Nevyužité místo ve skříní

Je-li police reálné skříně širší nebo vyšší, než odpovídající police standardní skříně, může vzniknout nevyužité místo, které vytváří špatný dojem. S tímto problémem si pracovník prodejny obvykle poradí jednoduše zvětšením některých produktbloků, nicméně může se stát že nezvolí nevhodnější možnou variantu.

#### 4.2.4 Těžko realizovatelné vizuální efekty

Pokud není možné v důsledku rozdílných rozměrů reálné a standardní skříně docílit stejného rozmístění produktů jako ve standardním planogramu a jsou provedeny některé úpravy, můžou v důsledku zaniknout vizuální efekty popsané v odstavci 4.3. Snaha o zachování vizuálních efektů metodou pokusu a omylu by byla příliš náročná pro pracovníky

prodejny. Není tedy možné očekávat jejich dodržení v případě, že prodejna obdrží standardní planogram, který není zcela odpovídající reálné situaci.



# 5 Návrh řešení

V této kapitole bude nejprve popsán návrh univerzálního planogramu vycházející z analýzy provedené v kapitole předchozí. Návrh se bude snažit splnit zejména následující cíle:

- Vytvoření univerzálního planogramu by mělo být řádově stejně náročné, jako vytvoření standardního planogramu.
- Mělo by být možné univerzální planogramy vytvářet manuálně i automaticky například na základě dat o prodejkách.
- Univerzální planogram by měl obsahovat dostatek informací, aby na jeho základě bylo možné automaticky generovat reálné planogramy respektující obecně platná pravidla rozmístování produktů popsaná v odstavci 4.2.
- Univerzální planogram by měl podporovat vizuální efekty popsané v odstavci 4.3.

Hlavní část kapitoly bude věnována návrhu rozmístovacího algoritmu, který bude splňovat následující cíle:

- Na základě univerzálního planogramu automaticky vytvoří reálný planogram pro danou reálnou skříň.
- Univerzální planogram bude možno kdykoliv změnit a na základě provedených změn bude automaticky generován reálný planogram tyto změny odrážející, tak aby uživatel měl okamžitou zpětnou vazbu o výsledném efektu provedených změn.
- Výsledný planogram nebude trpět nedostatky standardního planogramu popsanými v odstavci 4.4.
- Krom samotného planogramu bude výsledkem algoritmu také informace o produktech obsažených v univerzálním planogramu, které nebylo možné umístit v planogramu reálném a důvodu.

V neposlední řadě bude formálně definován korektní a optimální výsledek rozmístovacího algoritmu a bude proveden důkaz, že výsledek navrženého algoritmu je optimální.

## 5.1 Univerzální bloky

Jak již víme, jsou základním stavebním kamenem reálných planogramů vertikální produktbloky viz. 4.3.1. V univerzálním planogramu proto zavedeme abstrakci produktbloku nazvanou univerzální blok. Univerzální blok bude mít komplexnější sadu atributů, než produktblok, aby pomocí nich bylo možné vyjádřit všechny důležité informace. Univerzální blok bude pochopitelně základním stavebním kamenem univerzálního planogramu.

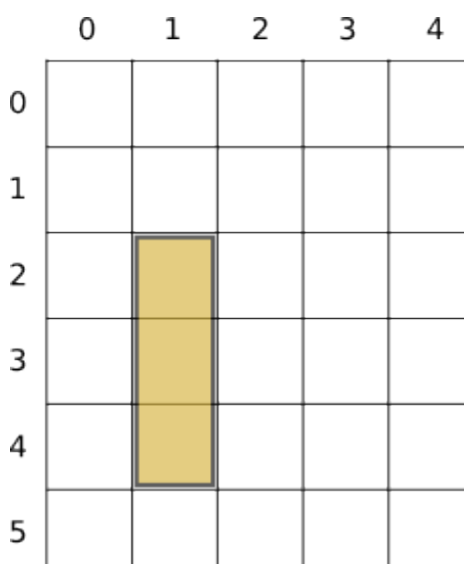
### 5.1.1 Poloha a délka univerzálního bloku v rozmíst'ovací mřížce

Poloha produktbloku je pochopitelně jedním z jeho nejdůležitějších atributů. Jelikož je cílem, aby univerzální planogram mohl sloužit pro více skříní různých rozměrů, zejména vnitřních šířek a počtů polic, není možné zadávat polohu univerzálního bloku fixně tj. v jednotkách délky. Z tohoto důvodu zavedeme koncept tzv. rozmíst'ovací mřížky.

Rozmíst'ovací mřížka je v podstatě jednoduchá tabulka, jejíž řádky odpovídají policím a sloupce poloze v rámci police. Poloha univerzálního bloku je pak dána souřadnicí v rozmíst'ovací mřížce tj. číslem příslušného řádku a sloupce. Počet řádků a sloupců mřížky volí uživatel v závislosti na jeho konkrétních potřebách. Vzájemnou polohou univerzálních bloků v rozmíst'ovací mřížce tak uživatel definuje vzájemnou polohu produktbloků v reálném planogramu odvozeném od daného univerzálního planogramu.

Délka univerzálního bloku, daná počtem po sobě jdoucích řádků rozmíst'ovací mřížky v nichž se univerzální blok nachází, udává, jakou délku by měl mít odpovídající vertikální produktblok.

Ilustrace 8 zobrazuje univerzální blok délky 3 umístěný na souřadnicích [1, 2] v rozmíst'ovací mřížce se šesti řádky a pěti sloupci.



Ilustrace 8: Souřadnicový systém rozmíst'ovací mřížky

### 5.1.2 Mapování polohy z rozmíst'ovací mřížky na polohu v planogramu

Nechť univerzální bloky A, B mají souřadnice v rozmíst'ovací mřížce  $[A_r, A_s]$ ,  $[B_r, B_s]$ , kde  $A_r$ ,  $B_r$  jsou čísla řádků a  $A_s$ ,  $B_s$  čísla sloupců a délky  $A_d$  a  $B_d$ , pak pro polohu odpovídajících produktbloků  $A_p$ ,  $B_p$  v reálném planogramu platí:

- Jestliže mají A a B alespoň jeden společný řádek tj.  $(A_r, A_{r+1}, \dots, A_{r+A_d-1})$  průnik  $(B_r, B_{r+1}, \dots, B_{r+B_d-1})$  je neprázdný, pak  $A_p$  je vlevo od  $B_p$  v reálném planogramu právě když A je vlevo od B v rozmíst'ovací mřížce tj. když  $A_s < B_s$ .
- Jestliže A a B společný řádek nemají, vzájemná poloha  $A_p$  a  $B_p$  není definovaná tj.

není důležité, zda nakonec A bude vlevo od B nebo vpravo.

Důležitou vlastností univerzálního planogramu je, že se univerzální bloky v rozmístovací mřížce nemohou překrývat tj. neexistuje buňka tabulky náležící víc jak jednomu univerzálnímu bloku.

### **5.1.3 Priorita**

Univerzální planogram musí nějakým způsobem řešit situaci, kdy ve skříní není dostatek místa (skříň je příliš úzká), na to aby se vešly všechny produktbloky odpovídající obsaženým univerzálním blokům a další situace, kde je třeba znát důležitost produktbloků.

Proto je dalším atributem univerzálního bloku priorita udávající jeho důležitost. Rozmístovací algoritmus pak musí zajistit, že se nestane, aby byl v reálném planogramu umístěn produktblok odpovídající univerzálnímu bloku s nižší prioritou na úkor produktbloku odpovídajícímu univerzálnímu bloku s vyšší prioritou.

### **5.1.4 Minimální počet viditelných kusů**

Specifikováním minimálního požadovaného počtu viditelných kusů pro daný univerzální blok uživatel definuje, kolik kusů daného produktu se má minimálně objevit v reálném planogramu. Do tohoto počtu se ovšem dle merchandisingových zvyklostí nezapočítávají kusy nad sebou ani kusy za sebou. Tj. počet viditelných kusů u odpovídajícího vertikálního produktbloku je roven počtu kusů vedle sebe vynásobenému počtem polic v nichž se tento vertikální produktblok ve skříní nachází tj. součinu jeho šířky a délky.

Společně s prioritou patří minimální počet viditelných kusů k hlavním atributům ovlivňujícím, zda bude odpovídající produktblok v planogramu obsažen. Je-li minimální počet kusů roven třem, nemůže se stát, že se blok v planogramu objeví pouze se dvěma viditelnými kusy. V kontextu priorit to znamená, že blok s vysokou prioritou a případným vyšším minimálním počtem viditelných kusů zabere místo a některé bloky s nižší prioritou pak nebudou vůbec umístěny a to i přes to, že kdyby byl minimální počet kusů bloku s vysokou prioritou menší, vešly by se i bloky s nižší prioritou.

### **5.1.5 Maximální počet viditelných kusů**

Maximální počet viditelných kusů omezuje velikost do níž může univerzální blok narůst. V kombinaci s minimálním počtem tak je možno nadefinovat rozmezí počtu viditelných kusů, který může daný blok v planogramu dosáhnout. Omezení maximálním počtem však není povinné.

Pokud není uživatelem specifikováno jinak, měl by se rozmístovací algoritmus snažit maximálně využít místo ve skříní tak, aby bylo umístěno co možná největší množství produktbloků s co nejvyššími prioritami v minimálním požadovaném počtu viditelných kusů. Ve chvíli, kdy již není možné umístit některý dosud neumístěný blok a stále zbývá místo, může algoritmus zvýšit počty viditelných kusů u bloků, které nedosáhli maximálního počtu

viditelných kusů. Takovéto rozšiřování produktbloků může být provedeno různými způsoby. V úvahu připadá zejména rovnoměrné zvětšování nebo snaha o zachování poměru velikostí. Pro merchandisery je podstatné zejména dodržení minimálního a maximálního počtu kusů. Metodiku rozdělování nadbytečného místa nebudeme přesně specifikovat, jelikož se může lišit v závislosti na situaci. Rozmísťovací algoritmus by v ideálním případě měl podporovat více variant, tak aby si uživatel mohl zvolit, kterou chce použít pro generování reálných planogramů.

### **5.1.6 Maximální počet kusů nad sebou a vzdálenost od stropu**

Pokud to druh produktu a jeho balení dovolují, umísťují se produkty v bloku v polici nad sebou tak, aby byla výška police co nejlépe využita. Obvykle je podstatné pouze zajistit dodržení určité požadované vzdálenosti od stropu potřebné k uchopení produktu při vyndávání z police. Někdy je však třeba počet kusu nad sebou omezit například z důvodu nosnosti balení, nebo z estetických důvodů. Proto jsou dalšími atributy, které může uživatel nastavit pro každý blok v rozmísťovací mřížce, maximální počet kusů nad sebou a požadovaná minimální vzdálenost od stropu police.

### **5.1.7 Požadovaný počet kusů nad sebou**

V případě, že uživatel tvoří univerzální planogram, na jehož základě budou generovány reálné planogramy pro skříň s pohyblivými deskami, musí pro každý univerzální blok zadat požadovaný počet kusů nad sebou. Na základě hodnot tohoto atributu v jednotlivých blocích a na základě požadované minimální vzdálenosti od stropu police, pak rozmísťovací algoritmus navrhne nejvhodnější pozice pro desky ve skříni oddělující jednotlivé police tak, aby vyhověl požadavkům na bloky a přitom umístil pokud možno všechny potřebné desky.

### **5.1.8 Výplň mezi jednotlivými kusy**

Pro úplnost by mělo být možné pro daný univerzální blok specifikovat také požadovanou vzdálenost mezi jednotlivými kusy v odpovídajícím produktbloku zleva a zprava. Vytváření mezer mezi jednotlivými kusy produktu v rámci jednoho produktbloku sice není příliš časté, ale někdy může být nezbytné, aby bylo možné produkt uchopit a vyndat z police.

### **5.1.9 Příslušnost k magnetické čáře**

Každý blok náleží (je přitahován) k nějaké magnetické čáře (viz. níže). Tedy jedním z atributů bloku je id příslušné magnetické čáry. Pokud uživatel nespecifikuje, žádnou magnetickou čáru, náleží všechny bloky fiktivní magnetické čáře umístěné zcela vlevo.

Tento atribut má spíše technický ráz. Při manuálním vytváření univerzálního planogramu, by mělo být id magnetické čáry univerzálním blokům nastavováno automaticky při umístování magnetických čar do rozmístovací mřížky uživatelským rozhraním.

## 5.2 Magnetické čáry

V univerzálním planogramu zavedeme abstrakci magnetických čar popsanych v odstavci 4.3.2 poskytující rozmístovacímu algoritmu informace potřebné pro generování reálných planogramů s tímto vizuálním efektem.

Je nutné rozlišit dva druhy magnetických čar. Magnetické čáry mohou být plovoucí, nebo fixní.

Fixní pozice magnetické čáry je udána její relativní pozicí ve skříní. Například fixní čára, která je umístěna v 50% šířky skříně se do výsledného planogramu promítne tak, že přesně prostředkem dané skříně bude procházet svislá linie k níž budou z obou stran zarovnány příslušné bloky.

Nevýhodou fixních magnetických čar je, že potenciálně plýtvají s místem. Může se totiž například u zmíněné čáry procházející středem stát, že napravo ani nalevo od čáry v některé polici již není dostatek místa na umístění nějakého produktu. Nicméně v případě, že by čára byla například na 45%, tak by se využilo volné místo nalevo a vpravo by se vytvořil prostor pro umístění dalšího produktu. Výhodou je přesná kontrola nad pozicí linie.

Plovoucí magnetická čára řeší nevýhodu fixní magnetické čáry na úkor ztráty přesné kontroly nad pozicí. Ve většině případů pro uživatele není ani tak důležitá přesná pozice linie ve skříní, ale spíše samotný fakt, že jsou určité bloky k této linii zarovnány. Plovoucí magnetická čára má tedy tu vlastnost, že jí rozmístovací algoritmus může posunout z původní pozice zadané uživatelem v případě, že se tím uvolní místo pro umístění dalšího produktu. Toto chování je velice výhodné pro velké skupiny skříní, pro které je bez plovoucího chování téměř nemožné docílit efektu magnetických čar.

Magnetická čára je v univerzálním planogramu definována následujícími atributy:

- **Relativní horizontální pozice ve skříní.** Nula procent odpovídá umístění zcela vlevo, sto procent umístění zcela vpravo.
- **Id** umožňující u univerzálních bloků nadefinování, ke které magnetické čáře náleží.
- **Pozice v rozmístovací mřížce** udávající, mezi kterými dvěma sloupci se čára nachází. Pozice 0 znamená, že je čára vlevo od prvního sloupce mřížky. Pozice 1 znamená, že je čára mezi prvním a druhým sloupcem. Pozice N znamená, že je čára napravo od posledního sloupce rozmístovací mřížky s N sloupci.

Pozice magnetické čáry v rozmístovací mřížce je důležitá, aby bylo možné určit, zda má být produktblok odpovídající univerzálnímu bloku náležícímu této čáře nalevo od čáry nebo napravo.

Je třeba si uvědomit nutnost některých dodatečných podmínek na univerzální planogram v souvislosti s magnetickými čarami:

- Relativní horizontální pozice čáry A ve skříní je menší než relativní horizontální pozice čáry B ve skříní, právě když je pozice čáry A v rozmístovací mřížce menší než pozice čáry B v rozmístovací mřížce.

- Neexistují dvě čáry se stejnou pozicí v rozmístovací mřížce.
- Napravo od univerzálního bloku přitahovaného směrem doprava nesmí být ve stejném řádku univerzální blok přitahovaný doleva. Jinak řečeno nesmí se stát, aby pomyslná přitažlivost mezi jedním univerzálním blokem a magnetickou čarou působila proti přitažlivosti mezi jiným blokem a čarou.

## 5.3 Priority polic

Vzhledem k tomu, že rozmístovací mřížka slouží jako součást univerzálního planogramu tj. rozmístovacího pravidla, které má být aplikováno při tvorbě planogramů pro celou skupinu skříní, může se stát, že mřížka má různý počet řádků, než kolik má konkrétní skříň polic.

V případě, že má mřížka stejný počet řádků, kolik má skříň polic, odpovídá první řádek první polici, druhý řádek druhé atd.. V případě, že má skříň víc polic, než kolik je řádků v mřížce, znamená to, že některé z polic ve skříní mají zůstat nevyplněné. Jedná se o nejspodnější přebytečné police.

Jestliže však dojde k situaci, že má mřížka víc řádků, než kolik je polic ve skříní, znamená to, že se uživatel snaží umístit produkty do více polic, než kolik jich je v dané skříní možné. Součástí univerzálního planogramu proto je priorizace řádků rozmístovací mřížky, pomocí které uživatel specifikuje, které police, jsou nejméně důležité v případě, že není k dispozici dostatečný počet polic. Tyto police pak nebudou umístěny.

## 5.4 Zarovnání bloků

Poslední vlastností, kterou potřebujeme mít možnost definovat v univerzálním planogramu, je jaký typ zarovnání viz. 4.3. má být aplikován na produktbloky dané police. Pro každý řádek rozmístovací mřížky tedy definujeme jeden ze čtyř podporovaných typů zarovnání.

## 5.5 Rozmístovací algoritmus

Rozmístovací algoritmus, který je schopen automaticky převést záměr merchandisera vyjádřený univerzálním planogramem na planogram reálný, je nejdůležitější součástí celého návrhu. V jeho existenci totiž spočívá hlavní výhoda univerzálních planogramů. Následující text se věnuje detailnímu popisu tohoto algoritmu.

### 5.5.1 Vstup

Rozmístovací algoritmus na svém vstupu, dostane univerzální planogram a reálnou

skříň, pro kterou má algoritmus vytvořit reálný planogram. Univerzální planogram bude reprezentován následujícími datovými strukturami:

- seznam univerzálních bloků  $S_B$
- seznam magnetických čar  $S_M$
- počet sloupců rozmisťovací mřížky  $C$
- počet řádků rozmisťovací mřížky  $P$
- priority řádků rozmisťovací mřížky (mapa indexů řádků na priority)

### 5.5.2 Odstranění bloků s příliš hlubokými produkty

Před níže popsanou procedurou přípravy skříně, je třeba odstranit bloky s produkty, jejichž hloubka je větší než hloubka skříně, aby neovlivňovaly výpočty požadovaných výšek polic při rozmisťování desek u skříní s pohyblivými policiemi.

### 5.5.3 Příprava skříně

Pro další běh algoritmu, je třeba přesně určit pozice desek ve skříní. Jak bylo popsáno dříve, nemusí být přesné pozice desek zadány pro skříň v případě, že jde o skříň s pohyblivými deskami. V tomto případě známe všechny možné pozice desek a jejich výšky v daných pozicích. Algoritmus tedy nejprve musí určit správné pozice desek, tak, aby co nejlépe vyhovovaly rozmisťovacímu pravidlu.

Množství umístitelných řádek je navíc omezeno počtem dostupných desek. Pokud je tedy počet řádků v rozmisťovací mřížce větší, než počet dostupných desek + 1, odebere se nejprve přebytečný počet řádků z rozmisťovací mřížky. Pochopitelně se odebírají řádky s nejnižšími prioritami.

Pro každou řádku z rozmisťovací mřížky je třeba určit minimální výšku odpovídající police ve skříní tak, aby se do ní na výšku vešly všechny produktbloky odpovídající univerzálním blokům, které jsou na této řádce. Tuto minimální výšku určíme, jako maximum z výšek potřebných pro jednotlivé produktbloky. Výška  $h(B)$  potřebná pro daný blok  $B$  se spočítá následovně:

$$h(B) = \text{pocetKusuNadSebou}(B) \cdot \text{vyskaProduktu}(B) + \text{vzdalenostOdStropu}(B)$$

Při zjišťování výšky produktu, je třeba vzít v úvahu, zda je produkt otočen o 90 stupňů, nebo není.

Tedy potřebná výška police  $h(R)$  odpovídající řádce  $R$  se spočítá dle předpisu:

$$h(R) = \max_{B \in R} h(B)$$

Dále je třeba zajistit, že pokud nebude dostatek místa pro všechny police, budou umístěny police odpovídající řádkům rozmisťovací mřížky s nejvyššími prioritami. Algoritmus rozmisťující desky na základě požadovaných výšek polic a priorit odpovídajících řádků rozmisťovací mřížky proto funguje způsobem popsaným v následujících odstavcích.

Algoritmus hledá posloupnost  $P = (R_1, \dots, R_n)$  řádků rozmisťovací mřížky

s následujícími vlastnostmi:

1. je-li řádek  $R_x$  před řádkem  $R_y$  v rozmisťovací mřížce, pak je  $R_x$  před  $R_y$  i v  $P$
2. je možné najít ve skříni pozice desek tak, aby vznikly od shora právě police  $p_1, \dots, p_n$  takové, že  $vyska(p_x) \leq h(R_x)$
3. nejvyšší z priorit řádků neobsažených v  $P$  je minimální možná, druhá nejvyšší z priorit řádků neobsažených v  $P$  je minimální možná, ... , k-tá nejvyšší z priorit řádků neobsažených v  $P$  je minimální možná

Algoritmus postupně konstruuje posloupnost  $P$  následujícím způsobem:

```
 $P$  = prázdná posloupnost  
 $P_M = (R_1, \dots, R_m)$  je posloupnost řádků rozmisťovací mřížky od  
prvního k poslednímu  
  
while ( $P_M$  je neprázdná) {  
  //zajistí platnost bodu 3  
   $R$  = řádka s nejvyšší prioritou z  $P_M$   
   $P_M = P_M / R$   
   $P$  = rozšiř( $P$ ,  $R$ ) //zajistí platnost bodu 1  
  if (! lzeUmístit( $P$ )) { //zajistí platnost bodu 2  
     $P = P / R$  //zajistí platnost bodu 3  
  }  
}
```

Procedura rozšiř( $P$ ,  $R$ ) jednoduše přidá řádek  $R$  do  $P$  a zatřídí ho, tak, aby zůstal splněn bod číslo 1 z požadavků na hledanou množinu  $P$ .

Procedura lzeUmístit( $P$ ) testuje, zda je možné vytvořit police odpovídající řádkům v  $P$ .

Postupuje směrem nahoru, aby případné přebývajícím místo bylo v nejvyšší polici. Ze všech možných pozic desek nalezne tu nejnižší možnou, pro kterou však výška vzniklé dolní police bude větší nebo rovna  $h(R_n)$ , kde  $R_n$  je poslední řádka v  $P$ .

Následně najde nejnižší možnou pozici desky, která je ovšem nad poslední umístěnou deskou takovou, že výška vzniklé police je větší nebo rovna  $h(R_{n-1})$ , kde  $R_{n-1}$  je předposlední řádka v  $P$ .

Takto postupuje, dokud je to možné. Pokud již zbývá umístit pouze první řádku z  $P$  a výška nejvyšší police je větší nebo rovna  $h(R_1)$ , vrátí procedura lzeUmístit( $P$ ) hodnotu true, jinak vrátí hodnotu false.

Nejjednodušší situaci máme při přípravě skříň v případě, že jde o skříň s fixními deskami. Zde je pouze třeba odstranit z rozmisťovací mřížky přebytečné řádky.



### 5.5.4 Odstranění přebytečných řádků

K odstraňování řádků z rozmisťovací mřížky tedy dochází v následujících situacích:

- Jde o skříň s pohyblivými deskami a počet dostupných desek + 1 je větší než počet řádků v rozmisťovací mřížce. Odstraní se přebytečné řádky s nejnižšími prioritami.
- Během algoritmu rozmístění pohyblivých desek vznikne posloupnost řádků  $P$ , která neobsahuje některé řádky rozmisťovací mřížky. Tyto řádky se odstraní.
- Jde o skříň s fixními deskami a počet polic v této skříni je menší než počet řádků v rozmisťovací mřížce. Odstraní se přebytečné řádky s nejnižšími prioritami.

Odstranění řádku znamená zejména opravu pozic a délek bloků a opravu počtu řádků rozmisťovací mřížky. Pokud se některý blok zkrátí na nulovou délku, znamená to, že všechny řádky, ve kterých se tento blok nacházel byly odstraněny a tedy i tento blok je odstraněn ze seznamu bloků.

### 5.5.5 Odstranění bloků s příliš vysokými produkty

Ve chvíli, kdy jsou známy přesné pozice desek ve skříni a tedy i výšky polic, je třeba odstranit bloky, jejichž požadované výšky jsou příliš velké, než aby bylo možné je umístit do příslušných polic. K takovéto situaci by nemělo dojít u skříní s pohyblivými deskami, nicméně může k ní dojít u skříní s fixními deskami.

### 5.5.6 Inicializace datových struktur pro expanzi bloků

Seznam magnetických čar  $S_M$ , který dostal algoritmus na vstupu je třeba setřídít, dle relativní pozice čáry ve skříni tak, aby čáry s menší relativní pozicí tj. ty které jsou více vlevo byly před čarami s větší relativní pozicí tj. těmi, které jsou více vpravo. Po setřídění tedy bude platit:

$$\forall i, j (pozice(S_M[i]) < pozice(S_M[j]))$$

Předpokladem, o jehož zajištění se stará uživatelské rozhraní pro tvorbu univerzálního planogramu je, že

$$\forall i, j (pozice(S_M[i]) \neq pozice(S_M[j]))$$

Rozmisťovací algoritmus při svém dalším běhu vytváří tzv. reálné bloky. Reálný blok je obrazem bloku v rozmisťovací mřížce tj. bloku obsaženého v  $S_B$ . Reálný blok má souřadnice v jednotkách délky vztažené k reálné skříni a také reálné rozměry v jednotkách délky. V dalším textu necht'  $B^R$  je reálný blok odpovídající bloku  $B$ . Inicializace reálných bloků vytvoří jejich seznam  $S_{B^R}$  následujícím způsobem:

```
 $S_{B^R}$  = prázdný seznam  
for ( $B : S_B$ ) {  
    vytvoř  $B^R$  a přidej  $B^R$  do  $S_{B^R}$ 
```

```
}  
setřid'  $S_{B^R}$  dle priorit
```

Tedy reálné bloky v  $S_{B^R}$  jsou setříděné dle priorit odpovídajících bloků od nejvyšší priority po nejnižší. Při vytvoření  $B^R$ , je třeba provést patřičnou inicializaci. Pokud se v následujícím textu bude mluvit o policích v nichž leží reálný blok  $B^R$ , máme na mysli police odpovídající řádkám rozmisťovací mřížky v nichž leží  $B$ . Řádky v nichž leží  $B$  jsou pochopitelně dány jeho polohou a délkou. Při inicializaci je třeba provést níže popsané kroky.

Do  $B^R.y$  uložíme vertikální souřadnici. Souřadnicový systém má počátek v levém horním vnitřním rohu skříně. Tato výšková souřadnice je rovna výšce stropu nejvyšší police v níž leží  $B^R$ .

Do  $B^R.height$  uložíme výšku tohoto reálného bloku. Ta je rovna vzdálenosti mezi dnem nejnižší police v níž leží  $B^R$  a  $B^R.y$ .

Do  $B^R.x$  uložíme horizontální souřadnici. Její poloha se spočítá z relativní polohy magnetické čáry, ke které je přitahován  $B$  a z velikosti vnitřní šířky skříně. Necht'  $r$  je relativní poloha příslušné magnetické čáry (obor hodnot je  $[0,1]$  a 0 odpovídá pozici zcela vlevo) a  $s$  je vnitřní šířka skříně, pak

$$B^R.x = r \cdot s$$

Šířku  $B^R.width$  inicializujeme nulou.

Dále je třeba určit, jakým směrem bude  $B^R$  expandovat tj. jestli bude expandovat doprava, nebo doleva, jelikož základní myšlenka následného rozmisťovacího algoritmu je postupná expanze reálných bloků v pořadí daném jejich prioritami. Pod pojmem expanze si předběžně můžeme představit rozšíření reálného bloku o šířku potřebnou pro umístění určitého počtu produktů ve směru expanze. Směr expanze je dán vzájemnou polohou bloku a magnetické čáry, ke které je přitahován.

```
if (  $B$  vlevo od příslušné magnetické čáry) {  
     $B^R.direction$  = left  
}  
else {  
     $B^R.direction$  = right  
}
```

V rámci inicializace je nutné určit levé a pravé sousedy  $B^R$ .

Reálný blok  $L^R$  je potenciálním levým sousedem  $B^R$ , jestliže jsou splněny následující podmínky:

- $L$  a  $B$  leží alespoň na jedné společné řádce v rozmisťovací mřížce
- $L$  a  $B$  přísluší ke stejné magnetické čáře
- $L$  je nalevo od  $B$  (bráno dle indexu sloupce v rozmisťovací mřížce)
- Reálný blok  $L^R$  je levým sousedem  $B^R$ , jestliže jsou splněny následující podmínky:
- $L^R$  je potenciálním levým sousedem  $B^R$

- Neexistuje reálný blok  $M^R$  různý od  $L^R$  a od  $B^R$  takový, že  $L^R$  je potenciálním levým sousedem  $M^R$  a  $M^R$  je potenciálním levým sousedem  $B^R$ . Jinak řečeno neexistuje žádný reálný blok mezi  $L^R$  a  $B^R$ .

Pravé sousedy definujeme obdobně.

Pro každý reálný blok  $B^R$  tedy při inicializaci uložíme do  $B^R.leftNeighbours$  jeho levé sousedy a do  $B^R.rightNeighbours$  jeho pravé sousedy. Určíme je jednoduchým otestováním výše uvedených podmínek.

Poslední částí inicializace je vytvoření tzv. magnetických skupin. Jedná se o skupiny reálných bloků náležících ke stejné magnetické čáře. Každý reálný blok  $B^R$  tedy náleží právě do jedné magnetické skupiny. Jelikož je  $B^{R,x}$  dána příslušností daného reálného bloku k magnetické čáře, mají všechny bloky ze stejné magnetické skupiny stejnou horizontální souřadnici  $B^{R,x}$ . Magnetickou skupinu reálných bloků přitahovaných (náležících) k magnetické čáře  $M$  značíme  $M^S$ .

Řekneme, že  $B^R$  je bezprostředně přitahován magnetickou čarou  $M$ , jestliže jsou splněny následující podmínky:

- $B^R \in M^S$
- Je-li  $B^R.direction = left$ , pak  $B^R$  není levým sousedem žádného reálného bloku z  $M^S$  tj. mezi  $B$  a magnetickou čarou  $M$  není žádný jiný blok v rozmístovací mřížce, který by měl s  $B$  společnou alespoň jednu řádku. Obdobně je-li  $B^R.direction = right$ .

## 5.5.7 Expanze bloků

Hlavní činností rozmístovacího algoritmu je expanze reálných bloků.

Po inicializaci máme rozmístěny reálné bloky ve skříně na horizontálních pozicích daných jejich příslušností do magnetických skupin. Část z nich má směr expanze nastaven na left, část na right. Všechny reálné bloky mají na počátku nulovou šířku.

Během fáze expanze se algoritmus snaží postupně zvětšovat  $B^R.width$  jednotlivých reálných bloků  $B^R$  o násobek šířky produktu umístěného v bloku  $B$  a případně měnit  $B^R.x$  tak, aby zůstaly zachovány všechny podmínky plynoucí z rozmístovacího pravidla tj. z univerzálního planogramu:

1. Žádné dva reálné bloky se nesmějí překrývat. Pozicí reálného bloku a jeho rozměry je definován obdélník umístěný ve skříně. Dva obdélníky příslušící dvěma reálným blokům nesmějí mít společnou dvourozměrnou oblast. Mohou mít společnou maximálně hranu.
2. Reálný blok nesmí přesahovat vnitřní okraje skříně.
3. Je-li  $L^R$  levým sousedem  $B^R$ , pak  $L^R.x + L^R.width \leq B^R.x$ . Tedy relace sousedství definuje vzájemnou polohu reálných bloků a tato vzájemná poloha je neměnná.
4. Jsou-li  $B_1^R$ ,  $B_2^R$  dva reálné bloky bezprostředně přitahované magnetickou čarou  $M$  expandující doleva, pak  $B_1^R.x + B_1^R.width = B_2^R.x + B_2^R.width$  tj.  $B_1^R$  a  $B_2^R$  mají stejnou horizontální souřadnici pravého okraje a jsou tedy k této souřadnici zarovnané.

5. Jsou-li  $B_1^R, B_2^R$  dva reálné bloky bezprostředně přitahované magnetickou čarou  $M$  expandující doprava, pak  $B_1^R.x = B_2^R.x$  t.j.  $B_1^R$  a  $B_2^R$  mají stejnou horizontální souřadnici levého okraje a jsou tedy k této souřadnici zarovnané.
6. Je-li  $B^R$  bezprostředně přitahován magnetickou čarou  $M$ ,  $B^R.direction = left$  a  $M$  je fixní, pak  $B^R.x + B^R.width = r \cdot s$ , kde  $r$  je relativní poloha  $M$  a  $s$  je vnitřní šířka skříně.
7. Je-li  $B^R$  bezprostředně přitahován magnetickou čarou  $M$ ,  $B^R.direction = right$  a  $M$  je fixní, pak  $B^R.x = r \cdot s$ , kde  $r$  je relativní poloha  $M$  a  $s$  je vnitřní šířka skříně.
8. Je-li  $B^R$  in  $M^S$ ,  $B^R.direction = left$  a  $B^R$  není bezprostředně přitahován k  $M$ , pak existuje  $P^R$  takový, že  $P^R$  je pravý soused  $B^R$  a  $B^R.x + B^R.width = P^R.x$
9. Je-li  $B^R$  in  $M^S$ ,  $B^R.direction = right$  a  $B^R$  není bezprostředně přitahován k  $M$ , pak existuje  $L^R$  takový, že  $L^R$  je levý soused  $B^R$  a  $B^R.x = L^R.x + L^R.width$ .

To ve spojení s podmínkou (8.) zaručuje, že reálné bloky ze stejné magnetické skupiny pokrývají ve skříni souvislou oblast.

### Definice regulérního rozmístění

*Jestliže rozmístění reálných bloků splňuje předchozí podmínky, řekneme, že se jedná o regulérní rozmístění.*

Regulérní rozmístění odpovídá požadavkům uživatele vyjádřenými rozmisťovacími pravidly co se týče vzájemné polohy bloků produktů ve skříni (reálných bloků). Bude tedy třeba ukázat, že algoritmus vytváří regulérní rozmístění.

Uživatel však má požadavky také na co nejlepší využití prostoru ve skříni. Abychom mohli později dokázat, že algoritmus generuje optimální rozmístění, musíme zadefinovat, co si pod pojmem optimálního rozmístění představujeme. Uživateli jde zejména o to, aby bylo rozmístění regulérní a aby byly přednostně umístěny bloky s vyššími prioritami.

### Definice optimálního rozmístění

- *Řekneme, že rozmístění je 0-optimální, jestliže je regulérní.*
- *Řekneme, že rozmístění je n-optimální, jestliže je (n-1)-optimální a platí:  $B^R.width = 0$  a priorita  $B^R$  je n-tá nejvyšší, pak neexistuje (n-1)-optimální rozmístění v němž  $B^R.width > 0$ .*
- *Řekneme, že rozmístění je optimální, jestliže pro  $\forall n \in \mathbb{N}$  je n-optimální.*

Později ukážeme, že algoritmus generuje optimální rozmístění, nejprve však je třeba popsat detailně samotný postup expanze reálných bloků.

Během první fáze expandování reálných bloků algoritmus postupuje následovně:

```
for (int i = 0; i != SBR.size; ++i) {
    // začátek kroku algoritmu
    if (lzeExpandovat(SBR[i], SBR[i].minWidth)) {
        expanduj(SBR[i], SBR[i].minWidth);
    }
    // konec kroku algoritmu
```

```
}
```

Během druhé fáze už jen obdobným způsobem rozšiřujeme reálné bloky nad rámec požadované minimální šířky, aby ve skříni nezůstalo zbytečně nevyužité místo. Existuje více možností jak toto rozšiřování provést. Můžeme například zvětšovat bloky vždy rozšířením o jeden kus, nebo se můžeme snažit důležitější bloky rozšiřovat rychleji o víc kusů, než bloky méně důležité.

Důležité je, že  $S_{b^R}$  je od inicializace seříděný dle priorit obsažených reálných bloků a tedy v první iteraci zjistíme, zda lze expandovat blok s nejvyšší prioritou a pokud ano tak jej expandujeme. V druhé iteraci provedeme totéž s blokem s druhou nejvyšší prioritou a tak postupujeme až nakonec expandujeme, je-li to možné, reálný blok s nejnižší prioritou.

Pro správnost algoritmu je tedy podstatné umět rozhodnout, zda je možné v danou chvíli expandovat daný reálný blok. V případě, že to možné je, musíme provést expanzi správným způsobem.

Procedura `lzeExpandovat( $B^R$ )` dostane na vstupu reálný blok  $B^R$  a zjistí, zda je možné provést jeho expanzi.

```
boolean lzeExpandovat(  $B^R$ , šířka ) {  
    return lzeExpandovatPřímo(  $B^R$ , šířka,  $B^R.direction$  ) ||  
           lzeExpandovatNepřímo(  $B^R$ , šířka,  $B^R.direction$  );  
}
```

Expanzi je možné provést buď přímo, nebo nepřímo. Přímou expanzi si můžeme představit tak, že blok zvětší svou velikost ve směru expanze a případně si uvolní místo posunutím jiných bloků v tomto směru. Nepřímá expanze může nastat v případě, že nelze provést expanzi přímou tj. posunutím bloků ve směru  $B^R.direction$  nelze získat dostatek místa. Nepřímá expanze pak vypadá tak, že blok získá maximum místa posunutím bloků ve směru expanze a zbylé místo získá posunutím bloků v opačném směru. Šířka  $B^R.minWidth$  se počítá z minimálního počtu kusů produktu v bloku B a z šířky produktu.

Procedura `lzeExpandovatPřímo( $B^R$ , width, směr)` zjistí, zda lze provést přímou expanzi o šířku width takto:

```
boolean lzeExpandovatPřímo(  $B^R$ , width, směr ) {  
    if (zvětšení bloku ve směru  $B^R.direction$  znamená  
        překročení hranice skříně) {  
        return false  
    }  
    S = seznam sousedů  $B^R$  ve směru expanze;  
    for (int i = 0; i != S.size; ++i) {  
        distance = vzdálenost(  $B^R$ , S[i] );  
        posun = max(0, width - distance);  
        if (! lzeExpandovatPřímo(S[i], posun, směr) {  
            return false;  
        }  
    }  
}
```

```

S = seznam kolidujících bloků z jiných mag. skupin;
for (int i = 0; i != S.size; ++i) {
    distance = vzdálenost( $B^R$ , S[i]);
    posun = max(0, width - distance);
    if (!lzePosunoutSkupinu(skupina(S[i]), posun, směr)
{
        return false;
    }
    return true;
}
}

```

Následuje popis procedury, která zjišťuje, zda je možné posunout magnetickou skupinu:

```

boolean lzePosunoutSkupinu(S, posun, směr) {
    if (skupina fixní) {
        return false;
    }
    for (int i = 0; i != S.size; ++i) {
        if (!lzeExpandovatPřimo(S[i], posun, směr) {
            return false;
        }
    }
    return true;
}

```

Abychom zjistili, zda může reálný blok  $B^R$  expandovat nepřímou, musíme umět zjistit, kolik místa má na přímou expanzi tj. kolik místa je možné uvolnit ve směru  $B^R.direction$ . K tomuto využijeme následující proceduru:

```

Number místoProExpanzi( $B^R$ , směr) {
    výsledek = vzdálenost  $B^R$  od okraje skříně v daném směru;
    S = sousedé  $B^R$  v daném směru;
    for (int i = 0; i != S.size; ++i) {
        výsledek = min(výsledek, vzdálenost( $B^R$ , S[i]) +
            místoProExpanzi(S[i], směr));
    }
    G = seznam mag. skupin ve směru od  $B^R$ ;
    for (int i = 0; i != G.size; ++i) {
        výsledek = min(výsledek, vzdálenost( $B^R$ , G[i]) +
            místoProPosunSkupiny(G[i], směr));
    }
}

```

Vzdálenost reálného bloku  $B^R$  a magnetické skupiny  $S$  v níž není tento blok

obsažen definujeme jako minimum ze vzdáleností  $B^R$  od bloků v  $S$ , které mají s  $B^R$  alespoň jednu společnou řádku v rozmístovací mřížce.

Místo pro posun celé skupiny je v případě fixní skupiny nula. V případě skupiny, která fixní není je to minimum z míst pro expanzi jednotlivých reálných bloků obsažených v této skupině.

Nyní již není těžké rozhodnout, zda je možné provést nepřímou expanzi:

```
boolean lzeExpandovatNepřimo( $B^R$ , width, směr) {
    posun = max(0, width - místoProExpanzi( $B^R$ , směr));
    skupina = magnetická skupina v níž je  $B^R$ ;
    return lzePosunoutSkupinu(skupina, posun, opačný směr);
}
```

Zbývá popsat, jak funguje procedura, která provede samotnou expanzi reálného bloku  $B^R$ , o které již víme, že je možné ji provést:

```
expanduj( $B^R$ , šířka) {
    směr =  $B^R$ .direction;
    prostorPřimo = místoProExpanzi( $B^R$ , směr);
    expanzePřimo = min(šířka, prostorPřimo);
    expanzeNepřimo = šířka - expanzePřimo;
    expandujPřimo( $B^R$ , expanzePřimo, směr);
    expandujNepřimo( $B^R$ , expanzeNepřimo, opačný směr);
}
```

Procedura tedy zjistí, kolik má místa pro přímou expanzi a provede ji. Pokud místo pro přímou expanzi není dostatečné, provede se navíc nepřímá expanze, tak aby v součtu šířka reálného bloku vzrostla o požadovanou hodnotu.

Přímá expanze reálného bloku znamená zvětšení šířky o požadovanou velikost v požadovaném směru. Aby nedošlo k tomu, že se začnou některé bloky překrývat, je třeba pro expanzi vytvořit místo posunutím sousedů a případně i celých sousedních magnetických skupin.

```
expandujPřimo( $B^R$ , šířka, směr) {
    S = sousedé  $B^R$  v daném směru;
    for (int i = 0; i != S.size; ++i) {
        posuňSousedu( $B^R$ , S[i], šířka, směr);
    }
    posuňSkupiny( $B^R$ , šířka, směr);
    rozšiřBlok( $B^R$ , šířka, směr);
}
```

Z využitých procedur je nejjednodušší procedura, která rozšíří blok v požadovaném směru o požadovanou šířku. Volá se na konci přímé expanze. Před tímto voláním se pro rozšíření vytvoří dostatek místa.

```

rozšiřBlok( $B^R$ , šířka, směr) {
    if (směr je doleva) {
         $B^R.x = B^R.x - \text{šířka};$ 
    }
     $B^R.width = B^R.width + \text{šířka};$ 
}

```

Posun souseda probíhá následovně:

```

posuňSousedu( $B^R$ , soused, šířka, směr) {
    posun = max(0, šířka - vzdálenost( $B^R$ , soused));
    if (posun > 0) {
        S = susedé souseda v daném směru;
        for (int i = 0; i != S.size; ++i) {
            posuňSousedu(soused, S[i], posun, směr);
        }
        posuňSkupiny(soused, posun, směr);
        posuňBlok(soused, posun, směr);
    }
}
posuňBlok(soused, posun, směr) {
    if (soused.direction = left) {
        soused.x = soused.x - posun;
    }
    else {
        soused.x = soused.x + posun;
    }
}

```

Posun magnetické skupiny je třeba provést v případě, že by posun bloku znamenal kolizi s blokem z jiné skupiny. Musíme tedy zkontrolovat všechny bloky z jiných magnetických skupin a v případě potřeby je posunout.

```

posuňSkupiny( $B^R$ , šířka, směr) {
    S = seznam reálných bloků patřících do skupin v daném
        směru od skupiny v níž je  $B^R$ ;
    for (int i = 0; i != S.size; ++i) {
        if (vzdálenost( $B^R$ , S[i]) < šířka) {
            posun = šířka - vzdálenost( $B^R$ , S[i]);
            G = skupina v níž je  $B^R$ ;
            posuňSkupinu(G, posun, směr);
        }
    }
}
posuňSkupinu(G, posun, směr) {
    S = seznam reálných bloků patřících do skupiny G;
    for (int i = 0; i != S.size; ++i) {

```



```

    posuňSkupiny(S[i], posun, směr);
    posuňBlok(S[i], posun, směr);
}
}

```

Zbývá popsat, jak probíhá nepřímá expanze reálného bloku  $B^R$ .

```

expandujNepřimo(B^R, šířka, směr) {
    G = magnetická skupina v níž je B^R;
    posuňSkupinu(G, šířka, směr);
    expandujPřimo(B^R, šířka, opačný směr) {
}
}

```

### 5.5.8 Důkaz regulérnosti výsledku

Dokážeme, že v každém kroku algoritmu je rozmístění regulérní a tedy i výsledné rozmístění je regulérní. Důkaz provedeme indukcí podle počtu kroků algoritmu.

V kroku 0 tj. na začátku máme rozmístění vzniklé inicializací reálných bloků. Ověříme tedy postupně podmínky regulérnosti:

1. Tato podmínka je splněna triviálně, jelikož všechny reálné bloky mají na začátku nulovou šířku a nemohou se tedy překrývat.
2. Reálný blok nepřesahuje vnitřní okraje skříně, jelikož hodnota  $B^R \cdot x$  je mezi nulou a vnitřní šířkou skříně.
3. Je-li  $L^R$  levým sousedem  $B^R$ , pak  $L^R \cdot x + L^R \cdot width \leq B^R \cdot x$ , jelikož  $L^R \cdot width = 0$  a navíc  $L^R \cdot x = B^R \cdot x$ , protože  $L^R$  i  $B^R$  jsou přitahovány ke stejné magnetické čáře (jinak by se nejednalo o sousedy).
4. Jsou-li  $B_1^R$ ,  $B_2^R$  dva reálné bloky bezprostředně přitahované magnetickou čarou  $M$  expandující doleva, pak  $B_1^R \cdot x + B_1^R \cdot width = B_2^R \cdot x + B_2^R \cdot width$ , jelikož  $B_1^R \cdot width = 0$ ,  $B_2^R \cdot width = 0$  a  $B_1^R \cdot x = B_2^R \cdot x$ , protože  $B_1^R$  i  $B_2^R$  jsou přitahovány ke stejné magnetické čáře.
5. Jsou-li  $B_1^R$ ,  $B_2^R$  dva reálné bloky bezprostředně přitahované magnetickou čarou  $M$  expandující doprava, pak  $B_1^R \cdot x = B_2^R \cdot x$ , protože  $B_1^R$  i  $B_2^R$  jsou přitahovány ke stejné magnetické čáře.
6. Splněno triviálně, jelikož  $B^R \cdot width = 0$ .
7. Splněno triviálně.
8. Je-li  $B^R$  in  $M^S$ ,  $B^R \cdot direction = left$  a  $B^R$  není bezprostředně přitahován k  $M$ , pak existuje  $P^R$  takový, že  $P^R$  je pravý soused, jinak by byl  $B^R$  bezprostředně přitahován. Navíc  $B^R \cdot x + B^R \cdot width = P^R \cdot x$ , jelikož  $B^R \cdot width = 0$  a oba reálné bloky jsou přitahovány ke stejné magnetické čáře a tedy  $B^R \cdot x = r \cdot s = P^R \cdot x$ .
9. Obdobně jako podmínka (8.)

Nechť podmínky regulérnosti platí až do kroku  $(n - 1)$ . Ukážeme, že pak platí i po skončení  $n$ -tého kroku:

1. Pozice a velikost reálných bloků se v  $n$ -tém kroce změní pouze tehdy, když se provede procedura `expanduj( $S_{B^r}[n]$ ,  $S_{B^r}[n].minWidth$ )`, která mění pozici bloků pouze v rámci volání podprocedur `expandujPřímou` a `posuňSkupinu`. Stačí tedy ukázat, že vykonání těchto procedur nezpůsobí, že se některé bloky začnou překrývat.

V rámci procedury `expandujPřímou` dojde ke změně velikosti a pozice expandovaného reálného bloku provedením podprocedury `rozšiřBlok`. Jelikož by tato změna, mohla způsobit, že se rozšířený blok začne překrývat s nějakým jiným blokem, je v rámci procedury `expandujPřímou` proveden posun všech bloků, se kterými by mohlo dojít k překryvu.

Nejprve tedy dojde k posunu všech sousedů bloku, který se chystáme rozšířit vykonáním procedury `posuňSousedu` pro každého souseda. Tato procedura si nejprve zjistí, o kolik je třeba souseda posunout. Pokud je soused dostatečně daleko od rozšiřovaného bloku, aby nedošlo k překryvu, není třeba posun provádět. V opačném případě je soused posunut, tak aby se nepřekrýval s rozšiřovaným blokem.

Posun souseda sice zabrání překryvu s rozšiřovaným blokem, nicméně může způsobit jiný překryv. Proto i procedura `posuňSousedu` před provedením požadovaného posunu nejprve rekurzivně zavolá sebe sama na všechny sousedy souseda.

Takto zajistíme, že po rozšíření reálného bloku nevznikne překryv mezi bloky ze stejné magnetické skupiny jako tento blok.

Zbývá tedy ještě ukázat, že nevznikne ani překryv mezi bloky z různých magnetických skupin nebo překryv v rámci jiné magnetické skupiny.

Před rozšířením bloku proto procedura `posuňSkupiny` zjistí, zda by mohlo dojít k překryvu mezi rozšiřovaným blokem a blokem z jiné magnetické skupiny. V takovém případě je posunuta celá magnetická skupina o potřebnou vzdálenost (je změněna pozice všech bloků ve skupině o stejnou vzdálenost ve stejném směru) provedením procedury `posuňSkupinu`.

Procedura `posuňSkupinu` pochopitelně kromě samotné změny pozice bloků ve skupině zajistí, že ani tento posun nezpůsobí překryv. Vzhledem k tomu, že jsou posunuty všechny bloky ze skupiny o stejnou vzdálenost, tak mezi nimi překryv vzniknout nemůže. Překryv může vzniknout pouze s blokem z jiné magnetické skupiny a proto je opět použita procedura `posuňSkupiny`, která zajistí, že překryv nevznikne.

Překryv mezi bloky z různých magnetických skupin, by mohl vzniknout ještě jako důsledek provedení procedury `posuňSousedu`, proto i zde je využita metoda `posuňSkupiny`, která zajistí, že se tak nestane.

Podařilo se nám tedy ukázat, že v rámci volání procedur `expandujPřímou` a `posuňSkupinu` nevznikne překryv a jelikož se jedná o jediné procedury v jejichž rámci se mění pozice bloků a díky indukčnímu předpokladu, dokázali jsme platnost první podmínky regulérnosti algoritmu.

2. Reálný blok nesmí přesahovat vnitřní okraje skříně. Procedura `lzeExpandovat` zjistí, zda by po provedení procedury `expanduj` nepřesahoval některý z reálných bloků okraje skříně. Pouze v případě, že k přesahu přes okraje skříně nedojde, je provedena procedura `expanduj`. Stačí tedy ukázat, že pokud procedura `lzeExpandovat` vrátí `true`,

nemůže procedura expanduj způsobit přesah.

Podíváme-li se na implementaci metody lzeExpandovat, je zřejmé, že je třeba ukázat, že procedura expanduj nemůže způsobit přesah, pokud vrátí true procedura lzeExpandovatNepřímou.

Zásadní význam má procedura místoProExpanzi, která spočítá, o kolik je možné expandovat přímo daný blok v daném směru. Podíváme-li se na implementaci procedury expanduj, zjistíme, že přesně o hodnotu vrácenou procedurou místoProExpanzi je v rámci procedury expanduj provedena podprocedura expandujPřímou.

Maximální vzdálenost o kterou je možno blok expandovat přímo v daném směru, je dána jeho vzdáleností od okraje skříně. Jak již víme z důkazu (1.), procedura expandujPřímou posouvá sousedy ve směru expanze o minimální potřebnou vzdálenost, tak aby nedošlo k překryvu dvou různých bloků. Z tohoto důvodu se rekurzivním voláním zjistí pro každého ze sousedů, jaké je jeho místo pro expanzi. Přesněji řečeno místo pro expanzi bloku není větší, než součet vzdálenosti tohoto bloku od jeho souseda a místa pro expanzi souseda. Obdobně platí, že místo pro expanzi bloku není větší, než součet vzdálenosti tohoto bloku od cizí magnetické skupiny a místa pro posun této skupiny. To odpovídá tomu, jak procedura expandujPřímou posouvá magnetické skupiny. Procedura místoProExpanzi proto vrátí minimum.

Procedura lzeExpandovatNepřímou tedy nejprve zjistí, kolik je místa pro přímou expanzi a v případě, že ho není dostatek, zjistí, zda lze posunout skupinu daného bloku o zbylou vzdálenost ve směru opačném. Skupinu lze posunout pouze, pokud není fixní a pokud lze přímo expandovat každý její blok o délku rovnou danému posunu v daném směru. Zbývá tedy zdůvodnit, že správně funguje i procedura lzeExpandovatPřímou. Tato procedura funguje, jelikož zjistí, zda by nedošlo k překročení hranice skříně samotným zvětšením daného bloku, ani posuny kolidujících magnetických bloků a skupin.

Jak je vidět, tak procedura lzeExpandovat vlastně odsimuluje, jak by vypadala případná expanze a v případě, že v nějakém okamžiku dojde místo a bylo by nutné překročit hranice skříně, vrátí false. Pokud k překročení hranice nedojde, vrátí true a následně je provedena opravdová expanze, kde se již nekontroluje, zda nedochází k překročení hranic skříně, jelikož se ví, že k němu nedojde.

3. Je-li  $L^R$  levým sousedem  $B^R$ , pak  $L^R \cdot x + L^R \cdot width \leq B^R \cdot x$ . Platnost této podmínky by mohla změnit pouze metoda expanduj, ale vzhledem k tomu, že jak již víme, tato metoda, posouvá sousedy a sousední skupiny expandovaného bloku tak, aby nedošlo k překryvu dvou bloků, nemůže tím spíše dojít k změně vzájemné polohy.
4. Jestliže blok expanduje směrem doleva, pak může posouvat bloky ze stejné magnetické skupiny v rámci podprocedury expandujPřímou. Posouvá však pouze levé sousedy a potenciální levé sousedy a tedy nemůže tímto narušit zarovnání, které je napravo od něj.

Zarovnání není narušeno ani vykonáním procedury posuňSkupiny, jelikož tato procedura posune vždy všechny bloky ve skupině o stejnou vzdálenost.

Posledním místem, kdy by mohlo dojít k porušení zarovnání je procedura expandujNepřímou, jelikož nepřímá expanze znamená posunutí bloku ve směru, kde je zarovnání jeho magnetické skupiny. Nicméně procedura expandujNepřímou se zachová

korektně, jelikož nejprve posune celou magnetickou skupinu o vzdálenost expanze, čímž neporuší zarovnání a poté provede přímou expanzi bloku již ve směru od zarovnání a jak již víme přímá expanze zarovnání neporuší.

5. Důkaz obdobně jako v předchozím bodě.
6. Jelikož při expanzi bloku nedochází k posunu cizích magnetických skupin, které jsou fixní, není možné aby byla podmínka expanzí narušena v jiné magnetické skupině, než je skupina expandovaného bloku. K posunu vlastní skupiny nedojde ani v rámci nepřímé expanze, je-li tato skupina fixní.

Je-li  $B^R$  bezprostředně přitahován magnetickou čarou  $M$ ,  $B^R.direction = left$  a  $M$  je fixní, pak mohou nastat následující případy během expanze:

- $B^R$  není expandovaným blokem a vzhledem k tomu, že je bezprostředně přitahován k  $M$ , není levým sousedem žádného bloku a nemůže se tedy stát, že by byl posunut v rámci expanze expandovaného bloku směrem doleva. Pokud expanze probíhá směrem doprava, tak také nemůže dojít k posunu  $B^R$ , jelikož buď expanduje blok, který již je napravo od  $B^R$  a tedy neovlivní jeho polohu, nebo expanduje blok z jiné magnetické skupiny a taková expanze jak již víme nemůže způsobit posun fixní magnetické skupiny v níž se nachází  $B^R$ . K expanzi bloku ze stejné magnetické skupiny, který by byl nalevo od  $B^R$  směrem doprava, nemůže dojít, jelikož by muselo jít o nepřímou expanzi a ta se u bloků z fixních magnetických skupin nekoná.
- $B^R$  je expandovaným blokem a tedy expanduje pouze přímo, jelikož je ve fixní magnetické skupině a tedy jeho šířka buď zůstane nulová, nebo se zvětší o  $B^R.width$  ve směru doleva a tedy bude nadále platit  $B^R.x + B^R.width = r.s$ .

7. Důkaz obdobně jako v předchozím bodě.
8. Pokud je posunuta celá magnetická skupina  $M^S$ , tak nemůže dojít k narušení platnosti podmínky, jelikož  $B^R$  i  $P^R$  jsou posunuty oba ve stejném směru o stejnou vzdálenost a tedy stále platí, že  $B^R.x + B^R.width = P^R.x$ .

K narušení podmínky  $B^R.x + B^R.width = P^R.x$  by mohlo dojít pouze v případě, že by expandoval doleva blok  $E^R$ , jehož je  $B^R$  potenciálním levým sousedem, nikoliv však  $P^R$ . V takovém případě by se mohlo stát, že již nebude platit  $B^R.x + B^R.width = P^R.x$ . V tom případě by ale musel existovat jiný pravý soused bloku  $B^R$  a to blok  $C^R$ , který by způsobil posunutí bloku  $B^R$  a platilo by tedy  $B^R.x + B^R.width = C^R.x$ . Tedy podmínka existence bloku daných vlastností zůstává platná i po provedení kroku  $n$ .

9. Důkaz obdobně jako v předchozím bodě.

### 5.5.9 Důkaz optimálnosti výsledku

Dokazujeme, že rozmístění je  $n$ -optimální pro  $\forall n \in \mathbb{N}$ .

Důkaz provedeme indukcí podle  $n$ .

Pro  $n = 0$ , tvrzení platí, jelikož výsledek algoritmu je regulérní a tedy 0-optimální.

Nechť tedy tvrzení platí až po  $n - 1$ . Blok s  $n$ -tou nejvyšší prioritou se algoritmus snažil expandovat v  $n$ -tém kroce.

V případě, že tento blok byl expandován, je  $B^R.width \neq 0$  a tvrzení tedy platí i pro  $n$ .

Pokud blok expandován nebyl, znamená to, že procedura  $lzeExpandovat(B^R)$  vrátila `false` a tedy nebylo možné provést ani přímou ani nepřímou expanzi a tedy ve skříni není dostatek místa pro tento blok společně s již expandovanými bloky do  $n$ -tého kroku. Z indukčního předpokladu, plyne, že bloky expandované do  $n$ -tého kroku musí být být expandované v každém  $(n-1)$ -optimálním rozmístění, jelikož jejich priority jsou menší než  $n$ . Není tedy možné nijak získat více místa pro expanzi bloku  $B^R$  a neexistuje tedy  $(n-1)$ -optimální rozmístění v němž  $B^R.width > 0$ .

### 5.5.10 Důvody pro neumístění produktů

Algoritmus by během svého průběhu měl zaznamenávat důvody neumístění produktů, aby bylo možné uživateli podat přehled, které produkty obsažené v univerzálním planogramu se neobjevili v planogramu reálném a proč. Algoritmus může neumístit produktblok z následujících důvodů:

- Produktblok odpovídá univerzálnímu bloku na řádku, který se neumísťuje z důvodu nedostatečného počtu polic v reálné skříni viz. 5.5.4.
- Police je příliš nízká pro daný produktblok viz. 5.5.5.
- Produktblok odpovídá reálnému bloku, jehož šířka zůstala po dokončení algoritmu nulová, jelikož ve skříni nebylo dostatek místa pro tento produktblok viz. procedura  $lzeExpandovat(B^R)$ .

# 6 Popis prototypové implementace

Pro demonstraci fungování výše popsaného návrhu tvorby univerzálních planogramů a rozmisťovacího algoritmu generujícího planogramy reálné je na přiloženém CD obsažen instalátor programu QuantPE pro Windows, Linux a MacOS X. Program QuantPE byl s mým přispěním vyvinut ve firmě ExTech s.r.o. jako jednoduchý nástroj pro tvorbu planogramů podobný ShelfLogic určený zejména pro tvorbu planogramů v malém měřítku a demonstrační účely.

QuantPE umožňuje:

- Vytvořit databázi produktů zadáním jednotlivých produktů ručně nebo s využitím importní utility.
- Zadat rozměry skříně v jednoduchém editoru.
- Vytvořit univerzální planogram a sledovat, jak z něj rozmisťovací algoritmus automaticky vytvoří reálný planogram pro skříně zadaných rozměrů.
- Uložit univerzální planogram společně se skříní na niž má být aplikován do souboru, který je možno později znovu otevřít a upravit.
- Exportovat výsledný planogram do PDF nebo jako prostý obrázek.

Uživatel může sledovat, jak rozmisťovací algoritmus automaticky přizpůsobuje výsledný planogram různým rozměrům skříně jednoduše tak, že rozměry změní.

Přiložené CD také obsahuje zdrojové kódy prototypové implementace univerzálního planogramu a rozmisťovacího algoritmu v jazyce Java.

## 6.1 Uživatelská dokumentace

### 6.1.1 Instalace QuantPE

Na přiloženém CD viz. příloha A jsou obsaženy instalátory QuantPE pro operační systémy Windows, Linux a MacOS X. Při instalaci postupujte následujícím způsobem:

#### **Windows**

- Spusťte QuantPE.exe soubor. Objeví se průvodce instalací. Klikněte na tlačítko "Next" a pokračujte podle pokynů instalátoru.
- Jakmile je instalace dokončena, klikněte na tlačítko "Finish"
- Pod volbou "Start" "Programy" naleznete položku "Quant". Klikněte na

"Planogrammer".

## Mac

- Spusťte QuantPE.exe soubor. Objeví se průvodce instalací. Klikněte na tlačítko "Next" a pokračujte podle pokynů instalátoru.
- Jakmile je instalace dokončena, klikněte na tlačítko "Finish"
- Pod volbou "Aplikace" vašeho programu Finder naleznete položku "Quant". Klikněte na "Planogrammer".

## Linux

- Spusťte stažený QuantPE.sh soubor.
- Z příkazové řádky pomocí: `sh [CESTA_K_INSTALATORU]/quant_planogrammer_setup_[VERZE].sh`
- Nebo v grafickém souborovém manažeru nastavte oprávnění "spustitelný" na soubor `quant_standalone_setup_[VERZE].sh` (nejčastěji v pravý klik/Vlastnosti), pak spusťte soubor dvojklikem
- Objeví se průvodce instalací. Klikněte na tlačítko "Next".
- Zvolte zapisovatelné adresáře pro místo instalace a symbolických linků.
- Jakmile je instalace dokončena, klikněte na tlačítko "Finish"
- Aplikaci můžete spustit z příkazové řádky pomocí "QuantExplorer" - pokud umístění vytvořených symbolických linků máte ve svém PATH nebo "[QUANT\_INSTALLATION\_DIR]/Planogrammer" – jinak. Nebo si zkopírujte záložku z [QUANT\_INSTALACNI\_ADRESAR]/Planogrammer.desktop na svoji plochu a klikněte na něj.

## 6.1.2 QuantPE s využitím demonstrační databáze

QuantPE obsahuje po nainstalování demonstrační databázi s dostatečným počtem produktů pro vyzkoušení principů tvorby univerzálních planogramů a jejich automatického převodu na planogramy reálné.

Po prvním spuštění je otevřen demonstrační dokument, který obsahuje ukázkový univerzální planogram. Pro odzkoušení principů popsaných v této práci by měl být tento ukázkový univerzální planogram plně postačující, jako dobrý startovní bod.

V záložce planogram viz. Ilustrace 9, která je vybrána jako defaultní, je v levé části vidět reálný planogram automaticky vygenerovaný pro zadané rozměry skříně dle planogramu univerzálního, který se nachází ve střední části obrazovky pod nadpisem „Globální rozmístění“. V pravé části obrazovky se pak zobrazují informace o umístěných a neumístěných produktech a pokrytí tržního podílu umístěnými produkty a brandy. Dolní část obrazovky obsahuje seznam produktů v databázi.

Nový produkt je možno přidat do univerzálního planogramu prostým přetažením ze

seznamu produktů do rozmisťovací mřížky univerzálního planogramu. V rozmisťovací mřížce tak automaticky vznikne blok obsahující přetažený produkt a reálný planogram se překreslí automaticky tak, aby odrážel nejnovější stav univerzálního planogramu.

id	jméno	brand	skupina	obrat	podíl	kumulovaný	b-podíl	V x S x H (mm)	náhled
36903	ACIFEIN porttbl...		ANALGETICA, AN...	1 780	1,19 %	1,19 %	1,19 %	48 x 113 x 19	
21083	ACYLPYRIN+ C T...		ANALGETICA, AN...	1 382	0,93 %	2,12 %	0,93 %	36 x 110 x 36	
20177	ADDITIVA VITAM...		VITAMINA, VITA...	2 596	1,74 %	3,86 %	1,74 %	150 x 37 x 37	
20178	ADDITIVA VITAM...		VITAMINA, VITA...	530	0,36 %	4,22 %	0,36 %	151 x 38 x 38	

Ilustrace 9: Záložka Planogram v QuantPE

Vlastnosti bloku je možno upravit dvojitým poklepáním na tento blok. Otevře se

Ilustrace 10: Vlastnosti bloku v QuantPE

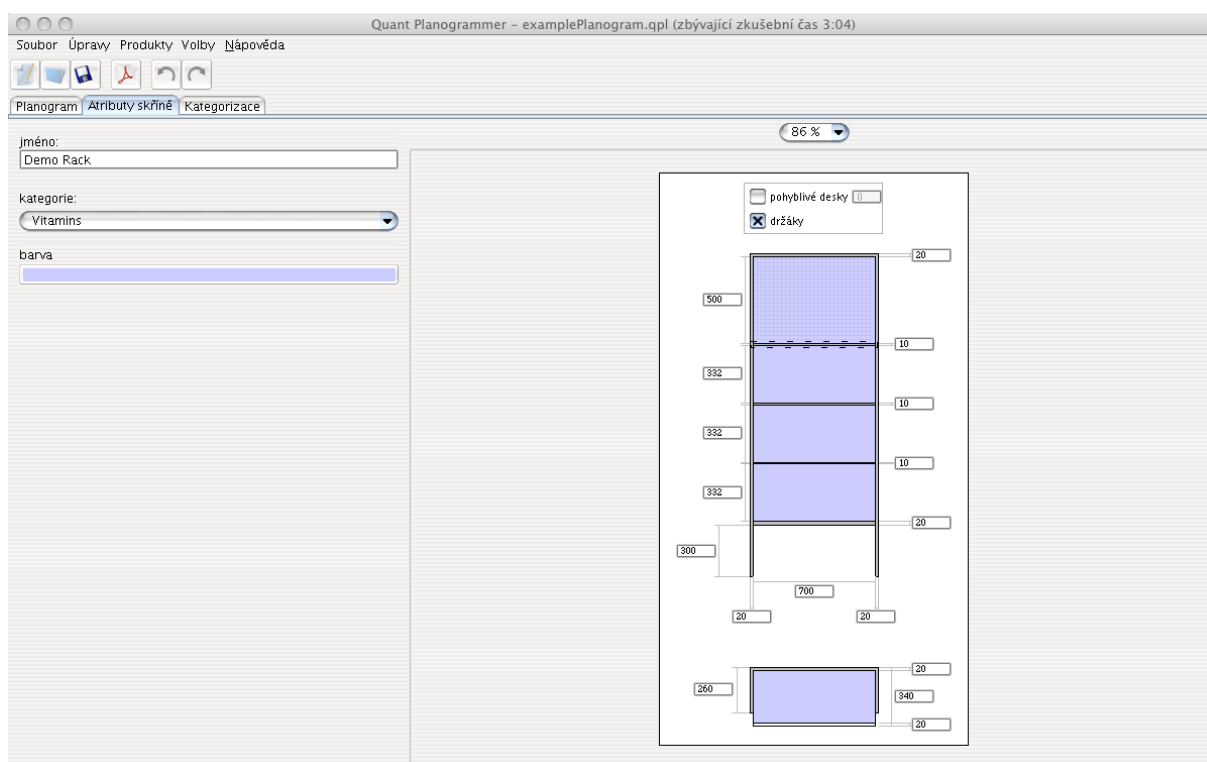


dialog viz. Ilustrace 10 umožňující měnit všechny důležité atributy, jak byly popsány v kapitole 5.1.

Magnetickou čáru je možno vložit využitím kontextového menu, které se objeví po kliknutí pravým tlačítkem myši v oblasti rozmisťovací mřížky. Jedná se o volbu „Vložit magnetickou čáru“.

Kontextové menu je možno využít také k přidání a odebrání řádku nebo sloupce rozmisťovací mřížky.

Rozměry skříně pro niž je generován reálný planogram na základě vytvořeného univerzálního planogramu se nastavují v záložce „Atributy skříně“ viz. Ilustrace 11. Schéma skříně nacházející se v pravé části obrazovky obsahuje vstupní pole pro odpovídající rozměry v milimetrech. Zadání nového rozměru se projeví po stisku klávesy „enter“. Počet polic se nastavuje přes kontextové menu schématu skříně. Po přepnutí zpět do záložky „Planogram“ je možno ihned sledovat největší výhodu univerzálních palnogramů a rozmisťovacího algoritmu, kterou je automaticky vygenerovaný reálný planogram pro nově nastavené rozměry skříně.



Ilustrace 11: Záložka Atributy skříně v QuantPE

Záložka „Kategorizace“ slouží ke správě databáze produktů a jejich zařazování do kategorií, jenž nejsou předmětem této práce a proto se jejich podrobnějším popisem nemá smysl zabývat.

Menu „Soubor“ umožňuje obvyklým způsobem ukládat, otevírat, či vytvářet nové dokumenty obsahující vždy jeden univerzální planogram a rozměry jedné skříně. Reálný planogram je možné vyexportovat jako rastrový obrázek, nebo jako PDF.

## 6.2 Programátorská dokumentace

Cílem následujících odstavců je představit nejdůležitější třídy a části zdrojového kódu prototypové implementace a ukázat základní příklady jejich využití, tak aby se čtenář mohl zorientovat v základních podporovaných postupech. Kompletní informace je pochopitelně obsažena v samotném kódu a doplňkových komentářích.

### 6.2.1 Model skříně

Rozměry skříně jsou reprezentovány třídou **RackModel** obsaženou v balíku **eu.kavanek.universal\_planograms.rack\_model**.

Následující výpis kódu demonstruje ukázkové nastavení rozměrů skříně:

```
// vytvori novou skrin
RackModel rack = new RackModel();

// nastaveni vnitřních rozmeru
rack.set(RackMeasure.INSIDE_SIZE_X, 800);
rack.set(RackMeasure.INSIDE_SIZE_Y, 300);
rack.set(RackMeasure.INSIDE_SIZE_Z, 2000);

// nastaveni tloušťky okraju skrine
rack.set(RackMeasure.BACK_BORDER_SIZE, 20);
rack.set(RackMeasure.TOP_BORDER_SIZE, 20);
rack.set(RackMeasure.BOTTOM_BORDER_SIZE, 20);
rack.set(RackMeasure.LEFT_BORDER_SIZE, 20);
rack.set(RackMeasure.RIGHT_BORDER_SIZE, 20);

// vyplneni skrine deskami vysokými 10 mm tak, aby
vzniklo 6 polic stejne vysky
rack.fillWithShelves(6, 10);
```

Podporované rozměry jsou reprezentovány třídou **RackMeasure**.

### 6.2.2 Třídy reprezentující universální planogram

Třídy reprezentující universální planogram jsou obsaženy v balíku:

**eu.kavanek.universal\_planograms.placement\_algorithm**.

Univerzální blok je v prototypové implementaci reprezentován třídou **PlacementBlockModel**. Obsahuje set a get metody pro nastavování a získání všech důležitých atributů univerzálního bloku, jak byly popsány v 5.1. Rozmístovací algoritmus na svém vstupu dostává seznam universálních bloků reprezentovaný třídou

**PlacementBlockModelList** obsaženou v tomtéž balíku.

Obdobně jsou reprezentovány magnetické čáry třídami **MagneticLineModel** a **MagneticLineModelList**.

Třída **ShelfIndexToShelfPriorityMap** slouží k namapování priorit polic na indexy polic, jak bylo popsáno v 5.3.

Informace o tom, jaký typ zarovnání má být použit pro kterou z polic je reprezentována třídou **GridLineControlsModel**. Podporované typy zarovnání představují třídy:

- **GridLineControlLeft**
- **GridLineControlRight**
- **GridLineControlJustify**
- **GridLineControlCenter**

odpovídající typům zarovnání popsaných v 5.4.

## 6.2.6 Implementace rozmisťovacího algoritmu

Rozmisťovací algoritmus je reprezentován třídou **PlacementAlgorithm** obsaženou v balíku:

**eu.kavanek.universal\_planograms.placement\_algorithm.**

Vstup algoritmu v podobě univerzálního planogramu a reálné skříně pro kterou má být vytvořen reálný planogram se předává v konstruktoru jako instance tříd:

- **PlacementBlockModelList** jako seznamu univerzálních bloků
- **MagneticLineModelList** jako seznam magnetických čar
- **Integer** jako počet sloupců a řádků rozmisťovací mřížky
- **ShelfIndexToShelfPriorityMap** jako priority polic
- **GridLineControlsModel** jako zarovnání produktů v policích
- **RackModel** jako rozměry reálné skříně.

Algoritmus provede výpočet zavoláním metody **calculateAll()**. Grafickou podobu výsledného planogramu je možno získat ve vektorové reprezentaci implementující rozhraní **SmartShape** viz. 6.2.7. Je také možno získat informace o umístěných a neumístěných produktech resp. reálných blocích zavoláním metod **getPlacedProducts()** a **getNotPlacedProducts()**.

## 6.2.7 Vektorová reprezentace planogramu

Pro potřeby spojené s grafickou reprezentací planogramů obsahuje prototypová implementace balík:

**eu.kavanek.universal\_planograms.smart\_shape.**

Je zde obsaženo rozhraní **SmartShape**, které reprezentuje dvojrozměrný objekt, který je možno zvětšovat, zmenšovat, otáčet apod. Takový objekt je možné pomocí metody **paint()** vykreslit na **java.awt.Graphics2D**.

Balík obsahuje několik užitečných tříd implementujících toto rozhraní včetně třídy **MultiShape**, která umožňuje skládat komplexní tvary z jednodušších. Pro reprezentaci planogramu tak jsou využity například následující třídy:

- **AbstractSmartShape** představující lomenou čáru použitou k vykreslení skříně
- **ImageSmartShape** představující obrázek použitý k vykreslení produktů
- **TextSmartShape** a jeho potomci představující text použitý k vykreslení popisků
- **MultiShape** použitý k poskládání celého planogramu do jednoho objektu implementujícího rozhraní **SmartShape**.

Třídy vytvořené pro grafickou reprezentaci planogramu jsou samostatně využitelné i v projektech, které se netýkají univerzálních planogramů.

# 7 Uplatnění navrženého řešení

Navržené řešení v podobě univerzálního planogramu a rozmisťovacího algoritmu našlo své uplatnění v několika produktech firem ExTech s.r.o. a QuantPharma GmbH. Cílem této kapitoly je podat stručný popis těchto produktů a způsobu, jakým využívají řešení navržené v této práci. Uvedeno bude také několik případových studií nasazení těchto systémů.

## 7.1 Systémy využívající univerzální planogramy

V následujícím textu je popsáno několik systémů, v nichž byly nalezeny univerzální planogramy uplatněné ještě před dokončením této práce.

### 7.1 QuantEE / i-merch

QuantEE distribuovaný také pod jménem i-merch, je komplexní merchandisingový systém, který byl vyvinut na základech položených projektem IPPS obhájěným jako softwarový projekt na MFF UK viz. [8]. QuantEE poskytuje komfortní řešení pro velká merchandisingová oddělení zahrnující:

- Editor prodejen v němž je možno tvořit přehledné půdorysy a zadávat rozměry skříní.
- Editor produktů s podporou automatických importů dat o prodejích a jejich analýzy.
- Editor kategorií umožňující category management.
- Editor projektů umožňující díky konceptu univerzálních planogramů vytvářet během velmi krátké doby přesné planogramy pro stovky prodejen.
- Webové rozhraní pro prodejny, ve kterém jsou aktuální a připravované planogramy přehledně zobrazeny spolu s dalšími doplňkovými informacemi, jako je vývoj tržeb jednotlivých skříní apod.
- Centralizovanou databázi umožňující jednotlivým klientským aplikacím sdílet data a spolupracovat na tvorbě planogramů z různých míst.

QuantEE využívá princip tvorby univerzálních planogramů ve své hlavní části, kterou je editor projektů. Projekt obsahuje skupiny skříní. Skupina skříní má společný univerzální planogram vytvořený uživatelem. Na jeho základě jsou automaticky generovány planogramy reálné pro všechny skříně ve skupině. Uživatel je automaticky informován o potenciálních problémech, jako je neumístění některého z produktů v některé z reálných skříní a má možnost tento problém odstranit modifikací kopie univerzálního planogramu tak, aby neovlivnil skříně ostatní.

Pomocí systému QuantEE zákazníci firmy ExTech s.r.o. úspěšně vytvořili již tisíce reálných planogramů, které byly využity ve stovkách prodejen a odezva merchandisingových

oddělení na koncept univerzálních palnogramů je velice pozitivní.

### **7.1.2 QuantSE**

QuantSE je obdobou QuantEE, která umožňuje pouze jednouživatelský režim. Celý systém běží samostatně na uživatelské počítači a data jsou uložena lokálně. Systém je určen zejména menším merchandisingovým oddělením, která nevyužijí výhody víceuživatelského přístupu, centralizované databáze a webového rozhraní pro prodejny. Planogramy vytvořené v QuantSE jsou distribuovány ve formě dokumentu PDF.

### **7.1.3 PPP**

PPP je systém vyvinutý společností QuantPharma GmbH ve spolupráci s Benchmark AG a Insight Health. Jedná se o plně webové rozhraní, které umožňuje nezávislým německým lékárnám generovat planogramy pro svůj OTC sortiment automaticky na základě nezávislých dat o prodejkách.

Automatický výpočet negeneruje přímo reálné planogramy, ale generuje planogramy univerzální. Lékařny totiž chtějí mít možnost provádět v automaticky vygenerovaných planogramech úpravy a uživatelské rozhraní tvorby univerzálních palnogramů se v tomto směru velmi osvědčilo. Navíc jeho implementace běžící ve webových prohlížečích pomocí technologie AJAX je poměrně snadná.

Univerzální planogramy jsou v PPP využity také pro tvorbu planogramů pro speciální akce výrobců. Lékařna totiž může pronajmout některou ze svých skříní výrobcem za určitou dohodnutou odměnu. Výrobce nevytváří reálný planogram manuálně pro každou takovou skřín, ale vytvoří pouze jeden univerzální planogram, který je pak s využitím rozmístovacího algoritmu převeden na reálné planogramy akčních skříní.

PPP byl představen na podzim roku 2008 na veletrhu ExpoPharma v Mnichově a od té doby se těší neustále rostoucímu počtu lékáren, které ho využívají.

### **7.1.4 Prototyp inteligentního rozhodování pro merchandising**

V diplomové práci [9] se Gergely Jakab zabývá návrhem algoritmu automaticky generujícího planogramy. Jako výstupní formát svého algoritmu zvolil univerzální planogramy tak, jak jsou implementovány v systému QuantSE mimo jiné pro jejich dobrou podporu estetiky a specifčnosti planogramů. Výhodou generování univerzálních planogramů namísto přímého generování reálných planogramů, je také umožnění jednoduchých doplňkových úprav merchandiserem a možnost využití jednoho vygenerovaného univerzálního planogramu pro více podobných skříní.

Merchandiser tak může použít vygenerovaný univerzální planogram, jako jakýsi počáteční návrh založený čistě na objektivních faktorech viz. 2.2.1 a provést v něm úpravy založené na subjektivních faktorech viz. 2.2.2. Úpravy nemusí provádět opakovaně pro více podobných skříní, ale pouze jednou, což znamená značné zefektivnění celého procesu

a časovou úsporu.

## 7.2 Případové studie

Vzhledem k tomu, že se firma ExTech využívající univerzálních planogramů soustředí na farmaceutický průmysl, obsahuje následující text několik příkladů nasazení systému QuantEE v tomto odvětví.

### 7.2.1 GEHE A-max

Jeden z předních distributorů léků v České Republice, společnost GEHE Pharma Praha viz. [www.gehe.cz](http://www.gehe.cz), nabízí v rámci svých marketingových aktivit lékárnám možnost účastnit se v akci A-max viz. [www.a-max.cz](http://www.a-max.cz).

Zúčastněná lékárna vyhradí pro akci jednu ze svých skříní a poskytne informace o jejich rozměrech společnosti GEHE, která každé dva měsíce vyjednává s výrobcí produktů speciální výhodné ceny na určité produkty. Výrobce tyto speciální ceny poskytne lékárnám zúčastněným v projektu jako odměnu za jejich vystavení v rezervované skříní. Marketingové oddělení společnosti GEHE je zodpovědné, za včasné zhotovení a distribuci planogramů pro akční skříně lékárnám.

Seznam vystavených produktů vzniká na základě individuálních dohod s výrobcí a mění se značně nepředvídatelně pro každou další akci. Výsledné planogramy musejí být co nejpřesnější, aby bylo zajištěno vystavení všech smluvně dohodnutých produktů. V případě dostatku prostoru jsou nasmlouvané produkty doplněny ještě dalšími podobnými produkty, aby skříně působila na zákazníky lékárny smysluplně a aby se neplýtvalo s cenným prostorem lékárny.

Realizace tohoto procesu s nástroji uvedenými v odstavci 2.3 je obtížně zvládnutelná a pro množství lékáren výrazně převyšující počet 100 znamenala neúnosnou zátěž. Díky využití systému QuantEE a principu univerzálních planogramů probíhá tvorba přesných planogramů víc jak dva roky, bez problémů. Pracovníci marketingového oddělení zvládají připravit přesné planogramy pro všechny zúčastněné lékárny vždy během několika málo dní.

### 7.2.2 Sanacorp „Meine Apotheke“

Merchandisingové oddělení společnosti Sanacorp viz. [www.sanacorp.de](http://www.sanacorp.de), velkého německého distributora léků, vytváří planogramy pro více jak 1000 spřátelených lékáren, které mají velmi rozmanité velikosti skříní. Pro každou lékárnu je třeba 16 planogramů odpovídajících různým kategoriím sortimentu jako jsou vitamíny, analgetika, léky proti chřípce apod.

Při rozhodování, jaké produkty umístit hrají v tomto případě subjektivní faktory a není tedy možné celý proces automatizovat. Tvořit potenciálně víc jak 16 000 různých planogramů s pomocí existujících merchandisingových systémů popsanych v odstavci 2.3 bylo kapacitně

nemožné a proto se jediným východiskem stala tvorba standardních planogramů.

Merchandisingové oddělení vytvářelo planogramy pro standardní skříně široké 50, 60, 70 a 80 cm ve variantách s 5 a 6 policemi vysokými 22 cm. Celkově tedy vznikalo 128 standardních planogramů a lékárna obdržela 16 rozměrově nejbližších jejím reálným skříním.

Nahrazením dosud užívaného merchandisingového systému systémem i-merch (QuantEE) se celý proces změnil na tvorbu univerzálních planogramů z nichž je automaticky vygenerováno víc než 16 000 reálných planogramů. Znamená to, že využití univerzálních planogramů umožnilo společnosti za stejnou dobu vytvořit 100x více přesných planogramů.

Na dotaz, v čem vidí hlavní přínos nahrazení bývalého merchandisingového systému systémem i-merch, odpověděla Marianne Diehl, vedoucí merchandisingového oddělení společnosti Sanacorp, následovně:

*Abychom podpořili lékárny v optimálním prezentování důležitých produktů s velkým obratem, vyvinuli jsme koncept optimalizace sortimentu. Na základě šířek skříní, obdržely lékárny odpovídající standardní planogramy. Reálná situace tak nebyla plně zobrazena. Díky systému i-merch však můžeme zajistit individuální přístup k lékárnám tak, že každá lékárna obdrží přesně planogramy odpovídající rozměrům jejich skříní.*

Originální znění tohoto vyjádření je obsaženo v příloze B.



## 8 Závěr

Cílem této práce bylo provést analýzu možností zlepšení konceptu tvorby standardních planogramů, tak aby byly odstraněny jeho hlavní nedostatky a zůstaly zachovány výhody, kvůli kterým je tento koncept velmi často využíván.

Na základě provedené analýzy požadavků, které je nutno splnit při tvorbě planogramů, byl vytvořen popsáný návrh univerzálního planogramu a rozmisťovacího algoritmu, který je na jeho základě schopen vytvořit automaticky planogram reálný včetně zdůvodnění správnosti algoritmu. Popsáný návrh dokázal při nasazení v praxi, že přináší zlepšení procesu tvorby planogramů co do samotné kvality výstupu a časových nároků a znamená konkurenční výhodu pro systémy, které jej využívají.

Součástí práce jsou zdrojové kódy prototypové implementace a software QuantPE určený k pohodlné prezentaci principu tvorby univerzálních planogramů.

V sedmé kapitole je popsáno několik systémů úspěšně využívajících zde navržené řešení.

Cíl práce se tedy podařilo splnit.

# Příloha A

## Obsah CD

K diplomové práci je přiloženo CD s následujícím obsahem:

- Text diplomové práce ve formátu PDF obsažený v souboru **/text/uni\_planogramy.pdf**.
- Zdrojový kód prototypové implementace univerzálních planogramů a rozmisťovacího algoritmu jako projekt vývojového prostředí Eclipse viz. [www.eclipse.org](http://www.eclipse.org) v adresáři **/src/eclipse\_projekt**. Z projektu je možno pohodlně vygenerovat soubor JAR použitelný jako knihovna pro další projekty. Případně je možné vygenerovat programátorskou dokumentaci ve formátu JavaDoc.
- Instalátory demonstračního systému QuantPE v adresáři **/install**.

# Příloha B

## **Originál vyjádření vedoucí merchandisingového oddělení společnosti Sanacorp**

*Um die Apotheken bei der Präsentation der absatzstarken Artikel optimal zu unterstützen, wurde das Konzept zur Sortimentsoptimierung entwickelt. Aufgrund der erhobenen Regalbreiten bekommen die Kunden sogenannte Standardlayouts, die den Maßen der Regale am ehesten entsprechen. Die reale Ist-Situation der Apotheken kann somit nicht ganz abgebildet werden. Im i-merch hingegen ist die apothekenindividuelle Realität der Regale gegeben und der Kunde bekommt genau die Planogramme zur Verfügung gestellt, die er eins zu eins umsetzen kann.*

*Marianne Diehl, Category Services - Sanacorp Pharmahandel GmbH*

# Seznam použité literatury

- [1] GLM: *DISPLAY & MERCHANDISING GUIDE*, ,  
<http://www.glmshows.com/press/DisplayHandbook/index.html>
- [2] Hana Chromková: *MERCHANDISING*, 2006,  
[http://www.pharmanews.cz/2006\\_02/merchan.html](http://www.pharmanews.cz/2006_02/merchan.html)
- [3] Team of authors: *The Essential Guide to Day to Day Category Management* , 2000,  
[http://www.ecrnet.org/04-publications/blue\\_books/pub\\_2000\\_essential\\_guide\\_daytoday\\_catman.pdf](http://www.ecrnet.org/04-publications/blue_books/pub_2000_essential_guide_daytoday_catman.pdf)
- [4] Joerg S Hofstetter, Dr., Chris Cave Jones: *The Case for ECR*, 2005,  
[http://www.ecrnet.org/04-publications/blue\\_books/pub\\_2005\\_case\\_for\\_ecr\\_full\\_report\(modified\).pdf](http://www.ecrnet.org/04-publications/blue_books/pub_2005_case_for_ecr_full_report(modified).pdf)
- [5] Hana Chromková: *MERCHANDISING*, 2006,  
[http://www.pharmanews.cz/2006\\_03/merchandis.html](http://www.pharmanews.cz/2006_03/merchandis.html)
- [6] Jon Schreibfeder: *Why Is Inventory Turnover Important?*, 1997,  
<http://www.effectiveinventory.com/article2.html>
- [7] Donna Geary: *Merchandising for Sales and Profits: Promoting Impulse Sales*, , [http://www.sideroad.com/Retail\\_Services/impulse-sales.html](http://www.sideroad.com/Retail_Services/impulse-sales.html)
- [8] Softwarový projekt: *Industry Product Placement System*, 2005,  
<http://urtax.ms.mff.cuni.cz/prk/zadani/ipps.pdf>
- [9] 9: Gergely Jakab, *Prototyp inteligentního rozhodování pro merchandising*, 2008

# Seznam ilustrací

Ilustrace 1: Příklad planogramu vytvořeného v systému QuantEE.....	6
Ilustrace 2: Planogram ve SmartDraw.....	10
Ilustrace 3: Prostředí programu ShelfLogic.....	10
Ilustrace 4: Schéma modelující podporovaný typ skříní.....	16
Ilustrace 5: Srovnání horizontálního a vertikálního produktbloku.....	19
Ilustrace 6: Magnetické čáry.....	20
Ilustrace 7: Typy zarovnání produktbloků.....	21
Ilustrace 8: Souřadnicový systém rozmisťovací mřížky.....	25
Ilustrace 9: Záložka Planogram v QuantPE.....	47
Ilustrace 10: Vlastnosti bloku v QuantPE.....	47
Ilustrace 11: Záložka Atributy skříně v QuantPE.....	48