Софийски университет „Св. Кл. Охридски”

Факултет по математика и информатика

Магистърска програма
„Софтуерни технологии”

Предмет: Обектно-ориентиран анализ и проектиране на софтуерни системи

Зимен семестър, 2022/2023 год.

Тема 33: Поддръжка на потокови диаграми (Data Flow Diagrams) във VP for UML

Ece

Автор:

Диана Василева Георгиева
фак. номер 3MI3400108

ноември, 2022

София

Съдържание

1 Въведение

Настоящият документ има за цел да представи поддръжката на потокови диаграми във VP for UML. Потоковите диаграми представят визуално движението на данните в една софтуерна система или процес. Заедно с редица други диаграми, те са част от езика за моделиране и визуализиране на дизайна на едно система – именно UML. Съществуват инструменти, които значително улесняват проектирането на диаграми, а VP (Visual Paradigm) е един от най-популярните и често използвани.

2 Характеристики и използване на VP за чертаене на потокови диаграми

В тази секция най-напред ще представим основните характеристики на потоковите диаграми и Visual Paradigm, а след това ще разгледаме по какъв начин тези диаграми се поддържат от Visual Paradigm.

2.1 Visual Paradigm (VP)

Visual Paradigm е инструмент, който поддържа UML и нотация за моделиране на бизнес процеси, част от групата за управление на обекти. Също така осигурява и генериране на отчети, както и възможност за инженеринг на код, включително и генериране на такъв. Поддържа 14 вида UML диаграми, сред които са:

- Клас диаграми
- Диаграми на случаите на употреба
- Потокови диаграми
- Диаграми на внедряването
- Пакетни диаграми
- и други.

Visual Paradigm се използва от малки и големи компании с цел подобряване и улесняване на бизнес процесите.

2.1.1 Основни характеристики

2.1.1.1 Онлайн диаграми

Чертаенето на онлайн диаграми се базира на cloud технологии и елиминира нуждата от сетъпване и допълнително конфигуриране. По този начин максимално улеснява създаването на диаграми и екипната колаборация. VP предоставя интуитивен интерфейс с възможност за drag-and-drop, както и над 1000 темплейти, които допълнително подпомагат създаването на диаграми. Поддържа над 100 вида различни диаграми. Уеб-базиран е – работи добре на различни интернет браузъри и платформи.

2.1.1.2 Екипна колаборация

Членовете на един екип могат да работят едновременно и да колаборират безпроблемно върху един и същ проект. VP използва cloud базирани хранилища, които съхраняват цялата екипна работа и я правят достъпна отвсякъде и по всяко време.

2.1.1.3 Визуално моделиране

Поддържа създаването на диаграми чрез drag-and-drop на елементи. Възможно е преизползването на елементи и диаграми, както и тяхното трансформиране. Предоставя се опция за форматиране и валидиране на синтаксиса.

2.2 Потокови диаграми

Потоковите диаграми са традиционен начин за илюстриране на движението на данните в една система. Онагледяват входната и изходната информация за системата, как тя се променя и как се съхранява.

Диаграмите на потока на данни могат да бъдат характеризирани като физически или логически.

Физическите потокови диаграми показват как ще бъде внедрена системата. Това включва софтуера, хардуера, файловете и се разработва по такъв начин, че описва изпълнението на логическия поток от данни. Логическите диаграми са фокусирани върху бизнеса и функционирането му. Описват потока от данни през системата и как това удовлетворява определена функционалност.

При използването на потокови диаграми се прилага подходът за декомпозиция отгоре-надолу. По този начин обикновено се започва с контекстна диаграма (ниво 0), която представя цялата система, чрез един процес. След това се продължава с ниво 1, което декомпозира основните функционалности на системата. Декомпозирането може да продължи на по-ниско ниво, като се приложи допълнителен анализ. Детайлността при декомпозирането зависи от нужните и сложността на дадената система, като най-често декомпозирането спира след ниво 3.

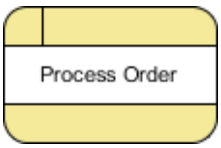
2.2.1.1 Компоненти

Ще разгледаме основните компоненти на потоковите диаграми.

- външен обект – може да репрезентира човек, система или подсистема. Това е мястото от където идва или отива определена информация. Външен е за системата от гледна точка на бизнес процесите и поради тази причина най-често се чертае в края на диаграмата. Външният обект е извън границата на системата и показва нейното взаимодействие с външния свят. Той получава или предоставя данни, но не ги обработва.

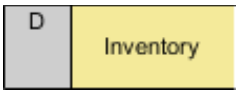
Фигура 1. Пример за нотация на външна единица [3]

- процес – бизнес функция, където информацията се обработва или трансформира. Един процес може да бъде декомпозиран на по-ниско ниво, с цел да представи как информацията се обработва в рамките на отделните процеси.



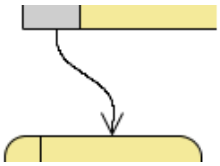
Фигура 2. Пример за нотация на процес [3]

- хранилище на данни – използва се в потоковата диаграма, когато данните трябва да бъдат съхранени. Хранилището може да бъде реализирано по различен начин – база от данни, файлове и т.н.



Фигура 3. Пример за нотация на хранилище [3]

- поток данни – показва движението на информацията от една част на системата в друга. Потоците обикновено свързват различни процеси, които обработват данните.



Фигура 4. Пример за нотация на поток данни [3]

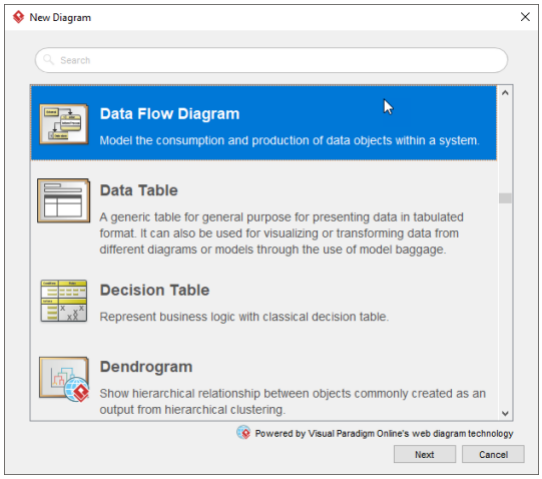
2.3 Въведение в чертаенето на потокови диаграми във VP

Ще разгледаме детайлно как VP поддържа потокови диаграми.

- Чертаене на контекстна диаграма във VP

1. За да създадем потокова диаграма във VP, трябва да изберем **Diagram -> New** от менюто.

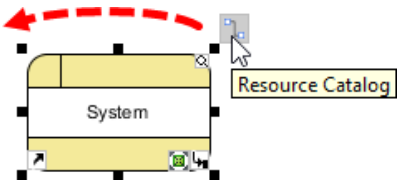
2. От прозорецът **New Diagram** селектираме **Data Flow Diagram** и натискаме бутонът **Next**.



Фигура 5. Създаване на Data Flow Diagram във VP [3]

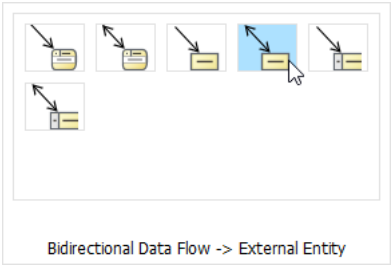
3. Въвеждаме име на диаграмата и натискаме **OK** за потвърждение.

4. За да начертаем процес трябва да изберем нотацията за процес и да зададем име. В настоящия пример именуваме процесът System.

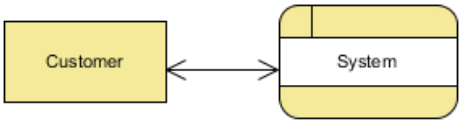


Фигура 6. Създаване на процес във VP [3]

5. Можем да създадем външна единица, като от Resource Catalog-а изберем **Bidirectional Data Flow -> External Entity**. В настоящия пример използваме Customer.



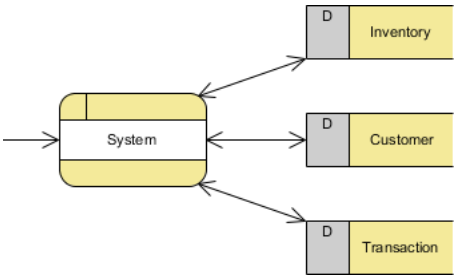
Фигура 7. Създаване на двупосочна връзка във VP



Фигура 8. Добавяне на външна единица към диаграмата [3]

6. Моделираме базата от данни, която се използва от системата, като за целта достъпваме Resource Catalog-а от System и чертаем двупосочна връзка между тях. Наименуваме хранилището Inventory.

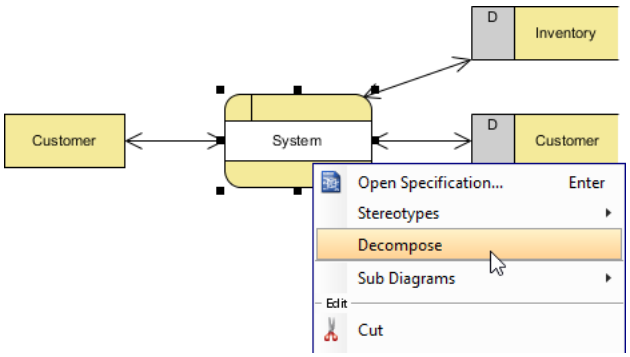
7. Създаваме още две хранилища с имена Customer и Transaction и по този начин завършваме нашата контекстна диаграма.



Фигура 9. Data Flow диаграма начертана с VP [3]

- Чертаене на потокова диаграма от ниво 1 във VP

1. Вместо да започваме изцяло от начало, може да преизползваме компонентите от създадената контекстна диаграма. За целта избираме **Decompose** от менюто след като натиснем десен бутон върху **System**.

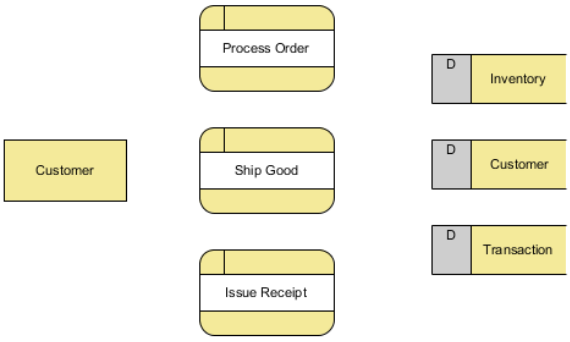


Фигура 10. Декомпозиране на Data Flow диаграма във VP [3]

2. В диаграмата от ниво 1 ще използваме хранилищата и външните обекти, които са свързани с процеса **System** в контекстната диаграма. Поначало диаграмата много прилича на контекстната и всички елементи остават непроменени с изключение на **System**, от където всъщност започва декомпозирането.

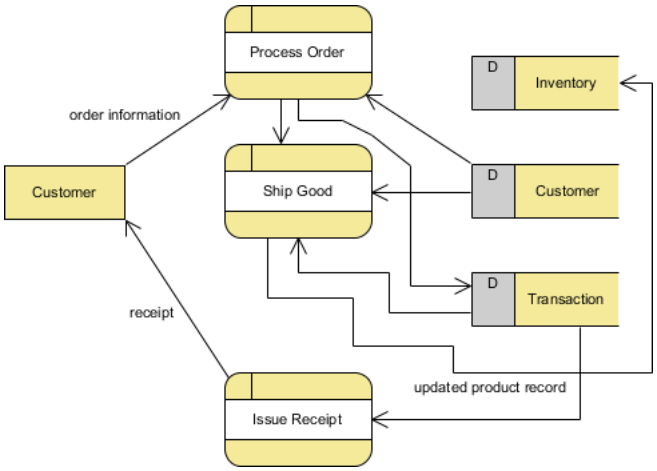
3. Може да преименуваме новата диаграма като селектираме **Rename** и въведем новото име.

4. Създаваме нови три процеса - **Process Order, Ship Good, Issue Receipt** на мястото на **System**.



Фигура 11. Създаване на процеси във VP [3]

5. Остава да начертаем връзките между компонентите на диаграмата.



Фигура 12. Финален изглед на диаграмата от ниво 1 [3]

Обработката на поръчка се извършва, използвайки информация от клиента (външен обект за системата) и хранилището. След това се създава запис в базата за транзакции. За да се осъществи доставката е необходима информацията за адреса на клиента от базата данни. След това процесът трябва да намали и съответната наличност на продуктите, които ще бъдат доставени. Когато поръчката пристигне при получателя се издава фактура на базата на информацията за съответната транзакция.

2.4 Други

Ако диаграмата изглежда претоварена и е трудна за четене/разбиране, VP предоставя възможност за подобрения. Дясно кликване върху диаграмата **-> Connectors -> Curve**, превръща връзките в диаграмата в заоблени линии, които лесно могат да бъдат пренаредени от нас, така че да не се пресичат и диаграмата да бъде по-лесна за четене.

3 Сравнителен анализ

Ще сравним VP с друг подобен инструмент, а именно – MagicDraw. В секция 3.1. ще сравним инструментите по различни критерии, а в секция 3.2. ще проследим приликите и разликите при поддръжката на потоковите диаграми.

3.1 Общо сравнение с MagicDraw

Критерий	VP	MagicDraw
Отворен код	Не	Не
Софтуерен лиценз	Commercial, Free Community Edition	Commercial / Free (educational)
Използван език за програмиране	Java, C++	HTML5 и JavaScript
Поддръжка на UML	Да	Да
Темплейти	Да	Да
Може да бъде интегриран с	Eclipse, NetBeans, IntelliJ, Visual Studio, Jira	Eclipse, EMF, NetBeans
Език, който се генерира	Java, C#, C++, PHP, Ada, Action Script	Java, C++, C#, CIL, CORBA IDL, DDL, EJB, XML Schema, WSDL

Таблица 1: Сравнение на VP с MagicDraw [5]

3.2 Сравнение на поддръжката на потокови диаграми

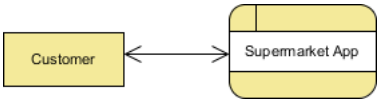
Както обстойно разглежда настоящият документ, VP предоставя възможност за чертаене на потокови диаграми. Всеки един от компонентите на потоковите диаграми се поддържа и създаването на диаграми от такъв вид е максимално улеснено.

MagicDraw от своя страна не предоставя такава възможност. Най-близкото, което е до потоковите диаграми е диаграма на активностите, която може да бъде използвана с такава цел, ако имитираме единиците и процесите.

4 Примери на използване

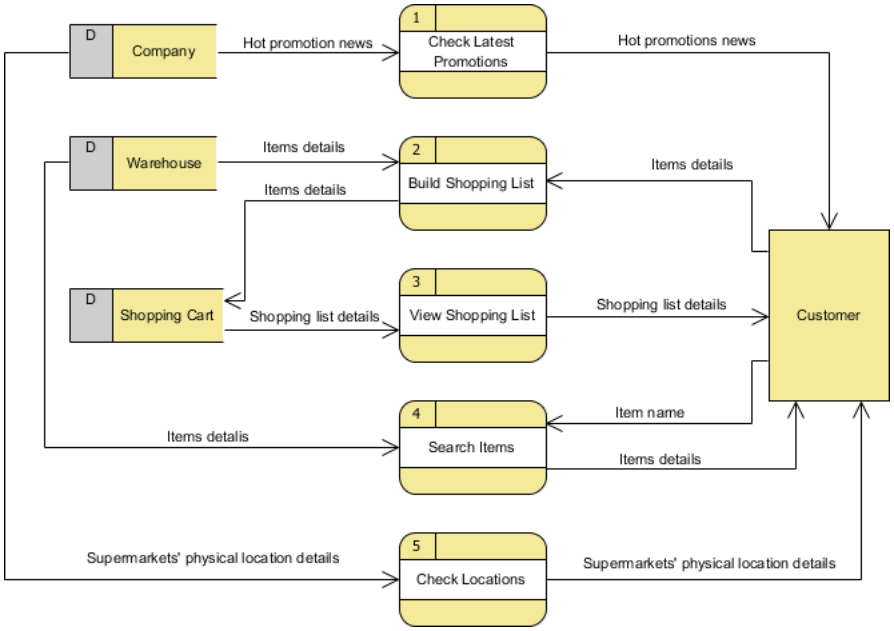
Ще разгледаме още един пример за потокова диаграма, начертана с помощта на VP. Примерът е за Супермаркет приложение.

Започваме с контекстната диаграма, която показва интеракцията на системата с външните обекти от най-високо ниво. В този пример има само един външен за системата обект – Customer (Клиент). Връзката между процеса и клиента е двупосочна.



Фигура 13. Получената контекстна диаграма [1]

Декомпозицията на контекстната диаграма води до получаването на потокова диаграма от ниво 1.



Фигура 14. Получената контекстна диаграма от ниво 1 [1]

Получената диаграма се състои от 5 процеса, 1 външна единица и 3 хранилища на данни. От нея виждаме, че клиентът може да получава новини за промоции. Клиентът може да си направи списък за пазаруване, като предостави информация за продуктите, които ще бъдат със хранени в едно от хранилищата. Той също така може да преглежда вече направения от него списък, да търси продукти в системата и да вижда физическите локации на магазини.

5 Добри практики и методи за използване

- Потоковите диаграми не трябва да бъдат претоварвани. Техният фокус е върху интеракциите на системата с външните системи, а не върху вършените комуникации между интерфейсите. Поради тази причина, обменянето на данни между интерфейсите и хранилището е определено да бъде извън границите на потоковите диаграми и не се илюстрира от тях.

- Потокът на данните и потокът на процеса не трябва да бъдат смесвани – потоковите диаграми имат за цел да визуализират обмена на данни в системата на едно абстрактно ниво и това не бива да бъде смесвано с допълнителни детайли, като например как точно този обмен се прави от самите процеси.

6 Заключение и очаквано бъдещо развитие

Потоковите диаграми са лесни за разбиране и подпомагат софтуерните процеси при разработването на системи. Тяхното чертане е съпроводено с използването на различни за целта инструменти като VP, които от своя страна предоставят и други важни характеристики, подпомагащи екипите и използваните от тях процеси. Очаква се тези инструменти да продължат да се надграждат и използват все повече от малки и големи компании.

7 Използвани литературни източници

1. Пример за Data Flow диаграма във VP - https://www.visual-paradigm.com/tutorials/data-flow-diagram-example-supermarket-app.jsp

2. Официална документация на VP - https://www.visual-paradigm.com/features/

3. Data Flow диаграми във VP - https://www.visual-paradigm.com/tutorials/data-flow-diagram-dfd.jsp

4. Описание на Data Flow диаграма - https://en.wikipedia.org/wiki/Data-flow_diagram

5. Сравняване на UML инструменти - https://socialcompare.com/en/comparison/uml-tools

6. Създаване на Data Flow диаграма в Magic Draw - https://community.nomagic.com/can-i-create-data-flow-diagram-in-the-magic-draw-t6035.html